



Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to University of Pune)

Structure & Syllabus of
B.Tech. (Computer Engineering)

Pattern 'B-14'
Effective from Academic Year 2016-17

Prepared by: - Board of Studies in Computer Engineering

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Signed by

Chairman – BOS

Chairman – Academic Board

Content

Sr. No.		Title	Page No.
1		Program Educational Objectives and Program Outcome of B.Tech (Computer Engineering)	8
2		Course Structure - Module I and II	11
	2.1	CS10102 Computer programming (Theory Course)	15
	2.2	CS10302 Computer Programming (Laboratory)	18
3		Course Structure - Module III	20
4		Course Syllabi for Courses - Module III	21
	4.1	CS20117 Discrete Structures and Graph theory(Theory Course)	23
	4.2	CS20111 Data Structures (Theory Course)	26
	4.3	CS20108 Computer Organization (Theory Course)	28
	4.4	CS20116 Problem Solving and Programming (Theory Course)	30
	4.5	CS20113 Digital Electronics and Logic Design (Theory Course)	33
	4.6	CS20216 Problem Solving and Programming (Tutorial)	35
	4.7	CS20113 Digital Electronics and Logic Design (Tutorial)	37
	4.8	CS20311 Data structures (Laboratory Course)	38
	4.9	CS20313 Digital Electronics (Laboratory Course)	39
	4.10	CS27401 Mini Project	41
	4.11	CS24306 PHP MYSQL OR	42
		CS24303 C#.NET (Skill Development Course)	44
	4.12	Elective –Soft Skills	-
	4.13	CS20401 Comprehensive Viva Voce	46
	4.14	Institute Elective	-
5		Course Structure - Module IV	47
6		Course Syllabi for Courses - Module IV	48
	6.1	CS21104 Mathematical Transformations and Applications (Theory Course)	50

	6.2	CS20110	Computer Graphics (Theory Course)	53
	6.3	CS20105	Principles of Programming Languages (Theory Course)	55
	6.4	CS21112	Data Communication (Theory Course)	57
	6.5	CS20114	Microprocessor and Interfacing (Theory Course)	59
	6.6	CS20114	Microprocessor and Interfacing (Tutorial)	61
	6.7	CS21104	Mathematical Transforms and Applications (Tutorial)	63
	6.8	CS20314	Microprocessor and Interfacing (Laboratory Course)	65
	6.9	CS20305	Principles of Programming Languages (Laboratory Course)	68
	6.10	CS20310	Computer Graphics (Laboratory Course)	70
	6.11	CS27402	Mini Project	71
	6.12	CS24302	ASP.NET OR	72
		CS24304	Python (Skill Development Course)	74
	6.13		Elective – Health & Hobby	-
	6.14	CS20402	Comprehensive Viva Voce	76
7		Course Structure - Module V		77
8		Course Syllabi for Courses - Module V		78
	8.1	CS30101	Operating Systems (Theory Course)	80
	8.2	CS30116	Computer Networks (Theory Course)	83
	8.3	CS30105	Theory of Computation (Theory Course)	86
	8.4	CS31113	Microprocessors and Microcontrollers (Theory Course)	89
	8.5	CS31115	Advanced Data Structures (Theory Course)	91
	8.6	CS30101	Operating Systems (Tutorial)	93
	8.7	CS31215	Advanced Data Structures (Tutorial)	95
	8.8	CS30303	Operating Systems (Laboratory Course)	96
	8.9	CS30316	Computer Networks (Laboratory Course)	98
	8.10	CS31313	Microprocessors and Microcontrollers (Laboratory Course)	100
	8.11	CS37401	Mini Project	101
	8.12		Professional Development Course (Institute Level)	-

	8.13	CS30401	Comprehensive Viva Voce	102
	8.14	CS37301	Seminar	103
	8.15	CS37302	Project Stage - 1	104
9		Course Structure - Module VI		107
10		Course Syllabi for Courses - Module VI		108
	10.1	CS30102	Software Engineering (Theory Course)	110
	10.2	CS30106	Database Management Systems (Theory Course)	112
	10.3	CS30108	Design and Analysis of Algorithm (Theory Course)	114
	10.4	CS30114	Systems Programming (Theory Course)	117
	10.5	CS31119	Object Oriented Modeling and Design (Theory Course)	119
	10.6	CS30102	Software Engineering (Tutorial)	122
	10.7	CS30108	Design and Analysis of Algorithm (Tutorial)	123
	10.8	CS30314	System programming (Laboratory Course)	125
	10.9	CS31319	Object Oriented Modeling and Design (Laboratory Course)	127
	10.10	CS30306	Database Management Systems (Laboratory Course)	129
	10.11	CS37402	Mini Project	131
	10.12		Professional Development Course (Institute Level)	-
	10.13	CS30402	Comprehensive Viva Voce	132
	10.14	CS37301	Seminar	133
11		Course Structure - Module VII		134
12		Course Syllabi for Courses - Module VII		135
	12.1	CS40115	Distributed Computing (Theory Course)	137
	12.2	CS40119	Business Intelligence and Analytics (Theory Course)	140
	12.3		*Elective Group I (Theory Course)	144
		CS42105	• Mobile Computing	
		CS42131	• Enterprise Systems	146
		CS42127	• Cloud Computing	149
		CS42103	• Parallel Computing on GPU	152
		CS42106	• Network Security	154

		CS42101	<ul style="list-style-type: none"> Advanced Computer Graphics 	157
	12.4		**Elective Group II (Theory Course)	
		CS42125	<ul style="list-style-type: none"> Randomized and Approximation Algorithms 	159
		CS42113	<ul style="list-style-type: none"> Digital Signal Processing 	162
		CS42119	<ul style="list-style-type: none"> Information Retrieval 	165
		CS42114	<ul style="list-style-type: none"> Product Design 	167
		CS42115	<ul style="list-style-type: none"> Human Computer Interaction 	170
		CS42134	<ul style="list-style-type: none"> Modeling and Simulation 	172
	12.5	CS40115	Distributed Computing (Tutorial Course)	174
	12.6		**Elective Group II (Tutorial Course)	
		CS42125	<ul style="list-style-type: none"> Randomized and Approximation Algorithms 	176
		CS42113	<ul style="list-style-type: none"> Digital Signal Processing 	178
		CS42119	<ul style="list-style-type: none"> Information Retrieval 	180
		CS42114	<ul style="list-style-type: none"> Product Design 	181
		CS42115	<ul style="list-style-type: none"> Human Computer Interaction 	182
		CS42134	<ul style="list-style-type: none"> Modeling and Simulation 	183
	12.7	CS40319	Business Intelligence and Analytics (Laboratory Course)	184
	12.8		*Elective Group I (Laboratory Course)	
		CS42305	<ul style="list-style-type: none"> Mobile Computing 	186
		CS42331	<ul style="list-style-type: none"> Enterprise Systems 	187
		CS42327	<ul style="list-style-type: none"> Cloud Computing 	189
		CS42303	<ul style="list-style-type: none"> Parallel Computing on GPU 	190
		CS42306	<ul style="list-style-type: none"> Network Security 	191
		CS42301	<ul style="list-style-type: none"> Advanced Computer Graphics 	193
	12.9	CS47303	Project Stage - II	195
13		Course Structure - Module VIII		197
14		Course Syllabi for Courses - Module VIII		198
	14.1	CS40106	Compiler Design (Theory Course)	200
	14.2	CS40110	Artificial Intelligence (Theory Course)	202
	14.3		\$Elective Group III (Theory Course)	
		CS42120	<ul style="list-style-type: none"> Data Mining 	204
		CS42128	<ul style="list-style-type: none"> Machine Learning 	206
		CS42104	<ul style="list-style-type: none"> Neural Networks 	208

		CS42111	<ul style="list-style-type: none"> Software testing and Quality Assurance 	210
		CS42124	<ul style="list-style-type: none"> Algorithmic Number Theory and Algebra 	213
	14.4		\$Elective Group IV (Theory Course)	
		CS42118	<ul style="list-style-type: none"> Geographical Information Systems 	216
		CS42129	<ul style="list-style-type: none"> Management Information Systems 	218
		CS42121	<ul style="list-style-type: none"> Advanced Computer Architecture 	221
		CS42116	<ul style="list-style-type: none"> Convergence Technologies 	224
		CS42133	<ul style="list-style-type: none"> Embedded Systems 	226
		CS42130	<ul style="list-style-type: none"> Image Processing 	228
	14.5		\$Elective Group III (Tutorial)	
		CS42120	<ul style="list-style-type: none"> Data Mining 	230
		CS42128	<ul style="list-style-type: none"> Machine Learning 	231
		CS42104	<ul style="list-style-type: none"> Neural Networks 	232
		CS42111	<ul style="list-style-type: none"> Software testing and Quality Assurance 	233
		CS42124	<ul style="list-style-type: none"> Algorithmic Number Theory and Algebra 	235
	14.6		\$Elective Group IV (Tutorial)	
		CS42118	<ul style="list-style-type: none"> Geographical Information Systems 	237
		CS42129	<ul style="list-style-type: none"> Management Information Systems 	238
		CS42121	<ul style="list-style-type: none"> Advanced Computer Architecture 	239
		CS42116	<ul style="list-style-type: none"> Convergence Technologies 	240
		CS42133	<ul style="list-style-type: none"> Embedded Systems 	241
		CS42130	<ul style="list-style-type: none"> Image Processing 	243
	14.7	CS40306	Compiler Design (Laboratory Course)	245
	14.8	CS40310	Artificial Intelligence (Laboratory Course)	247
	14.9	CS47308	Project Stage - III	248
15		Course Syllabi for PD Courses in TY B.Tech (Computer Engineering)		251
	15.1	CS33303	Advanced Java	252
	15.2	CS33312	PIC Microcontroller	254
	15.3	CS33313	Mobile Application Development	255
	15.4	CS33306	Ethical Hacking and Network Defense	257
	15.5	CS33310	Spring Framework	259

	15.6	CS33314	Struts Framework	261
	15.7	CS33311	Problem Solving and Programming	263
	15.8	CS33315	Big Data Technologies	265
	15.9	CS33307	Matlab	266

Program Educational Objectives (PEO)

B.Tech (Computer Engineering)

List of Programme Education Objectives [PEO] and Programme Outcomes [PO]

PEO	PEO Statement
PEO1	Preparation: Demonstrate application of sound engineering foundations to be a committed technology workforce
PEO2	Core competence: Apply mathematical and computing theory knowledge base to provide realistic computer engineering solutions
PEO3	Breadth: Exhibit problem solving skills and engineering practices to address problems faced by industry with innovative methods, tools and techniques
PEO4	Professionalism: Adopt professional and ethical practices adopting effective guidelines to acquire desired soft skills in societal and global context
PEO5	Learning Environment: Aim for continuing education and entrepreneurship in emerging areas of computing

List of Programme Outcomes [PO]

Graduates will be able

PO	PO Statement
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering

	solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
PSO	PSO Statement
PSO1	Select and incorporate appropriate computing theory principles, data structures and algorithms, programming paradigms to innovatively craft scientific solution addressing complex computing problems.
PSO2	Adapt to new frontiers of science, engineering and technology by getting acquainted with heterogeneous computing environments and platforms, computing hardware architectures and organizations through continuous experimentation.
PSO3	Conceive well-formed design specifications and constructs assimilating new design ideas and facts for identified real world problems using relevant development methodologies and practices, architecture styles and design patterns, modeling and simulation, and CASE tools.
PSO4	Exercise research and development aptitude focusing knowledge creation and dissemination through engineering artifacts construction, preparation and presentation of engineering evidences using procedures, techniques, guidelines, and standards considering technology migration and evolution.

FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

F.Y. B. Tech. Structure with effect from Academic Year 2015-16
Module 1

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
HS10109	Linear Algebra and Random Variables	Theory - Core	3	-	1	10	20	5	5	-	60	4
HS10103	Modern Physics	Theory – Core	3	-	1	10	20	5	5	-	60	4
CH10101	Chemistry	Theory – Core	3	-	-	15	20	5	-	-	60	3
ME10101	Engineering Graphics	Theory – Core	3	-	-	15	20	5	-	-	60	3
HS16101	Sociology	Theory – OE	2	-	-	15	20	5	-	-	60	2
HS16103	Psychology											
HS16105	Philosophy											
HS16107	EVS											
HS10301	Engineering Graphics Lab	Lab – Core	-	2	-	-	-	-		70	30	1
CH10301	Science Lab	Lab – Core	-	2	-	-	-	-		70	30	1
HS17401	Mini Project	Project	-	4	-	-	-	-		70	30	2
TOTAL			14	08	2							20

FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

F.Y. B. Tech. Structure with effect from Academic Year 2015-16

Module 2

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
HS10102	Differential Integral and Calculus	Theory - Core	3	-	1	10	20	5	5	-	60	4
HS10104	Engineering Mechanics	Theory – Core	3	-	1	10	20	5	5	-	60	4
HS10108	Electrical Engineering Fundamentals	Theory – Core	3	-	-	15	20	5	-	-	60	3
CS10102	Computer Programming	Theory – Core	3	-	-	15	20	5	-	-	60	3
HS16102 HS16104 HS16106 HS16108	Economics Management Technology Cost & Acc. Business Law	Theory – OE	2	-	-	15	20	5	-	-	60	2
CS10302	Computer Programming	Lab – Core	-	2	-	-	-	-		70	30	1
HS10306	Engineering Lab	Lab – Core	-	2	-	-	-	-		70	30	1
HS17402	Mini Project	Project	-	4	-	-	-	-		70	30	2
TOTAL			14	08	2							20

F.Y. B. Tech. Structure with effect from Academic Year 2015-16

Semester I – Irrespective of Module

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
HS10107	Communication Skill	Comm. Skill	-	2	-	-	-	-		70	30	1
HS153xx	General Proficiency	GP	-	2	-	-	-	-		70	30	1
HS15301	English I	Language	-	2	-	-	-	-		70	30	1
HS15302	French I											
HS15303	German I											
HS15304	Spanish I											
HS15305	Japanese I											
HS14301	Engineering Workshop	Workshop	-	2	-	-	-	-		70	30	1
TOTAL			-	8	-							4

F.Y. B. Tech. Structure with effect from Academic Year 2015-16

Semester II – Irrespective of Module

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
HS17301	General seminar I	Comm. Skill	-	2	-	-	-	-		70	30	1
HS153xx	General Proficiency	GP	-	2	-	-	-	-		70	30	1
HS15306	French II	Language	-	2	-	-	-	-		70	30	1
HS15307	German II											
HS15308	Spanish II											
HS15309	Japanese II											
HS15310	English II											
HS14302	Trade Workshop	Workshop	-	2	-	-	-	-		70	30	1
TOTAL			-	8	-							4

HS153xx : General Proficiency Courses as per following list

List of General Proficiency Courses

FY B Tech
AY 2015-16

Sr. No.	Course Code	Name of Course
1	HS15311	Flute
2	HS15312	Guitar
3	HS15313	Tabla
4	HS15314	Bharat Natyam
5	HS15317	Yoga
6	HS15318	Pranayam
7	HS15319	Aerobics
8	HS15321	Photography
9	HS15322	Digital Photography
10	HS15323	Volleyball
11	HS15324	Chess
12	HS15325	Taekwondo
13	HS15326	Film Appreciation
14	HS15327	Shares and Stocks
15	HS15328	Fundamentals of Banking
16	HS15329	Nutrition and Fitness
17	HS15330	Spirit of Entrepreneurship
18	HS15331	Memory Techniques
19	HS15332	Sanskrit Sambhashanam
20	HS15333	Numerology

CS10102:: COMPUTER PROGRAMMING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+2 Hrs)

Part A: Introduction to Programming: Problem solving using computers and logic design. Algorithms and their representations: flowcharts, pseudo code. Designing algorithms for problems like finding min-max, mean, median, mode, mensuration and roots of a quadratic equation. Concept of programming languages for implementing algorithms – levels of languages. Role of assemblers, compilers, linker, loader, interpreter in program execution.

Introduction to C: “Hello World” in C – editor, compiler, execution environment. C as a middle level language. Basic structure of C program, standard library and header files, Syntax and Semantics. Variable, constant (literal and named), Data types, variable declaration. Assignment. Operators: Arithmetic, logical, relational, Expressions, Precedence & Associativity. Input and output statements, escape sequences.

Part B: Bits and bytes – importance of digital representation in computers. Number System and algorithms for inter conversions. C programming on Linux and Windows.

Unit 2: (8+2 Hrs)

Part A: Flow of Control: Selection Statement: if, nested if –else, Conditional Expression, Switch statements. Iteration Statements: for loop, while loop, do -while loop, nested loop. Statements: go to, break & continue. Common programming errors. Application of C constructs in solving problems like generating arithmetic and geometric progression, prime numbers.

Arrays: Concept, declaration and initialization of arrays, accessing individual elements of array. Use of arrays in sorting, searching. Concept of 2-D array (Matrix), row major and column major representation of array, address calculation for accessing the individual element.

Part B: Static variables and constants in C language.

Unit 3: (8+2 Hrs)

Part A:

Functions: Need of functions, function declaration, definition and call. Inbuilt functions and user defined functions. Passing arguments to a function, returning values from a function. Scope of variable, local and global variable. Access specifiers. Passing arrays to functions.

Recursive Functions: Need of Recursion, direct recursion, indirect recursion, impact of recursion on local & global variables, examples of recursive programs – factorial, progressions, towers of Hanoi. Recursive vs Iterative solutions. Disadvantages of recursion.

Part B: Preprocessor and preprocessor directives: macro substitution, difference between macro and functions.

Unit 4: (8+2 Hrs)
Part A:

Pointers: Concept of pointers, relevance of data type in pointer variable, pointer arithmetic. Pointer to pointer. Pointers and functions (passing pointers to functions, returning pointers from functions). Pointers and arrays. Pointers and strings. Pointer constants. Array of pointers, pointer to array. Various alternatives of accessing arrays (1-D and 2-D) using pointers.

Strings: Strings as arrays, character array versus strings, reading strings, writing strings, user defined functions for string operations – copy, concatenate, length, reverse, converting case, appending, comparing two string, extracting a substring. Array of strings.

Part B: Const keyword in C, standard string library functions in string.h for string manipulation.

Unit 5: (8+2 Hrs)

Part A: Structures: Notion, declaration and initialization, structure variables, accessing and assigning values of the fields, "size of" operator, functions and structures, arrays of structures, nested structures, pointers and structures, passing structure to a function and returning structure from function. Dynamic memory allocation, type casting, Introduction to self referential structures, linked list as a dynamic alternative to arrays.

File Handling in C: file types, file opening modes, file handling I/O – fprintf, fscanf, fwrite, fread, fseek. File pointers. Implementing basic file operations in C.

Part B: Typedef keyword. Union, Nesting of Structure and Union. Enumerated data types.

Text Books

1. "An Introduction to Programming through C++ ", Abhiram Ranade, Mc Graw Hill Education. ISBN 978-9-33-290151-3
2. "Programming with C- Schaum's outline Series", B. Gottfried, Second edition, Tata McGraw Hill Publication, ISBN 0-07-463491-7,
3. "Let us C", Y. Kanetkar, Second Edition, BPB Publication. ISBN: 8176566217.

Reference Books

1. "Programming language – ANSI C", Brian W Kernighan and Dennis Ritchie, Second edition ISBN 0-13-110370-9,

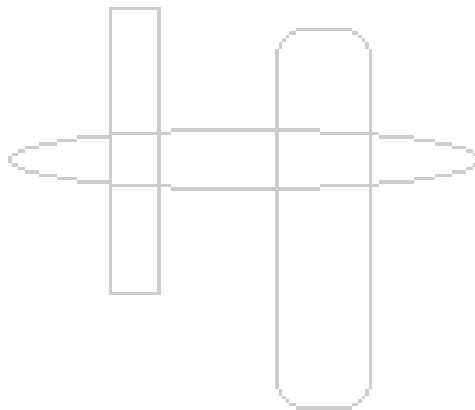
Additional Reading

1. "A first book of C- Fundamental of C Programming", Gary Bronson and Stephen Menconi, ISBN: 0314073361,

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. List procedural programming benefits to solve real world problems using generic prototype.
2. Develop visual representation of problem in hand.
3. Apply available algorithmic principles to general efficient solutions
4. Justify modular programming approach by making use of elementary as well as superior data structures.
5. Apply programming fundamentals solve real world problems using generic prototype.
6. Evaluate and manipulate given solutions in reengineered view



CS10302:: COMPUTER PROGRAMMING LAB

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practicals

1. Study of most important DOS/UNIX commands.
2. Write a program in C to find largest element / average of given N elements / sum / reverse of a given integer.
3. Write a program in C to implement a simple mathematical calculator
4. Write a program in C to read an integer and display each of the digits of an integer in English.
5. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
6. Write a program in C to perform Addition / Subtraction / Multiplication of two Matrices. Also determine whether the matrix is symmetric / skewed.
7. Write a program in C to carry out following operations on strings using string library Functions:
 - a. Length of a sting.
 - b. Copy a string.
 - c. Concatenation of strings.
8. Write a program in C to carry out following operations on strings without using string library functions
 - a. Compare two strings.
 - b. Reverse given string.
 - c. To check if the given string is a palindrome or not.
9. Write a program in C to carry out following operations on strings using pointers.
 - a. Length of a sting.
 - b. Concatenation of strings.
 - c. Copy of string
 - d. Compare two strings.
10. Write a C program that works with complex numbers using a structure. Perform the following operations:
 - a. Reading a complex number.

- b. Addition of two complex numbers.
- c. Writing a complex number.
- d. Multiplication of two complex numbers.

11. Write a C program to create a database of students by using array of structure and perform following operations on it.

- a. Accept/modify record of student
- b. Search a particular record
- c. Display all records

12. Write a program in C that use both recursive and non-recursive functions to find the Factorial / GCD (greatest common divisor) of two given integers / Fibonacci series.

13. Write a program in C to sort n integers using bubble / merge sort.

14. Write a program in C to search a number in a given list using linear / binary search.

Text Books

1. *"Programming with C- Schaum's outline Series", B. Gottfried, Second edition, Tata McGraw Hill Publication, ISBN 0-07-463491-7,*
2. *"Let us C", Y. Kanetkar, Second Edition, BPB Publication. ISBN: 8176566217.*

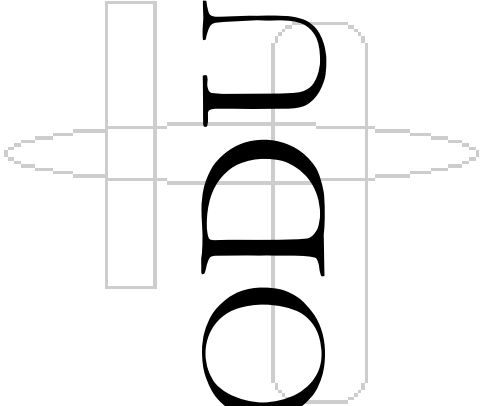
Reference Books

1. *"Programming language – ANSI C", Brain W Kernighan and Dennis Ritchie, Second edition ISBN 0-13-110370-9,*
2. *" Object Oriented Programming with C+ +", E. Balaguruswamy, Tata McGraw Hill Publication, ISBN 0-07-462038-x.*

Additional Reading

1. *"A first book of C- Fundamental of C Programming", Gary Bronson and Stephen Menconi, ISBN: 0314073361,*
2. *"C++ Program Design: An introduction to Programming and Object-Oriented Design", Cohoon and Davidson, 3rd Edition, Tata McGraw Hill. 2003, ISBN-13: 978-0-07-122649-3.*

MODULE III



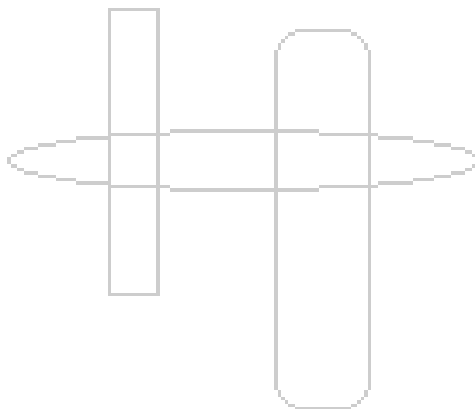
S.Y. B. Tech. Structure with effect from Academic Year 2015-16

Module III

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS20117	Discrete Structures and Graph Theory	S ₁	3	-		10	20	10		-	60	3
CS20111	Data Structures	S ₂	3	-	-	10	20	10	-	-	60	3
CS20108	Computer Organization	S ₃	3	-	-	10	20	10	-	-	60	3
CS20116	Problem Solving and Programming	S ₄	2	-	1	10	20	5	5	-	60	3
*CS20113 #CS20114	Digital Electronics and Logic Design Microprocessor and Interfacing	S ₅	3		1	10	20	5	5	-	60	4
CS20311	Data Structures	P ₁	-	4	-	-	-	-		30	70	2
*CS20313 #CS20314	Digital Electronics Microprocessor and Interfacing	P ₂		2						30	70	1
CS27402	Mini Project	MP ₃		4						30	70	2
CS24306 CS24303	PHP MYSQL OR C#.NET	SD ₃		2						30	70	1
*LAB ₃	Technical Writing	LAB ₃		2						70	30	1

#LAB ₃	General Seminar – 2	LAB ₃								70	30	1
CS20401	Comprehensive Viva Voce	CVV ₁										2
TOTAL			16	14	2							25

Students will register only in Semester IV irrespective of Module * Students will register only in Semester III irrespective of Module



CS20117: DISCRETE STRUCTURES AND GRAPH THEORY

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Elementary knowledge of 1) Numbers 2) Probability

Unit 1 (Logic and Proofs)

(8 hrs)

Part A:

Propositional logic, propositional equivalences, predicates and quantifiers, rules of inference, introduction to proofs- direct, trivial, contraposition, contradiction, counterexamples; Normal forms – DNF and CNF.

Part B: Program Correctness

Unit 2 (Elementary Discrete Structures & Basic Counting)

(8 hrs)

Part A:

Elementary set theory, relations, functions, principle of mathematical induction, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions), Permutations with indistinguishable objects, Binomial coefficients and identities.

Part B:

Generalized permutations and combinations (distinguishable/indistinguishable objects)

Unit 3 (Advanced Counting Techniques)

(8 hrs)

Part A:

Pigeon-Hole Principle : Some elegant applications, Inclusion Exclusion Principle : Counting with Venn Diagrams, (*some examples from* counting Derangements, number of primes upto n , number of onto functions, Euler's phi function), Recurrence relations, modeling using recurrence relations (*some examples from:* Fibonacci numbers, Catalan numbers, Derangements, Tower of Hanoi), solution of linear recurrence relations with constant coefficients (homogenous and non-homogenous), generating functions and coefficients.

Part B:

Turan's generalization of Mantel's theorem, Mobius inversion formula and some applications.

Unit 4 (Modular Arithmetic)**(8 hrs)****Part A.**

Number theory – Division Algorithm, Euclid's Algorithm, extended Euclid's algorithm, modular inversion, Fundamental Theorem of Arithmetic, Congruence's, Fermat's little theorem, Euler's phi function, Chinese remainder theorem, Diffie-Hellman and RSA algorithms.

Part B: Fast exponentiation**Unit 5 (Graph Theory)****(8 hrs)****Part A:**

Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, connected components, degree of a vertex, paths, cycles in graph, Euler and Hamiltonian tours/graphs, Trees, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Theorem on bound on number of edges, Graph colorings

Part B:

Hall's marriage theorem, perfect matching's in graph, Tutte's theorem, Konig's theorem,

Text Books:

1. *Discrete Mathematics and its applications* by Kenneth Rosen (William C Brown Publisher)
2. *Applied Combinatorics* by Alan Tucker (Wiley Publishing company)
3. *Combinatorics: Topics, techniques, algorithms* by Peter J. Cameron (Cambridge University Press)
4. *Graph Theory* by Reinhard Diestel (Springer Verlag Publishing Company)
5. *Introduction to Graph Theory* by Douglas B. West (Prentice-Hall publishers)
6. *Probability, Statistics and Queuing theory*", Allen, A.O., Academic Press.
7. *'Elementary Number Theory'*, David Burton
8. *Probability and Statistics for Engineers*, G. S. S. Bhishma Rao, SciTech Publications (India) Pvt. Ltd.

Reference Books:

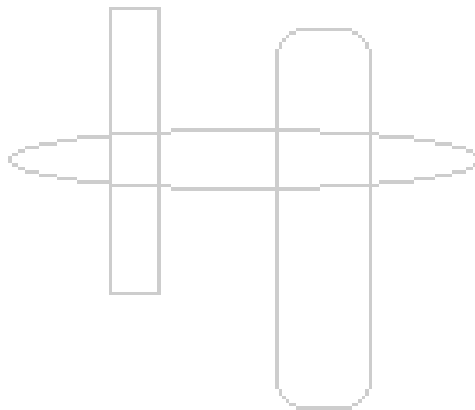
1. *A Course in Combinatorics* by J. H. Van Lint, R. M. Wilson (Cambridge University Press)
2. *Algebra* by Michael Artin (Pearson Prentice Hall)
3. *Probability and Statistics*, William Mendenhall, Robert J. Beaver, Barbara M. Beaver, Cengage Learning India Pvt. Ltd.

4. *Probability and Statistics for Scientist and Engineers, Rao V. Dukkupati, New Age International Publishers.*

Course Outcomes:

Upon completion of the course, graduates will be able to –

- 1) **Reason** mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
- 2) **Describe** the elementary properties of modular arithmetic and their applications in Computer Science like cryptography.
- 3) **Summarize** graph theory fundamentals and their applications
- 4) **Develop** recurrence relations for a wide variety of interesting problems
- 5) **Express** mathematical properties via the formal language of propositional and predicate logic
- 6) **Demonstrate** use of pigeon-hole and inclusion-exclusion principle in solving elegant and important problems



CS20111:: Data Structures

FF No. : 654 A

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Fundamentals of Data Structures, Sorting & Searching(8 Hrs)

Part A

Introduction, Types: Primitive, Non primitive, Linear, Nonlinear, Static, Dynamic Data structures, Internal Sorting Techniques: Selection, Bubble, Insertion, Radix, Shell, Merge, Quick sort, Time & Space Complexity Analysis Searching: Linear, Binary, Fibonacci, Hashing techniques: Hash table, Hash functions, Collision and Overflow Handling Evaluation of Overflow techniques, Heaps: Max Heap, Min Heap, Heap Sort

Part B: External Sorting, Sparse Matrix: Addition and Fast transpose

Unit 2: Elementary Data Structures

(8 Hrs)

Part A

Stack: Fundamentals of stack, representation using array, Applications of stack: Recursion, Validity of parentheses, Expression conversions and evaluations etc., Queue: Fundamentals of queue, representation using array, Types of queue, Applications of Queue: Job Scheduling, Josephus problem etc.

Part B: Priority Queue

Unit 3: Linked Lists

(8 Hrs)

Part A

Singly Linked Lists, Doubly linked Lists, Circular linked lists, Generalized linked lists, Applications: Stack & Queue using linked list, Polynomial Manipulation using linked list & Generalized linked list.

Part B: Dynamic memory allocation for matrices and operations on matrices, Sparse matrix representation

Unit 4: Trees

(8 Hrs)

Part A

Basic terminology, representation using array and linked list, Tree Traversals: Recursive and Non recursive, Operations on binary tree: Finding Height, Leaf nodes, counting no of nodes etc., Construction of binary tree from traversals, Binary Search trees (BST): Insertion, deletion of a node from BST, Optimal Binary Search tree (OBST), Threaded Binary tree (TBT): Creation and traversals on TBT, Height Balanced Tree (AVL): Rotations on AVL tree, M-way search trees: Btrees, B⁺ tree

Part B: Red-Black Trees, Game trees, Expression tree

Unit 5 : Graph

(8 Hrs)

Part A

Terminology and representation, Traversals, Connected components and Spanning trees: Prim's and Kruskal's Algorithm, Shortest Paths and Transitive Closures: Single Source all destinations (Dijkstra's Algorithm), All pair Shortest Path Algorithm, Activity Network, Topological Sort and Critical Path

Part B: Multistage Graphs

Text Books:

1. "Fundamentals of Data Structures in C", E. Horwitz, S. Sahani, Anderson-Freed, Universities Press, Second Edition, 2008, ISBN 978-81-7371-605-8.
2. "Data structures using C and C++", Y. Langsam, M.J. Augenstein, A.M. Tenenbaum, Pearson Education, Second Edition, 2002, ISBN 81-7808-729-4.

Reference Books :

1. "An Introduction to data Structures with applications", J. Tremblay, P. soresan, TMH Publication, 2nd Edition, 1984. ISBN-0-07-462471-7.
2. "Fundamentals of Computer Algorithms", E. Horwitz, S. Sahani, S. Rajasekaran, Galgotia book source, New Delhi, 2005, ISBN 81-7515-257-5.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1	To interpret and diagnose the properties of data structures with their memory representations.
2	Handle operations like searching, insertion, deletion, traversing mechanism etc. on data structures like stack and Queue.
3	Use linear and nonlinear data structures like stacks, queues, linked list etc
4	Demonstrate the use of tree and its applications.
5	Analyze the real world problems using Graph Data Structure.
6	Apply an appropriate data structure and algorithm to solve a problem

CS20108:: COMPUTER ORGANIZATION

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1 :Structure of a Computer System (8 Hrs)

Part A: Organization & Architecture, Structure & Function, Brief History of computers, Von Neumann Architecture, Bus Structure, Elements of Bus Design, Integer Representation: Fixed point & Signed numbers. Integer Arithmetic: 2's Complement arithmetic, multiplication, Booth's Algorithm, Division Restoring Algorithm, Non Restoring algorithm, Floating point representation: IEEE Standards for Floating point representations (Single Precision Format).

Part B: PCI Bus. IEEE standards for Floating point representations (Double Precision Format).

Unit 2 :Processor Organization (CPU) (8 Hrs)

Part A: CPU Architecture, Register Organization, Instruction types, Types of operands, Instruction formats, addressing modes and address translation. Instruction cycles, Instruction Pipelining, Pipelining Hazards, Performance Considerations – Effect of Instruction Hazards, Number of pipeline stages. RISC Processors: RISC- Features, CISC Features, Comparison of RISC & CISC Superscalar Processors. Register Organization in 8086/88, 80386Dx microprocessors, Single-core vs multicore processor organization.

Part B: Introduction to i7 processor.

Unit 3: Control Unit (8 Hrs)

Part A: Fundamental Concepts: Single Bus CPU organization, Register transfers, Performing an arithmetic/ logic operations, fetching a word from memory, storing a word in memory, Execution of a complete instruction. Micro-operations, Hardwired Control, Example- Multiplier CU. Micro-programmed Control: Microinstructions, Microinstruction-sequencing: Sequencing techniques, Micro-program sequencing.

Part B: Multiple-bus organization. Applications of micro programming.

Unit 4: Memory Organization (8 Hrs)

Part A: Need, Hierarchical memory system, Characteristics, Size, Access time, Read Cycle time and address space. Main Memory Organization: ROM, RAM, EPROM, DRAM (example of 64kb memory using 8kb chips), SDRAM, DDR3, Cache memory Organization: Address mapping, Replacement Algorithms, Cache Coherence, MESI protocol. Virtual Memory: Segmentation, Paging, interleaved memories.

Part B: Secondary Storage: Magnetic Disk, Optical memory, CDROM, RAID

Unit 5: Advanced Computer Organizations (8 Hrs)

Part A: The Intel IA-64 Architecture: General Organization, Prediction, Speculation Software pipelining, Instruction set architecture. Introduction to parallel processing: Trends towards parallel processing, architectural classification schemes (Flynn's classification) Introduction to multicore processor AMD/ NVIDIA GPU architecture.

Part B: Itanium Organization

Text Books

1. *"Computer Organization and Architecture: Designing for Performance"*, William Stallings, Pearson Prentice Hall Publication, ISBN 81-7758-9 93-8, 7th Edition.
2. *"Computer Organization"*, C. Hamacher, V. Zvonko, S. Zaky, Tata McGraw Hill Publication, ISBN 007-120411-3, 5th Edition.

Reference Books

1. *"Computer Architecture and Parallel Processing"*, Hwang and Briggs, Tata McGraw Hill Publication ISBN 13: 9780070315563.
2. *"Structured Computer Organization"*, A. Tanenbaum, Prentice Hall Publication, ISBN 81 – 203 – 1553 – 7, 4th Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

- 1 Operationalize arithmetic and control unit based on computer architectures.
- 2 Analyze impact of circuit driven and program driven analogies to assemble realizable solutions.
- 3 Design processor and memory with due consideration to tradeoffs and performance issues.
- 4 Suggest performance bound solutions in order to demonstrate variety of technologies.
- 5 Recognize historical scenario of computing unit's development with regards to technological evolution.
- 6 Illustrate organization of digital computers with basic principles and operations of its components.

CS20116:: PROBLEM SOLVING AND PROGRAMMING

Credits: 02

Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites: Computer Programming

Unit 1 Introduction

(6+1 Hrs)

Part A:

General Problem solving techniques: Examples of problems that are solved using different approaches: ask questions, look for things that are familiar, solve by analogy, means-ends analysis, divide and conquer (top-down approach or stepwise refinement), building block approach, merging solutions, working backwards from a solution, binary doubling strategy, iterative vs recursive solutions, parallel techniques

Program Verification: computer model for program execution, correctness of programs, input and output assertions, implication and symbolic execution, verification of different types of program segments – straight line program segment, having branch, having loops, loop invariants, using arrays. Proof of termination. Debugging programs and program testing. The challenge of Binary Search.

Efficiency of programs: considerations during design, considerations during implementation, reducing time and space requirements of programs, choosing right data structures and data types – arrays, lists, tables, bitmaps. When to use what?

Part B. Input validation – GIGO, input validation loop, defensive programming

Unit 2: Basic Algorithms for Integers

(6+1 Hrs)

Part A:

Fundamental Algorithms – exchanging values of two variables, generating Fibonacci sequence, reversing digits of an integer, base conversion, character to number conversion

Factoring Methods: finding square root of a number, finding GCD of two numbers, generation of pseudo random numbers, generating prime numbers, finding prime factors of an integer, raising a number to a large power.

Part B. Simple modular arithmetic - last k digits of number a^b , finding recurring decimal expansion for rational number a/b

Unit 3: Numerical Methods

(6+1 Hrs)

Part A.

Computing common mathematical functions – Taylor series, Systems of linear and non-linear equations: Newton Raphson method, Bisection method, Secant method, False position method, how a necklace reposes. Numerical integration . Solution of Differential equations - Runge Kutta method.

Part B. Simpson's rule, Trapezoidal rule

Unit 4: Recursion and Arrays

(6+1 Hrs)

Part A.

Problem solving using recursion - Virahanka numbers, Game of Nim. Sample generation, combination generation, permutation generation. Structural recursion: maintaining an ordered set, generating trees and layout of mathematical formulae.

Array Techniques: array order reversals, removal of duplicates from an ordered array, partitioning an array, finding the kth smallest element, largest monotone subsequence. Using parallel arrays and higher dimensional arrays. Algebraic equations - Gaussian elimination

Part B. 8 queens problem, Gauss-Siedel method

Unit 5 Text Processing

(6+1 Hrs)

Part A.

String processing and pattern searching: text line length adjustment, left and right justification of text, keyword searching in text, text line editing, linear pattern search, sublinear pattern search, character-by-character text processing. Some standard text processing problems.

Part B. Big number arithmetic using strings for representing numbers – for example multiplying 100 digit numbers or finding factorial of 100

Text Books

1. *"Problem Solving and Programming Concepts", Maureen Sprankle and Jim Hubbard, Prentice Hall*
2. *"How to Solve It by Computer", R. G. Dromey, PHI*
3. *"Starting out with programming logic and design", 3rd edition, Tony Gaddis, Pearson*

publications

4. *“Programming Logic and Design Introductory”, sixth edition, Joyce, Farrell, Course Technology, CENGAGE Learning*
5. *“An Introduction to programming through C++”, Abhiram Ranade, McGraw Hill Education*

Reference Books

1. *“Art of programming contest”, 2nd edition, Ahmed Shamsul Arefin, ACM Solver Training Series, Gyankosh Prokashoni, Bangladesh*
2. *“Data Structures and Algorithms in Java”, Robert Lafore, Sams Publications*
3. *“Problem Solving with C++”, 7th edition, Walter Savitch, Pearson*
4. *“Programming Pearls”, Second Edition, Jon Bentley, Pearson Education Asia*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1	Apply logical ability to solve the problems using suitable technique.
2	Construct various algorithms based on different data types using fundamental and factoring methods
3	Develop programming methods to solve various numerical and differential equations.
4	Solve problems using recursive and iterative techniques.
5	Implement robust programming solutions for problems such as string processing and pattern searching
6	Select appropriate programming paradigm, data structures and algorithm to solve complex computing problem.

CS20113:: DIGITAL ELECTRONICS AND LOGIC DESIGN

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Number system and Codes **(8+1 Hrs)**

Part A: Introduction, Binary number System, Sign-Magnitude representation, One's and Two's complement representation, Binary arithmetic, 2's complement arithmetic, Octal number System, Hexadecimal number System, BCD code, Excess-3 code, Gray code. Code conversion, Boolean algebra: Basic theorems and properties, K-Map: Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP forms using K- Map.

Part B. Minimization of POS forms using K- Map, Quine-McCluskey Method, Half adder and subtractor, full adder and subtractor.

Unit 2: Combinational & Sequential logic circuits **(9+1 Hrs)**

Part A: Design examples- multiplexers: code converter, demultiplexers, encoder: Priority encoders, decoders: 74138, BCD adder and subtractor, parity generator and checker, Digital comparator: 7485.

Introduction of flip-flop (F.F), 1 bit memory cell, clocked S-R F.F., J-K F.F. race around condition, M/S J-K F.F, flip-flop truth table, excitation table, flip-flop conversion, flip-flop characteristics. T and D F.F, Design of 4 – bit UP-Down ripple counter using J-K flip-flop, Design of Synchronous 3 bit up/down counter, mod-n counters (IC -7490, 7493).

Part B: ALU 74181, BCD-to-7-segment Decoder, adder with look ahead carry generator.

Unit 3 : Design of sequential circuits **(8+1 Hrs)**

Part A: Moore/Mealy M/c's: representation techniques, state diagrams, state tables, state reduction, state assignment, implementation using flip-flops. Applications like sequence generator and detector. Shift register (modes of operation), 4 bit bi-directional universal shift register, application of shift registers (Ring counter, Sequence generator, Johnson's counter.)ASM charts, notations, design of simple controller, multiplexer controller method, RTL notations and implementation.

Part B. Examples on ASM, RTL.

Unit 4: Logic Families **(8+1 Hrs)**

Part A. Characteristics of Digital ICs: Speed, Power dissipation, fan-out, current and voltage parameters, noise margin, operating temperature etc., TTL: Operation of TTL NAND gate, Standard TTL, TTL Characteristics, Active pull-up, Wired-AND, totem pole, open collector, Unconnected Inputs. CMOS Logic: CMOS Inverter, CMOS NAND and NOR, CMOS characteristics. Wired-logic, Unconnected Inputs, Open-Drain Outputs, Comparison of TTL and CMOS, interfacing TTL to CMOS

Part B. Interfacing CMOS to TTL, Tri-state logic: tri-state buffers, inverters, Study of Data sheets of 7400 Series ICs: (Basic and Universal logic gates)

Unit 5: Programmable Logic Devices

(8+1 Hrs)

Part A: Programmable Logic array: Input, Output Buffers, AND, OR, Invert/Non-Invert Matrix, Programming the PLA, Applications of PLAs to implement combinational and sequential logic circuits Introduction to :FPGA, CPLD.

Introduction to VHDL: Modeling Digital systems, modeling languages, modeling concepts.

Part B: Comparison of FPGA and CPLD, VHDL Programs.

Text Books

1. "Modern Digital Electronics", R.P. Jain, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 - 4, 3rd Edition.
2. "A VHDL Primer", J. Bhaskar, Englewood Cliffs, Prentice Hall, 1994, ISBN-13: 978-0131814479, 2nd Edition.

Reference Books

1. "Digital Design", M. Mano, Pearson Education, 2002, ISBN - 81 - 7808 - 555 - 0, 3rd Edition.
2. "Digital Principles and Applications", A. Malvino, D. Leach, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 - 05, 5th Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1	Optimize logical equations using reduction techniques
2	Design different types of code convertors
3	Construct Combinational and Sequential circuits
4	Validate the internal structure of combinational circuits
5	Develop applications of sequential circuits
6	Describe Programmable Logic Devices

CS20216:: PROBLEM SOLVING AND PROGRAMMING

Credits: 01

Teaching Scheme: -Tutorial 1Hr/Week

Prerequisites: Computer Programming

List of Contents

A TERM-WORK containing the record of the following:

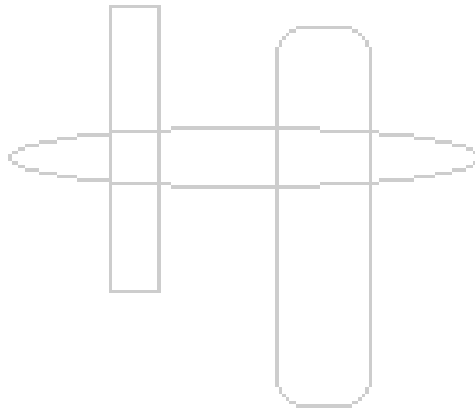
1. Implementing logic to draw basic shapes on computer using some library to take care of drawing details (basic shapes like square, circle, polygons etc., basic animation, projectile motion, bouncing ball etc.)
2. Visualizing simulations (e.g. cosmological simulation) using a library to take care of drawing details
3. Modular arithmetic (finding last k digits of number a^b , finding recurring decimal expansion for rational number a/b , etc.)
4. Factoring methods (generation of pseudo random numbers, smallest divisor of an integer, computing n^{th} Fibonacci number, etc.)
5. Taylor Series expansion (with an emphasis on program specification, testing, proof of correctness, invariants, debugging, good coding practices) taking sine, cosine, e etc. as examples
6. Applications of recursion (drawing recursive pictures, layout of mathematical formulae, finding min-max, etc.)
7. Array based techniques (Gaussian elimination, Gauss-Siedel method, etc.)
8. Text processing and pattern searching (finding LCS in strings, finding duplicate words, spell checker, etc.)

Text Books

1. *"Problem Solving and Programming Concepts"*, Maureen Sprankle and Jim Hubbard, Prentice Hall
2. *"How to Solve It by Computer"*, R. G. Dromey, PHI
3. *"Starting out with programming logic and design"*, 3rd edition, Tony Gaddis, Pearson publications
4. *"Programming Logic and Design Introductory"*, sixth edition, Joyce, Farrell, Course Technology, CENGAGE Learning
5. *"An Introduction to programming through C++"*, Abhiram Ranade, McGraw Hill Education

Reference Books

1. *“Art of programming contest”, 2nd edition, Ahmed Shamsul Arefin, ACM Solver Training Series, Gyankosh Prokashoni, Bangladesh*
2. *“Data Structures and Algorithms in Java”, Robert Lafore, Sams Publications*
3. *“Problem Solving with C++”, 7th edition, Walter Savitch, Pearson*
4. *“Programming Pearls”, Second Edition, Jon Bentley, Pearson Education Asia*



CS20113::DIGITAL ELECTRONICS AND LOGIC DESIGN **FF No. : 654 C**

Credits:-01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

1. Introduction to electronic workbench, solving 2-3 combinational examples.
2. Introduction to 74XX , 54XX ICs and basic component of logic family
3. Diode as a switch.
4. Design logic gates using diodes.
5. Introduction to basic component of logic family-Transistor.
6. Characteristics and configuration of Transistor.
7. Transistor as a switch
8. Design logic gates using Transistor
9. Introduction to basic component of logic family-MOSFET
10. Study of voltage and current characteristics of digital ICs.
11. Design of state transition diagram using ASM.
12. Design of datapath and control path using RTL. (Two examples)

Text Books

1. *"Modern Digital Electronics"*, R.P. Jain, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4, 3rd Edition.
2. *"Integrated Electronics Analog and Digital Circuits and Systems"*, J. Millman, C. Halkias, McGraw Hill, 1988, ISBN 0 – 07 – Y85493 –9, 2nd Edition.

Reference Books

1. *"Digital Design"*, M. Mano, Pearson Education, 2002, ISBN - 81 - 7808 - 555 – 0, 3rd Edition.
2. *"Digital Principles and Applications"*, A. Malvino, D. Leach, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05, 5th Edition.
3. *"Electronic Devices"*, Floyd, Pearson Education, 2001, ISBN 81– 7808–355-8, 5th Edition.

Additional Reading

1. J. Bignell, R. Donovan, *"Digital Electronics"*, DELMAR Thomson Learning, 4th Edition, 2001, ISBN 981 - 240 - 352 – 3.

CS20311:: DATA STRUCTURES

FF No. : 654 B

Credits: 02

Teaching Scheme: - Laboratory 4 Hrs/Week

Prerequisites: Computer Programming

List of Lab Experiments

1. Implementation Quick and Merge Sort.
2. Implementation of Heap Sort
3. Simulation of Hash table using any Collision Resolution and Overflow Handling Technique.
4. Simulation of Recursion using STACK.
5. Implementation of Expression conversion and Evaluation.
6. Simulation of Job Scheduling Using Queue
7. Polynomial Manipulation using Queue
8. Implementation of various operations on singly linked list.
9. Implement Generalized linked list and various operations on it.
10. Implement various operations on Doubly Linked list.
11. Implement various operations on Binary Search tree.
12. Implement Threaded Binary tree and its traversals.
13. Implement Btree, B⁺tree and operation on it.
14. Implement AVL tree and rotations on it.
15. Implement BFS, DFS on Graph.
16. Implement Prim's and Kruskal's Algorithm.
17. Implement Dijkstra's algorithm
18. Implement all pair shortest path problem.

Text Books:

1. *"Fundamentals of Data Structures in C"*, E. Horwitz, S. Sahani, Anderson-Freed, Universities Press, Second Edition, 2008, ISBN 978-81-7371-605-8.
2. *"Data structures using C and C++"*, Y. Langsam, M.J. Augenstein, A.M. Tenenbaum, Pearson Education, Second Edition, 2002, ISBN 81-7808-729-4.

Reference Books :

1. *"An Introduction to data Structures with applications"*, J. Tremblay, P. soresan, TMH Publication, 2nd Edition, 1984. ISBN-0-07-462471-7.
2. *"Fundamentals of Computer Algorithms"*, E. Horwitz , S. Sahani, S. Rajasekaran, Galgotia book source, New Delhi, 2005, ISBN 81-7515-257-5.

CS20313:: DIGITAL ELECTRONICS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

I Combinational Logic Design

1. Verification of Logical Gates and Boolean Algebra.
2. Code converters, e.g. Excess-3 to BCD and vice versa using logical gates.
3. Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Decoder – e.g. 2 bit comparator (IC 74138).
5. BCD adder –using IC 7483

II Sequential Circuit Design (Any six)

1. Conversion of flip-flops. e.g. JK to D, T.
2. Ripple (asynchronous) mod –N counter using J-K F-F.
3. Ripple (asynchronous) mod –N counter using IC 7490.
4. Synchronous 2 bit Up /down counter using JK flip-flop.
5. Sequence generator using JK flip-flop
6. Pseudo random number generator using 74194.(universal shift register)
7. Sequence detector (Moore ckt) using JK flip-flop

8. Sequence detector (Mealy ckt) using JK flip-flop

III ASM (Any one)

1. Simple ASM using multiplexer controller method using Simulator.
2. Design of simple combinational circuit: half adder and subtractor using VHDL language.

Text Books

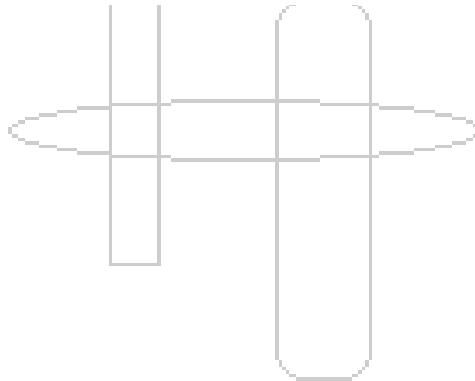
1. *"Modern Digital Electronics"*, R.P. Jain, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4, 3rd Edition.
2. *"A VHDL Primer"*, J. Bhaskar, Englewood Cliffs, Prentice Hall, 1994, ISBN-13: 978-0131814479, 2nd Edition.

Reference Books

1. *"Digital Design"*, M. Mano, Pearson Education, 2002, ISBN - 81 - 7808 - 555 – 0, 3rd Edition.
2. *"Digital Principles and Applications"*, A. Malvino, D. Leach, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05, 5th Edition.

Additional Reading

1. J. Bignell, R. Donovan, *"Digital Electronics"*, DELMAR Thomson Learning, 4th Edition, 2001, ISBN 981 - 240 - 352 – 3.



CS27401:MINIPROJECT

Credits: 02

Guidelines:

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

1. Utilize various algorithms, programming constructs, complexity issues, data structures in building a variety of software solutions.(PO1).
2. Evaluate real world problems and convert them to applications using various programming skills.(P02)
3. Devise data and solution design with sufficient details(PO3)
4. Conduct rapid application development using set of modern tools and frameworks.(PO5)
5. Apply ethical principles in utilizing their skills and knowledge for developing software applications.
(PO8 &PO9)
6. Learn and utilize engineering knowledge, apart from regular curriculum, achieving acumen for self learning and skill development.(PSO2)

FF No. : 654 B

CS24306 PHP MYSQL

Credits: 01

Teaching Scheme: Lab 2 Hours / Week

List of Practical's:

1. Download, Install and Configure Netbeans IDE 8.x/eclipse with XAMPP/WAMP/IIS/Tomcat 6.x and MySQL server for PHP
2. Design a Web page using HTML5 and CSS.
3. Design a PHP page to demonstrate the use of variables, functions, conditional and looping constructs.
4. Design a Web form using PHP and apply validation.
5. Design a Web page to demonstrate the use of session and cookie.
6. Implement user defined exception handling for the Web page.
7. Design a database application in PHP using MySQL.
8. Design a client agent to send an email in PHP
9. Design File upload and download program in PHP
10. Design Web page filters in PHP
11. Design Web page using AJAX and PHP
12. Mini project

Text Books:

1. *"Beginning PHP6, Apache, MySQL® Web Development"* by Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, Wiley Publishing, Inc., ISBN: 978-0-470-39114-3, 2009.

Reference Books:

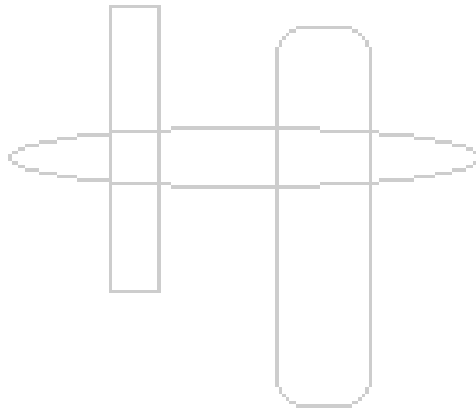
1. *HTML & CSS : The Complete Reference*, Thomas Powell, Tata McGraw -Hill Publication ISBN: 978-0-07-174170-5, Fifth edition, 2010.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Students will be able to simplify the data manipulation using MYSQL database and Flat files.
2. Students will be able to design the web pages using HTML,AJAX and apply CSS formatting styles for look and feel
3. Students will be able to refine dynamic web pages with user defined exception, session, cookies and filters

4. Students will be able to choose suitable IDE for the development of Desktop/Mobile devices web applications
5. Students will be able to deliver realistic and extensible web application product for the benefit of individuals and society
6. Students will be able to practice and utilize PHP programming paradigms and principles of Web development practices



CS24303: C#.NET

Credits: 01

Teaching Scheme: - Lab 2 Hrs/Week

Prerequisites: C programming

List of Practical

1. Study of .Net framework and its important components.
2. Implementing Arrays, Strings and System collections in C#.
3. Implementing Classes, objects, constructors in C#
4. Design a simple C#.net application for calculator.
5. Design a simple C#.net application for notepad/word pad using menu editor.
6. Handling multiple forms in C#.
7. Design Database application.
8. Report Generation and Grid view representation.
9. File handling and Exception Handling in C#
10. Publishing and Deployment of windows application in C#.net/Developing DLLs
11. Mini project.

Text Books

1. *"Programming in C#", E Balagurusamy, Mc Graw Hill Publication, Edition 3, ISBN:9780070702073, 29-JUN-10.*
2. *"C# 4.0 The Complete Reference", Herbert Schildt, Mc Graw Hill Publication, Edition 1, ISBN: 9780070703681, 07-MAY-10.*

Reference Books

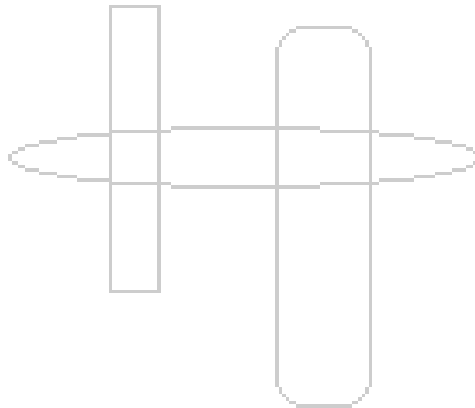
1. *"C# 3.0: A Beginner's Guide", Herbert Schildt, Mc Graw Hill Publication, Edition 2, ISBN: 9780070248946, 25-SEP-08.*
3. *"Ajax: The Complete Reference", Thomas Powell, Mc Graw Hill Publication, Edition 1, ISBN: 9780070248496, 20-MAR-08*

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Analyze problems based on wide variety of programming constructs available in .Net Framework.

2. Design range of applications such as console-based, windows-based and deploy them as end deliverables.
3. Evaluate complex issues in programming and development using object oriented paradigm.
4. Develop component based applications and projects using Visual Studio. Net.
5. Incorporate social issues while designing software solutions to real world problems.
6. Create projects complying with architectural standards and apt use of constructs laid by Microsoft .Net Framework.



CS20401:: Comprehensive Viva Voce

Credits: 2

Guidelines:

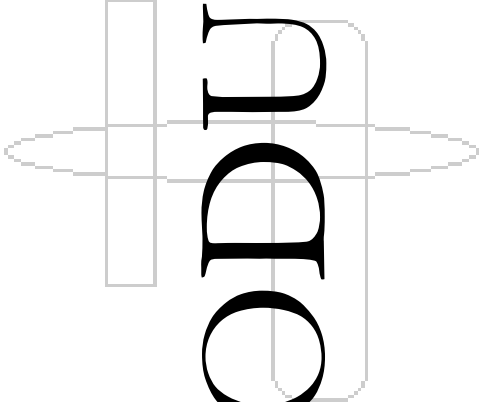
1. Student has to present for CVV exam with assignment file duly signed by subject teacher.
2. Student has to give oral/viva exam based on two core subjects they have studied throughout the semester.
3. Examiners examine students by asking questions based on theory and lab syllabus. Students will prepare two subjects and respective subject expert faculty will ask questions to students.
4. Each student will be evaluated by experienced faculty from department and other reputed engineering colleges.
5. To qualify in CVV, student has to pass both the subjects under CVV.

Course Outcomes

Upon completion of the course, graduates will be able to –

1. Describe properties of data structures and operating modes of microprocessor.
2. Explain stack, queues and microprocessor based system.
3. Design nonlinear data structures, combinational and sequential circuits.
4. Construct applications of trees and sequential circuits.
5. Apply the processor knowledge for future development of microprocessor.
6. Use appropriate data structure and deploy computing solution to solve real world problem.

MODULE IV



FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

S.Y. B. Tech. Structure with effect from Academic Year 2015-16

Module 1V

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS21104	Mathematical Transforms and Applications	S ₁	3	-	1	10	20	5	5	-	60	4
CS20110	Computer Graphics	S ₂	3	-	-	15	20	5	-	-	60	3
CS20105	Principles of Programming Languages	S ₃	3	-	-	15	20	5	-	-	60	3
CS21112	Data Communication	S ₄	2	-	-	10	20	5	5	-	60	2
*CS20113 #CS20114	Digital Electronics and Logic Design Microprocessor and Interfacing	S ₅	3	-	1	10	20	5	5	-	60	4
CS20310	Computer Graphics	P ₁	-	2	-	-	-	-		30	70	1
CS20305	Principles of Programming Languages	P ₂		2						30	70	1

*CS20313 #CS20314	Digital Electronics Microprocessor and Interfacing	P ₃		2						30	70	1
CS27402	Mini Project	MP ₃		2						30	70	2
CS24302 CS24304	ASP.net OR Python	SD ₃		2						70	30	1
*LAB ₄	Technical Writing	LAB ₃								70	30	1
#LAB ₃	General Seminar – 2	LAB ₃								70	30	1
CS20402	Comprehensive Viva Voce	CVV ₁										2
TOTAL			16	14	2							25

* Students will register only in Semester III irrespective of Module

Students will register only in Semester IV irrespective of Module

CS21104:: MATHEMATICAL TRANSFORMS AND APPLICATIONS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Engineering Mathematics- I and Engineering Mathematics - II

Unit 1: (08 Hrs)

Linear Differential equations of higher order

Part A. Homogeneous Linear Differential Equations of Second Order, Higher Order Homogeneous & Non Homogeneous Linear Differential Equations with Constant Coefficients, Complementary solution, particular integral by general method, undetermined coefficients and Variation of Parameters method, Euler – Cauchy Equation, Legendre equation and its solution, system of differential equations by matrix method.

Part B. Electrical circuits and Coupled circuits

Unit 2: (08 Hrs)

Complex Analysis

Part A: Complex differentiation, Analytical functions, Cauchy-Riemann equations, Complex Integration, Cauchy's Integral Theorem and formula, Residue Theorem and applications to Engineering Problems, Power series, Taylor series, Laurent series, Radius of convergence.

Part B: Bilinear Transformations and Conformal mapping.

(08 Hrs)

Unit 3:

Laplace Transform

Part A: Introduction to Laplace Transform and its properties. Laplace Transform of Unit step function, Delta function and periodic function. Inverse Laplace Transform and its evaluation. Laplace Transform of special functions.

Part B: Application of Laplace transform to simultaneous differential equations.

Unit 4:

Fourier Transform

(08 Hrs)

A. Introduction to signals and their properties, mathematical operations on signals, Complex Fourier series and frequency spectrum, Fourier integrals, Fourier cosine and sine transforms, Fourier transforms, properties of Fourier transform, Discrete Fourier transform, Properties.

B. Applications of FT and DFT

(08 Hrs)

**Unit 5:
Z Transform**

A. Introduction to Z Transform, Standard Z transforms, Region of Convergence, properties of Z-Transform, Convolution Theorem for Z transform Inverse Z-Transform by Partial Fraction, Use of standard transform, Inversion Integral Method, Poles and Zeros of the Rational Z-Transform

B. Difference equations by Z transform method.

Text Books

1. "Advanced Engineering Mathematics", Erwin Kreyszig, John Wiley and sons, inc.
2. "Higher Engineering Mathematics", B V Ramana, Tata McGraw-Hill, 2007.
3. "Advanced Engineering Mathematics", R.K. Jain, S.R.K. Iyengar, Narosa Publications.
4. "Signals and Systems", Alan V. Oppenheim (Author), Alan S. Willsky ; 2nd edition, Pearson Education Ltd.

Reference Books

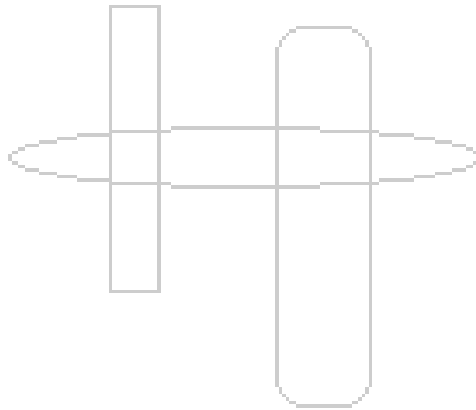
1. Michael D. Greenberg; Advanced Engineering Mathematics; Pearson Education Asia
2. Dennis G. Zill, Michael R. Cullen; Advanced Engineering Mathematics; Narosa Publishing House
3. Peter V. O'Neil; Advanced Engineering Mathematics; 5th edition, Thomson Brooks/Cole.
4. Robert A. Gabel, Richard A. Roberts; Signals and linear systems; John Wiley & sons.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Solve linear differential equations with constant coefficients by selecting suitable solution method
2. Describe the concept and consequences of analyticity and integration of functions of complex variables.
3. Utilize Laplace and Z transforms for BIBO, stability, causality properties of continuous and discrete LTI systems.
4. Summarize analytic, algebraic properties of Fourier, Laplace transforms and physical interpretation of these properties in time and frequency domain.

5. Compute Fourier transform, discrete Fourier transform, Z transform of given continuous and discrete signals respectively.
6. Translate a physical problem into a mathematical model to find solution of the given model.



CS20110::COMPUTER GRAPHICS

FF No. : 654 A

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: C programming

Unit 1 : Basic Concepts

(8 +1 Hrs)

Part A: Graphics Primitives: Introduction to computer graphics, Display adapters, Display modes, Pixel, Frame Buffer, Display file structure, Display file interpreter, Raster scan & random scan displays, Aspect Ratio.

Mathematical foundations: Lines and line representations, Vectors, Intersection of lines, Normalized Device Coordinates.

Scan conversions: DDA and Bresenham's line drawing algorithms and Bresenham and Midpoint circle drawing algorithms, Arcs and Sectors, Aliasing and Antialiasing, Character Generation techniques.

Part B: Display devices, Interactive devices, Data generating devices, Thick lines.

Unit 2 : Polygons and 2D Transformation

(9+1 Hrs)

Part A: Polygons: Introduction, Types of polygons, Inside-outside test of polygon, Polygon filling: Flood fill, Boundary fill, Edge fill, Scan line fill algorithm.

2D Transformations: Introduction, Basic transformations such as- Scaling, Rotation, Translation, Homogeneous coordinates for transformations, Other transformations like – Reflection, Shearing Transformations, Transformations about an arbitrary point, Inverse transformations. Numerical problems on transformation

Part B: Fence fill algorithm, Problems on 2D transformation.

Unit 3 : Segments and Clipping

(9+1 Hrs)

Part A: Segment: Introduction, Segment table, Segment Creation, Closing, Delete and renaming, Image transformation, Display structures used for segment.

Windowing and Clipping: Introduction, viewing transformation, Line clipping: Cohen Sutherland algorithm, Mid-point line clipping algorithm, Polygon clipping: Sutherland Hodgeman algorithm, Weiler Atherton algorithm, Text clipping, Interior and Exterior clipping.

Part B: Liang-Barsky algorithm, Cyrus Beck algorithm.

Unit 4 : 3D Transformations and Projections

(7+1 Hrs)

Part A: 3D Transformations: Introduction, 3D point representation, Left handed system, Right handed system, Basic 3D transformations- Scaling, Rotation, Translation, Matrix representation, Derivation of Rotation matrices along the main axis, Rotation about an arbitrary axis, Reflection transformation with respect to main axes.

Projection: Projection concept, parallel and perspective projections, Viewing parameters, 3D windowing and clipping.

Part B: Reflection transformation with respect to planes, Problems based on 3D transformations

Unit 5 : Hidden Surfaces, Curves and Fractals**(7+1 Hrs)**

Part A: Introduction, Back-face removal algorithm, Z buffers, Painters algorithm, Warnock algorithm, binary space partition method.

Curve generation, Curve continuity, Interpolation, LeGrange interpolating algorithms, Spline curve representation, B Spline Curves, Bezier Curves, Fractals, Hilbert curve, Triadic Koch Curve, Fractal lines.

Introduction to light, Light Illumination models (Diffuse, Ambient, Specular), Point source illumination, Shading Algorithms (Phong, Gourad), Color, RGB Color Model.

Part B: Scan line algorithm for Depth Comparison, Fractal surfaces, CMY and HSV color model

Text Books:

1. "Computer Graphics", D. Hearn, M. Baker, 2nd Edition, Pearson Education, 2002, ISBN 81-7808-794-4.

2. "Procedural Elements for Computer Graphics", D. Rogers, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0-07-047371-4.

Reference Books:

1. "Computer Graphics", S. Harrington, 2nd Edition, McGraw-Hill Publications, ISBN 0 - 07 - 100472 -6.

2. "Computer Graphics Principles and Practice", J. Foley, V. Dam, S. Feiner, J. Hughes, 2nd Edition, Pearson Education, 2003, ISBN 81-7808-038-9.

Course Outcomes:

Upon completion of the course, the graduates will be able to ...

1	Apply mathematics and computer programming to computer graphics applications and problem solutions.
2	Utilize algorithms to draw, fill and perform 2D transformations on basic geometrical shapes.
3	Construct animation based demonstrating system using segments and clipping algorithms.
4	Systematically identify and solve numerical problems of transformations and projections.
5	Interpret the curves, fractals and hidden surfaces for representation of interactive graphics systems.
6	Devise and frame new set of algorithmic principles for efficient solutions useful to computing community.

CS20105::PRINCIPLES OF PROGRAMMING LANGUAGES

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1:

Programming language concepts (8+2 Hrs)

Part A: Language design principles. characteristics of a good programming language. Data and control abstractions in a programming language. Programming paradigms – Imperative/Procedural, Object Oriented, Functional Programming, Logic Programming.

Concepts of Imperative Programming:

Variables: lvalue and rvalue. Memory leak, invalid pointer references. Types and type checking (strong vs weak, static vs dynamic). Binding. scope, local and global variables. Activation Records, Function Calls, Recursion, Parameter passing methods, Stack-Based Storage management. Heap based storage management.

Part B: Compilers and Interpreters. Just In Time interpreters.

Unit 2: (8+2 Hrs)

Object – Oriented Programming (Java-I)

Part A: Encapsulation: Classes and Objects, Methods and Constructors. Information hiding: access modifiers and packages. Static keyword: class variables and instance variables, class methods and instance methods. this and static. Inheritance, Types of inheritance, Constructors in Derived Classes. Overriding & Hiding Fields & Methods. Polymorphism – static and dynamic. Abstract Classes & methods, Final Classes & methods. Interfaces. Exception Handling - exceptions, checked & unchecked exceptions, user-defined exceptions. Similarities and differences between C++ and Java: destructors, access modifiers, inheritance, polymorphism, garbage collection.

Part B: Pointers in C++, Destructors in C++.

Unit 3: (8+1 Hrs)

Object – Oriented Programming (Java-II)

Part A: Multithreading – Thread life Cycle, thread Priority, Thread Methods. Inter-thread Communication. Producer-Consumer using Java. Java I/O – Introduction to Streams, Readers and Writers. File Management / Processing, primitive / Object Data Processing. Java GUI: Applet vs Application. AWT vs Swing. Layout Manager. Components. Label, Button, Choice, List, Event Handling (mouse, key).

Part B: Exception Handling in C++, multiple inheritance in C++.

Unit 4: (8+2 Hrs)

Functional Programming using SCHEME

Part A: Introduction to lambda calculus. The Scheme programming Language: Atoms, Lists, lambda expressions. Functions as first class objects. Control structures, Recursion and continuations, operations on objects, basic input output, Exceptions and

conditions, lazy evaluation and streams.

Part B: Haskell

Unit 5: (8+2 Hrs)

Introduction to SCALA

Part A: What is SCALA. Classes and Objects, Types. Control structures, composition and inheritance. Packages. Pattern matching. Collections API. Working with XML. Actors and concurrency. GUI programming.

Part B: Combining SCALA and JAVA.

Text Books:

“Programming Languages Design and Implementation”, T. W. Pratt, M.V.Zelkowitz, Publications, ISBN 10: 0130276782, 4th Edition

Java: The Complete Reference, Herbert Schildt, TMG Publication, ISBN 9780070636774 , 7th Edition

R Kent Dybvig, The Scheme Programming Language, Fourth Edition, MIT Press, 1990, ISBN 978-0-262-51298-5

Martin Odersky, Lex Spoon, and Bill Venners, Programming in SCALA, Second Edition, 2010. Artima.

Cay Horstmann, Scala for the Impatient, Addison-Wesley, 2012.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1	Analyze real world problems based on their knowledge of various programming paradigms and utilize the appropriate one.
2	Design and develop software artifacts using best practices in programming.
3	Evaluate the given problem and develop solution using object oriented programming paradigm.
4	Create solutions to problems using Integrated Development Environments and modern software tools.
5	Utilize their knowledge of programming paradigms to continue learning newer programming languages and constructs.
6	Select and utilize appropriate programming constructs and paradigms to devise effective software solutions.

CS21112 :: DATA COMMUNICATION

Credits: 02

Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites:

Unit 1: (6+1 Hrs)

Introduction to Electronic Communication

Part A: The importance of Communication, Elements of communication system, Types of electronics communication, Electromagnetic spectrum, Bandwidth, Signal Types, Noise: internal, External, Noise calculation, Nyquist theorem Shannon-Hartley theorem.

Part B: Survey of communication applications. Numerical based on Shannon-Hartley theorem, Bandwidth calculation.

Unit 2: (6+1 Hrs)

Modulation Techniques

Part A: Principles of Amplitude Modulation, Modulation index and percentage of modulation, AM power distribution, Single sideband communication, AM transmitters and Receivers. Phase modulation.

Part B: FM vs. AM, FM vs. PM, AM vs. PM, Numerical based on AM, FM.

Unit 3: (5+2 Hrs)

Multiplexing and Communication

Part A: Introduction, FDM, TDM, WDM, CDMA, Frequency modulation principles, sideband and modulation index. Pulse code modulation, Delta modulation, Adaptive delta modulation, Differential PCM, PAM,

Part B: Parallel transmission, serial transmission.

Unit 4: (5+1 Hrs)

Digital Modulations

Part A: Sampling theorem, Modems, Null modems, ASK, PSK, FSK, QPSK, Line coding schemes. Frequency hopping spread spectrum, Direct sequence spread spectrum. Encoding Schemes: NRZ, NRZI, Unipolar, Bipolar, Manchester, Differential Manchester.

Part B: Cellular Telephone System, Examples on Encoding, decoding.

Unit 5:

(6+1 Hrs)

Transmission and Propagation

Part A: Transmission modes, Antenna Fundamentals, Radio Frequency Wave propagation, Attenuation and distortion sources with examples. Dipole antenna, sectorized antenna. Error detection and correction: CRC, Hamming code, Checksum, Block coding

Part B: Transmission media wired and wireless, Microwave antenna.

Text Books

1. *"Electronic Communication Systems", by George Kennedy, Bernard Davis, Tata McGraw Hill Publication, ISBN 0-07-463682-0, Edition 4th*
2. *"Data Communications and Networking" by Behrouz Forouzan, McGraw Hill Publication, ISBN 0-07-063414-9, Edition 4th*

Reference Books

1. *"Communication Electronics- Principles and Applications", by Frenzel, Tata McGraw Hill Publication, ISBN 0-07-048398-1, Edition 3rd*
2. *"Principles of Communication Systems", by Herbert Taub and Donald Schilling, McGraw Hill Publication, ISBN 0-07-062955-2, Edition 2nd.*

Additional Reading

1. *"Computer Networks", by Andrew S. Tenenbaum, Prentice Hall of India, ISBN 81-203-2175-8, 4th Edition.*
2. *"Data and Computer Communications", by Stallings W, Prentice Hall of India Pvt. Ltd., 2002, ISBN 81-203-2067-0, Sixth Edition.*
3. *"Digital and Analog Communication Systems", by Shanmugam K, John Wiley & Sons (Asia) Pvt. Ltd. ISBN 9971-51-146-0.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Apply modulation and multiplexing techniques to optimize the channel requirements.
2. Compute the bandwidth, throughput, channel efficiency for different multiplexing and modulation techniques.
3. Interpret the given data in problems related to Data communication to generate solutions for it.
4. Relate the knowledge and techniques learned in course with the components used in the real world communication technology.
5. Deliver the communication requirements and mathematical approaches for development in the communication technology.
6. Grasp the theory of data communication to equipped oneself with the knowledge of communication.

FF No. : 654 A

CS20114:: MICROPROCESSOR AND INTERFACING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Digital Electronics and Logic Design.

Unit 1: (8+1 Hrs)
8086 Microprocessor

Part A: Introduction to 80x86 microprocessor, Internal Architecture, Generation of physical address, Minimum & Maximum Mode, Ready and Reset pin significance ,study of 8086 supporting chips 8282(Latch), 8284(Clock Generator), 8286(Transceiver), 8288(Bus Controller). Timing Diagram Read Write Machine Cycles, Real Mode, General Purpose Instructions.

Part B: Instruction Set

Unit 2: (8+1 Hrs)
Assembly Language Programming & Interrupt Structure

Part A: Address Translation, Addressing Modes, Introduction to Assembly Language Programming, Examples on Programming. Interrupt Structure , Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR ,NMI , Interrupt Response, Execution of an ISR, Priority of Interrupts.

Part B: Examples on Assembly Language Programming

Unit 3: (8+1 Hrs)
Interfacing with 8086 – I

Part A: 8259 (Programmable Interrupt Controller): Features, Block Diagram, Control & status registers, Interfacing & Programming ,

8255 (Programmable peripheral interface 8255)-block diagram, control word, interfacing ADC (Successive Approximation Method) , DAC (R – 2R ladder Network.)

Part B: Interfacing of stepper motor, seven segment display, (8255)

Unit 4: (8+1 Hrs)
Interfacing with 8086 – II

Part A: 8253/8254 –(Programmable Interval timer/counter) block diagram, control word & interfacing, Mode0, Mode1, Mode3 of timer,

8251(USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming.

Part B: Programming of 8251, Programming of Timer Mode 1, Mode2, Mode 3, Mode4, Mode5

Unit 5: (8+1 Hrs)

DMA Controller and NDP Co processor

Part A: Concept of DMA, 8237 DMA Controller: Features, Block Diagram & programming detail.

8087(NDP) - Features, Block Diagram, Control & status registers, typical Instruction set & programming detail.

Part B: Programming of 8237 and 8087(NDP).

Text Books

1. *"Microprocessors and Interfacing"*, Douglas Hall, Tata McGraw Hill Publications, ISBN 0-07-025742-6, 2nd Edition.
2. *"Advanced 80386, programming techniques "*, James Turley, Tata McGraw Hill Publications, ISBN – 0-07-881342-5

Reference Books

1. *"Advanced MS DOS Programming"*, Ray Duncan, BPB Publications ISBN 0 – 07 – 048677 – 8, 2nd Edition.
2. *"Microprocessor and Peripheral Handbook"*, INTEL –VOL I

Additional Reading

1. *"Assembly Language Programming"*, Peter Abel, Pearson Education Publications, ISBN 10:013030655, 5th Edition.
2. *Intel Pentium Manual.*

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Study different operating modes of Microprocessor
2. Design Microprocessor systems using peripheral components.
3. Develop experiments to study individual Microprocessor interconnects
4. Use simulation tool and assembler for optimising system design
5. Apply the processor knowledge to develop Microprocessor Systems
6. Design and deploy computing solution to Real World Problems

CS20114:: MICROPROCESSOR AND INTERFACING

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

A TERM-WORK containing the record of the following:

Assignments :

1. Understanding of different Memory Models
2. List various assembler directives,
3. Understand concepts of editor, assembler, linker, loader.
4. 8086 assembly language programming, to understand the basic concepts of various functions(01,02,08,09,0A) of INT 21h
5. List various debugging commands.
6. Interface 8086 microprocessor with 4KB RAM in minimum mode. Apply memory banking. Draw memory address map and explain address decoding logic.
7. Interface 8086 microprocessor with 16KB ROM in maximum mode. Draw memory address map and explain address decoding logic.
8. Design specified time delay (delay time calculation).
9. Near, Far procedures (string example).
10. Use of string instructions
11. Study of Mother Board

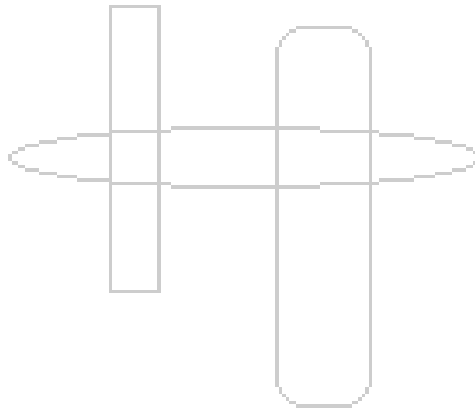
Note: Students should perform vi & vii assignments on drawing sheet

Text Books

1. *“Microprocessors and Interfacing”, Douglas Hall, Tata McGraw Hill Publications, ISBN 0-07-025742-6, 2nd Edition.*
2. *“Assembly Language Programming”, Peter Abel, Pearson Education Publications, ISBN 10:013030655, 5th Edition*

Reference Books

1. *“Advanced MS DOS Programming”, Ray Duncan, BPB Publications ISBN 0 – 07 – 048677 – 8, 2nd Edition.*
2. *“Microprocessor and Peripheral Handbook”, INTEL –VOL I*



CS21104:: MATHEMATICAL TRANSFORMS AND APPLICATIONS**Credits:** 01**Teaching Scheme:** - Tutorial 1 Hr/Week**Prerequisites::** Syllabus covered unit-wise in theory lectures.**List of Contents**

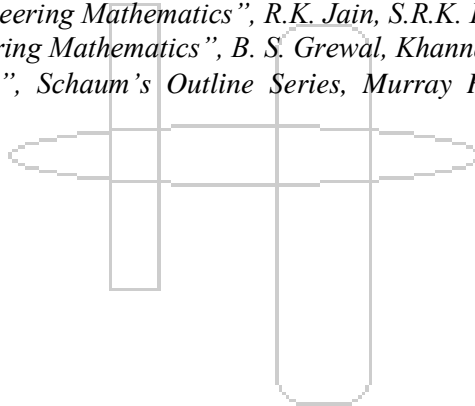
In this module students will work on problems to practice and apply methods introduced in the theory lectures. Discussions of problems in small groups is always encouraged and facilitated. Students are asked to submit weekly home work assignments and provide them immediate feedback and support materials.

- | | |
|------------------------|---|
| Tutorial No. 1: | Summary on higher order linear differential equations, solution of homogeneous and non homogeneous equations, complementary solution. |
| Tutorial No. 2: | Summary on particular solution by method of variation by parameters
Method of undetermined coefficients and problems solving. |
| Tutorial No. 3: | Summary on Euler – Cauchy and Legendre Equation, simultaneous equations and problems solving. |
| Tutorial No. 4: | Summary on Functions of complex variables, Differentiation of functions of complex variables, Analytic functions, Harmonic functions, Harmonic conjugate. |
| Tutorial No. 5 | Summary on Integration of functions of complex variables, integration along a path, Cauchy's theorem, Cauchy's integral formula, Cauchy's residue theorem and problems solving. |
| Tutorial No. 6: | Power Series, Taylor Series, Laurent Series |
| Tutorial No. 7: | Summary on Laplace transform, Laplace transform of standard functions, Properties of Laplace and problems solving. |
| Tutorial No. 8: | Summary on Properties of Laplace transform, Laplace transform of Unit step function, Dirac Delta function, Periodic functions and problems solving. |
| Tutorial No. 9: | Summary on Inverse Laplace transform, properties of inverse Laplace transform, solution of differential equations by Laplace transform method and problems solving. |

- Tutorial No. 10:** Summary on Fourier series, Complex form of Fourier series, Fourier integral representation and problems solving,
- Tutorial No. 11:** Summary on Fourier transform, Sine transform, Cosine transform and corresponding inverse and problems solving.
- Tutorial No. 12:** Summary on Discrete Fourier Transform, properties and problems solving.
- Tutorial No. 13:** Summary on Summary on Z transform, properties of Z transform, inverse Z transform, methods of solution and problems solving,
- Tutorial No. 14:** Poles and Zeros of the Rational Z-Transform, Solution of difference equations by Z transform.

Text Books

1. *“Advanced Engineering Mathematics”, Erwin Kreyszig, John Wiley and sons, inc.*
2. *“Advanced Engineering Mathematics”, R.K. Jain, S.R.K. Iyengar, Narosa Publications.*
3. *“Higher Engineering Mathematics”, B. S. Grewal, Khanna Publishers.*
4. *“Vector Analysis”, Schaum’s Outline Series, Murray R Spiegel, Seymour Lipschutz, Dennis Spellman*



CS20314:: MICROPROCESSOR AND INTERFACING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

Group A: - (Any 6)

1. Study of 8086 Architecture and Execution of sample programs.
2. Write 8086 Assembly language program to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
3. Write 8086 ALP to perform block transfer operation. (Don't use string operations) Data bytes in a block stored in one array transfer to another array. Use debugger to show execution of program.
4. Write 8086 ALP to find and count negative number from the array of signed number stored in memory.
5. Write 8086 Assembly language program (ALP) to arrange the numbers stored in the array in ascending as well as descending order. Assume that the first location in the array hold the number of elements in the array and successive memory location will have actual array elements. Write a separate subroutine to arrange the numbers in ascending and descending order. Accept a key from the user.
 - a. If user enters 0, Arrange in ascending
 - b. If user enters 1, Arrange in descending
6. Write 8086 Alp to convert 2_digit HEX number into equivalent BCD number.
7. Write 8086 ALP to convert 2_digit BCD number into equivalent HEX number.

Group B: - (Any 6)

1. Write 8086 Assembly language program (ALP) for following operations on the string entered by the user.
 - a. Concatenation of two strings
 - b. Find number of words, characters
2. Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC.
3. Write 8086 ALP to interface DAC & generate following waveforms on oscilloscope. Comment on types of DAC's and write detailed specifications of the DAC used
 - i) Square wave -- Variable Duty Cycle & frequency.
 - ii) Stair case wave
 - iii) Triangular wave
4. Write 8086 ALP to rotate a stepper motor for
 - a. one clockwise rotation
 - b. one anti clockwise rotation

Write routines to accelerate and de-accelerate the motor

Modify your program to rotate stepper motor for given angle and given direction.

5. Write 8086 ALP to program 8253 in Mode 0 . Generate a square wave with a pulse of 10 mS.
6. Write 8086 ALP to initialize 8279 & to display characters in right entry mode. Provide also the facility to display "SECOMP"/.
 - a. Character in left entry mode
 - b. Rolling Display
 - c. Flashing Display
7. Perform an experiment to establish communication between two USART's. Initialize USART-A in asynchronous transmitter mode and interface USART-B by initializing it in asynchronous receiver mode.

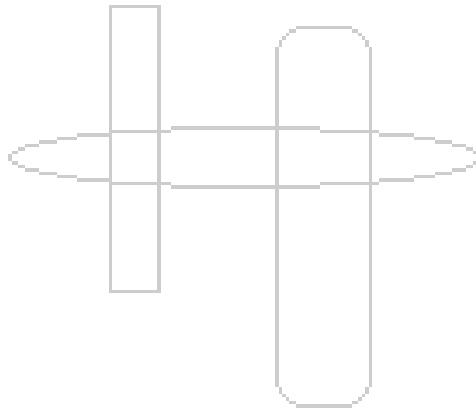
Note: - Students should perform any 6 assignments from group A and any 6 assignments from group B.

Text Books

1. *"Microprocessors and Interfacing", Douglas Hall, Tata McGraw Hill Publications, ISBN 0-07-025742-6, 2nd Edition.*
2. *"Assembly Language Programming", Peter Abel, Pearson Education Publications, ISBN 10:013030655, 5th Edition*

Reference Books

1. *“Advanced MS DOS Programming”, Ray Duncan, BPB Publications ISBN 0 – 07 – 048677 – 8, 2nd Edition.*
2. *“Microprocessor and Peripheral Handbook”, INTEL –VOL I*



CS20305::PRINCIPLES OF PROGRAMMING LANGUAGES

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C

List of Practical

1. Write a C++ program to implement the concept of objects, classes, constructors, destructors.
2. Write a C++ program to implement the concept of Inheritance and polymorphism.
3. Write a C++ program to use the concept of generic programming (generic functions and generic classes)
4. Write a JAVA program to implement the concept of class, constructor, instance variable & class variable.
5. Write a JAVA program to implement the concept of inheritance, interface & package.
6. Write a Java program to implement the concepts of static polymorphism (function overloading) and dynamic polymorphism (using function overriding)
7. Write a JAVA program to use multithreading
8. Write a JAVA program for file handling.
9. Write a Java program with Graphical User Interface.
10. Write programs to implement list operations in Scheme programming language.
11. Write program in Scala to implement basic programming constructs
12. Mini project.

Text Books

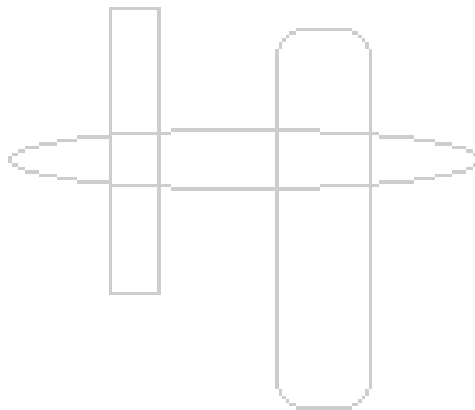
1. *"Object Oriented Programming with C++", E. Balaguruswamy, Tata McGraw-Edition*
2. *"Java: The Complete Reference", Herbert Schildt, TMG Publication, ISBN Hill Publishing Company Ltd, New Delhi ISBN 0 - 07 - 462038 – X, 49780070636774 , 7th Edition*

Reference Books

1. *"The Waite Group's Object oriented Programming in C++", R. Lafore, Galgotia Publications, ISBN 81-7515-269-9, 3rd Edition*
2. *"Core Java 2 Volume – I ", Cay S Horstmann, Gary Cornell, Pearson Education, ISBN 9788131719459, 8th Edition*
3. *R Kent Dybvig, The Scheme Programming Language, Fourth Edition, MIT Press, 1990, ISBN 978-0-262-51298-5*
4. *Martin Odersky, Lex Spoon, and Bill Venners, Programming in SCALA, Second Edition, 2010. Artima.*

Additional Reading

1. *"Core Java Volume .II " Cornell G, Horstmann C S, Sunsoft Press, ISBN 81-7808-018-4*



Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C

List of Practical

1. Write a Program to implement DDA and Bresenham Line drawing algorithm.
2. Write a Program to implement Bresenham's Circle drawing algorithm.
3. Write a Program to implement Polygon fill algorithm.
4. Write a Program to implement Scaling, rotation and translation of a 2D object.
5. Write a program to achieve animation by using segmentation.
6. Write a Program to implement Cohen Sutherland line clipping algorithm.
7. Write a Program to implement Polygon clipping algorithm.
8. Write a Program to implement Scaling, reflection about planes and axes of a 3D object.
9. Write a Program to draw a Koch curve, fractal line and surface.

Text Books

1. "Computer Graphics", S. Harrington, 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0-07-100472-6.
2. "Procedural Elements for Computer Graphics", D. Rogers, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0-07-047371-4.

Reference Books

1. "Computer Graphics Principles and Practice", J. Foley, V. Dam, S. Feiner, J. Hughes, 2nd Edition, Pearson Education, 2003, ISBN 81-7808-038-9.
2. "Computer Graphics – C Version", D. Hearn, M. Baker, 2nd Edition, Pearson Education, 2002, ISBN 81-7808-794-4.

CS27402:MINIPROJECT

Credits: 02

Guidelines:

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

1. Utilize various algorithms, programming constructs, complexity issues, data structures in building a variety of software solutions.(PO1).
2. Evaluate real world problems and convert them to applications using various programming skills.(P02)
3. Devise data and solution design with sufficient details(PO3)
4. Conduct rapid application development using set of modern tools and frameworks.(PO5)
5. Apply ethical principles in utilizing their skills and knowledge for developing software applications.(PO8 &PO9)
6. Learn and utilize engineering knowledge, apart from regular curriculum, achieving acumen for self learning and skill development.(PSO2)

CS24302:: ASP.NET

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C#.NET or VB.NET

List of Practical

1. Design simple web application using ASP.NET.
2. Design web application with different validations.
3. Design on line database application.
4. Design data report application.
5. Design web application for uploading files on web.
6. Design AJAX application.
7. Design localized web application.
8. Design WPF browser application.
9. Authentication and authorization in asp..
10. Deployment and publishing web sites.
11. Mini project.

Text Books

1. *“Beginning ASP.NET 3.5: In C# and VB”*, Imar Spaanjaars, Wrox publication, ISBN: 978-0-470-18759-3, March 2008.
2. *“ASP.NET 3.5: A Beginner's Guide”*, William Sanders, McGraw Hill Publication, ISBN: 007159194X / 9780071591942, September 2008.

Reference Books

1. *“ASP.NET AJAX Programmer's Reference: with ASP.NET 2.0”*, Shahram Khosravi, Wrox Publication, ISBN: 978-0-470-10998-4, Sept.2007.
2. *“Professional ASP.NET 2.0”*, B. Evjen, S.Hanselman, F.Muhammad, S. S. Sivakumar, D. Rader, Wrox Publication, ISBN: 978-0-7645-7610-2, Nov. 2005.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Analyze problems based on wide variety of programming constructs available in .Net Framework.
2. Design web applications with rich UI and bug free experience using the ASP.NET standard Validation controls.
3. Evaluate problems and alternative web solutions using ASP.Net for a wide variety of business and organizational contexts.
4. Build projects complying with architectural standards laid by Microsoft .Net Framework.
5. Acquire hands-on experience on .Net Technologies to hold responsible positions in government and industry sectors.
6. Develop foundation of web applications.

CS24304:: Python

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C programming.

List of Practical

1. Syntax basics, Arithmetic/String Operations, Input/Output
2. Control Flow constructs: If-else, Relational and Logical Operators
3. Iteration: While loop, For loop
4. Collections: Lists, Tuples
5. Collections: Sets, Dictionary
6. Functions and Modules: sys, math, time
7. File Handling: Data streams, Access modes, Read/Write/Seek
8. Exception handling: hierarchy, raise, assert
9. OOP: Classes, Objects
10. GUI programming: TkInter

Text Books

1. *Exploring Python*, Timothy Budd, Mc Graw Hill Publication, ISBN:9780073523378, August 2010.
2. *Beginning Python*, Peter C. Norton, Alex Samuel, Dave Aitel, Eric Foster-Johnson, Leonard Richardson, Jason Diamond, Aleatha Parker, Michael Roberts, ISBN: 978-0-7645-9654-4, August 2005.

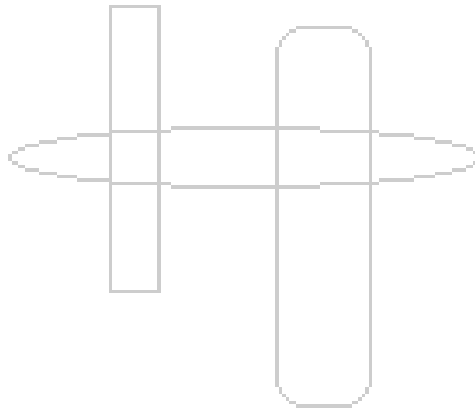
Reference Books

1. *Python: Create - Modify - Reuse*, James O. Knowlton, Wrox Publication, ISBN: 978-0-470-25932-0, July 2008.
2. *Professional Python Frameworks: Web 2.0 Programming*, Dana Moore, Raymond Budd, William Wright, Wrox Publication, ISBN: 978-0-470-13809-0, October 2007.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Formulate a concrete problem definition for a given problem with consideration for Python specific data structures and capabilities.
2. Develop functional, reliable and user friendly Python programs for given problem statement and constraints.
3. Evaluate various solution alternatives in terms of correctness, space and time complexity and usability.
4. Utilize state-of-the-art tools and techniques for efficient software development.
5. Defend the efficiency of proposed solution in terms of the choice of data structures and algorithms.
6. Apply the problem solving and programming skills learned through the course for tackling relevant pressing issues in public and private sectors.



FF No. : 654 B

CS20402: Comprehensive Viva Voce

Credits: 2

Guidelines:

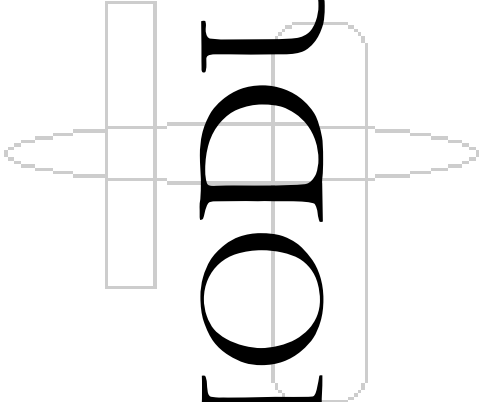
1. Student has to present for CVV exam with assignment file duly signed by subject teacher.
2. Student has to give oral/viva exam based on two core subjects they have studied throughout the semester.
3. Examiners examine students by asking questions based on theory and lab syllabus. Students will prepare two subjects and respective subject expert faculty will ask questions to students.
4. Each student will be evaluated by experienced faculty from department and other reputed engineering colleges.
5. To qualify in CVV, student has to pass both the subjects under CVV.

Course Outcomes

Upon completion of the course, graduates will be able to –

1. Describe programming to computer graphics applications and operating modes of microprocessor.
2. Explain polygons, 2-D transformation and microprocessor based system.
3. Design animation based system, combinational and sequential circuits.
4. Construct applications of 3-D transformation, projections and sequential circuits.
5. Apply the principles of programming knowledge for future development of microprocessor and applications.
6. Use appropriate set of algorithmic principles and deploy computing solution to solve real world problem.

MODULE V



FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

T.Y. B. Tech. Structure with effect from Academic Year 2015-16

Module V

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS30101	Operating Systems	S ₁	3	-	1	10	20	5	5	-	60	4
CS30116	Computer Networks	S ₂	3	-	-	15	20	5	-	-	60	3
CS30105	Theory of Computation	S ₃	3	-	-	15	20	5	-	-	60	3
CS31113	Microprocessors and Microcontrollers	S ₄	2	-	-	15	20	5	-	-	60	3
CS31115	Advanced Data Structures	S ₅	2	-	1	10	20	5	5	-	60	3
CS30303	Operating Systems	P ₁	-	2	-	-	-	-		30	70	1
CS30316	Computer Networks	P ₂		2						30	70	1
CS31313	Microprocessors and Microcontrollers	P ₃		2						30	70	1

	ers											
CS37401	Mini Project (T.Y. Semester I) Irrespective of Module	MP ₅		4						30	70	2
PD ₁	Institute Level Elective	* PD ₁		2						30	70	1
CS30401	Comprehensi ve Viva Voce	CVV ₃								30	70	2
CS37301	Seminar(T.Y. Semester I) Irrespective of Module	SM ₁		4						30	70	2
CS37302	Project Stage 1 (T.Y. Semester II) Irrespective of Module	PS ₁		4						30	70	2
TOTAL			14	16	2							26

CS30101:: OPERATING SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures and Algorithms, Computer Organization.

Unit 1: (8+1 Hrs) **Introduction to OS**

Part A: Introduction to OS: What is OS, Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls.

Types of OS: Batch, Multiprogramming, Time sharing, Parallel, Distributed & Real-time OS.

Structures of OS: Monolithic, Layered, Ringed, Virtualization-Virtual Machines, Hypervisor, Exokernels, Client-server model, Microkernels.

Shell: Linux commands and shells, shell programming, AWK programming.

Introduction to Mobile OS: Architecture & Overview of Android OS.

Part B: Overview of Linux and Windows 2000 architecture

Unit 2: (8+1 Hrs) **Process Management**

Part A: Processes: Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control

Threads: Multithreading models, Thread implementations – user level and kernel level threads.

Symmetric Multiprocessing.

Concurrency: Issues with concurrency, Principles of Concurrency

Mutual Exclusion: H/W approaches, S/W approach, OS/Programming Language support: Semaphores, Mutex, Monitors. **Classical Problems of Synchronization:** Readers-Writers problem, Producer Consumer problem, Dining Philosopher problem.

Part B: IPC: Shared memory, Message passing. Sleeping Barber problem.

Unit 3: (8+1 Hrs) **Scheduling and Deadlock**

Part A: Uniprocessor Scheduling: Types of Scheduling - Preemptive, Non-preemptive, Long-term, Medium-term, Short-term, Scheduling Criteria.

Scheduling Algorithms: FCFS, SJF, RR, Virtual Round Robin, Priority

Multiprocessor Scheduling: Granularity, Design Issues, Process Scheduling

Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery.

Part B: Thread Scheduling, Real Time Scheduling.

Unit 4:

(8+1 Hrs)

Memory Management

Part A: Memory Management concepts: Memory Management requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy Systems. Placement Strategies: First Fit, Best Fit, and Worst Fit, Fragmentation, Swapping, Paging, Segmentation, Address translation.

Virtual Memory: Concepts, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, VM with Segmentation.

OS policies for Virtual Memory: Fetch, Placement, Replacement, Resident Set management, Cleaning Policy, Load Control.

Page Replacement Policies: FIFO, LRU, Optimal, Random. Optimization with Page Buffering. Swapping issues: Thrashing.

Part B: VM with combined paging and segmentation, Working Set Model.

Unit 5:

(8+1 Hrs)

I/O and File Management

Part A: I/O management: I/O Devices - Types, Characteristics of Serial and Parallel devices, OS design issues for I/O management, I/O Buffering.

Disk Scheduling: FCFS, SCAN, C-SCAN, SSTF.

File Management: Concepts, File Organization, File Directories, File Sharing. Record Blocking, Secondary Storage Management, Free Space management, Security.

File System: Structure, Implementation, Memory mapped files, Special Purpose File Systems

Case study: Process Management, Concurrency, Scheduling, Memory Management, I/O Management, File Management(VFS) in LINUX

Part B: Organization of I/O functions, Disk Caches.

Text Books

1. *“Operating Systems”, Stalling William, Pearson Education, ISBN: 0-13-031999-6, 4th Edition.*
2. *“Operating System Concepts”, Silberschatz A., Galvin P., Gagne G., John Wiley and Sons, ISBN:9971-51-388-9, 6th Edition.*

Reference Books

1. *“Modern Operating Systems”, Tanenbaum Andrew S., Prentice Hall India, ISBN:81-7808-447-3, 2nd Edition.*
2. *“Unix Concepts and Applications”, Das Sumitabha, Tata McGraw Hill, ISBN: 0-07-053475-6, 3rd Edition.*

Additional Reading

1. *“Operating System Principles”, Silberschatz A., Galvin P., Gagne G, John Wiley and Sons, ISBN: 9812-53-176-9, 7th Edition.*
2. *“Design of the Unix Operating System”, Bach Maurice, Pearson Education, ISBN: 81-7808-731-6, 1st Edition.*
3. *“Unix Shell Programming”, Yashavant Kanetkar, BPB Publications, ISBN: 81-7029-753-2, 1st Edition.*
4. *“Unix And Shell Programming”, Forouzan B. A., Gilberg R. F., Australia, Thomson Brooks Cole, ISBN: 981-243-127-6 , 1st Edition.*
5. *“The Symbian OS Architecture Sourcebook: Design and Evaluation of a Mobile Phone OS”, Ben Morris, John Wiley & Sons, ISBN: 978-0-470-01846-0, 2007.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify the mechanisms and strategies of an Operating System in order to solve real world problems.
2. Develop solutions based on Operating system concepts in various contexts.
3. Automate the administrative tasks by means of modern tools in Operating System.
4. Examine the functions of a contemporary Operating system with respect to convenience, efficiency and the ability to evolve
5. Engage in a team towards development of a prototype Operating System.
6. Construct solutions to real world problems by applying the standard techniques used by Operating Systems for similar issues.

CS30116:: COMPUTER NETWORKS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Communication.

Unit 1: (8+1 Hrs)

Introduction to Computer Networks and Logical Link Control

Part A: LAN, MAN, WAN, PAN, Internet, Intranet, Internetwork, Protocol Hierarchy, Design Issues for the Layers, OSI Model, TCP/IP Model. Network Devices: Hub, Switch, Router and Access Point, Data Link Layer Design Issues: Services Provided to the Network Layer, Framing, Error Control and Flow Control. Error Detection and correction codes: Hamming Code and CRC. Elementary DLL Protocols: Unrestricted Simplex, Stop-and-Wait and Noisy channel. Sliding Window protocols, HDLC Protocol.

Part B: Point-to-Point Protocol (PPP), MPLS, Bridges, Gateways, Network Cables

Unit 2: (8+1 Hrs)

Medium Access Control

Part A: Channel allocation: Static and Dynamic allocation, Multiple Access Protocols: Pure ALOHA, Slotted ALOHA, CSMA, WDMA, Ethernet: Cabling, MAC Sub-layer protocol: DIX and IEEE 802.3 Frame Formats, Collision Detection, Binary Exponential Back-off Algorithm, Switched Ethernet, Fast Ethernet, Wireless 802.11a/b/g/n LANS, MACA, Broadband wireless: 802.16

Part B: Gigabit Ethernet, Layer-II Switch and Bluetooth

Unit 3: (9+2 Hrs)

Network Layer

Part A: Design Issues, Switching Techniques: Circuit and Packet Switching, Connectionless and Connection-oriented Services, Virtual Circuit and Datagram Subnets. Autonomous system. Organization of the Internet: ISP, Content Providers, Routers, Routing versus forwarding, Routing Algorithms: Optimality principle, shortest path routing, flooding, Distance Vector routing, link state routing, hierarchical routing. Congestion Control and QOS: General Principles, Congestion prevention policies, Load shading, Jitter Control, Quality of Service, Internetworking. Network layer Protocols: ARP, RARP, IP protocol, IPV6, ICMP, Unicast Routing Algorithms: RIP, OSPF, BGP.

Part B: Broadcast and Multicast routing, Routing for mobile hosts, IGMP, Mobile IP, VLAN

Unit 4: (8+1 Hrs)

Transport Layer

Part A: Services and service primitives, Elements of Transport protocol: Addressing, Connection establishment and release, flow control and buffering, Multiplexing, Crash recovery, UDP: Introduction, TCP: Introduction, Model, protocol, header, connection establishment and release, connection management, Transmission policy, congestion control, timer management, RPC, Transport layer in Mobile network.

Part B: Real Time Streaming Protocol RTSP, RTP, RTCP

Unit 5:

(7+1 Hrs)

Application Layer

Part A: Domain Name System (DNS), Naming and Address Schemes, DNS servers, E-mail: MIME, SMTP and POP3. Remote login, File Transfer Protocol (FTP), SNMP, DHCP and BOOTP. CDN, Working of Bit Torrent, Cloud computing: Architectures and working principle.

Part B: World Wide Web, HTTP1.0, HTTP 1.1, Web Caching, Wireless Web.

Text Books

1. *"Computer Networks", by Tanenbaum A. S., Pearson Education , 2008, ISBN-978-81-7758-165-2, 4th Edition,*
2. *"Computer Networking- A Top-Down Approach", by James F. Kurose and Keith W Ross, Person Education, ISBN- 978-81-317-9054-0, 5th Edition.*

Reference Books

1. *"Data Communications and Networking", by Forouzan B. A, Tata McGraw-Hill Publications, 2006, ISBN-0-07-063414-9, 4th edition.*
2. *"Communication Networks- Fundamental Concepts and Key Architectures", by Leon-Garcia-Wadjaja, Tata McGraw-Hill Publications, ISBN-978-0072463521.*

Additional Reading

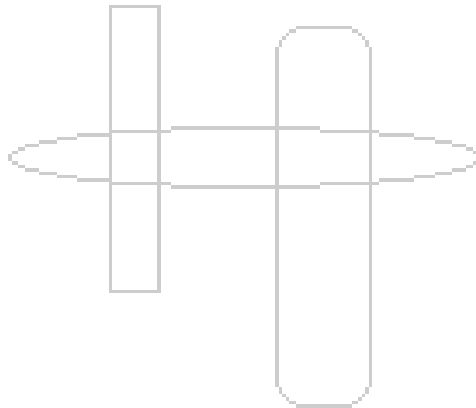
1. *"Computer Networks and Internet", by Comer D., Pearson Education, ISBN-81-297-0330-0, 2nd Edition.*
2. *"Computer Networks- A Systems Approach", by Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, ISBN-978-81-312-1045-1, 4th Edition.*
3. *"Cloud Application Architectures: Building Applications and Infrastructure in the Cloud", George Reese, O'Reilly Publication, 2009, ISBN 978-0-596-80163-2. 1st Edition.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency.
2. Design mechanisms to demonstrate LAN behavior utilizing network architecture, protocols, and network components.

3. Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols.
4. Illustrate applications of Computer Network capabilities, selection and usage for various sectors of user community.
5. Demonstrate sustainable engineering practice indicating the scientific purpose and utility of communication frameworks and standards.
6. Develop Client-Server architectures and prototypes by the means of correct standards and technology.



CS30105:: THEORY OF COMPUTATION

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures.

Unit 1: (8+1 Hrs)
Automata Theory

Part A: Introduction to Finite Automata, Structural Representations, Automata and Complexity, Central Concepts to Automata Theory: Alphabets, Strings, Languages and Problems, Deterministic finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence and Minimization of Automata, Conversion of NFA with epsilon to DFA Equivalence of Moore and Mealy Machine. Applications and Limitation of FA.

Part B: FA with output: Moore and Mealy machine.

Unit 2: (8+1 Hrs)
Regular Expressions (RE) and Languages

Part A: Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene's Theorem, Regular expression to DFA, DFA to Regular expression, Non Regular Languages, Pumping Lemma for regular Languages, Myhill-Nerode theorem, Closure properties of Regular Languages, Applications of RE: Regular expressions in Unix, GREP utilities of Unix, Lexical analysis and finding patterns in text.

Part B: Decision properties of Regular Languages.

Unit 3: (11+1 Hrs)
Context Free Grammars (CFG) and Push Down Automata(PDA)

Part A: Context Free Grammars: Definition, Examples, Derivation, Languages of Grammar, Derivation trees, Ambiguity in Grammar, Ambiguous and Unambiguous CFG, Inherent ambiguity, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure properties of CFLs, Decision Properties of CFLs (Emptiness, Finiteness and Membership), Chomsky Hierarchy. Pumping lemma for CFLs

Push Down Automata: Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof).

Part B: Regular grammars, left linear and right linear regular grammars, regular grammar and finite automata.

Unit 4: (7+1 Hrs)

Introduction to Turing Machines

Part A: Turing Machines: Basic model, definition and representation, Instantaneous Description, Language acceptance by TM.

Robustness of Turing Machine model and equivalence with various variants: Two-way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, TM as enumerator.

Recursive and Recursively Enumerable languages and their *closure properties*.

Part B: Comparison between Finite Automata, Push Down Automata, and Turing Machines.

Unit 5: (6+1 Hrs)

Introduction to Undecidability

Part A: Universal Turing Machines, Church-Turing Thesis and intuitive notion of Algorithm.

Introduction to countable and uncountable sets (countability of set of natural numbers, integers, rationals. Uncountability of set of real numbers, points in plane), Encoding for Turing machines and countability of set of all Turing machines. *Existence* of Turing unrecognizable languages.

Undecidability of Halting problem, Post Correspondence Problem. Example of a Turing unrecognizable language. Decision properties of R. RE languages and Rice's theorem.

Part B: Hilbert's tenth problem, undecidability of tiling problem

Text Books

1. "Introduction to Automata Theory, Languages and Computation", Hopcroft J, Motwani R, Ullman, Addison-Wesley, ISBN 81-7808-347-7, Second Edition .

Reference Books

- 1 "Introduction to Theory of Computation", Michael Sipser, Course Technology, ISBN-10: 053494728X, Third Edition.
- 2 "Introduction to Languages and the Theory of Computation", J. Martin, Tata McGraw-Hill, ISBN 0-07-049939-x, Third edition, 2003.
- 3 "Introduction to Computer Theory", Cohen D., Wiley Publications, 0-471-51010-6, Second Edition.

Additional Reading

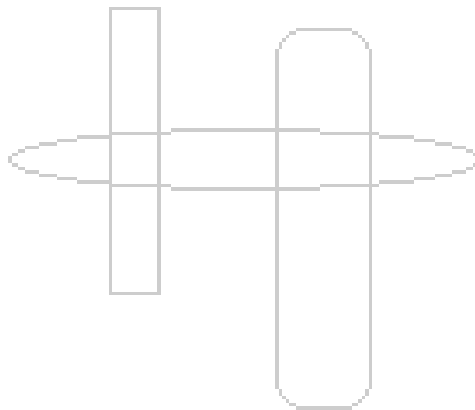
1. "Elements of The theory of Computation", H.R.Lewis, C.H.Papadimitriou, Pearson Education, ISBN 81-7808-487-2, Second Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Describe various automata theoretic models for recognizing formal languages.
2. Comprehend general purpose powers and computability issues related to state machines and grammars.

3. Discriminate the expressive powers of various automata theoretic and formal language theoretic models and illustrate significance of non determinism for automata theoretic models.
4. Explain the relevance of Church-Turing thesis, and comprehend the theoretical limit of computation (independent of software or hardware used) via the concept of undecidability.
5. To infer the applicability of various automata theoretic models for recognizing formal languages
6. Select appropriate computing theory principles, models for the solution of complex computing problems



FF No. : 654 A

CS31113:: Microprocessors and Microcontroller

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Microprocessors and Interfacing.

Unit 1: (8+1 Hrs)

Introduction to Pentium microprocessor

Part A: Historical evolution of 80286, 386, 486 processors. Pentium features and Architectures, Pentium Real mode, Pentium RISC features, Pentium super-scalar architecture - Pipelining, Instruction paring rules, Branch prediction, Instruction and Data caches. The Floating point Unit features & data types.

Part B: Pipelining stages.

Unit 2: (8+1 Hrs)

BUS cycles and Memory organization

Part A: Initialization and configuration, Bus operations – Reset, Non pipelined and pipelined (read and write). Memory organization and I/O organization. Data transfer mechanism – 8bit, 16bit, 32bit. Data bus interface. Pentium programmers model, Register set, Addressing modes, Data Types, BUS cycle.

Part B: Pentium Instruction Set.

Unit 3: (8+1 Hrs)

Microcontroller

Part A: System hardware, Interrupt structure and Applications, Processor - Architecture, Programmer's model, Modes of operation, Interrupt Structure and Applications. Management of Power Supply, Clocking Unit, Real Time Clock and Timers, Reset Circuitry and Watchdog Timer. Structural Units of Processor, Processor and Memory Selection, Memory Map, Interfacing Processors, Memories and I/O. Processor and Memory Selection.

Part B: Programmer's model

(8+1 Hrs)

Unit 4:

Microcontroller I/O interfacing

Part A: Digital I/O interfacing, Serial Data Communication using RS-232C. I/O devices- ADC / DAC and Stepper Motor, Optical Devices such as LED / LCD Display devices, Timers/Counters.

Part B: Design of Delay Routine using Hardware timers.

Unit 5: (8+1 Hrs)

Protected Mode Architecture of Pentium

Part A: Introduction, segmentation, support registers, related instructions, descriptors, memory management through segmentation, logical to linear address translations, protection by segmentation, privilege-level, protection, related instructions, inter-privilege level, transfer control, Paging-support registers, related data structures, linear to physical address translation, TLB, page level protection.

Part B: Programming of Protected mode

Text Books

1. Antonakos J., *"The Pentium Microprocessor"*, Pearson Education, 2004, 2nd Edition.
2. Deshmukh A., *"Microcontrollers - Theory and Applications"*, Tata McGraw-Hill, 2004.

Reference Books

1. ARM data Sheet.
2. Intel data Sheet.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Study Operating Modes of Advanced Microprocessors.
2. Study Application and Development of Microprocessor Components
3. Use advanced Tools for Program implementations
4. Develop Effective Engineering Solutions to solve Real World Problems
5. Utilize the Structures to effectively solve Computing Problems in Teams
6. Design Effective Automation Solutions using Advanced Microprocessors

CS31115:: ADVANCED DATA STRUCTURES

Credits: 02

Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites: Data Structures

Unit 1: (6+1 Hrs)
Hashing

Part A: Dynamic Hashing: Motivation for Dynamic Hashing, Dynamic Hashing using Directories, directory less Dynamic Hashing. Bloom Filters: An application-differential files, Bloom Filter Design.

Part B: Alternate hash functions (mid-square, folding, digit analysis), Double Hashing

Unit 2: (7+1 Hrs)
Priority Queue and Advance Heaps

Part A: Double Ended Priority queues, Leftist Trees: Height Biased, Weight Biased. Binomial Heaps: Cost Amortization, Definition of Binomial Heaps, Insertion, Melding two Binomial Heaps, deletion of min element. Fibonacci Heaps: Definition, Deletion from an F-heap, Decrease key, Cascading Cut.

Part B: Comparative study of different priority queue implementations using – binary heaps, leftist trees, binomial heaps, Fibonacci heaps with respect to the following operations – insert, delete, find-min, extract-min, decrease-key, meld

Unit 3: (6+1 Hrs)
Advanced Binary Search Trees

Part A: Red-Black Trees: Definition, Representation, Searching, Insertion, Deletion. Splay Trees: Bottom-up, top-down.

Part B: Insertion and Deletion in B Trees and B+ Trees.

Unit 4: (6+1 Hrs)
Digital Search Structures

Part A: Digital Search Trees: Definition, Search, Insert and Delete. Binary Tries, Compressed Binary Tries. Multiway Tries: Definition, searching a Trie, sampling strategies, Insertion, Deletion, Height of a Trie. Prefix Search and applications. Suffix Trees.

Part B: Space required and alternative node structures for a Trie.

Unit 5:

(7+1 Hrs)

Data structures for Disjoint Sets and Linear Programming

Part A: Data structures for Disjoint Sets: Disjoint Set Operations, Linked list representation of disjoint sets, Disjoint set forests. Linear Programming: Overview, Applications, Algorithms, Standard and Slack forms, Formulating problems as linear programs, Simplex Algorithm.

Part A: Linear Programming duality

Text Books

1. *“Fundamentals of Data Structures in C”*, E. Horwitz , S. Sahani, Anderson-Freed, Universities Press, Second Edition, 2008, ISBN 978-81-7371-605-8.
2. *“Introduction to Algorithms”*, T. Cormen, R.Rivest, C. Stein, C. Leiserson, PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Demonstrate memory efficient solution in order to apply basic principle of programming perspective.
2. Analyze complexity issues for space and time bound scaled objective for programming solution.
3. Infer memory utilization with modularization useful for programming pragmatics.
4. Verify and narrate test condition in employing debugging technique to assist problem solving.
5. Elaborate on unknown well organized problems based on realizable solutions
6. Contribute breadth of modularization in order to understand and implement realistic solution.

CS30101:: OPERATING SYSTEMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:: Data Structures and Algorithms, Computer Organization.

List of Contents

A TERM-WORK containing the record of the following:

1. Execution of Advance Unix commands.
2. Write a shell program to sort an array of numbers using any sort method.
3. Execution of AWK related commands.
4. Implement the solution for Reader-Writer problem using Threads and Semaphores/Mutex.
5. Implement the solution for Producer-Consumer (Bounded Buffer) problem using Threads and Semaphore/ Mutex .
6. Implement the solution for Dining-Philosopher problem using Threads and Semaphore.
7. Implementation of resource allocation graph (RAG).
8. Implement the solution for Banker's Algorithm for deadlock avoidance.
9. Draw the Gantt charts and compute the finish time, turnaround time and waiting time for the following algorithms:
 - a. First come First serve
 - b. Shortest Job First (Preemptive and Non-Preemptive)
 - c. Priority (Preemptive and Non-Preemptive)
 - d. Round Robin
10. Calculate the number of page faults for a reference string for the following page

replacement algorithms:

- a. Optimal
- b. FIFO
- c. LRU

11. Calculate the total distance traversed by the disk arm to satisfy the pending requests for the following disk scheduling algorithms:

- a. FCFS
- b. SSTF
- c. SCAN
- d. C-SCAN

Text Books

1. *“Operating Systems”, Stalling William, Pearson Education, ISBN: 0-13-031999-6, 4th Edition.*
2. *“Operating System Concepts”, Silberschatz A., Galvin P., Gagne G., John Wiley and Sons, ISBN: 9971-51-388-9, 6th Edition.*

Reference Books

1. *“Modern Operating Systems”, Tanenbaum Andrew S., PHI, ISBN: 81-7808-447-3, 2nd Edition.*
2. *“Unix Concepts and Applications”, Das Sumitabha, TMH, ISBN: 0-07-053475-6, 3rd Edition.*

Additional Reading:

1. *“Operating System Principles”, Silberschatz A., Galvin P., Gagne G, John Wiley And Sons, ISBN: 9812-53-176-9, 7th Edition.*
2. *“Design of the Unix Operating System”, Bach Maurice, Pearson Education, ISBN: 81-7808-731-6, 1st Edition.*
3. *“Unix Shell Programming”, Yashavant Kanetkar, BPB Publications, ISBN: 81-7029-753-2, 1st Edition.*
4. *“Unix And Shell Programming”, Forouzan B. A., Gilberg R. F., Australia, Thomson Brooks Cole, ISBN: 981-243-127-6, 1st Edition.*

CS31215:: ADVANCED DATA STRUCTURES

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Data Structures

List of Contents

1. Example to indicate the limitations of static hashing and how it is addressed using dynamic hashing
2. Implement an application that uses bloom filter
3. Improve the performance of Dijkstra's shortest path algorithm using Fibonacci heaps
4. Implement a binomial heap and compare its amortized complexity with binary heap
5. Implement a data store using Red Black trees as the underlying data structure
6. Implement an application that makes use of a prefix tree (trie) – address book, spell checker, auto completion etc.
7. Implement a solution for LCS problem (Longest Common Subsequence) using suffix trees
8. Improve the performance of Kruskal's MST algorithm using disjoint set data structure
9. Write a program to solve a linear programming problem using simplex algorithm
10. Simulate a real-world application (search engine, file system, etc.) using one or more advanced data structures.

Text Books

1. *"Fundamentals of Data Structures in C", E. Horwitz , S. Sahani, Anderson-Freed, Universities Press, Second Edition, 2008, ISBN 978-81-7371-605-8.*
2. *"Introduction to Algorithms", T. Cormen, R.Rivest, C. Stein, C. Leiserson, PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.*

Reference Books

1. *"Advanced Data structures", Peter Brass, Cambridge Publication, 1st Edition, 2008 ,ISBN – 978-0-521-88037-4.*
2. *"Data Structures and algorithms with Object Oriented design Patterns in Java", Bruno R. Preiss, wiley Publication, 1st Edition, 2000. ISBN: 978-0-471-34613-5.*

CS30303:: OPERATING SYSTEMS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Structures and Algorithms, Computer Organization

List of Practical

Part A:

1. Implementation of a multiprogramming operating system:

- a. Stage I:
 - i. CPU/ Machine Simulation
 - ii. Supervisor Call through interrupt
- b. Stage II:
 - i. Paging
 - ii. Error Handling
 - iii. Interrupt Generation and Servicing
 - iv. Process Data Structure
- c. Stage III:
 - i. Multiprogramming
 - ii. Virtual Memory
 - iii. Process Scheduling and Synchronization
 - iv. Inter-Process Communication
 - v. I/O Handling, Spooling and Buffering

Text Books

1. *"The Logical Design of Operating Systems"*, Shaw, Alan C, Prentice Hall, 1974.
2. *"Computer Networks"*, by Tanenbaum A. S., Pearson Education , 2008, ISBN-978-81-7758-165-2, 4th Edition,

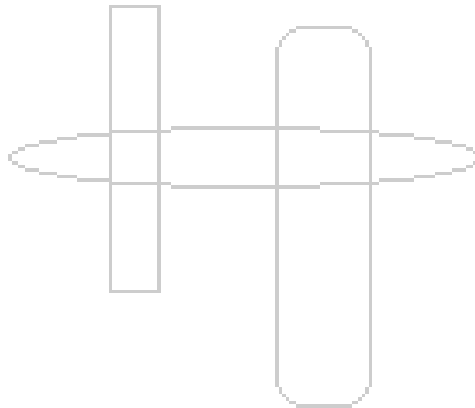
Reference Books

1. *"Operating System Concepts"*, by Silberschatz A., Galvin P., Gagne G., John Wiley and Sons, ISBN: 9971-51-388-9, 6th Edition.
2. *"Communication Networks- Fundamental Concepts and Key Architectures"*, by Leon-Garcia-Wadjaja, Tata McGraw-Hill Publications, ISBN-978-0072463521.

Additional Reading

1. *"Operating System Principles"*, Silberschatz A., Galvin P., Gagne G, John Wiley and Sons, ISBN: 9812-53-176-9, 7th Edition.
2. *"Design of the Unix Operating System"*, Bach Maurice, Pearson Education, ISBN: 81-7808-731-6, 1st Edition.

3. *“Unix Shell Programming”, Yashavant Kanetkar, BPB Publications, ISBN: 81-7029-753-2, 1st Edition.*
4. *“Unix And Shell Programming”, Forouzan B. A., Gilberg R. F., Australia, Thomson Brooks Cole, ISBN: 981-243-127-6, 1st Edition.*
5. *“Modern Operating Systems”, Tanenbaum Andrew S., PHI, ISBN: 81-7808-447-3, 2nd Edition.*
6. *“Unix Concepts and Applications”, Das Sumitabha, TMH, ISBN: 0-07-053475-6, 3rd Edition.*
7. *“Computer Networks- A Systems Approach”, By Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, 4th Edition, ISBN-978-81-312-1045-1.*



CS30316:: Computer Networks

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Communication

List of Practical

1. Set up a small network of 2 to 4 computers using Hub/Switch. It includes installation of LAN Cards, Preparation of Cables, Assigning IP addresses and sharing C drive.
2. File Transfer using PC To PC Communication.
3. Installation and configuration web server using Apache Tomcat for Linux / IIS for Windows 7. Access the web pages from another computer.
4. Studying Linux and Windows network commands. [ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap, etc]
5. Program for calculating CRC using Modulo-2 and Polynomial methods.
6. Simulate the sliding window protocols Go Back N and Selective Repeat.
7. File Transfer between two computers using TCP sockets.
8. Multiuser chat application using UDP sockets.
9. To create TCP/IP packet using standard TCP/IP include files and send it to other machine
10. Program to find active and passive ports on nearby host using sockets.
11. Study of protocol and packet analyzer using wireshark.

12. Installing and configuring DHCP server for Linux/Windows.

Text Books

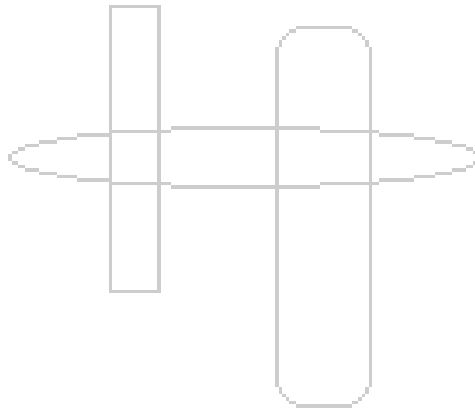
1. *“Computer Networks”, by Tanenbaum A. S., Pearson Education , 2008, ISBN-978-81-7758-165-2, 4th Edition,*
2. *““Computer Networking- A Top-Down Approach”, by James F. Kurose and Keith W Ross, Person Education, ISBN- 978-81-317-9054-0, 5th Edition.*

Reference Books

1. *“Data Communications and Networking”, by Forouzan B. A, Tata McGraw-Hill Publications, 2006, ISBN-0-07-063414-9, 4th edition.*
2. *“Communication Networks- Fundamental Concepts and Key Architectures”, by Leon-Garcia-Wadjaja, Tata McGraw-Hill Publications, ISBN-978-0072463521.*

Additional Reading

1. *“Computer Networks and Internet”, by Comer D., Pearson Education, ISBN-81-297-0330-0, 2nd Edition.*
2. *“Computer Networks- A Systems Approach”, by Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, ISBN-978-81-312-1045-1, 4th Edition.*



FF No. : 654 B

CS31313:: Microprocessors & Microcontrollers

Credits: 01

Teaching Scheme: -Laboratory 2 Hrs/Week

Prerequisites: Microprocessor and Interfacings

List of Practical

Group A- Pentium

1. Write an ALP to simulate TYPE command using PSP.
2. Write an ALP to simulate COPY command using PSP.
3. Write an ALP to do the Following
 - a) Capture MSW
 - b) Display and Analyse Contents of GDTR IDTR LDTR
4. Write an ALP / in line code for displaying boot record of hard disk.
5. Write a Program to Demonstrate Code Cache Design
6. Write ALP for DPMI.
7. Write ALP for Mouse interface.
8. Study of Pentium motherboard.
9. Write ALP to implement multitasking using Pentium programming.

Group B- ARM-7

1. Write a program to interface Switch and LED.
2. Write a program to interface Timer.
3. Write a program to interface LCD.
4. Write a program to interface serial port.
5. Write a program to interface ADC.
6. Write a program to interface Stepper motor.

Text Books

1. Antonakos J., "The Pentium Microprocessor", Pearson Education, 2004, 2nd Edition.

Reference Books

1. Intel data sheet.
2. DOS data manual.

CS37401::MINI PROJECT

Credits: 02

Guidelines:

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

- 1.Evaluate solutions to real world problems and convert them to applications using various programming skills. (PO2)
- 2.Devise software solutions with necessary design documents(PO3)
- 3.Conduct rapid application development using set of modern tools and frameworks(PO5)
- 4.Work in well formed teams with proper communication and share responsibilities in project development according to skill sets and individual strengths.(PO4)
- 5.Concisely estimate cost and effort constraints on problem specific domains.(PO10 & PO11)
- 6.Develop software solutions based on standard engineering specifications and documents such as requirement analysis, flow diagrams, verification and validations.PSO3

CS30401: Comprehensive Viva Voce

Credits: 2

Guidelines:

1. Student has to present for CVV exam with assignment file duly signed by subject teacher.
2. Student has to give oral/viva exam based on two core subjects they have studied throughout the semester.
3. Examiners examine students by asking questions based on theory and lab syllabus. Students will prepare two subjects and respective subject expert faculty will ask questions to students.
4. Each student will be evaluated by experienced faculty from department and other reputed engineering colleges.
5. To qualify in CVV, student has to pass both the subjects under CVV.

Course Outcomes

Upon completion of the course, graduates will be able to –

1. Justify the selection of scheduling algorithm for process and memory management
2. Investigate the design attribute of Operating System module
3. Use different networking and OS tools
4. Demonstrate the networking protocols and OS modules
5. Adapt IEEE, ISO, ITU standards and OS for requirement of next generation.
6. Design the client server prototypes for networks and OS for present needs.

FF No:654 E

CS37301::SEMINAR

Credits: 02

Teaching Scheme: - Lab 2 Hrs/Week

Guidelines:

Seminar is a course requirement wherein under the guidance of a faculty member a student is expected to do an in depth study in a specialized area by doing literature survey, understanding different aspects of the problem and arriving at a status report in that area. Students are expected to choose a topic in CSE based on current trends or industry practices. While doing a seminar, the student is expected to learn investigation methodologies, study relevant research papers, correlate work of various authors/researchers critically, study concepts, techniques, prevailing results etc., analyze it and present a seminar report. Evaluation will be based on relevance of topic, understanding of the problem, literature Survey, presentation, communication skills, answering queries and reporting or documenting procedure.

Course Outcomes:

Upon completion of the course, graduates will be able to -

- 1 Conduct a thorough literature survey of identify technical topic.
- 2 Scope and visibly identify technological trade off in computer engineering leading to significant topics.
- 3 Adapt skills, professional ethics & technical standardization to solve real world problems.
- 4 Present technical topic in written form with technical report or document.
- 5 Relate real world problem scenarios in computer engineering.
- 6 Select and utilize appropriate study concept, techniques, relevant research paper and investigation methodologies to devise effective current trends or industry practices solution.

CS37302::PROJECT STAGE I

Credits: 02

Guidelines:

Aim

This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

- a. Transform existing Ideas into conceptual models.
- b. Transform conceptual models into determinable models.
- c. Use determinable models to obtain system specifications.
- d. Select optimum specifications and create physical models.
- e. Apply the results from physical models to create real target systems.

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
6. The Project Work will be assessed jointly by a panel of examiners. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.

7. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.
8. The Student Project Groups are expected to work on the recommendations given by the panel of examiners.

Assessment Scheme

Sr. No.	Content	Marks
1	Concept	20
2	System Requirement Specification	10
3	System Analysis	30
4	System Design Block Diagram	30
5	Presentation of the Project Work	10

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify Real World Problems
2. Select Design Pattern to Best approach the Solution
3. Construct Model to deliver Solutions for Real World Problem
4. Optimise the design to enable effective reduction in Resource utilisation.
5. Create System Requirements to define the Solution
6. Design Effective System requirement of Enabling implementation of Solution

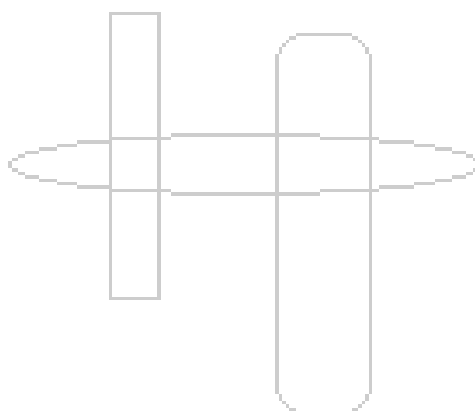
Note:

The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members.

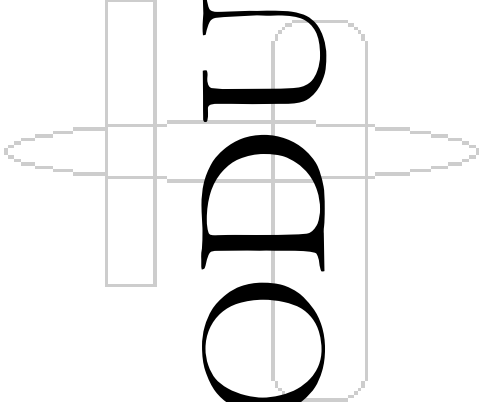
The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted. Following is the list of recommended domains for Project Work:

Computer Networks	Image Processing
Operating Systems	Artificial intelligence

Network Security	Expert Systems
Digital Signal Processing	Object Oriented Systems
Systems Programming	Modeling and Design
Real Time Systems	System Testing
Embedded systems	Storage Management
Cluster Computing	Client-Server Computing
Mobile & Wireless Communications	Cloud Computing
Multimedia Systems	Protocol Engineering



MODULE VI



FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

T.Y. B. Tech. Structure with effect from Academic Year 2015-16

Module VI

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS30102	Software Engineering	S ₁	3	-	1	10	20	5	5	-	60	4
CS30106	Database Management Systems	S ₂	3	-	-	15	20	5	-	-	60	3
CS30108	Design and Analysis of Algorithms	S ₃	3	-	1	10	20	5	5	-	60	4
CS31119	Object Oriented Modeling and Design	S ₄	3	-	-	15	20	5	-	-	60	3
CS30114	Systems Programming	S ₅	2	-	-	15	20	5	-	-	60	2
CS30314	System Programming	P ₁	-	2	-	-	-	-		30	70	1
CS30306	Database Management Systems	P ₂	-	2						30	70	1
CS30319	Object	P ₃	-	2						30	70	1

	Oriented Modeling and Design											
CS37402	Mini Project (T.Y. Semester I) Irrespective of Module	-	4	-		2				30	70	2
CS37301	Seminar(T.Y. Semester I) Irrespective of Module	-	4	-		2				30	70	2
PD	Institute Level Elective	-	2	-		2				30	70	1
CS30402	Comprehensive Viva Voce	-	-	-						30	70	2
CS37302	Project Stage 1 (T.Y. Semester II) Irrespective of Module	-	4	-						30	70	2
TOTAL			14	16	2							24

CS 30102: Software Engineering

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1: Software Engineering Paradigms (8 Hrs)

Part A: Overview of Software Engineering, Software Process Framework, Process Patterns, Personal and Team Process Models, Process Models: Code-and-Fix, Waterfall Model, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Agile process, Extreme Programming, Cleanroom Methodology, Component-Based Software Engineering, CMMI, Impact of Processes and Outcomes, Process Selection and applicability, Software Engineering Principles and Practices

Part B: DSDM, Component-Assembly Process Model, Best Practices in Technology Selection, Formal Methods

Unit 2: Requirement Engineering (8 Hrs)

Part A: Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non-Functional, Domain Engineering activities, Requirements Characteristics and Characterization, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, Requirement Traceability, Requirement Prioritization, Relationship of Requirement Engineering to other Framework Activities, System Scope Determination and Feasibility Study, Statement of Work Generation

Part B: Requirements Verification and Validation, Requirement Maturity, Technical Reviews

Unit 3: System Analysis and Design Foundations (8 Hrs)

Part A: System Attributes, Properties, and Characteristics, System Roles and Stakeholders, System Acceptability, Levels of Abstractions, System Interfaces, Organizational Roles and Missions, Problem, Opportunity and Solution Spaces, System Environment and Mission Analysis, System Operational Model, Analysis Concepts, Analysis Methods, Analysis Modeling Techniques, Data Flow diagrams, System Analysis Scenarios and Model Generation, Context Models, Behavioral Models, Data Models, Object Models, Structured Methods Design Concepts, The Design Model, Design Qualities, Characteristics of Design activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics

Part B: User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation

Unit 4: System Architecture Determination (8 Hrs)

Part A: Foundations of Software Architecture, Reference Architectures, Architectural Design: Software Architecture, Data Design and Architectural Design, Views,

Viewpoints, Perspectives, Conceptual Architecture View, Module Architecture View, Execution Architecture View, Code Architecture View, Architecture styles: Repository, Layered, Pipe-Filter, Call-Return, Peer-Peer, Publish-Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, Heterogeneity in Architecture

Part B: Architecture Analysis Techniques, Zachman Framework, Architecture Assessment

Unit 5: Project Management Principles (8 Hrs)

Part A: Project Management Activities, Structures and Frameworks, Teamwork, Leadership, Project Planning, Project Scheduling, Risk Analysis, Introduction to Function Points, Empirical Estimation, COCOMO II model, Configuration Management Planning, Change Management, Version and Release Management, Foundations of Software Testing: Terms, Testing Cycle, Outcomes, Principles

Part B: Classic Mistakes, Complex Systems, Critical Systems, Software Safety

Text Books:

1. Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.
2. Roger S Pressman, 'Software Engineering: A Practitioner's Approach', McGraw Hill, 6/e, 2005

Reference Books :

1. Paul Clements, Felix Bachmann, Len Bass, David Garlan, *Documenting Software Architectures: Views and Beyond* Addison-Wesley Professional 2003, ISBN-10: 0201703726, ISBN-13: 9780201703726
2. Soren Lauesen, *Software requirements: Styles and techniques*, Addison Wesley, ISBN 0201745704, 2002
3. Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*, John Wiley & Sons, Inc., ISBN-13 978-0-471-39333-7, 2006

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system architecture and design model using architectural styles and design guidelines.
3. Formulate system specifications required to compose software artifacts using appropriate CASE tools.
4. Propose and demonstrate realistic solutions supported by well-formed reports and documentation.
5. Conform to project management principles and demonstrate cohesive teamwork skills avoiding classic mistakes and emphasizing on software safety.
6. Analyze the target system properties and recommend solution alternative by practicing project planning, scheduling, estimation and risk management activities.

FF No. : 654 A

CS30106:: DATABASE MANAGEMENT SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data structures

Unit 1: Introduction and Data Models

(9+2 Hrs)

Part A: Introduction: Need of Database Management System, Evolution, Data Abstraction, Data Independence, System Architecture of DBMS;
Data Models: Entity Relationship (ER) Model, Extended ER Model, Relational Data Model, Object Oriented Data model, Semi-structured Data Model: DTD or XML Schema

Part B: Spreadsheet Model, Codd's Twelve Rules for Relational DBMS, Life Cycle of a Relational Database

Unit 2: Database Design Theory

(7+2 Hrs)

Part A: Normalization: Need, Functional Dependency, Inference Rules, FD Closure, Minimal Cover, Decomposition Properties, Normal Forms (upto BCNF), Multi-valued Dependency (4NF), Relational Synthesis Algorithm

Part B: Join and Inclusion Dependency, 5NF, DKNF, Trade - off

Unit 3: Query Languages

(6+2 Hrs)

Part A: Formal Relational Query Languages: Relational Algebra, Tuple Relational Calculus;

SQL: DDL, DML, Select Queries, Join Queries, Subqueries;

PL/SQL: Procedure, Function, Trigger;

Query by Example;

Part B: Domain Relational Calculus, DCL-Security and Authorization, Date-Timestamp, String and Numerical Functions, Mapping of Relational Algebra to SQL

Unit 4: Storage and Querying

(9+2 Hrs)

Part A: Storage: Storage and File structure, Files with Fixed / Variable Length Records, Hashed Files;

Indexing: Indexed Files, Single Level and Multi Level Indexes, B⁺ Trees;

Query Processing: Steps, Algorithms for Selection, Join Operation;

Query Optimization: Transformation of Relational Expressions, Choice of Evaluation Plans; Query Execution Cost;

Part B: SAN, Files with Sparse / Dense Index; Query Processing: Sort Operation, Impact of Indices on Query Performance;

Unit 5: Transaction Management and Emerging Trends **(9+2 Hrs)**

Part A: Transaction: ACID Properties, Concurrency Control Protocols: Lock-based, Multiple Granularity, Multiversion Scheme; Failure and Recovery;
NoSQL: RDBMS vs NoSQL, BASE properties, NoSQL Categories; NewSQL;
Emerging Trends: Distributed Databases, Distributed Data Storage, Distributed Query Processing; Parallel Databases, Architectures, Speedup and Scaleup, Decomposition, Data Replication; Time Series Databases, Spatial and Geographic Databases;
Part B: Design of Core DBMS Functions, Timestamp based Concurrency Control Protocol, ARIES Recovery Technique, Personal Databases

Text Books

1. "Database System Concepts", Silberschatz, Korth, Sudarshan, McGraw Hill International Edition, ISBN- 0-07-228363-7, 5th Edition.
2. "Fundamentals of Database Systems", Elmasri, Navathe, Pearson Education, ISBN 81-297-0228- 2, 5th Edition.
3. "Database Systems", Thomas Connolly, Carolyn Begg, Pearson Education, ISBN, 81-7808-861- 4, 3rd Edition.

Reference Books

1. "Database Management Systems", Ramakrishnan, Gehrke, McGraw-Hill International Edition, ISBN 0-07-115110-9, 3rd Edition.
2. "Introduction to database systems", C. J. Date, Narosa Publishing House: 1995, 3rd Edition.
3. "Getting Started with NoSQL: Your guide to the world and technology of NoSQL", by Gaurav Vaish
4. "Understanding the New SQL: A Complete Guide (The Morgan Kaufmann Series in Data Management Systems)", Jim Melton, Alan R. Simon

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Design data models to enforce data requirements and operational constraints of an organization
2. Construct refined logical database model with consideration of data semantics and dependency.
3. Develop a database system using relational database query languages, PL/SQL and NoSQL
4. Acquaint themselves with emerging trends in data storage and processing.
5. Engage in a team to develop front end and back end solution for a real world scenario.
6. Build a database system and demonstrate competence with the fundamental tasks involved with its modelling, designing, and implementation.

CS30108:: DESIGN AND ANALYSIS OF ALGORITHMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1: (9+1 Hrs)

Overview of Time Complexity analysis, Divide and Conquer

Part A: Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst case time and space complexity of algorithms. Overview of searching, sorting algorithms (binary search, insertion sort, heap sort, bubble sort). Adversary lower bounds (for comparison based sorting, for finding second minima etc). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity and to prove correctness of algorithms. Amortized complexity of algorithms.

Divide and Conquer: Analyzing Quick sort, Merge sort. Strassen's matrix multiplication, Finding Median, Counting Inversions, efficient algorithms for Integer arithmetic, Application of Divide and Conquer to Geometric Problems. Master's theorem and its uses.

Part B: Discrete Ham-Sandwich theorem, Computing convex hull of points in plane, Solution for Josephus problem.

Unit 2: (8+1 Hrs)

Dynamic Programming and Backtracking Strategies

Part A: Dynamic Programming: General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients. Matrix Chain multiplication. Optimal binary search tree (OBST) construction, 0/1-Knapsack, Traveling Salesperson Problem, Shortest path in a Graph, Sequence Alignment problem, Scheduling problem.

Backtracking: General Strategy, n-Queen's problem, Graph Coloring, Hamiltonian Cycles, 0/1 Knapsack, Subset sum problem.

Part B: String Editing Problem, Patience-sorting and $O(n \log n)$ algorithm for longest increasing sub-sequence problem. Solution for Peg-solitaire game.

Unit 3: (7+1 Hrs)

Greedy, Branch & Bound, Transform and Conquer techniques

Part A: Greedy Method : Analysis of Minimum Spanning Tree and Shortest Path algorithms, Huffman coding, Scheduling Algorithms. Optimal Storage Problem.

Branch and Bound: General Strategy, 0/1 Knapsack, Traveling Salesperson Problem. Problem solving based on transform and conquer technique (Gaussian elimination, Horner's rule and fast exponentiation etc). Heuristic based algorithms (Knight tour).

Part B: Postage stamp problem, $n*n*n$ Queens problem, testing 2-colorability of graphs efficiently

Unit 4: (6+1 Hrs)

Introduction to Complexity Theory and NP-Completeness

Part A: Overview of deterministic and non deterministic Algorithms. Time Complexity classes P, NP, coNP, and their interrelation, EXP. Space complexity class PSPACE. Notion of polynomial time many one reductions reduction. Notion of NP-hardness and NP-completeness. Cook's Theorem and implication to P versus NP question. NP-hardness of halting problem.

NP-Complete problems- Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem.

Part B: Decision Vs Search versions of problems in class NP, some problems in NP intersection coNP (linear programming, primality testing, perfect matchings in bipartite graphs).

Unit 5: (10+1 Hrs)

Introduction to Randomized, approximation and online algorithms.

Part A: Randomized algorithms: Introduction to Las-Vegas and Monte-Carlo Algorithms. Abundance of witnesses/solutions and application of randomization, solving SAT for formulas with "many" satisfying assignments, Randomized Quick Sort, Karger's algorithm for Min Cut problem, Coupon Collector problem.

Approximation algorithms: Introduction to NP-optimization problems, factor-2 approximation algorithm for Vertex Cover, hardness of approximation of Travelling Sales Person Problem(TSP), factor-2 approximation algorithm for *metric* TSP, approximation algorithm for set-cover.

Part B: Birthday paradox, probabilistic recurrences, generation of large primes and Prime Number Theorem. Approximation algorithm for bin packing problem.

Text Books

1. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" ,PHI 2nd edition, 2001. ISBN 81-203-2141-3
2. Jon Kleinberg, Eva Tardos "Algorithm Design", Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6
3. Horowitz, Sahani, "Fundamentals of computer Algorithms", Galgotia. 2nd Edition, 1998.ISBN 81-7515-257-5

Reference Books

1. Bressard, Bratley "Fundamentals of Algorithmics." ,PHI, 2nd Edition,1996, ISBN 81-203-1131-0

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Formulate computational problems in abstract and mathematically precise manner
2. Design efficient algorithms for computational problems using appropriate algorithmic paradigm.
3. Analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.
4. Apply randomization and approximation techniques to design efficient algorithms for some potentially NP-hard computational problems.
5. Establish NP-completeness of some decision problems and grasp the significance of the notion of NP-completeness and its relation with intractability of the decision problems.
6. Incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solution for complex computing problems.

CS30114:: SYSTEMS PROGRAMMING

Credits: 02

Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites:

- Data Structures
- Computer Organization
- Microprocessors

Unit 1: **(5+1 Hrs)**

Introduction to System Programming

Part A: Introduction: Components of System Software, Language Processing Activities, Fundamentals of Language Processing.

Assemblers: Elements of Assembly language programming. Simple assembler scheme, Structure of an assembler, Design of single and two pass assembler.

Macro Processors: Macro Definition and call, Macro expansion, Nested Macro Calls, Advanced Macro Facilities, design of Macro Preprocessor. Booting Procedure for DOS & Windows,

Part B: RISC machines, Machine dependent and machine-independent Assembler features.

Unit 2: **(5+1 Hrs)**

Compilers, Loaders and Linkers

Part A: Compilers: Introduction to Compiler phases, Introduction to cross compiler, Features of machine dependent and independent compilers, Overview of types of compilers, Interpreters.

Linkers: Relocation and linking concepts, Static and dynamic linker, subroutine linkages.

Loaders: Introduction to Loader, Loader Schemes: Compile and go, General Loader Scheme, Absolute loaders, relocating loaders, direct linking loaders.

Part B: Instruction description, Pseudo operations, Instruction Mapping, MSDOS Linker, Sun OS linker.

Unit 3: **(5+1 Hrs)**

Essential concepts of Systems programming for Linux as Open Source OS.

Part A: Introduction and essential concepts of LINUX system programming: System Programming, APIs and ABIs, standards, Program segments/sections; The ELF Format, Linking and loading, Linux dynamic libraries (shared objects), Multitasking and paging, Address translation, Memory Protection, Comparison with Windows.

Part B: Dynamic linking, API compatibility, Dynamically linked libraries, Overall architecture and limitations.

Unit 4:

(5+1 Hrs)

Encoding, Decoding and Device drivers

Part A: Encoding and decoding schemes for the X-86 processor, Advanced System Programming Concepts: Operating system interfaces, Stack smashing, Dynamic Linking Libraries. Types of Drivers, Driver History, Driver Issues, Kernel Level Device drivers, Virtual device drivers(VxD), Device Driver Stack Buses and Physical Devices, Static Device drivers, Dynamic Device drivers, PnP, Device Namespace, and Named Devices.

Part B: Library Description for IA-32/Intel64.

Unit 5:

(4+1 Hrs)

TSR Programming

Part A: DOS: Internals of DOS, DOS loading, DOS memory map, Internal commands, External commands, command interpreter, POST details, POST sequence, PSP (structure details), '.EXE' and '.COM' file structures, conversion of .EXE to .COM file. BIOS: what and why, BIOS calls: INT 10H calls, DOS calls: INT 21H calls. Difference between DOS and BIOS, TSRs: types, Structure, details of TSR loading, examples, writing TSRs.

Part B: .Net Framework and Direct X: History, Components, Compatibility, Architecture.

Text Books

1. "Systems Programming & Operating Systems", D M Dhamdhere, Tata McGraw Hill Publications, ISBN - 0074635794
2. "Systems Programming", John J Donovan, ISBN - 0070176035

Reference Books

1. "Linux System Programming", Robert Love, O'Reilly, ISBN 978-0-596-00958-8

Course Outcomes

Upon completion of the course, the graduates will be able to -

1. Discriminate among different System software and their functionalities.
2. Design Device Drivers, TSR programs and DLL for real world applications.
3. Interpret the methods and techniques about instructions Encoding and Decoding for implementing system-level programs.
4. Deliver the knowledge and techniques in order to bridge the gap between the society and technology.
5. Adapt the skills and ethics to solve critical problems about System design and provide solutions to real world problems.
6. Develop approaches and methods for implementing different system-level software's.

CS31119 : Object Oriented Modeling and Design

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1: Introduction to Modeling

(8 Hrs)

Part A: The importance of modeling, Principles of Modeling, UML Building blocks: things, relationships and diagrams, Architectural views: use case, design, implementation, process and deployment, Levels of detail: visualization, specification and construction, Object properties: Abstraction, Encapsulation, Modularity, Hierarchy, Stereotypes, Tagged Values, Overview of Methodologies: OOAD, OOSE, OMT, Concerns and Aspects in Modeling, UML 2.0 Diagram set

Part B: Methodology: Coad-Yordon, Responsibility-Driven Design, OPM, Catalysis, Aspect-Oriented Modeling

Unit 2: Model Driven Development

(8 Hrs)

Part A: Overview of Model Driven Development and Model Driven Engineering, Model Transformation, Introduction to Model Driven Architecture: MDA Terms and Concepts, Model Mappings, Marking Models, Executable Models, MOF, CWM, Introduction to XML, XMI, Introduction to UML Metamodel, Extensibility Mechanisms and its usage, Introduction to OCL, Model Based Software Engineering

Part B: Domain-Specific Modeling: Fundamentals and Architecture, MDA Applications

Unit 3: Behavior Specification

(8 Hrs)

Part A: Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Using Use Cases, The Domain Perspective, Data Dictionary: Finding the Objects, Responsibilities, Collaborators, and Attributes, CRC Cards, Class Models and Use Case Models, Judging the Domain Model, Capturing system behavior in use cases

Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization

Part B: Study of other Behavioral Diagrams: Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Unit 4: Design Specification

(8 Hrs)

Part A: Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes:

entity, boundary and control , Modeling associations and collections, Preserving referential integrity , Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors
Forward Engineering and Reverse Engineering Concepts
Part B: Application of UML in Real Time and Embedded System, Application of UML in Web Engineering, UML Profiles for other technology disciplines

Unit 5: Design Patterns

(8 Hrs)

Part A: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor
Part B: Antipatterns, Applications of Design Patterns, Archetype Patterns

Text Books:

1. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049
2. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.

Reference Books:

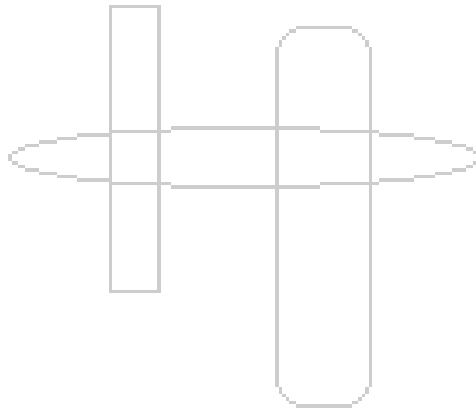
1. Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X
2. Grady Booch, James Rumbaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
3. Erich Gamma, Richard Helm, Ralph Johnson, "Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Steven Kelly, Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using object-oriented analysis principles and Model Driven Development practices.
2. Compose system design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented design principles and Model Driven Engineering practices.
3. Construct and justify the evolutionary system models generated using UML-supported modelling tools.

4. Prepare and present well-documented system profiles to the engineering and social community.
5. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance.
6. Frame system dynamics and construct system specifications in order to realize system artifacts.



CS 30102: Software Engineering

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites: Data Structures

List of Contents

A TERM-WORK containing the record of the following:

1. To study Software Process Models and identify their applicability to various categories of projects.
2. To understand Requirement Elicitation Techniques and recognize types of requirements while preparing System Requirement Specification.
3. To narrate Requirement Definition Document and Software Requirement Specification for target System.
4. To develop all level Data Flow diagrams for the target system indicating problem partitions and solution structure.
5. To apply design principles with relevant architecture style and structure the solution accordingly.
6. To prepare estimation for the System Development using Function Point technique.

Text Books:

1. Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.
2. Roger S Pressman, 'Software Engineering: A Practitioner's Approach', McGraw Hill, 6/e, 2005

Reference Books :

1. Paul Clements, Felix Bachmann, Len Bass, David Garlan, *Documenting Software Architectures: Views and Beyond* Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726
2. Soren Lauesen, *Software requirements: Styles and techniques*, Addison Wesley, ISBN 0201745704, 2002
3. Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*, John Wiley & Sons, Inc., ISBN-13 978-0-471-39333-7, 2006

CS30108:: DESIGN AND ANALYSIS OF ALGORITHMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Data Structures and Files

List of Tutorials

- 1 Introduction to basic set theory, Mathematical Induction. Problem solving based on Induction and Recursion.
- 2 Formally proving correctness of algorithms using induction, loop invariants.
- 3 Studying asymptotic behavior of some non-standard functions like: H_n , $\log(n!)$, $\log^*(n)$, Ackerman function etc. Problem solving based on asymptotic notations.
- 4 Problem solving based on simple binary-search like technique (perfect power testing, finding square roots efficiently etc.)
- 5 Introduction to Adversary lower bound technique and problem solving based on it (E.g. Adversary lower bound for finding two smallest elements in an array)
- 6 Problem solving based on Divide and Conquer technique
- 7 Divide and Conquer technique for problems in Computational Geometry (like convex hull computation, finding closest pair of points, discrete analog of Ham-Sandwich theorem etc)
- 8 Fast Fourier Transform and efficient uni-variate polynomial multiplication.
- 9 Problem solving based on Dynamic Programming strategy
- 10 Introduction to matroids and relation with Greedy strategy
- 11 Efficient implementation of Union-Find data structure and applications.
- 12 Problem solving based on Greedy strategy (formally proving optimality of solution for various greedy based problems)
- 13 Programming assignment on some problem based on Backtracking strategy (E.g. Sudoku solvers, Hi-Q solver, graph coloring etc.)
- 14 Relation between search and decision versions of problems in complexity class NP and their self reducibility property (particularly for SAT, Graph-Isomorphism, Hamiltonian Cycle etc)
- 15 Interesting problems in the complexity class NP intersection coNP, e.g. Linear Programming, Matchings in bipartite graphs, Primality testing.
- 16 Problem solving on Elementary Probability theory.
- 17 Problem solving on Elementary Probability theory.
- 18 Algorithms for enumeration and uniform generation of combinatorial objects (typically for permutations, subsets of fixed size, trees, derangements etc).

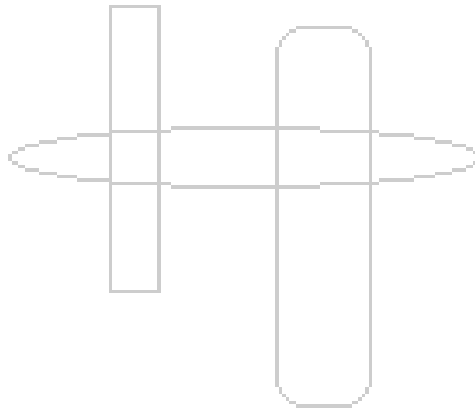
19 Problem solving based on design and analysis of approximation algorithms.

Text Books

1. Jon Kleinberg, Eva Tardos “Algorithm Design”, Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6
2. Horowitz, Sahani, “Fundamentals of computer Algorithms”, Galgotia. 2nd Edition, 1998. ISBN 81-7515-257-5

Reference Books

1. Bressard, Bratley “Fundamentals of Algorithmics.” ,PHI, 2nd Edition, 1996, ISBN 81-203-1131-0
2. Thomas H Cormen and Charles E.L Leiserson, “Introduction to Algorithm” ,PHI 2nd edition, 2001. ISBN 81-203-2141-3



CS30314:: SYSTEMS PROGRAMMING

Credits: 01

Teaching Scheme: - Lab 2 Hrs/Week

List of Assignments:

1. Expanding the Simple Macros with Generating different Parameter Tables and MDT.
2. Expanding the Nested Macros with Generating different Parameter Tables and MDT.
3. Design and implementation of 1 pass and 2 Pass assemblers with generating different data structure for it.
4. Design and implementation of an Editor: Design of a Line or Screen Editor using C Language.
5. Symbol table generation for input *.c file.
6. Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, white spaces.
7. Simulation of linkers.
8. Simulation of loaders.
9. Understanding the design for DLL on Linux shared library.
10. Use of different debugger tools.
11. Printer controller in device drivers.
12. Write a TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT and FUNCTION and display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

13. Write a TSR program in 8086 ALP to implement Screen Saver. Screen Saver should get activated if the keyboard is idle for 7 seconds. Access the video RAM directly in your routine.
14. Write a TSR program in 8086 ALP to handle the “Divide by zero” interrupt. Test your program with a small code, which causes the divide by zero interrupt.
15. Write a TSR program in ‘C’ that would change the color of the screen every 10 seconds.

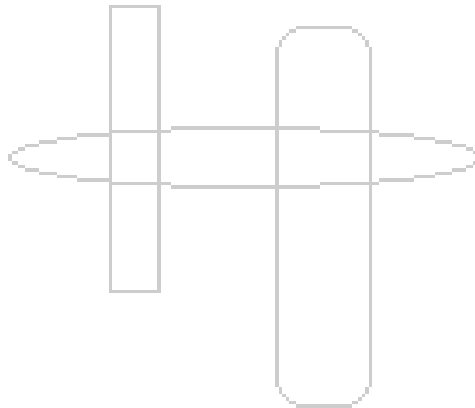
Note: It is expected that student must perform at least 2 assignments from assignment number 12 to 15.

Text Books

1. *“Systems Programming & Operating Systems”, D M Dhamdhare, Tata McGraw Hill Publications, ISBN - 0074635794*
2. *“Systems Programming”, John J Donovan, ISBN - 0070176035*

Reference Books

1. *“Linux System Programming”, Robert Love, O’Reilly, ISBN 978-0-596-00958-8*



CS31319: Object Oriented Modeling and Design

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Structures

List of Practical

1. To narrate Requirement Definition Document for the target system with following three areas:
 - a. Problem Identification
 - b. Problem Definition
 - c. Problem Statement

To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std guidelines.

2. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.

The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass

- a. Actors (External Users)
 - b. Transactions (Use Cases)
 - c. Event responses related to transactions with external agents.
 - d. Detection of System boundaries indicating scope of system.
3. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
 - a. Discrete, distinguishable entities (class).
 - b. Events (Individual stimulus from one object to another).
 - c. Conditional events and relationship representation.
4. To depict the state transition with the life history of objects of a given class model. The model should depict:
 - a. Possible ways the object can respond to events from other objects.
 - b. Determine of start, end, and transition states.

5. To depict the dynamic behavior using detailed Activity diagram.

6. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict

- a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
 - b. Identification of objects and their purpose.
 - c. Roles / responsibilities entities that determine system behavior.
7. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.
8. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. To implement the system according to specification with confirmation to design patterns.

Text Books:

1. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049
2. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.

Reference Books:

1. Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X
2. Grady Booch, James Rumbaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
3. Erich Gamma, Richard Helm, Ralph Johnson, "Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Steven Kelly, Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008

CS30306:: DATABASE MANAGEMENT SYSTEMS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Choose a database system you propose to work on throughout the course. Perform requirements analysis in detail for design of the database. Design an entity-relationship (ER) data model for the selected database system.
2. Convert above ER model to relational model, semi_structured data model. List functional dependencies. Normalize these relations up to 3NF/BCNF.
3. Consider a different database system. List functional dependencies [Include complex business logic.] Apply bottom - up approach using Relational Synthesis Algorithm for design of relational model for the chosen system. Verify decomposition properties.
4. Create tables with appropriate constraints for the relational schema. Create views, indices, and sequence. Alter the schema by adding/removing columns and constraints. Write DML queries.
5. Execute 'SELECT' queries using order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.
6. Write equijoin, non equijoin, self join and outer join queries. Write queries containing single row / multiple row / correlated subqueries using operators like =, in, any, all, exists etc. Write DML queries containing subqueries. Study a set of query processing strategies.
7. Write meaningful stored procedures in PL/SQL. Make use of cursors and different arguments. Write useful stored functions to perform complex computation. Write row level and statement level triggers in PL/SQL.
8. Implement a small database application for the above system using suitable front end and back end tool. Create a transaction by embedding SQL into an application program. Generate different useful reports.
9. Implementation of a small database using NoSQL and/or New SQL database system.

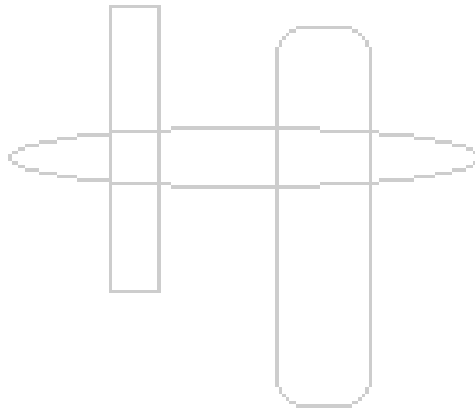
Text Books

1. *"Fundamentals of Database Systems", Elmasri and Navathe, Pearson Education, ISBN 81-297-0228-2, 5th Edition*

2. *"SQL, PL/SQL", Bayross Ivan, BPB Publications New Delhi, ISBN 81-7656-964-X, 3rd Edition*

Reference Books

1. *"Oracle: Applications Development", Bayross Ivan, BPB Publications, New Delhi, ISBN 81-7656-912-7*
2. *"Oracle SQL * Plus", Gennick Jonathan, Shroff Publishers & Distributors P Ltd Mumbai, ISBN 81-7366-606-7*
3. *"Getting Started with NoSQL: Your guide to the world and technology of NoSQL", by Gaurav Vaish*
4. *"Understanding the New SQL: A Complete Guide (The Morgan Kaufmann Series in Data Management Systems)", Jim Melton, Alan R. Simon*



CS37402::MINI PROJECT

Credits: 02

Guidelines:

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

- 1.Evaluate solutions to real world problems and convert them to applications using various programming skills. (PO2)
- 2.Devise software solutions with necessary design documents(PO3)
- 3.Conduct rapid application development using set of modern tools and frameworks(PO5)
- 4.Work in well formed teams with proper communication and share responsibilities in project development according to skill sets and individual strengths.(PO4)
- 5.Concisely estimate cost and effort constraints on problem specific domains.(PO10 & PO11)
6. Develop software solutions based on standard engineering specifications and documents such as requirement analysis, flow diagrams, verification and validations.PSO3

FF No. : 654 B

CS30402: Comprehensive Viva Voce

Credits:2

Guidelines:

1. Student has to present for CVV exam with assignment file duly signed by subject teacher.
2. Student has to give oral/viva exam based on two core subjects they have studied throughout the semester.
3. Examiners examine students by asking questions based on theory and lab syllabus. Students will prepare two subjects and respective subject expert faculty will ask questions to students.
4. Each student will be evaluated by experienced faculty from department and other reputed engineering colleges.
5. To qualify in CVV, student has to pass both the subjects under CVV.

Course Outcomes

Upon completion of the course, graduates will be able to –

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using object-oriented analysis principles and create data models.
2. Compose system design specifications indicating applicable viewpoints using object-oriented design principles and apply normalization techniques, transaction management processes.
3. Construct and justify the evolutionary system models generated using UML-supported modeling tools and solve relational queries.
4. Prepare and present well-documented system profiles to the engineering and social community.
5. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns and database application design.
6. Frame system dynamics and construct system specifications in order to realize system artifacts.

FF No:654 E

CS37301::SEMINAR

Credits: 02

Teaching Scheme: - Lab 2 Hrs/Week

Guidelines:

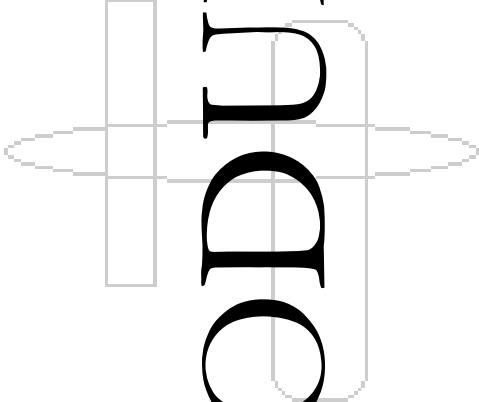
Seminar is a course requirement wherein under the guidance of a faculty member a student is expected to do an in depth study in a specialized area by doing literature survey, understanding different aspects of the problem and arriving at a status report in that area. Students are expected to choose a topic in CSE based on current trends or industry practices. While doing a seminar, the student is expected to learn investigation methodologies, study relevant research papers, correlate work of various authors/researchers critically, study concepts, techniques, prevailing results etc., analyze it and present a seminar report. Evaluation will be based on relevance of topic, understanding of the problem, literature Survey, presentation, communication skills, answering queries and reporting or documenting procedure.

Course Outcomes:

Upon completion of the course, graduates will be able to -

- 1 Scope and visibly identify technological trade-off in computer engineering leading to significant topics.
- 2 Conduct a thorough literature survey of identify technical topic.
- 3 Present technical topic in written form with technical report or document
- 4 Communicate effectively technical topic in verbal form with suitable demonstration
- 5 Access real world problem scenarios in computer engineering.
- 6 Demonstrate skills and competences with an awareness of technical standardization.

MODULE VII



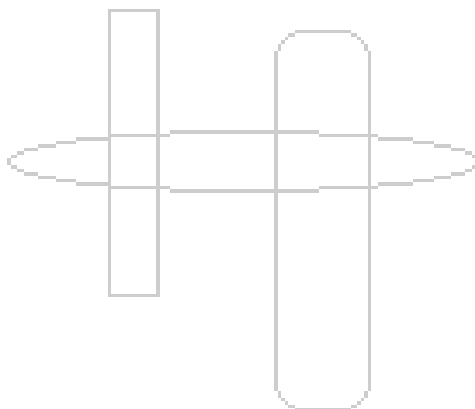
FF No. 653

Issue 4, Rev. 1 , Dt. 01/03/2014

B. Tech. Structure with effect from Academic Year 2015-16
Module VII

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS40115	Distributed Computing	S ₁	3	-	1	10	20	5	5	-	60	4
CS40119	Business Intelligence and Analytics	S ₂	3	-	-	15	20	5	-	-	60	3
CS42101 CS42127 CS42103 CS42105 CS42131 CS42106	ELECTIVE I Advanced Computer Graphics Cloud Computing Parallel Computing on GPU Mobile Computing Enterprise Systems Networks Security	S ₃	3	-	-	15	20	5	-	-	60	3
CS42125 CS42113 CS42119 CS42114 CS42115 CS42134	ELECTIVE II Randomized and Approximation Algorithms Digital Signal Processing Information Retrieval Product Design Human Computer Interaction Modeling and Simulation	S ₄	3	-	1	10	20	5	5	-	60	4
CS40314	Business Intelligence and Analytics	P ₁	-	2	-					30	70	1

	Elective I	P ₂	-	2	-					30	70	1
CS47303	PROJECT STAGE 2 (B.Tech. Semester I) Irrespective of Module	PS ₂	-	8	-					30	70	4
TOTAL			12	12	2							20



CS40115:: DISTRIBUTED COMPUTING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Operating Systems.

Unit 1: (8+1 Hrs) **Introduction**

Part A: Introduction to Distributed Systems, Motivation, Examples of Distributed Systems, Hardware and Software Concepts, Design issues.

System models: Introduction, Architectural Model, Fundamental Models.

Introduction to Hadoop/MapReduce.

Part B: Conceptual design of a Distributed File System based on concepts learned in Unit 1:. Reading: Hadoop/MapReduce.

Unit 2: (8+1 Hrs) **Communication**

Part A: Interprocess Communication: Communication primitives: Blocking/non-blocking, synchronous/asynchronous primitives, Message Oriented Communication, Stream Oriented Communication.

RPC: RPC Model, Transparencies in RPC, Implementation, Stub Generation, RPC Messages, Server Management, Call Semantics, Communication Protocols, Distributed Objects: Remote Method Invocation.

Part B: Case studies of existing RPC implementation. Design a RPC/RMI mechanism based on the case studies and concepts learned in class.

Unit 3: (8+1 Hrs) **Synchronization**

Part A: Time and Global States: Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State, Event Ordering.

Co-ordination: Election Algorithm: Bully Algorithm, Ring Algorithm, Mutual Exclusion: Requirements, Performance metrics, Centralized Approach, Lamport's algorithm, Distributed Approach (Ricart and Agrawala)

Distributed Deadlock Algorithms for Avoidance, Prevention, and Detection: Classification of distributed deadlock detection algorithms, Centralized Approach, Hierarchical Approach, WFG Based Fully Distributed, Deadlock Recovery.

Part B: Comparative analysis of time synchronization/mutual exclusion/election algorithms implementations in well-established distributed systems.

Unit 4:

(8+1 Hrs)

Fault Tolerance

Part A: Introduction to Fault Tolerance, Failure Models, Failure Masking by Redundancy: Triple Modular Redundancy.

Process Resilience: Design Issues, Failure Masking and Replication, Agreement in Faulty Systems: Two Army Problem, Byzantine Generals Problem.

Reliable Client Server Communication, Reliable Group Communication.

Part B: Identify the issues that can arise in your DFS prototype in case of various failures. Based on the concepts learned in Unit 1:V design a fault tolerance mechanism for the DFS.

Develop test-case scenarios to assess the system's resilience.

Unit 5:

(8+1 Hrs)

Distributed Transaction and Distributed Shared Memory

Part A: Distributed Transaction: Transaction Model, Classification, Implementation, Concurrency Control: Serializability, 2 Phase Locking, Strict 2 PL, Distributed Commit: 2 Phase Commit, Recovery

Distributed Shared Memory: Introduction, Advantages, Disadvantages, Architecture of DSM Systems, Design and Implementation Issues of DSM: Granularity, Structure of Shared Memory Space, Memory Consistency Models, Replacement Strategies, Thrashing,

Part B: Distributed File Systems :Data Intensive Computing, Google FS, BigTable

Text Books

1. *"Distributed Computing: Principles, Algorithms, and Systems"*, Ajay Kshemkalyani, Mukesh Singhal, Cambridge University Press, ISBN-13: 978-0-521-87634-6, 1st Edition.
2. *"Distributed Systems Principles and Paradigms"*, Andrew S. Tanenbaum & Maarten Van Steen, Prentice Hall India, ISBN 81-7808-789-8, 1st Edition.

Reference Books

1. *"Distributed Operating Systems Concepts and Design"*, Pradeep K. Sinha, Prentice Hall India, ISBN: 81-203-1380-1, 1st Edition
2. *"Distributed Systems – Concepts and Design"*, George Coulouris, Jean Dollimore & Tim Kindberg, Addison-Wesley, ISBN: 0-201-61918-0, 3rd Edition

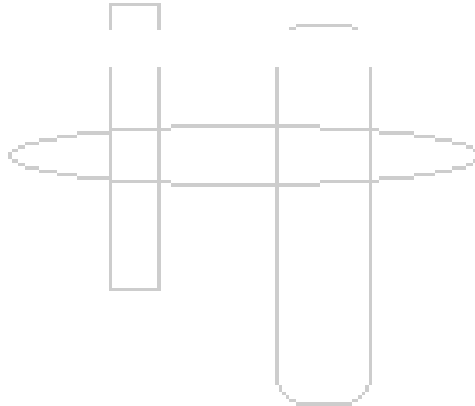
Additional Reading

1. *"Advanced Concepts In Operating Systems"*, Mukesh Singhal, Niranjana G. Shivaratri, McGrawHill, ISBN-13: 978-0-07-047268-6.
2. *"Distributed and Parallel Systems: From Cluster to Grid Computing"*, Peter Kacsuk, Thomas Fahringer, Zolt Nemeth, Springer, ISBN: 978-0-378-69857-1, 2007.
3. *"Distributed Computing: Principles and Applications"*, M. L. Liu, Addison-Wesley, ISBN-10: 0201796449, 2004.
4. *"Distributed Computing::Fundamentals, Simulations and Advanced Topics"*, Hagit Attiya, Jennifer Welch, A John Wiley & Sons, Inc, ISBN 0-471-45324-2.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Develop the solutions for Communication, Synchronization, Mutual exclusion and Deadlock handling in distributed computing.
2. Identify the basic principles, design requirements and challenges associated with implementing large-scale distributed system.
3. Design of distributed application using modern tools in Distributed System.
4. Construct an optimal and cost-effective solution without compromising the security and reliability of the system with respect to convenience, efficiency and the ability to evolve.
5. Build a basic prototype distributed system for different applications in a team.
6. Apply principles of distributed computing while building appropriate variations of existing solutions to meet the development contexts.



FF No. : 654 A

CS40119:: BUSINESS INTELLIGENCE AND ANALYTICS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Database Management Systems.

Unit 1: (8+1 Hrs)

Part A:

Introduction: What is business intelligence and analytics? Need for BI&A, how businesses use BI&A. Evolution of BI&A. Interplay among Business Intelligence, Business Analytics, Data Science, Data Mining, Data Analytics, Data Warehousing, Statistics and Machine Learning. Drawing insights from data: DIKW pyramid, Categorization of Analytics: DIPP. Concept of a model and its characterization.

Data Warehousing & Multi-Dimensional Modeling: What is a data warehouse, need for a data warehouse, four characteristics of a data warehouse, components/architecture of a data warehouse, data marts, building a data warehouse. OLTP vs OLAP.

Multidimensional Modeling: ER Modeling vs Multi-Dimensional Modeling , Data cubes, OLAP operations, lattice of cuboids, multi-dimensional schemas (star, snowflake, galaxy), facts, dimensions, concept hierarchies, types of dimensions, measures

Data Integration: different types of digital data and their sources, ETL (extract-transform-load)

Enterprise Reporting: Metrics, Measurement, Measures, KPIs, Dashboards, Reports, Scorecards

Part B: Examples of business analyses – funnel analysis, distribution channel analysis and performance analysis; levels of decision making (strategic, tactical and operational BI); types of facts; types of measures; different OLAP architectures

Unit 2: Descriptive Analytics - I (8+1 Hrs)

Part A:

Data Preprocessing: mechanisms of data collection and challenges involved therein. Typical preprocessing operations: combining values into one, handling incomplete or incorrect data, handling missing values, recoding values, subsetting, sorting, transforming scale, determining percentiles, data manipulation, removing noise, removing inconsistencies, transformations, standardizing, normalizing - min-max normalization, z-score standardization, rules of standardizing data.

Data visualization: role of visualization in analytics, different techniques for visualizing data based on the nature of data and what kind of insights need to be drawn

Descriptive Statistics: role of statistics in analytics, types of data (scales of measurement - NOIR), data distributions, measures of variability (range, quartile, five number summary, variance, std dev, coeff of variation), analyzing distributions, Chebychev's Inequality, measures of shape (skewness, kurtosis), measures of association (covariance, correlation), outliers

Part B: Data quality and its various aspects, measures of central tendency (arithmetic mean, geometric mean, harmonic mean, median, mode)

Unit 3: Descriptive Analytics - II

(7+1 Hrs)

Part A:

Inferential Statistics: Role of probability in analytics. Need for sampling, generating samples, sampling and non-sampling error. Sampling Distribution of Mean, Central Limit Theorem, Standard Error.

Estimation: Point and Interval Estimates, Confidence Intervals, level of confidence, sample size.

Hypothesis Testing: basic concepts, Errors in hypothesis testing, Power of test, Level of significance, p-value, general procedure for hypothesis testing. Parametric tests – z test, t test, chi-square test. Hypothesis testing of means: two tailed and one-tailed tests. Chi-square test for independence and goodness of fit. Hypothesis testing for comparing two related samples. Limitations of hypothesis testing. Picking up the right test for a given scenario.

Part B: Common probability distributions and their characteristics, parametric vs non-parametric tests, ANOVA

Unit 4: Predictive Analytics

(8+1 Hrs)

Part A:

Similarity Measures: Design of recommender systems - user based and item based collaborative filtering

Modelling: Data Modelling Basics, Logic driven modeling, data driven modeling

Regression Analysis: Correlation and regression, Simple Linear Regression Model,

Least Squares Method. Making Data Models more flexible, making data models more selective, dealing with Categorical variables, Interpretation of regression coefficients, fine tuning data models (assessing the fit, model fitting), Coefficient of determination, Significance tests, Residual analysis, Prediction intervals. Model evaluation techniques. Assumptions of regression analysis.

Introduction to Time Series Analysis and Forecasting: Time series patterns, forecast accuracy, moving averages and exponential smoothing, casual models, using regression analysis for forecasting, determining best forecast model to use, ARIMA models

Part B: Basic *what-if* spreadsheet models, different types of regression and which is used when, modeling non-linear relationships, heteroscedasticity, multicollinearity

Unit 5: Prescriptive Analytics & Emerging Trends in BI&A **(8+1 Hrs)**

Part A:

Optimization Analytics: Overview of simulation and risk analysis, Linear Optimization Models (linear programming), Integer Linear Optimization models (integer programming), Non-linear optimization models (portfolio theory), Monte Carlo Simulation, Decision Analysis

BigData Analytics: What is Big Data, sources of BigData, MapReduce, Hadoop, statistical analysis of big data, visualizing bigdata

Familiarity with the following areas: multi variate analytics, text analytics, web analytics, social media analytics, in-memory analytics – row vs columnar databases, in-memory databases

Part B: NoSQL databases for unstructured data and its role in analytics, analytics in the cloud, embedded BI, mobile analytics, mobile BI, Data Stream Analytics, Customer Centricity, Patient Centricity, IoT, Self-Service BI, Fog networks

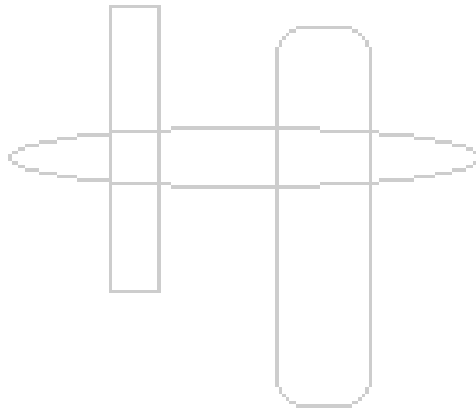
Text Books

1. *"Business Analytics"* by James R Evans, Pearson
2. *"Data Mining: Concepts and Techniques"*, Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition
3. *"Fundamentals of Business Analytics"*, by R. N. Prasad, Seema Acharya, ISBN: 978-81-256-3203-2, Wiley-India
4. *"Essentials of Business Analytics"* by Camm, Cochran, Fry, Ohlmann, Anderson, Sweeney, Williams, Cengage Learning
5. *"The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence"* 2010, Ralph Kimball, Margy Ross, Wiley Publications

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Utilize both descriptive and inferential statistical concepts in deriving useful insights from data
2. Evaluate various data pre-processing techniques to enable better data analysis in case of complex problems
3. Construct an end-to-end analytics solution for business intelligence involving various data sources, ETL, multi-dimensional modelling, OLAP, reporting and analytics
4. Demonstrate appropriate data visualization techniques for effective communication
5. Follow the research and technological advancement in the field of business intelligence and analytics
6. Apply various descriptive, predictive and prescriptive data analytics techniques to address real world challenges by making use of available data



FF No. : 654 A

CS42105:: MOBILE COMPUTING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

- Communication Engineering
- Computer Network

Unit 1: (8+1 Hrs)
Cellular Network

Part A: Introduction, Generations (1G,2G,3G,4G), Why cellular networks? Basic cellular system, Frequency allocation, System capacity and frequency re-use, Concept of frequency reuse channels, Co-channel interference and its reduction factor. Types of non co-channel interference, Cell splitting.

Part B: Signal and signal propagation, IEEE 802.11 standard, 802.11a,b,g, 802.15,

Unit 2: (8+1 Hrs)
GSM Communication

Part A: System Architecture: GSM Radio subsystem, Interfaces, Network and switching subsystem, Operation subsystem. GSM channels, GSM protocol architecture, Location tracking and call setup, Security, Data services N/W signaling, GSM mobility management, Administration and maintenance.

Handoff- Initialization of handoff, Delaying handoff, Forcing handoff, Power different handoff. Mobile assisted handoff, Intersystem handoff.

Part B: Survey of GSM network, Hard, Soft Handoff.

Unit 3: (9+1 Hrs)
GSM Bearer Services

Part A: SMS architecture protocol, Hierarchy, VOIP services for mobile networks.

WAP: model and architecture, Gateway, protocol stack.

Telecommunication system: GPRS, wireless in local loop, DECT, EDGE, UMTS, Paging systems, CDPD.

Part B: Wireless application environment, Bluetooth.

Unit 4: (8+1 Hrs)
Mobile Network and Transport layer

Part A: Mobile Network layer: Mobile IP, Mobile node, IP packet delivery, Agent discovery, Registration, Tunneling and encapsulation, Mobile ad-hoc networks.

Mobile Transport layer: Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/recovery, Transmission/time-out freezing, Selective retransmission, Transaction-oriented TCP.

Part B: TCP over 2.5/3G wireless networks.

Unit 5:

(7+1 Hrs)

Mobile Databases

Part A: Database hoarding, Data caching, Data cache and web cache maintenance in mobile environments, Client-Server computing and adaptation, Query processing, Data recovery process, Issues relating to quality of service, Digital audio broadcasting: DAB System, DAB objects, Object transfer protocol, DVB: DVB system.

Part A: Mobile Billing

Text Books

1. *"Mobile Communications"*, Jochen Schiller, 2nd edition, Pearson education, ISBN- 81-297-0350-5
2. *"Mobile Communication"*, G.K.Behera, Lopamudra Das, Scitech publications, ISBN - 9788183711791

Reference Books

1. *"Wireless and Mobile Network Architectures"*, Yi-Bing Lin, Imrich Chlamtac, Wiley publication, ISBN-9971-51-366-8
2. *"Mobile Computing"*, Raj Kamal, Oxford University press, ISBN 978-0-19-568677-7

Course Outcomes:

Upon completion of the course, the students will be able to-

1. Estimate performance parameters for designing the Cellular Network which comply with 2G and 3G Cellular Network Standards.
2. Formulate conceptual Telecommunication system to be deployed to fulfil bandwidth capacity planning.
3. Design the mobile network considering futuristic requirements which conforms to the technology safety standards.
4. Justify the Mobile Network performance parameters and design decisions.
5. Adapt to the requirements of next generation mobile network and mobile applications.
6. Develop scalable utility based Mobile Applications for user communities incorporating portability.

FF No. : 654 A

CS42131: Enterprise Systems

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Software Engineering, Database Management System

Unit 1:: Business Process Management

(8 Hrs)

Part A: Business Process Modeling Foundation: Conceptual Model and Terminology, Abstraction Concepts, From Business Functions to Business Processes, Activity Models and Activity Instances , Process Models and Process Instances , Process Interactions, Modeling Process Data /Organization / Operation, Business Process Flexibility

Process Orchestrations: Control Flow Patterns, Workflow Nets, Business Process Modeling Notation

Process Choreographies: Motivation and Terminology, Development Phases, Process Choreography Design and Implementation, Service Interaction Patterns

Properties of Business Processes: Data Dependencies, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness, Soundness Criteria Overview

Part B: Petri Nets, Event-driven Process Chains, Yet Another Workflow Language, Graph-Based Workflow Language

Unit 2:: SOA Fundamentals

(8 Hrs)

Part A: Service-Oriented Computing and SOA , Introduction to Service-Oriented Computing, The Evolution of SOA, Principles of Service-Oriented Computing, Goals and Benefits of Service-Oriented Computing, Service-Oriented Computing, Problems Solved by Service-Oriented Computing, Challenges Introduced by Service-Oriented Computing, Effects of Service-Oriented Computing on the Enterprise, Origins and Influences of Service-Oriented Computing, Understanding Design Principles, Principle Profiles, Design Pattern References, Principles and Design Granularity

Service Contracts (Standardization and Design): Contracts principles, Types of Service Contract Standardization, Contracts and Service Design, Versioning, Technology / Development Tool Dependencies

Service Coupling (Intra-Service and Consumer Dependencies): Coupling principles, Service Contract Coupling Types, Service Consumer Coupling Types, Service Loose Coupling and Service Design, Enterprise Service Bus, Web Services and Primitive SOA, Web Services and Contemporary SOA, Service Layers

Part B: XML; HTTP; SOAP

Unit 3:: SOA Design Principles

(8 Hrs)

Part A: Service Abstraction (Information Hiding and Meta Abstraction Types): Abstraction principles, Types of Meta Abstraction, Measuring Service Abstraction, Service Abstraction and Service Design, Risks Associated with Service Abstraction

Service Reusability (Commercial and Agnostic Design): Reuse Principle, Service Reuse in SOA, Service Reusability and Service Design

Service Autonomy (Processing Boundaries and Control): Autonomy Principle, Types of Service Autonomy, Measuring Service Autonomy, Service Contract Autonomy (services with normalized contracts) Autonomy and Service Design
Service Statelessness (State Management Deferral and Stateless Design): State Management, Measuring Service Statelessness, Statelessness and Service Design
SOA Delivery Strategies, Service-Oriented Analysis: Introduction, Service Modeling, Service-Oriented Design: Introduction, SOA Composition Guidelines), Service Design
Part B: Importance of WSDL, SOAP, The use of registries via UDDI

Unit 4:: SOA Technology and Implementation (8 Hrs)

Part A: Service Discoverability (Interpretability and Communication): Discoverability, Types of Discovery and Discoverability, Measuring Service Discoverability, Discoverability and Service

Service Composability (Composition Member Design and Complex Compositions): Composition, Composition Concepts and Terminology, Complex Service Composition, Measuring Service Composability and Composition Effectiveness Potential, Composition and Service Design, Service-Oriented and Object-Oriented, Mapping Service-Oriented Principles to Strategic Goals

Part B: SOA Platforms, SOA support in .NET and J2EE platforms

Unit 5:: Enterprise Architecture (8 Hrs)

Part A: Introduction to Enterprise Architecture, State of the Art and Foundations of Enterprise Architecture, Communication of Enterprise Architecture, Language for Enterprise Modeling, Viewpoints and Visualization, Architecture Analysis, Architecture Alignment, Tool Support, Domain-Driven Architecture, Resource-Oriented Architecture, Defining EAI, The EAI Process, Data-Level EAI, Application Interface-Level EAI, Method-Level EAI, User Interface-Level EAI

Part B: Enterprise Architecture Frameworks: The Open Group Architecture Framework (TOGAF), Zachman Framework for Enterprise Architecture Extended Enterprise Architecture Framework (E2AF), Department of Defense Architecture Framework (DoDAF), Federal Enterprise Architecture Framework (FEAF), Treasury Enterprise Architecture Framework (TEAF)

Text Books:

1. *Mathias Weske, Business Process Management, Concepts, Languages, Architectures, ISBN 978-3-540-73521-2 Springer Berlin Heidelberg New York, 2007*
2. *Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design. ISBN: 0-13-185858-0, Publisher: Prentice Hall PTR, 2005*

Reference Books :

1. *Thomas Erl, SOA Principles of Service Design, Pearson Education, Inc., ISBN 0-13-234482-3, 2007*
2. *Eric A. Marks, Michael Bell., Executive's guide to service-oriented architecture, John Wiley & Sons, Inc. ISBN-13: 978-0-471-76894-4, 2006*
3. *Daniel Minoli, Enterprise Architecture A to Z, Frameworks, Business Process Modeling, SOA, and Infrastructure Technology, Auerbach Publications, Taylor & Francis Group, ISBN 978-0-8493-8517-9, 2008*

4. *Setrag Khoshafian, Service Oriented Enterprises, Auerbach Publications, Taylor & Francis Group, ISBN 0-8493-5360-2, 2007*
5. *Mike Rosen, Boris Lublinsky, Kevin T. Smith, Marc J. Balcer, Applied SOA: Service-Oriented Architecture and Design Strategies, Wiley Publishing, Inc., ISBN: 978-0-470-22365-9, 2008*
6. *Marc Lankhorst et al., Enterprise Architecture at Work, Modelling, Communication and Analysis, Second Edition, ISBN 978-3-642-01309-6, Springer-Verlag Berlin Heidelberg 2009*
7. *David S. Linthicum, Enterprise Application Integration Addison-Wesley Professional 2003, ISBN-10: 1402052626*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Model business requirements and business processes using BPMN 2.0 standard encompassing Process Orchestrations and Choreographies.
2. Discover the set of services with composite services creation and designing services to facilitate integration and understand interrelationships among SOA, Web Services, OOD and IT infrastructure.
3. Explore the concepts, guidelines and technology for service orchestration to integrate a Business Process Management Solution in an Enterprise SOA in societal context.
4. Prepare well-formed specifications and reports for service composition and delivery to the stakeholders.
5. Understand case studies and lessons learned with utilization of Enterprise Architecture Integration and Frameworks knowledge towards planning and implementing complex enterprise projects.
6. Create sustainable Enterprise System design supported by enterprise modelling, architecture analysis and alignment.

CS42127:: CLOUD COMPUTING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks, Operating Systems, Web Technologies

Unit 1: Introduction

(6+1 Hrs)

Part A: Overview of computing paradigm: Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing. Evolution of cloud computing - Business driver for adopting cloud computing

Introduction to Cloud Computing: Cloud Computing - Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers. Properties, Characteristics & Disadvantages - Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing. Role of Open Standards.

Part B: Study of top 5 cloud providers and top 5 applications on cloud

Unit 2: Architecture and IaaS

(8+1 Hrs)

Part A: Cloud Computing Architecture: Cloud computing stack - Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services. Service Models (XaaS) - Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS). Deployment Models, Public cloud, Private cloud, Hybrid cloud, Community cloud

Infrastructure as a Service (IaaS): Introduction to IaaS - IaaS definition, Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine(VM). Resource Virtualization - Server, Storage, Network. Virtual Machine(resource) provisioning and manageability, storage as a service, Data storage in cloud computing(storage as a service). Renting, EC2 Compute Unit, Platform and Storage, pricing, customers.

Part B: Study top 5 open source tools for building private cloud along with pros/cons

Unit 3: PaaS and SaaS

(7+1 Hrs)

Part A: Platform as a Service (PaaS): Introduction to PaaS - What is PaaS, Service Oriented Architecture (SOA). Cloud Platform and Management - computation, storage

Software as a Service (SaaS): Introduction to SaaS, Web services, Web 2.0, Web OS, Case Study on SaaS

Part B: Innovative applications of cloud computing

Unit 4: Cloud Management and Security

(10+1 Hrs)

Part A: Service Management in Cloud Computing: Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling: Benefitting enormously, Managing Data - Looking at Data, Scalability & Cloud Services, Database & Data Stores in Cloud, Large Scale Data Processing

Cloud Security: Infrastructure Security - Network level security, Host level security, Application level security. Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location: Identity & Access Management, Access Control, Trust, Reputation, Risk, Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations

Part B: Study the cloud deployment of a big enterprise

Unit 5: Cloud Solutions

(9+1 Hrs)

Part A: Case study on Open Source and Commercial Clouds – Amazon EC2, Google Compute Engine, Microsoft Azure, Cloudfoundry, OpenStack

Part B: How the CDNs (Content Delivery Networks) make use of the cloud?

Text Books

1. *Cloud Computing for Dummies* by Judith Hurwitz, R. Bloor, M.Kanfman, F.Halper (Wiley India Edition).
2. *Enterprise Cloud Computing* by Gautam Shroff, Cambridge.
3. *Cloud Security* by Ronald Krutz and Russell Dean Vines, Wiley-India

Reference Books

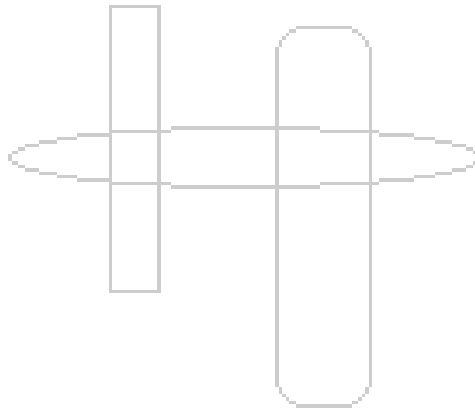
1. *Google Apps* by Scott Granneman, Pearson.
2. *Cloud Security & Privacy* by Tim Malhar, S.Kumaraswamy, S.Latif (SPD, O'REILLY)
3. *Cloud Computing : A Practical Approach*, Antohy T Velte, et.al McGraw Hill,

4. *Cloud Computing Bible* by Barrie Sosinsky, Wiley India.
5. *Cloud Computing*, Michael Miller, Que Publishing

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Illustrate the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
2. Investigate the cloud computing model to be deployed for a given business case
3. Choose the appropriate technologies, algorithms, and approaches for reducing carbon footprint
4. Differentiate real world SaaS solution architectures collaboratively
5. Employ cost engineering principles while building the SaaS solution
6. Build a SaaS solution for a real world problem with collaborative efforts



CS42103:: PARALLEL COMPUTING ON GPU

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+1 Hrs)

Parallel programming basics

Part A: Parallel programming definition, motivation, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn's classification of multi-processors, Motivation for heterogeneous programming, Definition of thread and process, programming parallel computers- extend a compiler, extend a sequential programming language, add a parallel programming layer, create a parallel language.

Part B: multiprocessor and multicomputer systems, interconnection networks

Unit 2: (8+1 Hrs)

Introduction to GPU, architecture and parallel algorithms

Part A: Introduction to GPU computing, motivation, Modern GPU architecture case study: NVIDIA Fermi Tesla C2050/Kepler K20, GPU memories- global, shared, texture memory and their properties and uses, roles of CPU and GPU in parallel computing, GPU computing domain areas and success. Parallel algorithm design. Speedup and scalability.

Part B: parallel algorithm design for data clustering, theory of locality of reference

Unit 3: (8+1 Hrs)

Compute Unified Device Architecture (CUDA)

Part A: CUDA Architecture, CUDA programming model, execution model, thread organization: Concept of grid, block and thread, thread index generation, warp; memory model: Introduction to global, shared, local memories, usage of cache, texture cache, constant memory, memory banks and bank conflicts, memory coalescing,. CUDA structure and API details. CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc).

Part B: atomic operations in CUDA

Unit 4: (8+1 Hrs)

Problem solving using GPUs

Part A: Single vss double precision, solving problems that involves Vectors, Matrices, Binomial coefficients, Bernstein coefficients and etc. Instructor will choose the problems from several domains.

Part B: Study problems given by instructor

Unit 5:

(8+1 Hrs)

Parallel reduction and Tools

Part A: Reduction operation using prefix sum example. Performance issues in algorithms- deciding parallelization of a part of algorithm and selecting the highest parallelism, Need of profilers, Introduction to CUDA Tools: MemCheck and & Visual Profiler.

Part B: Memory leaks and associated problems

Text Books

1. *CUDA: Programming Massively Parallel Processors: A Hands-On Approach*. Authors: David Kirk, Wen-mei Hwu © ELSEVIER Inc.
2. *CUDA by Example: An Introduction to General-Purpose GPU Programming* by Jason Sanders and Edward Kandrot
3. *Parallel Programming in C with MPI and OpenMP* by Michael J. Quinn, Tata McGraw-Hill Edition
4. *Advanced computer architecture* by Kai Hwong, Tata McGraw-Hill Edition, 2001

References:

1. <http://developer.nvidia.com/>
2. *www tutorials on introduction to parallel computing*
3. *Other references suggested by instructor*

Course Outcomes

Upon completion of the course, graduates will be able to -

1. Recognize various parallel computing architectures and their fundamentals
2. Investigate GPU parallel solutions to complex real world problems
3. Compare serial and parallel executions with professional ethics
4. Code and optimize the parallel programs on GPU using CUDA
5. Apply parallel computing methods to research oriented problems and measure the success of CUDA projects
6. Design and develop new solutions to research problems

CS42106:: NETWORKS SECURITY

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks.

Unit 1: (8+2 Hrs)

Introduction

Part A: Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services.

Trustworthiness, Ethical issues and practices, Tradeoffs of balancing key security properties - Confidentiality, Integrity, Availability.

Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack.

Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks

Part B: SYN-Flooding, SQL- injection, DNS poisoning, Sniffing, Sabotage flashing attack, Security of Embedded Devices.

Unit 2: (8+2 Hrs)

Private key cryptography

Part A:Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).

Role of random numbers in security, Importance of prime numbers

Data Encryption Standard: Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.

Part B: Chinese remainder theorem, Elementary Ciphers (Substitution, Transposition and their Properties), Frequency analysis

Unit 3: (8+2 Hrs)

Public key cryptography

Part A: RSA: RSA algorithm, Key generation in RSA, attacks on RSA.

Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.

Elliptic Curve Cryptography (ECC): Elliptic Curve over real numbers, Elliptic Curve over \mathbb{Z}_p , Elliptic Curve arithmetic. Diffie-Hellman key exchange using ECC.

Message authentication and Hash Function.

Part B: Diffie-Hellman key exchange on EC Groups, digital signature, EC- based digital signature, Digital Forensics.

Unit 4: (8+2 Hrs)

Authentication and access control

Part A: Authentication: One-Way Authentication, Mutual Authentication, Dictionary Attacks, Centralized Authentication, The Needham-Schroeder Protocol.

Authentication Applications: Kerberos, X.509 authentication service, public key infrastructure.

Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control.

Part B: Authentication and authorization tools, Biometrics.

Unit 5:

(8+2 Hrs)

Security application and design

Part A: Network layer security: IPSec for IPV4 and IPV6.

Transport layer security: SSL and TLS.

Application layer security: Security services, S/MIME, PGP, PEM, Https, Honey pots.

Security design: End-to-end security, Security composability, Open design, Cost and tradeoffs

Part B: Cyber laws, Cyber Security, IDS, SNORT, Firewall, Cloud Security Tradeoffs.

Text Books

1. *“Cryptography and Network Security-Principles and Practices”* by William Stallings, Pearson Education, 2006, ISBN 81-7758-774-9, 4th Edition.
2. *“Network Security and Cryptography”*, by Bernard Menezes, Cengage Learning, 2010, ISBN 81-315-1349-1, 1st Edition.

Reference Books

1. *“Computer Security: Art and Science”*, by Matt Bishop, Pearson Education, 2002, ISBN 0201440997, 1st Edition.
2. *“Network security, private communication in a public world”*, by Charlie Kaufman, Radia Perlman and Mike Spencer, Prentice Hall, 2002, ISBN 9780130460196, 2nd Edition.

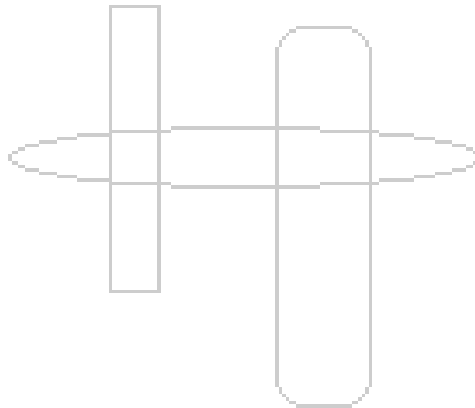
Additional Reading

1. *“Security architecture, design deployment and operations”*, by Christopher M. King, Curtis Patton and RSA press, McGraw-Hill, 2001, ISBN 0072133856, 1st Edition.
2. *“Inside Network Perimeter Security”* by Stephen Northcott, Leny Zeltser, et al, Pearson Education Asia, ISBN 8178087618, 1st Edition.

Course Outcomes

Upon completion of the course, the students will be able to:

1. Identify the various types of attacks by analyzing the behavior of data in the networks
2. Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization
3. Investigate the vulnerabilities in the existing system for the development of new system by following laws of security.
4. Perform authentication of individuals or groups with cost effective Engineering solutions.
5. Examine the importance of network security applications for current and future needs of organizations.
6. Estimate needs of security for a system by researching current environment on a continuous basis for the benefit of society



CS42101:: ADVANCED COMPUTER GRAPHICS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Graphics

Unit 1: (8+1 Hrs) **OpenGL**

Part A: OpenGL over windows, SDK, Extensions, GLUT, GLU, OpenGL primitives, Programming language: Blending, 3D viewing(camera analogy), Lighting model, Culling, Fog, Texture mapping.

Part B: OpenGL over Linux, pBuffer rendering, Shadowing Techniques.

Unit 2: (8+1 Hrs) **Graphics Hardware**

Part A: Graphics card, Components of graphics card, General-Purpose computation, GPU, GPGPU, CUDA architecture, CUDA Programming: Type qualifiers, Execution configuration, vector types, functions, APIs.

Part B: How graphics card works? GeForce 6800 series GPU architecture.

Unit 3: (8+1 Hrs) **Advanced Rendering Techniques**

Part A: Point based rendering, Mesh Simplification, Spatial partitioning, Solid Modeling, Subdivision surfaces: Catmull-Clark subdivision, Subdivision rules, Visibility Computation: culling types, cells and portals, hardware support.

Part B: Splines , Tessellation, 3D viewing.

Unit 4: (8+1 Hrs) **Photorealistic and Volume Rendering**

Part A: Global Illumination, Normals and reflections, Ray-Tracing algorithm and mathematical formulation, Photon mapping algorithm, Monte Carlo algorithm, Volume graphics overview, Direct volume rendering, Marching square algorithm.

Part B: Monte Carlo mathematical formulation, Marching cubes algorithm.

Unit 5: (8+1 Hrs) **Texture Synthesis and Image Processing**

Part A: Texture synthesis, Image processing: Digital image representation, Image data structures, Sampling and Quantization, Image enhancement in spatial domain.

Part B: Image compression, Image synthesis.

Text Books

1. *"Procedural Elements for Computer Graphics"*, Rogers David F., Tata McGraw Hill, 2001, ISBN 0-07-047371-4.
2. *"OpenGL Programming Guide: The Official Guide to Learning OpenGL"*, Mason Woo, Jackie, Tom Davis, Version 2.1, 6th Edition, Pearson Education, ISBN 978-81-317-2184-1.

Reference Books

1. *"Computer Graphics"*, M Paulin Baker, Donald Hearn, PHI, 2001, ISBN 81-203-09440-8.
2. *"Digital Image Processing"*, Rafael C. Gonzalez, Richard E. Woods, 2nd Edition, Prentice Hall, ISBN: 0-201-18075-8.

Additional Reading

- "Computer graphics: A Programming approach"*, Harrington Steven, Tata McGraw Hill, ISBN 0-07-026753-7.
- "Advanced Animation and Rendering Techniques ÷ Theory and Practice"*, Alan H. Watt and Mark Watt, Addison-Wesley, acm press, ISBN: 0201544121.
- "CUDA by Example: An Introduction to General-Purpose GPU Programming"*, Jason Sanders, Edward Kandrot, Addison-Wesley, ISBN-13: 978-0-13-138768-3.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify data structures and algorithms of 3D modeling, visibility and rendering.(PO1)
2. Develop algorithms/programs for 2D and 3D graphics using OpenGL (PO5)
3. Demonstrate graphics hardware for its functions and programming (PO6)
4. Interpret algorithms of global illumination and 3D volume rendering(PO9)
5. Adapt use of image or texture in 2D/3D graphics in currently available applications.(PO12)
6. Build realistic 2D/3D graphics models based on either technological availability or through creativity. (PSO4).

CS42125:: Randomized and Approximation Algorithms

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Unit 1: Basic probability theory

(8+1 Hrs)

Part A:

Introduction to randomization in computation and some simple randomized algorithms.
Basic discrete probability theory: basic counting, definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes' rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, probabilistic recurrences.

Uniform generation of combinatorial structures. Indicator random variables and their role in algorithm analysis.

Part B:

Review Las Vegas and Monte-Carlo algorithms (with examples: randomized quick sort, Karger's min-cut algorithm.)

Unit 2: Tail inequalities, randomized complexity classes

(8+1 Hrs)

Part A:

Tail Inequalities and applications

Moments and deviation, occupancy problem, Markov and Chebyshev inequalities and some applications, randomized selection, weak law of large numbers, stable marriage problem and principle of deferred decision, coupon collector problem and sharp threshold, Chernoff's bound and some applications, set balancing.

Complexity classes

Basic complexity classes P, NP, RP, Co-RP, ZPP, BPP and their interrelations, probability amplification in RP and BPP, randomness and nonuniformity, Adleman's theorem.

Part B:

Yao's min-max principle and lower bound for randomized computations.

Unit 3: Algebraic techniques

(8+1 Hrs)

Part A:

Polynomial identity testing, Schwartz-Zippel lemma and applications (with examples verifying matrix multiplication, testing equality of strings, perfect matching problem for bipartite graphs), Mulmuley-Vazirani-Vazirani isolation lemma and application to matching problem. Number theoretic algorithms (finding quadratic non-residues, primality testing), introduction to probabilistic methods.

Part B:

Application of randomized algorithms in geometric problems (Convex hulls, half space intersection, Delaunay triangulations, diameter of point set).

Unit 4: Markov Chains and Random Walks

(8+1 Hrs)

Part A:

Markov chains: definition, representations, randomized algorithm for 2-SAT and 3-SAT, classifying states of Markov chains, Gambler's ruin, stationary distributions. Random walks on undirected graphs, cover time, hitting time, commute time, graph connectivity, electrical networks, introduction to expander graphs.

Part B:

Expanders and rapidly mixing random walks.

Unit 5: Approximation Algorithms

(8+1 Hrs)

Part A:

Introduction to approximation algorithms, NP-hard optimization problems, lower bounding OPT, example of set-cover ($O(\log n)$ factor approx-algorithm based on greedy strategy, layering), Shortest super-string problem, Knapsack and FPTAS algorithms. Linear programming based algorithms, LP relaxation, LP duality. LP rounding strategy and primal-dual schema, set-cover and some other examples using LP based techniques, maximum satisfiability.

Part B:

Review Vertex cover problem, Traveling Salesman Problem (TSP), general TSP hard to approximate, metric TSP.

Advanced Topics:

(depending on the availability of time some of the following topics may be included)
Role of expander graphs in derandomization, hardness of approximation, statement of PCP theorem, approximation algorithms for classical problems on integer lattices.

Text books:

1. *Randomized Algorithms* by Rajeev Motwani and Prabhakar Raghavan (Cambridge University Press)
2. *Probability and computing* by Michael Mitzenmacher, Eli Upfal
3. *Approximation Algorithms* by Vijay V. Vazirani (Springer)
4. *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (MIT Press)

References :

- *A course in probability theory* by Kai Lai Chung (Academic Press)
- *An Introduction to Probability Theory and Its Applications Vol I* by William Feller (Wiley)

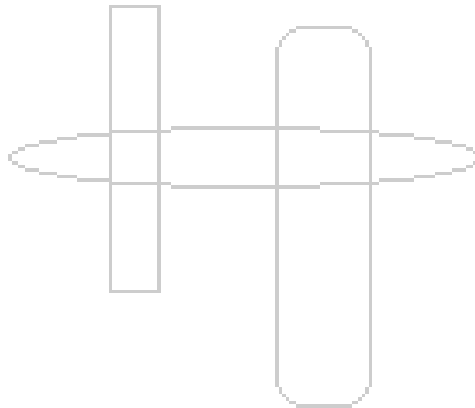
For some of the topics some online lecture-notes or original papers are used as reference.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Describe the basic principles of discrete probability and combinatorics in the context of randomized algorithms.

2. Design Las-Vegas, Monte-Carlo randomized algorithms for various complex computational problems
3. Analyze time complexity and success probability of randomized algorithms for computational problems with practical relevance.
4. Design approximation algorithms for NP-complete problems using suitable paradigm
5. Illustrate role of advanced algebraic techniques such as Schwartz-Zippel Lemma, Isolation Lemma, Markov chains and random walks on graphs in randomized algorithms design
6. Comprehend the applicability of randomized and approximation algorithms for solution of potentially hard computational problems.



CS42113:: DIGITAL SIGNAL PROCESSING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: **(8+1 Hrs)**

Introduction to signals and systems

Part A: Classification of Signals: Analog, Discrete-time and Digital, Basic sequences and sequence operations. Discrete-time (D.T) systems, Properties of D. T. Systems and Classification Linear Time Invariant Systems, impulse response, linear convolution and its properties, , DTS described by difference equations, Recursive and Non recursive, Relaxed and Non relaxed systems, Periodic Sampling, Sampling Theorem, Frequency Domain representation of sampling, reconstruction of a band limited Signal, A to D conversion Process: Sampling, quantization and encoding.

Part B: Properties of LTI systems, parallel and cascade connection, Correlation of DTS.

Unit 2: **(8+1 Hrs)**

Z and Fourier transforms

Part A: Z-transform, ROC and its properties, Inverse z transform by inspection, partial fraction and complex inversion, Z transform properties: multiplication by exponential sequence, Linearity, time shifting, frequency shifting, time reversal, convolution theorem initial value theorem, final value theorem, System Functions for LTI Systems: Stability and causality, inverse systems, significance of poles/zeros, Unilateral Z-transform: solution of difference equation with and without initial condition. Representation of Sequences by Fourier Transform, F. T. theorems: convolution theorem, and windowing theorem.

Part B: Symmetry properties of F. T, F. T. theorems: Linearity, time shifting, frequency shifting, time reversal, differentiation.

Unit 3: **(8+1 Hrs)**

Frequency analysis of Signals and Systems

Part A: Frequency Response of LTI Systems: Ideal frequency selective filters, magnitude and phase response, group delay, Frequency Response for Rational System Functions: Frequency Response of a single zero or pole, Frequency response from pole-zero plot using simple geometric construction

Sampling the F.T., Fourier representation of finite-duration sequences, The Discrete Fourier Transform, Properties of DFT: circular shift, duality, symmetry, Circular Convolution, Linear Convolution using DFT, Effective computation of DFT and FFT, DIT FFT, Overlap and save algorithm, Inverse DFT using FFT(DIF)

Part B: Properties of DFT: Linearity,DIF FFT, Goertzel Algorithm, Inverse DFT using FFT(DIT)

Unit 4: (9+1 Hrs)

Design of Digital Filters

Part A: Concept of filtering, Ideal filters and approximations, specifications, IIR filter design from continuous time filters: Characteristics of Butterworth, Chebyshev approximations, impulse invariant and bilinear transformation techniques, Design examples, FIR filter design using windows: properties of commonly used windows, systems with Linear phase, Generalized Linear phase systems, Four Types of GLPS (TypeI), Design Examples, Design using Kaiser window, Comparison of IIR and FIR Filters .

Part B: Four Types of GLPS (Type II,III,IV),Examples on Filter Design (IIR & FIR)

Unit 5: (7+1 Hrs)

Realization of Filters

Part A: Block diagrams and Signal flow graph representation of LCCDE, Basic structures for IIR Systems: direct form, cascade form, parallel form, feedback in IIR systems, Basic Structures for FIR Systems: direct form, cascade form, structures for linear phase FIR Systems,

Detail Study of DSP chip architecture as an example of ADSP 21XX series and their desirable features.

Part B: Instruction set of ADSP 21XX series processor and some examples.

Text Books

1. "Digital signal processing", Proakis J., Manolakis D., Prentice Hall Publications ISBN 9780072957167, 3rd Edition.
2. "Discrete time signal processing", Oppenheim A., Schafer R., Buck J., Prentice Hall Publications, ISBN 0-849-31093-8 2nd Edition

Reference Books

1. "Digital Signal Processing: A Computer Based Approach", Mitra S., Tata McGraw-Hill, ISBN 0-07-044705-5
2. "Digital Signal Processing", Vallavraj A., S Salivahanan, Tata McGraw- Hill, ISBN 0-07-463996, 2nd Edition

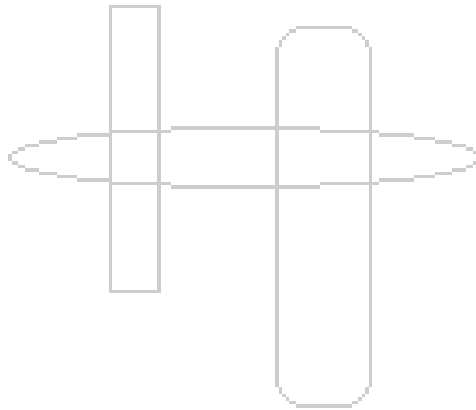
Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Assess the stability and the possibility of implementing a system.
2. Design faster algorithms for signal domain conversion which ensures expected operation on various new and upcoming architectures.
3. Synthesize filters for a given complex tasks.
4. Design a stable and optimized system which requires minimum components to

implement.

5. Analyze a signal in different mathematic domains to understand its characteristics using mathematical tools.
6. Practice filters design and deployment on different ADSP processors.



CS42119:: INFORMATION RETRIEVAL

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures, Database Management Systems

Unit 1: Information Retrieval Introduction and Models (9 Hrs)

Part A: Introduction, Definition, Objectives, Search and Browse Capabilities; A Formal Document Representation, Characterization of IR Models, Boolean Retrieval, Extended Boolean Models, Vector Space Model, Probabilistic Model, Measure of Relatedness, Term Weighting, Naive Bayes Text Classification, Document and Term Clustering, Flat and Hierarchical Clustering, Matrix Decomposition, Latent Semantic Indexing

Part B: Bayesian Model, Data Structures and Techniques for Ranking, Models for Browsing, Zipf's law, Heap's law

Unit 2: Query Processing and Retrieval Evaluation (8 Hrs)

Part A: Digital libraries, Morphological, Lexical Analysis, Stemming Algorithms, Thesaurus Construction, Ontology, Metadata, Query Languages, Similarity Measures and Ranking, Relevance Feedback, Query Expansion, Retrieval Performance, Evaluation Measures for Ranked and Unranked Results

Part B: Porter's Stemming Algorithm, Automatic Local/ Global Analysis, Information Summarization and Visualization, Archiving and Preservation

Unit 3: Indexing and Searching (8 Hrs)

Part A: Automatic Indexing, Inverted Files, Structures Used, Signature Files, Compression, Partitioning, Tries, Suffix Trees and Suffix Arrays, Index Construction, Distributed Indexing, Index Compression, Sequential Searching, Pattern Matching, String Matching allowing Errors, Regular Expressions and Extended Patterns, Pattern Matching using Indices, Structural Queries

B. Fast Inversion (FAST-INV) Algorithm, Algorithms on PAT Tree, Faceted Search

Unit 4: Parallel, Distributed IR and Web Searching (8 Hrs)

Part A: Parallel IR, Index Construction, Distributed IR, Characterizing the Web, Search Engines, Browsing, Metasearchers, Searching using Hyperlinks, Crawling, Link Analysis, Architectures (Agents, Buses, Wrappers/Mediators)

Part B: Watermarking, PageRank Algorithm, HillTop Algorithm

Unit 5: Multimedia IR (7 Hrs)

Part A: Multimedia Data Modeling, Query Languages, A Generic Multimedia Indexing (GEMINI) Approach, One Dimensional Time Series, Two Dimensional Color Images, Automatic Feature Extraction, Operations on images, Motion detection, Object recognition, Automatic image annotation and retrieval, Audio, Graph, Video Retrieval

Part B: Hashing Algorithms, Image Features and Similarity Functions

Text Books

1. *“Modern Information Retrieval The Concepts and Technology behind Search”*, Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Pearson Education: New Delhi, 2007
2. *“Introduction to Information Retrieval”*, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2012

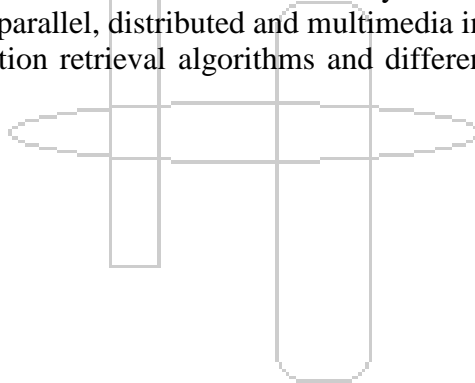
Reference Books

1. *“Information Storage and Retrieval Systems Theory and Implementation”*, Gerald Kowalski, Mark Maybury, Springer Pvt. Ltd., 2006
2. *“Information Retrieval Data Structures & Algorithms”* William Frakes, Ricardo Baeza-Yates, Pearson Education, 2008
3. *“Information Retrieval”*, C. J. Van Rijsbergen, Information Retrieval Group, University of Glasgow, online at <http://www.dcs.gla.ac.uk/Keith/Preface.html>

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Construct various indexes using suitable techniques
2. Validate retrieval performance of an information retrieval system
3. Apply sequential search and pattern matching techniques
4. Describe various models for information retrieval system
5. Illustrate working of parallel, distributed and multimedia information retrieval system
6. Use various information retrieval algorithms and different types of queries to address real world challenges



CS42114:: PRODUCT DESIGN

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1:

(9+1 Hrs)

Introduction and User Studies

Part A: Product Design, Products, Difference between Product and Service, Understanding of Problem Areas, Visual Design, User Segments, Specialized Users, Children, Physically Challenged and Colorblind Users, Illiterate and Old Users, Left-Handers, Accommodating Specialized Users into Product Design, User Studies, User Persona, Guidelines for User Studies, Developing Questionnaires, Interviewing Users, Observing Users, Photographic Studies of Products in Use, Role of Creativity in Understanding of Latent Needs.

Part B: Selection of a Product for Design Study such as Antivirus, Operating System, Mobile Phones, Web Browsers, Accounting Software etc.

Unit 2:

(7+2 Hrs)

Design and Usability Evaluation

Part A: Design Methodology for Products, Services and Events, Design of Integrated Systems, Contextual Enquiry, Design Process, Three Pillars of Design, Guidelines for Product Development, Logical User Centered Iterative Design (LUCID), Ethnography, Ethnographic Observations and Research, Participatory Design, Design Issues, Human Information Processing, Human Memory, Fitt's Law, Hick's Law, Heuristic Evaluation, Cognitive Walkthroughs, User Testing Using Think Aloud Protocol, Task Analysis, GOMS, Measurable Human Factors.

Part B: Ethnographic and Cultural Study of Selected Product.

Unit 3:

(8+2 Hrs)

Categorization of Products

Part A: Products for Future Use, Products to be Used in Groups, Devices used in Public Places, Products that Enrich User Experience, Embedded Products, Designer Products, Interfaces, Complexity of Interfaces, Design of Multi-Modal Interfaces, Expressive Interfaces, Natural Interfaces, Tangible Interfaces, Faulty Interfaces.

Part B: Classification of Selected Products and Possible Variances with Extended Features.

Unit 4:

(8+2 Hrs)

Design Management and Professional Practice

Part A: Designer, Designer Attributes, Role of a Designer, Designer Classification, Setting Up a Design Office, Location, Size, Legal Framework, Infrastructure, Organization of Supplies and Human Resources, Work Environment, Finding Clients, Business Correspondence, Advertising, Professionalism, Ethics, Need for Ethics, Unethical Practices, Cyber Crimes, Human Factors, Cognitive Psychology, Human Information Processing, Managing Teamwork, R&D Activities, Design Evaluation, IPR, Copyrights, Patents and Trademarks, Design Registration Laws and Procedure.

Part B: Identification of IPR (Copyrights, Patents and Trademarks) Issues with Selected Products.

Unit 5: **(8+1 Hrs)**

Product Life and Marketing

Part A: Defining Business Policy, SWOT Analysis, Study of Product Life Cycle, Stages -Introduction, Growth, Maturity, Decline, Evaluation Parameters, Monitoring of Sale and Competition, Assessing Market Potential, Market Research, Consumer Research and Demographic Aspects, Establishing Market Segment and Their Dimensions, Developing a Strategy to Introduce New Products, Developing a Product Plan, Pricing Policy, Factors Influencing Prices, Factors Influenced by Prices, Product Positioning.

Part B: Analysis of Prices and Related Pricing Policy for Selected Product.

Text Books

1. *"The Design of Everyday Things"*, Donald Norman, Basic Books, ISBN 100-465-06710-7, 2002 Edition.
2. *"Product Design and Development"*, Karl Ulrich and Steven Eppinger, McGraw-Hill, 2004, ISBN 978-0072471465, 3rd Edition.

Reference Books

1. *"Design Secrets: Products 2: 50 Real-Life Projects Uncovered"*, Industrial Designers Society of America, Edited by Lynn Haller, Cheryl Dangel Cullen, Publisher: Rockport Publishers, Oct 2004, ISBN 978-1592530717.
2. *"The Design of Future Things"*, Donald A. Norman, Basic Books, ISBN 978-0-465-00227-6, 2007 Edition.

Additional Reading

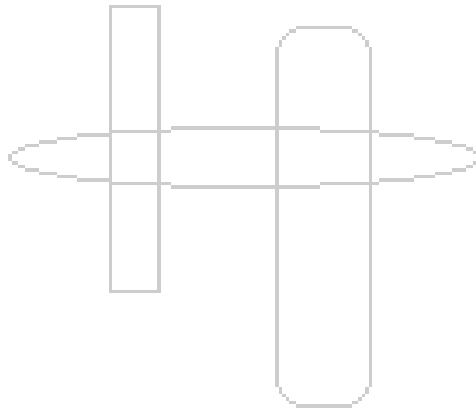
1. *"What is a Designer: Things, Places, Messages"*, Norman Potter, Publisher: Princeton Architectural Press, 2002, ISBN 978-0907259169, 4th Edition.
2. *"History of Modern Design"* David Raizman, Prentice Hall, 2004, ISBN 978-0131830400.

Course Outcomes:

Upon completion of the course, graduate will be able to -

1. Develop an IT product concept by applying suitable design principles, models and guidelines along with creativity.
2. Propose an implementation-centric product design with due consideration to ethnographic, cultural and accessibility aspects

3. Formulate all possible vital product features required for a specific category of product
4. Propose effective pricing policy along with multilingual documentation and support for the product
5. Build a set of skills required for responsible positions such as Product Designer, Interaction Designer and Creative Technologist
6. Follow required processes and standards while designing IT products



CS42115:: HUMAN COMPUTER INTERACTION

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (7+1 Hrs)

Introduction

Part A: Human, Definition of Human Computer Interaction, Interdisciplinary Nature, Goals, Human Factors, Measurable Factors – Learn ability, Speed, Efficiency, Satisfaction. Early Focus on Users, Ergonomics, Usability, Types of Usability, User Interface (UI), Contexts - Web, Business, Mobile, Gaming Applications, Categorization of Applications based on Human Factors, Accessibility and Security.

Part B: Identification of Application Category and Related Features for Selected Product / System.

Unit 2: (9+2 Hrs)

Principles, Models & Guidelines

Part A: Eight Golden Rules of Interface Design, Principles of Good Design, Faulty Designs, Miller's Principle, Norman's Action Model, Gulf of Execution and Evaluation, Errors – Mistakes, Slips, Lapses and Violations, Guidelines for Data Display, Guidelines for Data Entry, Conceptual, Semantic, Syntactic and Lexical Model, Task Analysis, GOMS, Keystroke-Level Model, User Persona, UI Standards and GUI Libraries.

Part B: Task / Error Analysis for Selected Product / System.

Unit 3: (8+2 Hrs)

Design Process and Interaction Styles

Part A: Design, Three Pillars of Design, Process of Design, Ethnographic Observations, Contextual Inquiry, Iterative Design, Participatory Design, Navigation Design, Visual Design, - Layout, Color, Fonts, Labeling, LUCID, Scenarios, Interaction Styles - Direct Manipulation, Menu Selection, Form-Fillin, Commands, Natural Language, Internationalization, Interaction Design Patterns.

Part B: UI Design for Selected Product/System.

Unit 4: (8+2 Hrs)

Evaluation Techniques and Interface Categories

Part A: Expert-based Evaluation, User-based Evaluation, Heuristic Evaluation, Cognitive Walkthrough, Semiotic Analysis, Expert Reviews, Usability Testing, User Surveys, Interviews, Think Aloud, Acceptance Tests, Statistical Methods, Touch Interfaces,

Public Place Interfaces, Wearable Interfaces, Tangible Interfaces, Intelligent Interfaces, Ubiquitous and Context-Aware Interaction.

Part B: Usability Evaluation of Selected Product/System.

Unit 5:

(8+1 Hrs)

Documentation and Groupware

Part A: Classification of Documents, Printed Manuals, Reading from Displays, Online Help, Tutorial, Error / Warning Messages, Groupware, Goals / Dimensions of Cooperation, Asynchronous Interactions, Synchronous Interactions, Online Communities, Communityware, Social Psychology, Social Networks, Social Networking Sites.

Part B: Documentation Design for Selected Product/System.

Text Books

1. *"Human-Computer Interaction"*, Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, Pearson Education, ISBN 81- 297-0409-9, 3rd Edition.
2. *"Designing the User Interface"*, Ben Shneiderman, Pearson Education, ISBN 81-7808-262-4, 3rd Edition.

Additional Reading

1. *"The Design of Everyday Things"*, Donald Norman, Basic Books, ISBN 100-465-06710-7, 2002 Edition.
2. *"The Essential Guide to User Interface Design"*, Wilbert O. Galitz, Wiley-dreamtech India (P) Ltd., ISBN 81-265-0280-0, 2nd Edition.
3. *"Human-Computer-Interaction in the New Millennium"*, John M. Carroll, Pearson Education, ISBN 81-7808-549-6.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify human factors and usability issues related with computing applications
2. Differentiate computing applications into categories based on human factors
3. Design a user interface by applying suitable design principles, models and usability guidelines
4. Integrate ethno-cultural and accessibility computing aspects into the user interface design.
5. Display the impact of usability evaluation and testing in computing applications
6. Follow required processes and standards while designing user interfaces

CS42134::Modeling and Simulation

Credits: 3

Teaching Scheme: 3 Hours / Week

Unit 1: Process of Modeling and Simulation (8 Hours)

Part A: What is M&S, Need for Abstraction, Relationship between modeling and simulation Process of modeling: Problem identification and formulation, Real system data collection, Model development, Validation, Experiment design, Simulation runs and Results interpretation.

Part B: Application areas: optimization, decision making support, forecasting, safety considerations, training and education

Unit 2: Formal models and modeling techniques (8 Hours)

Part A: Monte Carlo methods, Stochastic processes, Queuing theory: Little's Theorem and applications, M/M/1 Queuing System, Markov Chain Formulation, the M/M/m, M/M/m/m and other Markov Systems, Non-markovian queue, Analytical Model Vs. Simulation model, Petri nets and colored Petri nets, Game theory, State spaces and transitions.

Part B: Graph structures: directed graphs, trees, networks

Unit 3: Discrete Event Simulation (8 Hours)

Part A: Deterministic vs. stochastic simulation, Static vs. Dynamic Simulation, Constructing dynamic stochastic simulation models, Time keeping, Event Scheduling, State transition, Time driven and event driven models, Pseudo-random number generation.

Part B: SimPy basics

Unit 4: Agent-based simulation (8 Hours)

Part A: Modeling Complex Systems, Agents, environments, ABMS: When and Why, Agent based model design, Autonomous Agents, Agent Interaction, Topologies and Neighborhoods, Tools for ABMS: Repast, Swarm, NetLogo, Applications: Social networks, Organizations, Markets, Flows, Epidemiology, Diffusion.

Part B: NetLogo basics

Unit 5: Advanced Topics (8 Hours)

Part A: Intro to Parallel Discrete Event Simulation: Parallel and Distributed Platforms, Model scalability, Virtual Reality, Virtual Worlds, Intro to Rare Event Simulation.

Part B: PDES Challenges

Text Books:

1. *Discrete Event Simulation: A First Course*, L. Leemis and S. Park, 2006, Prentice-Hall.
2. *Agent-Based Models*, Nigel Gilbert, 2008, SAGE Publications.
3. *System Simulation and Modeling*, Sankar Sengupta, 2013, Pearson Education.

Reference Books:

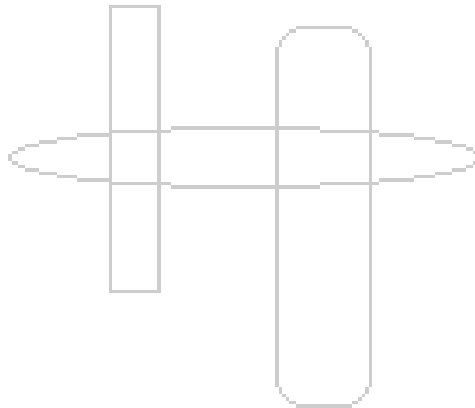
1. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, J. Banks, 1998, John Wiley & Sons.

2. *Parallel and Distributed Simulation Systems, Fujimoto R.M., 2000, John Wiley & Sons.*

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Develop a model for a given problem using appropriate modeling and simulation technique/formalism.
2. Analyze complex real world systems to identify the key components of the model.
3. Utilize state-of-the-art development methodologies for efficient model development.
4. Apply the modeling and simulation approach to assess the solution alternatives in terms of the environmental impact and sustainability.
5. Contribute towards increased utilization of modeling and simulation as a problem solving approach for issues which affect the society.
6. Function effectively as a part of collaborative and interdisciplinary M&S projects.



CS40215:: DISTRIBUTED COMPUTING

Credits: 01

Teaching Scheme: -Tutorial 1 Hr/Week

Prerequisites: Operating Systems.

List of Contents

A TERM-WORK containing the record of the following:

1. Design and develop a basic prototype distributed system (e.g. a DFS).
2. Design and implement client server application using RPC/ RMI mechanism (Java)
3. Design and implement a clock synchronization algorithm for prototype DS.
4. Implement Ring or Bully election algorithm for prototype DS.
5. Ricart Agrawala's distributed algorithm for mutual exclusion.
6. Problem solving of Wait-die and Wait –wound scheme for deadlock prevention.
7. Simulate Wait for Graph based Centralized or Hierarchical or Distributed algorithm for deadlock detection.
8. Implementation of 2PC / Byzantine Generals Problem
9. Simulate any one of the Consistency models.

Text Books

1. *“Distributed Computing: Principles, Algorithms, and Systems”*, Ajay Kshemkalyani, Mukesh Singhal, Cambridge University Press, ISBN-13: 978-0-521-87634-6, 1st Edition.
2. *“Distributed Systems Principles and Paradigms”*, Andrew S. Tanenbaum & Maarten Van Steen, Prentice Hall India, ISBN 81-7808-789-8, 1st Edition.

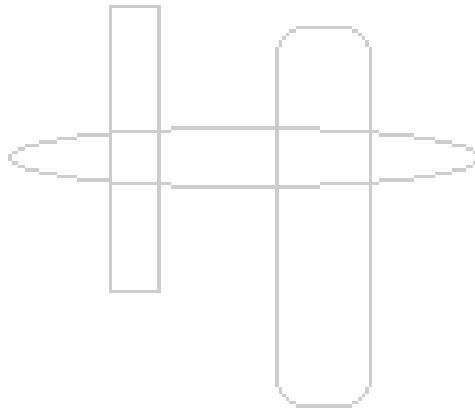
Reference Books

1. *“Distributed Operating Systems Concepts and Design”*, Pradeep K. Sinha, Prentice Hall India, ISBN: 81-203-1380-1, 1st Edition
2. *“Distributed Systems – Concepts and Design”*, George Coulouris, Jean Dollimore & Tim

Kindberg, Addison-Wesley, ISBN: 0-201-61918-0, 3rd Edition

Additional Reading

1. “Advanced Concepts In Operating Systems”, Mukesh Singhal, Niranjana G. Shivaratri, McGrawHill, ISBN-13: 978-0-07-047268-6.
2. “Distributed and Parallel Systems: From Cluster to Grid Computing”, Peter Kacsuk, Thomas Fahringer, Zsolt Nemeth, Springer, ISBN: 978-0-378-69857-1, 2007.
3. “Distributed Computing: Principles and Applications”, M. L. Liu, Addison-Wesley, ISBN-10: 0201796449, 2004.
4. “Distributed Computing: Fundamentals, Simulations and Advanced Topics”, Hagit Attiya, Jennifer Welch, John Wiley & Sons, Inc, ISBN 0-471-45324-2,



CS42125:: Randomized and Approximation Algorithms

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Prior introduction to basic probability theory is helpful though it is not presumed. First course on algorithms.

List of Contents

A TERM-WORK containing the record of the following:

Assignments :

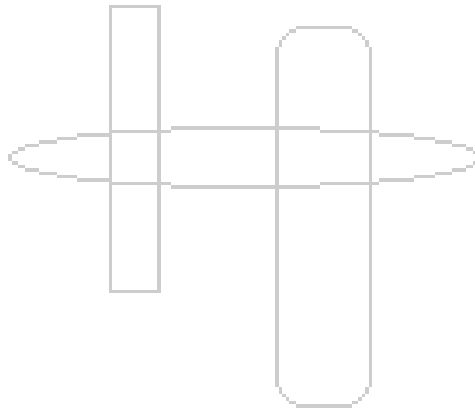
1. Problem solving based on basic probability (random variables, mean, variance)
2. Problem solving based on basic probability (linearity of expectation, some applications, conditional probability)
3. Use of indicator random variables in algorithm analysis
4. Tail Inequalities and applications (Markov, Chebyshev)
5. Tail Inequalities and applications (Chernoff bound, variance of sum of co-related random variables)
6. Random walks, rapid mixing and some applications like sampling points from convex bodies
7. Schwartz-Zippel Lemma and applications
8. Complexity of perfect matching problem
9. Randomization in Number theoretic and geometric algorithms
10. Approximation algorithms, lower bounding OPT
11. Linear programming and LP-duality, LP based approximation algorithms
12. PCP theorem and hardness of approximation

Text Books

1. *Randomized Algorithms* by Rajeev Motwani and Prabhakar Raghavan (Cambridge University Press)
2. *Approximation Algorithms* by Vijay V. Vazirani (Springer)
3. *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (MIT Press) Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition.

Reference Books

1. *A course in probability theory* by Kai Lai Chung (Academic Press)
2. *An Introduction to Probability Theory and Its Applications Vol I* by William Feller (Wiley)
3. *Structural Complexity I* by Error! Hyperlink reference not valid. Error! Hyperlink reference not valid. Error! Hyperlink reference not valid., Error! Hyperlink reference not valid., Error! Hyperlink reference not valid. (Springer) (for basic complexity theory)
4. *Computational Complexity: Modern approach* by Sanjeev Arora and Boaz Barak (Cambridge University Press) book is freely available on authors homepage <http://www.cs.princeton.edu/theory/complexity/>



CS42113:: DIGITAL SIGNAL PROCESSING

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites:

List of Tutorials

1. Sampling theorem: Write a program in Matlab to verify sampling theorem
2. Linear Convolution: Write a program in Matlab to perform linear convolution of 2 user defined signals. Verify your answer using Matlab functions.
3. Cross co-relation: Get two signals from user and write a code in Matlab to perform cross co-relation. Verify your answer using Matlab functions.
4. Auto co-relation: Get a signal from user and write a program in Matlab to verify the following two properties of the Auto co-relation.
 - A. Auto co-relation attains its maximum value at zero lag
 - B. Auto co-relation is an even function
5. Pole Zero: For a given Z transform of signal, plot poles and zeroes on Z space using Matlab and verify that when for a system, if pole is near unit circle, rate of decay of the response in time domain is much more gradual than a system with pole near origin.
6. Magnitude and Phase response of a system: For a given transfer function, plot magnitude and phase response of the system using Matlab
7. DFT: Write a program to calculate 8 point DFT of a user defined signal using Matlab. Verify your answer using Matlab functions.
8. Circular Convolution: Get two signals from user and write a program in Matlab to calculate their circular convolution. Verify your answer using Matlab functions.
9. FFT: Write a program to implement a Radix 2, 8 Point DIT FFT algorithm using Matlab and verify your answer using in built functions.
10. IIR Filter design: Get the filter specification from the user and write a program to calculate the filter coefficients in C/C++

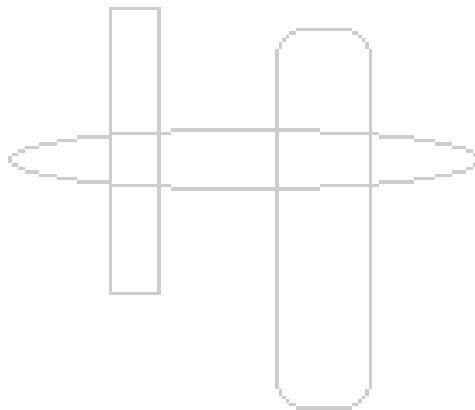
11. FIR Filter design: Get the filter specification from the user and write a program to calculate the filter coefficients in C/C++

Text Books

1. *“Digital signal processing”, Proakis J., Manolakis D., Prentice Hall Publications ISBN 9780072957167, 3rd Edition.*
2. *“Discrete time signal processing”, Oppenheim A., Schafer R., Buck J., Prentice Hall Publications, ISBN 0-849-31093-8 2nd Edition*

Reference Books

1. *“Digital Signal Processing: A Computer Based Approach”, Mitra S., Tata McGraw-Hill, ISBN 0-07-044705-5*
2. *“Digital Signal Processing”, Vallavraj A., S Salivahanan, Tata McGraw- Hill*



CS42219:: INFORMATION RETRIEVAL

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites: Data Structures, Database Management Systems

List of Contents

A TERM-WORK containing the record of the following:

A. Assignments :

1. Problem solving for Boolean model.
2. Problem solving for Vector Space model.
3. Study of Stemming Algorithms and Thesaurus Construction
4. Problem solving for Index creation: Inverted Files, Signature Files, Suffix Trees and Suffix Arrays
5. Problem solving for Sequential Searching and Pattern Matching techniques
6. Problem solving for Latent Semantic Indexing for Text Classification
7. Build an Information Retrieval system using Apache **Lucene**

Text Books

1. *“Modern Information Retrieval”*, by Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Addison Wesley Longman Publishing Co. Inc, ISBN:020139829X, 1999.
2. *“Introduction to Information Retrieval”*, by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. ISBN: 0521865719, 2008.

Reference Books

1. *“Information Retrieval”*, C. J. Van Rijsbergen, Information Retrieval Group, University of Glasgow, online at <http://www.dcs.gla.ac.uk/Keith/Preface.html>.
2. *“Information Retrieval Data Structures & Algorithms”* by Bill Frakes, Ricardo Baeza-Yates, Prentice-Hall, ISBN 0-13-463837-9, 1999.
3. *Apache **Lucene** Tutorial*

CS42114:: PRODUCT DESIGN

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

A TERM-WORK containing the record of the following:

1. Design a questionnaire for study of selected product.
2. Study specialized user population and their impact on the design of selected product.
3. Perform a GOMS analysis for any task(s) related with selected product.
4. Study faulty interfaces/interactions related with selected product.
5. Analyze the cybercrimes related with selected product and suggest preventive measures.

Text Books

1. *"The Design of Everyday Things"*, Donald Norman, Basic Books, ISBN 100-465- 06710-7, 2002 Edition.
2. *"Product Design and Development"*, Karl Ulrich and Steven Eppinger, McGraw-Hill, 2004, ISBN 978-0072471465, 3rd Edition.

Reference Books

1. *"Design Secrets: Products 2: 50 Real-Life Projects Uncovered"*, Industrial Designers Society of America, Edited by Lynn Haller, Cheryl Dangel Cullen, Publisher: Rockport Publishers, Oct 2004, ISBN 978-1592530717.
2. *"The Design of Future Things"*, Donald A. Norman, Basic Books, ISBN 978-0-465-00227-6, 2007 Edition.

Additional Reading

1. *"What is a Designer: Things, Places, Messages"*, Norman Potter, Publisher: Princeton Architectural Press, 2002, ISBN 978-0907259169, 4th Edition.
2. *"History of Modern Design"* David Raizman, Prentice Hall, 2004, ISBN 978-0131830400.

CS42115:: HUMAN COMPUTER INTERACTION

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: :

List of Contents

A TERM-WORK containing the record of the following:

1. Identify specialized users and related facilities for a selected product / system and make necessary suggestions for its improved accessibility design.
2. Design user persona for the users of selected product / system.
3. Conduct a contextual inquiry for selected product/system.
4. Design an interface prototype for selected product/system.
5. Evaluate an interface using usability evaluation technique.

Text Books

1. *“Human-Computer Interaction”*, Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, Pearson Education, ISBN 81- 297-0409-9, 3rd Edition.
2. *“Designing the User Interface”*, Ben Shneiderman, Pearson Education, ISBN 81-7808-262-4, 3rd Edition.

CS42234:Modeling and Simulation

Credits:1

Teaching Scheme: Tutorial 1 Hours / Week

List of Contents:

1. Develop a simple deterministic simulation to determine the loan tenure for Rs.X principal amount when the customer pays Rs.Y per month. Assume the fixed interest rate of 10% per year.
2. Develop a Monte Carlo simulation model for profit estimation before introducing a new product in the market. Consider the uncertainty in terms of sales, reduction costs, competitive pricing and other market dynamics.
3. Develop a discrete event simulation of a typical fast-food restaurant. Restaurant configuration, business factors and customer behaviour factors should be tunable parameters.
4. Agent based simulation : Marketing/Diffusion: Word of Mouth publicity, Epidemiology: spread of disease.
5. Develop a parallel discrete event simulation for a network of routers using conservative event processing.

Text Books:

1. *Discrete Event Simulation: A First Course*, L. Leemis and S. Park, 2006, Prentice-Hall.
2. *Agent-Based Models*, Nigel Gilbert, 2008, SAGE Publications.
3. *System Simulation and Modeling*, Sankar Sengupta, 2013, Pearson Education.

Reference Books:

1. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, J. Banks, 1998, John Wiley & Sons.
2. *Parallel and Distributed Simulation Systems*, Fujimoto R.M., 2000, John Wiley & Sons.

CS40319:: BUSINESS INTELLIGENCE AND ANALYTICS

Credits: 01

Teaching Scheme: -Practical 2Hr/Week

Prerequisites: Database Management Systems.

List of Contents

A TERM-WORK containing the record of the following:

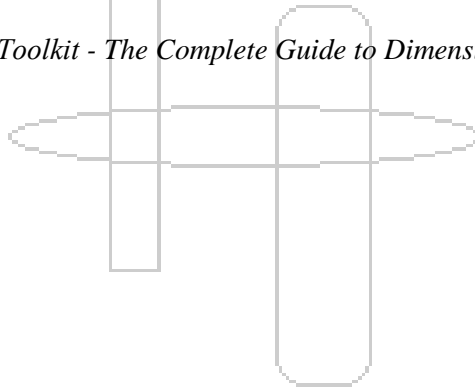
1. Getting started: understand what business do with their data with a scenario based application using QlikView tool
2. Designing an end to end warehousing solution for a real world scenario involving multi-dimensional modeling, designing data cube, doing ETL, OLAP and reporting
3. Getting started with R
4. Using R for data preprocessing, exploratory analysis, visualization, correlation and regression analysis, hypothesis testing, chi square test
5. Data analysis case study using R for a readily available data set
6. BigData Analytics - MapReduce and exposure to Hadoop, Using R over Hadoop
7. [Optional] A group mini-project: take a real world data analysis problem and solve it using the above learned concepts
 - a. Getting Data from varied sources
 - b. Data massaging to prepare it for analysis
 - c. Generating visualizations to interpret descriptive analysis
 - d. Implementing sampling and estimation techniques
 - e. Regression analysis on data
 - f. Hypothesis testing

Text Books

1. *"Data mining and business analytics with R"*, Johannes Ledolter, Wiley
2. *"Business Analytics"* by James R Evans, Pearson
3. *"Data Mining: Concepts and Techniques"*, Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition
4. *"Fundamentals of Business Analytics"*, by R. N. Prasad, Seema Acharya, ISBN: 978-81-256-3203-2, Wiley-India
5. *"Essentials of Business Analytics"* by Camm, Cochran, Fry, Ohlmann, Anderson, Sweeney, Williams, Cengage Learning
6. *"The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence"* 2010, Ralph Kimball, Margy Ross, Wiley Publications

Reference Books

1. *"Business Intelligence for Dummies"*
2. *"Advanced Engineering Mathematics"* by Erwin Kreyszig, John Wiley & Sons Inc., 10th Edition
3. *"Applied Business Statistics: Making Better Business Decisions"* 7th Edition by Ken Black, Wiley India
4. *"Forecasting: Principles and Practices"*, Rob J Hyndman, George Athanasopoulos, OTexts
5. *"Information Dashboard Design: The effective visual communication of data"*, Stephen Few, O'Reilly
6. *"The Data Warehouse Toolkit - The Complete Guide to Dimensional Modeling (2nd Ed 2002 Wiley)"* - Kimball & Ross



CS42305:: MOBILE COMPUTING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Computer Networks

List of Practical

Assignments should be implemented on android operating systems.

1. Design simple GUI application with activity and intents e.g. calculator.
2. Design an application to handle address book and call log.
3. Design an application to display map with current location.
4. Design an application to send SMS and emails.
5. Design a client server application to communicate with database server.

Course Project: The project needs to be done in the group of 2 or 3 where students need to select real world problem and present a solution. During the evaluation of the project the emphasis is on: (a) how problem selection is done and (b) is the application user friendly.

Text Books

1. *"Beginning Android Application Development"*, Wei-Meng Lee, Wrox publication, ISBN: 978-1-118-01711-1, March 2011.
2. *"Mobile Communications"*, Jochen Schiller, 2nd edition, Pearson education, ISBN- 81-297-0350-5.

Reference Books

1. *"Wireless and Mobile Network Architectures"*, Yi-Bing Lin, Imrich Chlamtac, Wiley publication, ISBN-9971-51-366-8.
2. *"Mobile Computing"*, Raj Kamal, Oxford University press, ISBN 978-0-19-568677-7

CS42331::Enterprise Systems

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Software Engineering, Database Management System

List of Practical

1. To narrate concise Requirement Definition Document and System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Standard guidelines clearly indicating the requirements considerations.
2. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system along with identification of Business Processes and develop full detail Business Process diagrams.
3. To develop Domain-driven vocabulary of the target system indicating domain lexicon and context-based terminologies.
4. To identify and categorize the target system services with detailed service specifications modeled with component diagram incorporating appropriate architectural style and coupling.
5. To design the service layers and tiers modeled with deployment diagram accommodating abstraction, autonomy, statelessness and reuse.
6. To map the service levels and primitives to appropriate Strategies for data processing using XML / XQuery/ JSON / JAXB.
7. To produce, invoke, compose Web Services using SOAP, WSDL and UDDI.
8. To implement and integrate the components of the target system using .NET / J2EE platforms adhering to Service specifications.
9. To create the balanced scorecard for the target system indicating the standards and principles applied.

Text Books:

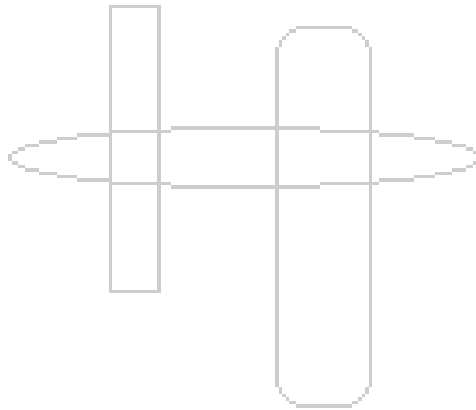
1. *Mathias Weske, Business Process Management, Concepts, Languages, Architectures, ISBN 978-3-540-73521-2 Springer Berlin Heidelberg New York, 2007*
2. *Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design. ISBN: 0-13-185858-0, Publisher: Prentice Hall PTR, 2005*

Reference Books :

1. *Thomas Erl, SOA Principles of Service Design, Pearson Education, Inc., ISBN 0-13-234482-3, 2007*
2. *Eric A. Marks, Michael Bell., Executive's guide to service-oriented architecture, John Wiley & Sons, Inc. ISBN-13: 978-0-471-76894-4, 2006*
3. *Daniel Minoli, Enterprise Architecture A to Z, Frameworks, Business Process Modeling, SOA, and Infrastructure Technology, Auerbach Publications, Taylor & Francis Group, ISBN 978-0-8493-8517-9, 2008*
4. *Setrag Khoshafian, Service Oriented Enterprises, Auerbach Publications, Taylor & Francis Group, ISBN 0-8493-5360-2, 2007*
5. *Mike Rosen, Boris Lublinsky, Kevin T. Smith, Marc J. Balcer, Applied SOA: Service-Structure and syllabus of S.Y. B.Tech Computer Engineering, Pattern A-14, A.Y. 2015-16*

Oriented Architecture and Design Strategies, Wiley Publishing, Inc., ISBN: 978-0-470-22365-9, 2008

6. *Marc Lankhorst et al., Enterprise Architecture at Work, Modelling, Communication and Analysis, Second Edition, ISBN 978-3-642-01309-6, Springer-Verlag Berlin Heidelberg 2009*
7. *David S. Linthicum, Enterprise Application Integration Addison-Wesley Professional 2003, ISBN-10: 1402052626*



FF No. : 654 B

CS42327 :: CLOUD COMPUTING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Computer Networks, web Technology, Operating Systems.

List of Contents

A TERM-WORK containing the record of the following:

1. Hands on virtualization using XenServer
2. Hands on containerisation using Docker
3. Deployment and Configuration options in Amazon (AWS)
4. Deployment and Configuration options in Google Cloud
5. Deployment and Configuration options in Microsoft Azure
6. Building a 'HelloWorld' app for the cloud
7. Deploying the 'HelloWorld' app for the cloud

Text Books

1. *Cloud Computing for Dummies* by Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper (Wiley India Edition)
2. *Enterprise Cloud Computing* by Gautam Shroff, Cambridge
3. *Cloud Security* by Ronald Krutz and Russell Dean Vines, Wiley-India

Reference Books

1. *Google Apps* by Scott Granneman, Pearson
2. *Cloud Security & Privacy* by Tim Malhar, S.Kumaraswammy, S.Latif (SPD, O'REILLY)
3. *Cloud Computing : A Practical Approach*, Antohy T Velte, et.al McGraw Hill,
4. *Cloud Computing Bible* by Barrie Sosinsky, Wiley India

CS42303:: PARALLEL COMPUTING ON GPU

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Parallel GPU implementation of vector-vector operations
2. Parallel GPU implementation of vector-Matrix operations
3. Parallel computation of binomial coefficient matrix
4. Parallel GPU implementation of Matrix-Matrix operations
5. Assignment focusing on optimization of data transfer between CPU and GPU:using page locked host memory and to avoid the data transfer
6. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.
7. Parallel GPU implementation involving kernel looping.
8. Parallel computation of set of multi-indices on GPU.

Text Books

1. *CUDA: Programming Massively Parallel Processors: A Hands-On Approach.* Authors: David Kirk, Wen-mei Hwu © ELSEVIER Inc.
2. *CUDA by Example: An Introduction to General-Purpose GPU Programming* by Jason Sanders and Edward Kandrot
3. *Parallel Programming in C-with MPI and OpenMP* by Michael J. Quinn, Tata McGraw-Hill Edition
4. *Advanced computer architecture* by Kai Hwong, Tata McGraw-Hill Edition, 2001

References:

<http://developer.nvidia.com/>
www tutorials on introduction to parallel computing

Other references suggested by instructor

CS42306:: NETWORKS SECURITY

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practicals

1. Demonstrate: SQL injection, Cross-site scripting, buffer overflow.
2. Implement packet sniffer.
3. Implementation of Caesar and Vigenere Cipher
4. Implementation of Playfair Cipher
5. Implementation of Hill Cipher
6. Implementation of S-DES.
7. Implementation of S-AES
8. Implementation of RSA.
9. Implementation of Diffie-Hellman key exchange
10. Implementation of ECC algorithm.

Text Books

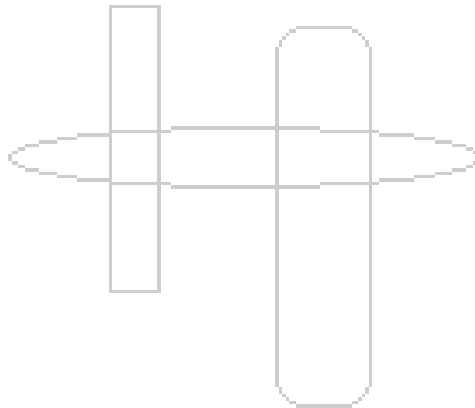
1. *“Network Security and Cryptography”, by Bernard Menezes, Cengage Learning, 2010, ISBN 81-315-1349-1, 1st Edition.*
2. *“Cryptography and Network Security-Principles and Practices” by William Stallings, Pearson Education, 2006, ISBN 81-7758-774-9, 4th Edition.*

Reference Books

1. *“Computer Security: Art and Science”*, by Matt Bishop, Pearson Education, 2002, ISBN 0201440997, 1st Edition.
2. *“Network security, private communication in a public world”*, by Charlie Kaufman, Radia Perlman and Mike Spencer, Prentice Hall, 2002, ISBN 9780130460196, 2nd Edition.

Additional Reading

1. *“Security architecture, design deployment and operations”*, by Christopher M. King, Curtis Patton and RSA press, McGraw-Hill, 2001, ISBN 0072133856, 1st Edition.
2. *“Inside Network Perimeter Security”* by Stephen Northcott, Leny Zeltser, et al, Pearson Education Asia, ISBN 8178087618, 1st Edition.



CS42301:: ADVANCED COMPUTER GRAPHICS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Implement an OpenGL program to draw different 2D shapes.
2. Implement an OpenGL program to draw 2 overlapped shapes and use alpha blending.
3. Implement an OpenGL program to draw 3D cube and apply transformations.
4. Implement an OpenGL program to draw 12 spheres and apply different light effects.
5. Implement an OpenGL program to draw scene and apply fog effect.
6. Implement an OpenGL program to draw 3D cube and apply different textures on different faces.
7. Implement CUDA program for the prefix addition.
8. Implement CUDA program for the multiply two matrices.
9. Draw histogram of 256-color bmp image.
10. Implement a program for edge detection using Gaussian filter.

Text Books

1. *"Procedural Elements for Computer Graphics", Rogers David F., Tata McGraw Hill, 2001, ISBN 0-07-047371-4.*
2. *"OpenGL Programming Guide: The Official Guide to Learning OpenGL", Mason_Woo, Jackie, Tom Davis, Version 2.1, 6th Edition, Pearson Education, ISBN 978-81-317-2184-*

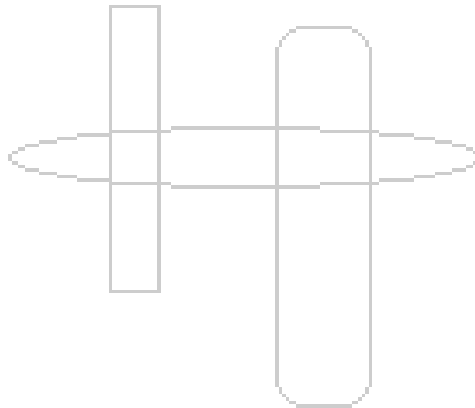
1.

Reference Books

1. *"Computer Graphics", M Paulin Baker, Donald Hearn, PHI, 2001, ISBN 81-203-09440-8.*
2. *"Digital Image Processing", Rafael C. Gonzalez , Richard E. Woods, 2nd Edition, Prentice Hall, ISBN: 0-201-18075-8.*

Additional Reading

1. *"Advanced Animation and Rendering Techniques : Theory and Practice", Alan H. Watt and Mark Watt, Addison-Wesley, acm press, ISBN: 0201544121*
2. *"CUDA by Example: An Introduction to General-Purpose GPU Programming" , Jason Sanders, Edward Kandrot, Addison-Wesley, ISBN-13: 978-0-13-138768-3.*



CS47303:: PROJECT STAGE 2

Credits: 04

Teaching Scheme: - Practical 8 Hrs/Week

Aim

This course addresses the issues associated with the successful management of a software development project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career. The Project Work can lead to:

- a. Transform existing systems into conceptual models.
- b. Transform conceptual models into determinable models.
- c. Use determinable models to obtain system specifications.
- d. Select optimum specifications and create physical models.
- e. Apply the results from physical models to create real target systems.

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
3. The project must be open-ended – meaning that there is not a known correct answer to the design problem. Students are expected to apply their creativity (simply copying or re-creating something that already exists is not acceptable).
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting Requirement Definition Document, Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
6. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.

7. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.
8. The Student Project Groups are expected to work on the recommendations given by the panel of examiners. In no case any variation in Project Theme will be permitted.

Assessment Scheme

Sr. No.	Content	Marks
1	System Requirement Specification	20
2	Feasibility Study	10
3	System Analysis	30
4	System Design	30
5	Presentation of the Project Work	10

Note:

The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members.

The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.

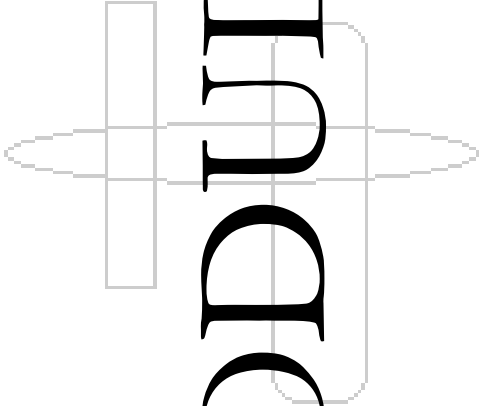
Following is the list of recommended domains for Project Work:

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify the Design within Specification and Available Resources
2. Optimise the Design Specification within Available resources
3. Maximise Resource utilisation by avoiding wastage and idea Duplication
4. Cooperate and Communicate with Interdisciplinary teams for effective Solutions
5. Study the Financial implications of your Design
6. Use the knowledge of available Resources for Designing the Solution

MODULE VIII



FF No. 653

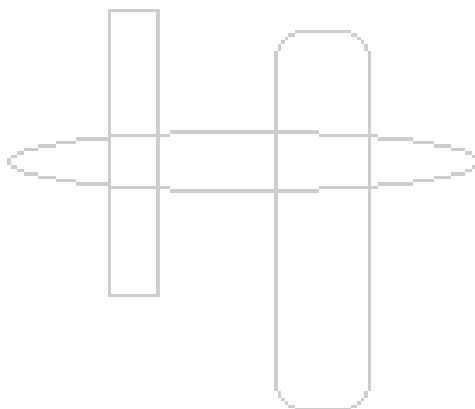
Issue 4, Rev. 1 , Dt. 01/03/2014

B. Tech. Structure with effect from Academic Year 2015-16

Module VIII

Code	Subject	Type	Teaching Scheme			Assessment Scheme						Credits
			L	P	Tut.	ISA					ESA	
						Test 1	Test 2	HA	Tut.	CA	ESE	
CS40110	Artificial Intelligence	S ₁	3	-	-	15	20	5	-	-	60	3
CS40106	Compiler Design	S ₂	3	-	-	15	20	5	-	-	60	3
CS42120 CS42128 CS42104 CS42124 CS42111	ELECTIVE III Data Mining Machine Learning Neural Networks Algorithmic Number Theory and Algebra Software Testing and Quality Assurance	S ₃	3	-	1	10	20	5	5	-	60	4
CS42118 CS42129 CS42121 CS42116 CS42133 CS42130	ELECTIVE IV Geographical Information Systems Management Information Systems Advanced Computer Architecture Convergence Technologies Embedded Systems Image Processing	S ₄	3	-	1	10	20	5	5	-	60	4
CS40306	Compiler Design	P ₁	-	2	-					30	70	1

CS40310	Artificial Intelligence	P ₂	-	2	-					30	70	1
CS47308	PROJECT STAGE 3 (B.Tech Semester II) Irrespective of Module	PS ₂	-	12	-					30	70	6
TOTAL			12	16	2							22



CS40106:: COMPILER DESIGN

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Theory of Computation.

Unit 1: (8+1 Hrs)

Lexical Analysis and introduction to Syntax Analysis

Part A: Introduction to Compiler, Phases and Passes, Bootstrapping, Cross Compiler, Role of a Lexical Analyzer, Specification and Recognition of Tokens, LEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers.

Part B: Implementing Scanners, operator precedence parsers.

Unit 2: (8+1 Hrs)

Syntax and Semantic Analysis

Part A: Bottom-Up Parsing, LR Parsers: constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, YACC, Type Checking, Type Conversion.

Part B: Symbol-Table Structure.

Unit 3: (8+1 Hrs)

Syntax-Directed Translation and Intermediate Code Generation

Part A: Syntax-Directed Definitions, Bottom-Up Evaluation, Top-Down Translation, Intermediate Representations, Intermediate Code Generation. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.

Part B: More about translation: Array references in arithmetic expressions, case statements.

Unit 4: (8+1 Hrs)

Code Generation

Part A: Issues in Code Generation, Basic Blocks and Flow Graphs, Next-use information, A simple Code generator, DAG representation of Basic Blocks, Peephole Optimization.

Part B: Generating code from dags.

Unit 5: (8+1 Hrs)

Code Optimization and Run Time Environments

Part A: Introduction, Principal Sources of Optimization, Optimization of basic Blocks, Introduction to Global Data Flow Analysis, Runtime Environments, Source Language issues.

Part B: Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing.

Text Books

1. *“Compilers: Principles, Techniques and Tools”, A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Addison Wesley, ISBN 978-81317-2101-8, Second Edition, 2007.*
2. *“Engineering a Compiler”, K. Cooper, L. Torczon, Morgan Kaufmann, ISBN 1-55860-698-X, First Edition, 2003.*

Reference Books

1. *“Advanced Compiler Design and Implementation”, S. S. Muchnik, Morgan Kaufmann, ISBN 8178672413, First Edition, 1997.*
2. *“Lex & Yacc”, J. R. Levine, T. Mason, D. Brown, “Lex & Yacc”, J. R. Levine, T. Mason, D. Brown, O’Reilly, ISBN 1-56592-000-7, Second Edition, 1992.*

Additional Reading

1. *“Compiler Construction: Principles and Practice”, K. Loudon, Course Technology, ISBN 0-534-93972-4, First Edition, 1997.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify and Interpret the different phases of a compiler and their functioning.
2. Design a well-structured system to ensure the syntactic and semantic correctness of a program.
3. Acquaint themselves with compiler writing tools.
4. Propose techniques and performance bound solutions to address real world problems in language processing.
5. Develop compiler for a subset of a programming language with collaborative efforts.
6. Apply classical compilation principles and techniques to implement various phases of compiler which conforms to the underlying programming language specifications.

FF No. : 654 A

CS40110: Artificial Intelligence

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Fundamentals of Artificial Intelligence (8 Hrs)

Part A: Introduction, A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation

Part B: Criteria for Success, Turing Test.

Unit 2: Searching (8 Hrs)

Part A: Depth First Search, Breadth First Search, Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Means-Ends Analysis. Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence

Part B: Applications of Minimax Algorithm.

Unit 3: Planning (8 Hrs)

Part A: Planning: Blocks world, STRIPS, Implementation using goal stack, Partial Order Planning, Hierarchical planning, and least commitment strategy. Conditional Planning, Continuous Planning

Part B: Planning Graph

Unit 4: Knowledge Representation (8 Hrs)

Part A: Knowledge based agents, Wumpus world. Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.

First order Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.

Basics of PROLOG: Representation, Structure, Backtracking, Expert System.

Part B: Case study of Expert System in PROLOG

Unit 5: Uncertainty

(8 Hrs)

Part A: Non Monotonic Reasoning, Logics for Non Monotonic Reasoning, Forward rules and Backward rules, Justification based Truth Maintenance Systems, Semantic Nets Statistical Reasoning, Probability and Bayes' theorem, Bayesian Network, Markov Networks, Hidden Markov Model

Part B: Basis of Utility Theory, Utility Functions.

Text Books

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill
2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.

Reference Books

1. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Patterson: "Introduction to AI and Expert Systems", PHI
4. Nilsson : "Principles of Artificial Intelligence", Morgan Kaufmann.
5. Carl Townsend, "Introduction to turbo Prolog", Paperback, 1987

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify problems which are amenable for resolution by AI methods, and identify suitable AI methods for solving a given problem.
2. Evaluation of different algorithms on well formulated problems along with stating valid conclusions that the evaluation supports.
3. Demonstrate knowledge of basic AI algorithms through implementation for sustainable solutions of applications.
4. Apply various symbolic knowledge representations to specific multidisciplinary domains and reasoning tasks of a software agent.
5. Formulate a given problem within the framework of different AI methods with focus on building lifelong learning ability.
6. Analyze research based problems using AI techniques

CS42120 :: DATA MINING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Database Management Systems.

Unit 1: (9+2 Hrs)

Data Mining and Preprocessing

Part A: Introduction: Need of Data Mining, Knowledge Discovery in Database (KDD), Architecture of Data Mining System; Data Objects and Attribute Types, Statistical Description of Data, Data Visualization

Data Preprocessing: Central Tendency, Data Dispersion, Data Summarization, Cleaning, Integration, Transformation, Reduction, Discretization, Binarization, Variable Transformation

Part B: Relation of Data Mining with Business Intelligence and Business Analytics, Integration of Data Mining with a Database or Data Warehouse System, Basics of Business Intelligence and Data Warehousing, Issues in Data Mining

Unit 2: (7+1 Hrs)

Mining Frequent Patterns, Association and Correlations

Part A: Frequent Itemset Mining: Interesting Item Set Mining: Market Basket Analysis, Generating Association Rules, Apriori Algorithm, A pattern growth approach for mining frequent item set, Mining frequent item-sets using vertical data, Evaluation of Association Patterns, From Association Analysis to Correlation Analysis

Part B: Sequential Pattern Mining Algorithms, Pattern mining in multi-level, multi-dimensional space

Unit 3: (9+2 Hrs)

Classification and Prediction

Part A: Classification: Decision Tree Classifier, Rule Based Classification, Bayesian Classification, Neural Network Classification: Back Propagation Algorithm, Lazy Learner: kNN Classifier, Support Vector Machine
Model Overfitting, Classifier Accuracy Measures, Techniques for Evaluating Classifier Accuracy, Ensemble Methods, Multiclass Problem

Prediction: Linear, Non-Linear Regression

Part B: Case-Based Reasoning, Associative Classification, Other Classification Techniques: Genetic Algorithm, Fuzzy Set Approach, Rough Set, Constraints Based Association Mining.

Unit 4: (8+2 Hrs)

Clustering and Outlier Detection

Part A: Cluster Analysis: Categories of Clustering methods, Different Types of Clusters, Partitioning methods: k-Means, k-Medoids; Hierarchical Clustering Methods: BIRCH, Chameleon; Grid Based Methods: STING; Density based Clustering: DBScan Cluster Evaluation

Outlier Analysis: Types of outlier, Proximity based approach: distance based , Density based approach

Part B: Grid Based Methods: CLIQUE, Density based Clustering: OPTICS, Deviation based outlier detection approach: grid based

Unit 5: (7+1 Hrs)

Data Mining on different Databases

Part A: Multimedia Data Mining, Web Mining, Text Mining, Spatial Data Mining, Mining on Social Networks, Multirelational Data Mining.

Part B: Graph Mining , Mining Time – Series Data, Data Mining Applications

Text Books

1. *“Data Mining: Concepts and Techniques”, Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition.*
2. *“Introduction to Data Mining”, Pang-Ning Tan, Vipin Kumar, Michael Steinbach, Pearson*

Reference Books

1. *“Data Mining and Analysis: Fundamental Concepts and Algorithms”, Mohammed J. Zaki, Wagner Meira Jr., Cambridge University Press*
2. *“Mastering Data Mining: The art and science of customer relationship management”, M Berry and G. Linoff, John Wiley, ISBN 9971-51-369-2, 2001 Edition.*
3. *“Data Mining : Theory and Practice” , Soman K P, Diwakar Shyam, Ajay V, New Delhi, Prentice Hall Of India, ISBN 81-203-2897-3, 2006 Edition.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Identify different preprocessing techniques required for analysis of given dataset
2. Derive useful facts from data using different data mining algorithms
3. Apply suitable data mining toolkit/s to predict useful patterns
4. Illustrate use of appropriate analysis technique/s given the data specifications
5. Participate in applying relevant technology for societal benefit
6. Solve real life problems using data analysis techniques.

CS42128:: MACHINE LEARNING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Artificial Intelligence.

Unit 1: (7 Hrs)
Fundamentals

Part-A:

Soft computing and hard computing, types of learning, activation functions, Learning Rules, Mc-Culloch-pitts Neuron model, single layer and multilayer perceptron, Backpropagation algorithm, Well posed learning problem, Designing a learning system, Perspectives and issues in soft computing,

Part B: Solving Problems based on above algorithms.

Unit 2: (7 Hrs)
Concept and decision tree Learning

Part-A:

Concept Learning, General-to-Specific Ordering: Task, search, Find S algorithm, Version space and the candidate elimination algorithm, list then eliminate algorithm, inductive bias, Decision Tree Learning: representation, Basic decision tree learning algorithm, Hypothesis space, Issues in decision tree learning.

Part B: Solving Problems based on above algorithms.

Unit 3: (6 Hrs)
Evaluating Hypothesis

Evaluating Hypotheses, Evaluating hypothesis accuracy, Sampling theory, Central limit theorem, hypothesis testing, comparing learning algorithms. Bayesian Learning, Bayes theorem, Maximum likelihood hypothesis, minimum description length principle, Gibbs algorithm, Bayesian belief networks, Expectation maximization algorithm.

Part B: Solving Problems based on above algorithms.

Unit 4:
Machine Learning Algorithm

Hidden markov model, Genetic algorithm, SVM, Kernel functions, Linear SVM, Nonlinear SVM, Regression analysis, ensemble classifiers, Validation, evaluation.

Part B: Solving Problems based on above algorithms.

Unit 5:

(7 Hrs)

Clustering Algorithm and recurrent Networks

Part-A: k-means algorithm, k-nearest neighbor learning, weighted majority algorithm, Hopfield Net, Hamming net, Maxnet, Kohonen self organizing map, Principal component Analysis (PCA), Applications of machine learning.

Part B: Solving Problems based on above algorithms. locally weighted regression,

Text Books

1. T. Mitchell, "Machine Learning", McGraw-Hill, 1997.
2. Anup Kumar Srivastava, *Soft Computing*, Alpha Science International limited. 2009.
3. *Introduction to neural networks*, S. N. Shivanandam, Mc-Graw Hill, 2013.

Reference Books

1. Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004.
2. Jacek M. Zurada, "Introduction to Artificial neural System", JAICO publishing house, 2002,.
3. Research papers suggested by the faculty.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Evaluation of different algorithms on well formulated problems along with stating valid conclusions that the evaluation support.
2. Application of Machine learning toolkits for implementation of different problem domains.
3. Demonstrate knowledge of supervised, unsupervised and reinforcement Machine Learning algorithms through implementation for sustainable solutions of applications.
4. Apply different Mathematical models used in machine learning to specific multidisciplinary domains.
5. Formulate a given problem within the framework of different machine learning methods with focus on building lifelong learning ability.
6. Analyze research based problems using Machine learning techniques.

CS42104:: NEURAL NETWORKS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+2 Hrs)
Introduction

Part A: Introduction and Role of Neural Network (NN), Applications: Signal Processing, Pattern recognition, Medicine, Speech Production/Recognition, Business etc., Artificial Neural Networks, Biological Neural Networks, Components of Neural Network, Network topologies, Linear Separability, Hebb Net, Perceptron, Adaline/Madaline, Paradigms of Learning: Unsupervised, Reinforced, Supervised, Competitive, Offline or Online, Activation Functions

Part B: Examples on Learning.

Unit 2: (8+1 Hrs)
The Perceptron, Backpropagation and other Neural Networks

Part A: Standard Backpropagation: Architecture, Algorithm, Applications, variations, Theoretical results, Boltzmann machine, Continuous Hopfield, Gaussian Machine, Cauchy machine neural networks

Unit 3:
Competitive Neural Network

Part A: Fixed weight Competitive Neural Network, Kohonen Self Organizing Maps, Learning Vector Quantization, Counter propagation: Fully Counter propagation neural network, Forward only Counter propagation Neural network.

Part B: Applications of Self organizing Maps

Unit 4: (8+1 Hrs)
Adaptive Resonance Theory(ART)

Part A: The ART Network Structure, Network Operation, Properties of ART, ART-1, ART2

Network, Architecture, Algorithm, Applications.

Part B: Case Study: ART2 Application

Unit 5: (8+2 Hrs)
Pattern Association

Part A: Training Algorithm for Pattern association, Heteroassociative Memory Neural Network, Auto associative Neural Network, Iterative Auto associative Neural Network, Discrete Hopfield Network, Bidirectional associative Memory (BAM),

Part B: Perceptron Case Study

Text Books

1. *“Fundamentals of Neural Networks: Architectures, Algorithms and Applications”*, Laurene Fausett, Prentice Hall Edition
2. *“Principles of Artificial Neural Networks”*, Daniel Graupe, World Scientific Publishing, ISBN 13-978-981-270-624-9, 2nd Edition.
3. *“Artificial Neural Networks”*, B. Yegnanarayana, PHI Learning Private Limited, ISBN 978-81-203-1253-1, 2010 Edition.

Reference Books

1. *“Introduction to Artificial Neural Systems”*, Jacek M. Zurada, Jaico Publishing House, ISBN 81-7224-650-1, 2006 Edition.
2. *“Neural Networks a Comprehensive Foundation”*, Simon Haykin, Pearson Education, ISBN 81-7808-300-, 2nd Edition.

Additional Reading

1. *“Neural Networks a Classical Approach”*, Satish Kumar, Tata McGraw-Hill Publishing Company Limited, ISBN 0-07-048292-6, 2004 Edition.
2. *“Neural Networks for Pattern Recognition”*, Christopher M. Bishop, OXFORD University Press, ISBN 0-19-566799-9.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. To Understand a number of models for supervised, unsupervised, and reinforcement neural networks systematically.
2. student should able to analyze different algorithms according to the properties of their inputs and outputs using different types of big data.
3. Design the most appropriate neural network for classification, Clustering, automatic detection and optimization.
4. Implement the algorithms in a software environment using MATLAB / Neural ware Professional and R-Programming Language.
5. To evaluate the neural network algorithms

CS42111: Software Testing and Quality Assurance

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Software Engineering, Database Management System

Unit 1:: Software Measurement

(8 Hrs)

Part A: Measurement Theory and Why to Measure the Software, Measurement and Models, Measurement Scales, Classification of Software Measures, Measurement Framework, Measuring Internal Product Attributes: Size, Measuring Internal Product Attributes : Structure, Halstead's Software Science, Product Quality Metrics, In-Process Quality Metrics, Software Reliability: Measurement and Prediction, Planning a Measurement Program, Measurement in Practice

Part B: The Rayleigh Model, Exponential Distribution and Reliability Growth Models, SRE process, Reliability Concepts: Hardware and Software, Deploying SRE

Unit 2:: Principles of Testing

(8 Hrs)

Part A: Testing Concepts: Purpose of Software Testing, Testing Principles, Goals of Testing, Testing aspects: Requirements, Test Scenarios, Test cases, Test scripts/procedures, Strategies for Software Testing, Testing Activities, Mistakes, Faults & Failures, Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods, Levels of Testing

White-Box Testing: Test Adequacy Criteria, Static Testing, Structural Testing, Code Complexity Testing, Mutation Testing, Data Flow Testing

Black-Box Testing: Test Case Design Criteria, Requirement Based Testing, Positive and Negative Testing, Boundary Value Analysis, Equivalence Partitioning State Based Testing, Domain Testing

Part B: Analysis of Flow Graphs, Complexity Measures and computations

Unit 3:: Functional Testing

(8 Hrs)

Part A: Test Plan, Test Management, Test Execution and Reporting, Test Specialist Skills, Tester's Workbench and Tool Categories, Test Maturity Model and Test Process Assessment, Debugging & Root Cause Analysis, Software Items, Component & Units, Test Bed, Traceability and Testability, Attributes of Testable Requirements, Test Matrix, Benefits of Formal Test Documentation, Types of Testing Documentation, Verification Testing, Validation Testing, Integration Testing, System and Acceptance Testing, GUI Testing, , Regression Testing, Selection, Minimization and Prioritization of Test Cases for Regression Testing, Creating Test Cases from Requirements and Use cases

Part B: Software Defects: Origins of Defects, Defect Classes, Defect Repository / Test Design, Developer/Tester Support for Defect Repository, Need for Testing

Unit 4:: Higher Order Testing

(8 Hrs)

Part A: Object Oriented Testing, Specification Based Testing, Performance Testing, Ad-hoc Testing, Usability and Accessibility Testing, Risk-based Testing, Exploratory Testing,

Scenario-based Testing, Random Testing Compatibility Testing, User Documentation Testing, Client –Server System Testing, RAD Testing, Configuration Testing, Testing internal Controls, Multiplatform Environment Testing, Security Testing, Web-based System Testing, Testing VS Test Automation, Tool evaluation and selection, Automation team roles, Architectures, Planning and implementing test automation process

Part B: IEEE Standards Related to Testing, ISO 12207 & IEEE/EIA Std12207on Testing, ISO 9000:2000 on Testing, ISO 9003, IEEE Std 1012, IEEE Std 1028

Unit 5:: Software Quality Assurance **(8 Hrs)**

Part A: Quality Concepts, Software Quality Assurance, Planning for SQA, Six Sigma Principles, Malcolm Baldrige Assessment, Edward Deming's Principles, Total Quality Management, Ishikawa's Seven Basic Tools, Using Agile Methods to Improve Software Testing, Building Agility into the Testing Process, Software Maintenance, Purpose of Inspection, Cost of Inspections, Defect Detection and Prevention, Feedback, Feed-forwards and Control, Flow of Inspection Process, Activities of Inspection, Economics of Inspections

Part B: Software Benchmarks and Baselines, Key Factors for Software Assessment and Benchmark Studies, Identifying Software Best and Worst Practices

Text Books:

1. Fenton, Pfleeger, "Software Metrics: A Rigorous and practical Approach", Thomson Brooks/Cole, ISBN 981-240-385-X.
2. Desikan, Ramesh, "Software Testing: principles and Practices", Pearson Education, ISBN 81-7758-121-X.

Reference Books :

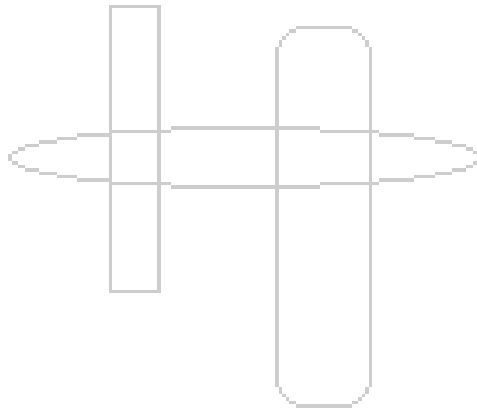
1. Burnstein, "Practical Software Testing", Springer International Edition, ISBN 81-8128-089-X
2. William E. Perry, " Effective Methods for Software Testing", John Wiley and Sons, ISBN 9971-51-345-5
3. Yogesh Singh, "Software Testing", Cambridge University Press, ISBN 978-1-107-01296-7
4. Nina Godbole, "Software Quality Assurance: Principles And Practice", Alpha Science International, Ltd (August 1, 2004) ISBN-10: 1842651765 ISBN-13: 978-1842651766
5. Ronald Radice, "Software Inspections", Tata McGraw Hill, ISBN 0-07-048340-X
6. Capers Jones, " Software Assessments, Benchmarks, and Best Practices" ,Burlington, Vermont Publisher: Addison-Wesley Professional 2000 ISBN-10: 0201485427 ,ISBN-13: 9780201485424

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Select and classify measurement scales and models, software metrics and measures addressing software quality and reliability.
2. Conduct unit and integration tests by determining test design, test automation, test coverage criteria using testing frameworks and test adequacy assessment using control flow, data flow, and program mutations.
3. Apply suitable higher order testing techniques and methods in order to achieve verified and validated software by following testing best practices.

4. Demonstrate the skillset as a tester to neutralize the consequences of wicked problems by narrating effective test cases and test procedures.
5. Adapt to various test processes, types of errors and fault models and methods of test generation from requirements for continuous quality improvement of the software system.
6. Apply software testing cycle in relation to software development and project management focusing incidents and risks management within a project towards efficient delivery of software solutions and implement improvements in the software development processes.



CS42124:: ALGORITHMIC NUMBER THEORY AND ALGEBRA**Credits:** 03**Teaching Scheme:** - Theory 3 Hrs/Week

Prerequisites: Basic introduction to linear algebra and abstract algebra (though it is not presumed) is helpful to better appreciate the contents of the course. First course on algorithms.

(8+2 Hrs)**Unit 1:****Basic Number Theory**

Part A: Infinitely many primes in \mathbb{Z} (some proofs), greatest common divisor, unique factorization of integers, modular arithmetic, (some results on modular arithmetic: Fermat's little theorem, Wilson's theorem,... etc), linear congruences and Chinese Remaindering Theorem, linear Diophantine equations, some arithmetic multiplicative functions (e.g. Euler's phi function, Mobius function, divisor function...), Dirichlet convolution, Mobius inversion. Quadratic residues, Legendre symbol, Gauss' law of Quadratic reciprocity. Distribution of primes, Prime number theorem.

Part B: More Diophantine equations (e.g. Pell's equation), more results on distribution of primes (proving lower and upper bound (asymptotic) on $\pi(x)$, where $\pi(x)$ is number of primes upto x)

Unit 2:**Basic abstract Algebra**

Part A: Groups, subgroups, Lagrange's theorem, group homomorphism, quotient groups, normal subgroups, cyclic groups, Abelian groups and their decomposition, rings, ideals, some examples of rings (e.g. polynomial ring $F[x]$, ring of integers modulo n), units and zero divisors in the ring, structure of unit group of \mathbb{Z}_n , integral domains, fields, vector spaces. Unique factorization in ring $k[x]$ where k is a field, Principal Ideal Domains (PID).

Basics of finite fields: existence of finite fields of size p^n for $n > 0$, p prime, isomorphism of finite fields of same size, structure of multiplicative group, field automorphisms, Frobenius maps.

Part B: Ring of Gaussian integers $\mathbb{Z}[i]$ and Lagrange's four square theorem, analogies between ring of integers and univariate polynomial ring, more properties of finite fields.

Unit 3:**Number theoretic algorithms**

Part A: Euclid's algorithm for GCD, Extended Euclid's algorithm, fast integer arithmetic, algorithms for modular arithmetic, fast exponentiation, finding modular inverse, finding order of element in a group, finding quadratic non-residues, algorithm for modular square-root, algorithms for discrete-log problem.

Univariate polynomial arithmetic (multiplication, gcd, etc), irreducibility testing for

polynomials in $F[x]$ where F is finite field, ring $F[x]/(f)$, modular inverse computation.

Part B: Chinese Remaindering and computing determinant of integer matrix, Euclid's algorithm and Fibonacci numbers (a detailed analysis of Euclid's algorithm), Some more algorithmic questions over finite fields.

Unit 4: **(9+2 Hrs)**
Polynomial Factorization

Part A: Univariate polynomial factorization over finite fields: revisit unique factorization, Randomized polynomial time algorithms:

- a. Berlekamp's algorithm: linear algebra preliminaries for Berlekamp's algorithm, Frobenius map, the algorithm, resultants and some implications of Berlekamp's algorithm, parallel algorithm for computing gcd of polynomials.
- b. Cantor-Zassenhaus algorithm: distinct degree and equal degree factorization, CRT and Cantor-Zassenhaus algorithm.

Geometry of Numbers and polynomial factorization over rationals: Basics on integer lattices, Minkowski's theorems, algorithmic questions over integer lattices (Closest Vector Problem and Shortest Vector Problem) Gauss' algorithm for SVP in 2 dimensions. Lenstra-Lenstra-Lovasz (LLL) approximation algorithm for SVP. Deterministic polynomial time algorithm for factorization of univariate polynomials over rational (using LLL and Hensel lifting lemma)

Part B: Reduction of polynomial factorization problem to root finding using Berlekamp's algorithm. Recent deterministic sub-exponential time algorithm for the

Unit 5: **(7+1 Hrs)**
Primality

Part A: Pratt's result: PRIMES in NP intersection coNP, randomized polynomial time algorithms:

- a. Miller-Rabin primality test: Carmichael numbers, derandomizing Miller-Rabin test using Riemann Hypothesis, computing a factor of a number given $\phi(n)$ where ϕ is Euler's totient function.
- b. Solovay-Strassen primality test: Jacobi symbol, quadratic reciprocity, Solovay-Strassen primality test.-

Prime number theorem and generating large primes (application to RSA crypto system)

Part B: Agarwal-Kayal-Saxena (deterministic polynomial time) primality test.

Text Books

1. *Modern Computer Algebra* by Joachim von zur Gathen, Jürgen Gerhard (Cambridge)

2. *A computational introduction to Number Theory and Algebra by Victor Shoup(Cambridge)*
3. *A classical introduction to modern number theory by Ireland and Rosen (Springer)*

Reference Books

1. *Topics in Algebra by I. N. Herstein (Wiley Publishing company)*
2. *Algebra by Michael Artin (Pearson Prentice Hall)*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. solve mathematical problems based on elementary number theory
2. design efficient algorithms for various number theoretic and algebraic computational problems
3. explain various issues arising in designing efficient algorithms for testing primality of a natural number
4. describe efficient algorithmic solutions for shortest vector problem over integer lattices and their applications
5. evaluate various known algorithms for multivariate polynomial factorization over finite fields and field of rationals
6. comprehend properties of various algebraic structures such as groups, rings, fields, vector-spaces and their role in design and analysis of number theoretic algorithms

CS42118:: GEOGRAPHICAL INFORMATION SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+1 Hrs)
GIS and Maps

Part A: Fundamentals of GIS: Roots of GIS, Overview of Information System, Four Ms, GIS Definition and Terminology, GIS Queries, GIS Architecture, Models of GIS, Framework for GIS, GIS Categories, Levels/Scales of Measurement. Map Language: Map as a Model, Classification of Maps, Spatial Referencing System, Map Projections, Commonly Used Map Projections, Grid Systems, Computer in Map Production, Digital Database in a GIS, Linkage of GIS to Remote Sensing

Part B: Selection of a GIS Application in Various Domains such as Weather Forecasting, Urban Planning, Agriculture, Defense, Network Applications.

Unit 2: (8+2 Hrs)
Remote Sensing Fundamentals

Part A: Remote Sensing - Basic Principles: Electromagnetic Remote Sensing, Energy Sources, Energy Interactions with Earth's Surface Materials, Microwave Remote Sensing: Radar Principle, Factors affecting Microwave Measurements, Radar Wavebands, SLAR Systems, Interpreting SAR Images, Geometrical Characteristics. Remote Sensing Platform and Sensors: Satellite System Parameters, Sensor Parameters, Imaging Sensor Systems, Earth Resources Satellites, Meteorological Satellites.

Part B: Study of Satellites such as IRS, OCEANSAT-1, IKONOS etc.

(8+2 Hrs)

Unit 3:
Image Processing

Part A: Digital Image Processing: Basic Character of Digital Images, Preprocessing, Registration, Enhancement, Spatial Filtering, Transformations, Classification, Image Classification and GIS. Visual Image Interpretation: Types of Pictorial Data Products, Image Interpretation Strategy, Image Interpretation Process, Overview of Image Interpretation Equipments.

Part B: Study of GIS Hardware and Software required specially for Image Processing.

Unit 4: (8+2 Hrs)
Spatial Data Modeling and Management

Part A: Spatial Data Modeling: Stages of GIS Data Modeling, Graphic Representation of Spatial Data, Raster GIS Models, Vector GIS Models, GIS Data Management: Database

Management Systems, GIS Data File Management, Database Models, Storage of GIS Data Object based Data Models, Temporal Topology, Organizational Strategy of DBMS in GIS.

Part B: Design a Spatial Database for a Selected Application.

Unit 5: **(8+1 Hrs)**
Data Input, Quality and Analysis

Part A: Data Input and Editing: The Data Stream, Data Input Models, Data Input Methods, GPS for GIS Data Capture, Data Editing. Data Quality Issues: Components of Data Quality, Accuracy, Precision and Resolution, Consistency, Completeness, Sources of Error in GIS, GIS Output, Modeling Errors and Error Evaluation. Data Analysis and Modeling: Format Conversion, Data Medium Conversion, Spatial Measurement Methods, Buffering Techniques, Overlay Analysis, Modeling Surfaces.

Part A: Identification of Data Inputs Outputs and Study of Required Analytical

Text Books

1. *“Remote Sensing and Geographical Information Systems”, M. Anji Reddy B S Publications, Second Edition, 2006.*
2. *“Introduction to Geographic information systems”, Kang-Tsung Chang, Tata-McGraw-Hill Publications, Third edition, 2006. ISBN: 978-0073101712.*

Reference Books

1. *“Principles of Geographical Information Systems”, Peter A Burroughs and McDonnell, Oxford University Press, 1998. ISBN 978-0198233657.*
2. *“The GIS Book”, George B Korte, Onward press (Thomson Learning), 5th Edition, 2001. ISBN 81-7800-112-8.*

Additional Reading

1. *Bolstad, Paul “GIS Fundamentals.”, Eider Press. 2nd edition 2005. ISBN 978-0-9717647-2-9.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Differentiate among map projections for geographical areas
2. Design spatial data model for geographical area encompassing related entities
3. Create database for geographical area encompassing related entities using GIS software
4. Propose effective remote sensing based solutions addressing requirements in domains such as urban planning, weather forecasting, defense and land management
5. Display the impact of data processes such as data input, quality, analysis and output in GIS applications
6. Build skill set required for responsible positions such as GIS Analyst, GIS Technician and GIS Specialist

CS42129: MANAGEMENT INFORMATION SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+2 Hrs)

Foundations of Information Systems

A. Introduction, Why Information Systems (IS)? IS Framework, System Concepts, Components of IS, Major Roles of IS, Trends in IS, Major Types of IS– Transaction Processing Systems (TPS), MIS, Decision Support Systems (DSS), etc, Organization Basic, Features of Organizations, Models of Organizations, Competitive Strategy Concepts, Strategic Uses of Information Technology, The Value Chain.

B. Selection of a Domain: Banking, Healthcare, Hotel, Telecom, Education, Agriculture, Shopping Mall, Automobile, Food Industry etc.

Unit 2: (8+2 Hrs)

Manufacturing and Service Systems

A. Functional Levels in Manufacturing Systems, Personnel Management, Financial Management, Production Management, Material Management, Marketing Management, MIS for Manufacturing Sector, Service Sector, Distinctive Services, Service Vs. Product, Service Process Cycle and Analysis, Customer Service Design, MIS for Service Sector, Insurance and Airline

B. Identification of Functional Levels, Services and Products in Selected Domain.

Unit 3: (8+2 Hrs)

e-Business

A. Enterprise Resource Planning (ERP), Benefits and Challenges of ERP, Trends in ERP, Supply Chain Management (SCM), The Role of SCM, Benefits and Challenges of SCM, Trends in SCM, Customer Relationship Management (CRM), The Three Phases of CRM, Benefits and Challenges of CRM, Trends in CRM, Electronic Commerce (e-Commerce), Scope of e-Commerce, Essential e-Commerce Processes, Electronic Payment Processes, B2C e-Commerce, B2B e-Commerce, Business Standards.

B. Study of Process to accommodate e-Business Approach in Selected Domain.

Unit 4: (8+2 Hrs)

Information Systems for Decision Support

A. Business and Decision Support, Decision Making Process, Components of DSS, MIS, Difference between DSS and MIS, Online Analytical Processing, Types of DSS, Using DSS, What-if analysis, Sensitivity analysis, Goal-seeking analysis, Optimization analysis, Data Mining for Decision Support, Executive Information Systems, Knowledge Management

Systems, Group Decision Support Systems (GDSS), Components of GDSS, Overview of a GDSS Meeting, Expert Systems.

B. Identify Decision-making Aspects in a Selected Domain with Appropriate Examples.

Unit 5:

(8+2 Hrs)

Challenges Ahead

Part A: Introduction to Security and Ethical Challenges, Ethical Responsibility of Business Professionals, Computer Crime, Hacking, Cyber Theft, Unauthorized Use at Work Software Piracy, Piracy of Intellectual Property, Computer Viruses and Worms, Security Management, Tools, Encryption, Firewalls, e-Mail Monitoring, Biometric Security, Disaster Recovery, Fault Tolerant Systems, etc, Global Management of Information Technology, Cultural, Political and Geo-economic Challenges, Global Business/IT Strategies.

Part B: Study of Cybercrimes and Preventive Measures w. r. t. Selected Domain.

Text Books

1. "Management Information Systems: Managing Information Technology in the Business Enterprise", James O'Brien, Tata McGraw-Hill Publishing Company Limited, 10th Edition, ISBN 0-07-058739-6.
2. "Management Information Systems", Jawadekar Waman S, Tata McGraw-Hill Publishing Company Limited, ISBN 0-07-044575-3, 2nd Edition.

Reference Books

1. "Management Information Systems: Managing the Digital Firm", Kenneth C. Laudon, and Jane P. Laudon, Prentice-Hall of India, ISBN 81-203--2908-2, 9th Edition.
2. "Management Information Systems", Oz Effy, Singapore, Thomson, ISBN 81-315-0174-4, 5th Edition.

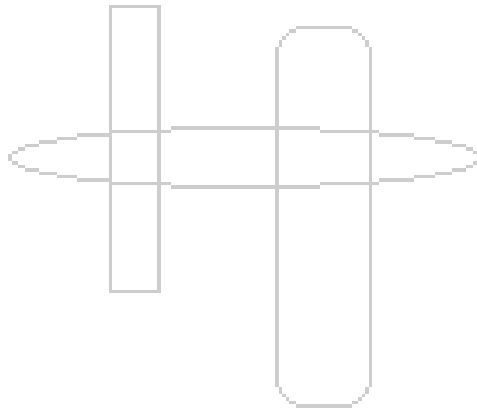
Additional Reading

1. "Management Information Systems", Shajahan S, Priyadharshini R, New Age International, ISBN 81-224-1549-0.
2. "Management Information Systems", Arora Ashok, Bhatia Akshaya, Excel Pub, ISBN 81-7446-188-4.
3. "Information Systems A management Perspective", Alter Steven, Addison Wiley, ISBN 0-201-35109-9, 3rd Edition.
4. "Information Systems For Modern Management", Murdick R G, Ross J E, Claggett J R, Prentice Hall Of India, ISBN 81-203-0397-0, 3rd Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Estimate the functional complexities in manufacturing and/or service sectors for implementation of Management Information Systems.
2. Design solutions differentiating management information systems based on their features and applicability.
3. Initiate ethically responsible behaviour as a professional.
4. Build a set of professional skills required for responsible positions such as System Analyst, Business Consultant and Information System Manager.
5. Respond positively to cultural, political, economical and organizational challenges during process of project management.
6. Follow required domainspecific processes and standards for management information systems.



FF No. : 654 A

CS42121: ADVANCED COMPUTER ARCHITECTURE

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Organization

Unit 1:

(8+1 Hrs)

Overview of Parallel Processing

Part A: Overview of Parallel Processing and Pipelining Processing, study and comparison of uni-processors and parallel processors.

Necessity of high performance, Constraints of conventional architecture, Parallelism in uni-processor system, Evolution of parallel processors, future trends, Architectural Classification, Applications of parallel processing, Instruction level Parallelism and Thread Level Parallelism

Part B: Explicitly Parallel Instruction Computing (EPIC) Architecture, Performance Metrics and Measures, Speedup Performance Laws.

Unit 2:

(8+1 Hrs)

Pipelining Processing

Part A: Principles and implementation of Pipelining, Classification of pipelining processors, Pipeline Architecture, Study and comparison of processors with and without pipelining. General pipelining reservation table, Design aspect of Arithmetic and Instruction pipelining, Pipelining hazards and resolving techniques, Data buffering techniques, Job sequencing and Collision, Advanced pipelining techniques, loop unrolling techniques, out of order execution, software scheduling, trace scheduling.

Part B: Advances in pipeline architectures. Implementation issues of a program on any pipelined processor their analysis.

Unit 3:

(8+1 Hrs)

SIMD Computer Organization and Parallel Algorithms For Array Processors

Part A: Study and comparison of Vector and array processors, Vector and Array Processor, Basic vector architecture, Issues in Vector Processing, Vector performance modeling, vectorizers and optimizers, Case study: Cray Arch. Masking and Data network mechanism, Inter PE Communication, Interconnection networks of SIMD, Static Vs Dynamic network, cube hyper cube and Mesh Interconnection network. Matrix Multiplication. Sorting, SIMD computer organization.

Part B: Implementation issues of Matrix multiplication and sorting on array processor and their analysis.

Unit 4:

(8+1 Hrs)

Multiprocessor

Part A: Microprocessor Architectures, study and comparison of Loosely and Tightly coupled multiprocessors.

Loosely and Tightly coupled multiprocessors, Processor characteristics of multiprocessors, Inter Processor communication network, Time shared bus, Crossbar switch, Multiport Memory Model, Memory contention and arbitration techniques, Cache coherency and bus snooping, Massively Parallel Processors (MPP), Cow's and NOW's Cluster and Network of Work Stations), Chip Multiprocessing (CMP).

Part B: Implementation issues of a program on multiprocessor system

Unit 5:

(8+1 Hrs)

Parallel Programming Techniques

Part A: Study of Architecture of Multithreaded processors, Latency hiding techniques, Principles of multithreading, Issues and solutions.

Message passing program development, Synchronous and asynchronous message passing, Message passing parallel programming, Shared Memory Programming, Data Parallel Programming.

Part B: Implementation issues of a multithreaded program.

Text Books

1. Kai Hwang, Faye A. Briggs, "Computer Architecture and Parallel Processing" McGraw-Hill international Edition.
2. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill.

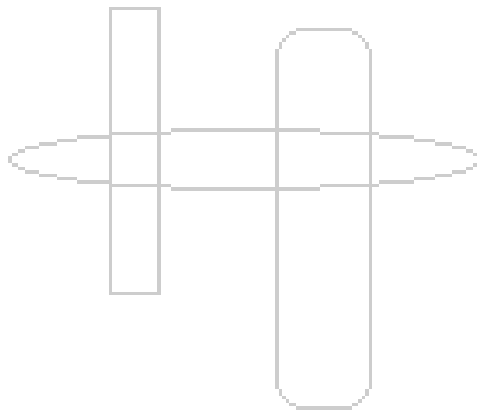
Reference Books

1. Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
2. William Stallings, "Computer Organization and Architecture, Designing for performance" Prentice Hall, Sixth edition.
3. Kai Hwang, Scalable Parallel Computing.
4. Harrold Stone, High performance computer Architecture.
5. Richard Y. Kain, Advanced Computer Architecture
6. <http://www.intel.com/products/processor> (for Intel Itanium Processor).

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Create and use instruction level parallelism, memory level parallelism, and thread level parallelism..
2. Design pipelined processor architecture and resolve different pipeline hazards.
3. Design parallel algorithms for Array processors
- 4 Mitigate design issues for multiprocessor systems, including cache coherency issue
- 5 Develop applications using multithreading architecture
- 6 Analyze architectures performance and select among different ones for particular use.



CS42116 :: CONVERGENCE TECHNOLOGIES

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks.

Unit 1: (7+2 Hrs)

Introduction to Convergence

Part A: what is network Convergence, the promise of network convergence, networking issues and convergence, Voice and data network characteristics, benefits of IP centric network, challenges of converged network, introduction to VOIP, applications of converged networks, VOIP implementation challenges.

Part B: voice and data network growth factor, effects of network convergence on businesses.

Unit 2: (9+1 Hrs)

Protocols and Standards for Convergence

Part A: Protocols Supporting VOIP: Multicast IP, RTP, RTCP, RSVP, RTSP, SDP, SAP, SIP.

Subscriber Lines: T1/T3, DS0, DS1, DS3, E1/E3.

Signaling Standards: H.323, SIP. Gateways, Gatekeepers.

Part B: MGCP, Audio and Video Codecs.

Unit 3: (9+1 Hrs)

Switching networks

Part A: ISDN: conceptual view of ISDN, transmission structure, user-network interface configuration, ISDN Protocol Architecture, ISDN connection, Addressing, Interworking, PRI, BRI, LAPD,

Basic Call control, SS7.

B-ISDN standards, Broadband services, B-ISDN architecture, B-ISDN protocol reference model.

Part B: ISDN standards, SDH.

Unit 4: (7+1 Hrs)

Frame Relay and SMDS

Part A: Frame Relay Circuits, Frame mode protocol architecture, frame mode call control, LAPF, Congestion in frame relay networks, approaches to congestion control, Traffic rate management, Explicit congestion avoidance, implicit congestion control.

SMDS: introduction to SMDS, SMDS interface protocol, SMDS addressing.

Part B: Comparison of SMDS with other LAN technologies.

Unit 5: (8+1 Hrs)

ATM technology

Part A: ATM VPI& VCI, Creation of virtual channel, Definitions of Virtual circuit and permanent virtual circuit, ATM reference model, step-by-step PVC example of how ATM network processes cells, AAL, Adaption layer from voice over ATM perspective AAL1,AAL2, AAL3, Connection admission control (CAC). Cell Loss Priority (CLP), ATM and convergence technology.

Part B: ATM versus Frame relay, ATM versus SONET.

Text Books

1. *“ISDN and Broadband ISDN with frame relay and ATM”* by William Stallings, Pearson Education, 2003, ISBN 81-7808-422-8, 4th Edition.
2. *“Voice over IP Technologies”* by Mark A. Miller, P.E., Wiley Publications, 2002, ISBN 81-265-0286-X, 1st Edition.

Reference Books

1. *“Computer Communications and Networking Technologies”*, by Michael A. Gallo, William M. Hancock, Cengage Learning, 2002, ISBN81-315-0364-X, 1st Edition.
2. *“ATM network concepts and protocols”*, by Sumeet Kasera and Pankaj Sethi, Tata McGraw Hill, 2001, ISBN 0-07-463776-2, 1st Edition.

Additional Reading

1. *“VOIP”*, by Ulyess Black, Pearson Education, 2001, ISBN 0130652040, 2nd Edition.
2. *“Multimedia communication system techniques standards and networks”*, by K.R. Rao, Zoran Bojkovic and Dragorad Milovanovic, Pearson Education, 2002, ISBN 0-13031398-X, 1st Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Design a network to support voice and data services.
2. Use of Standard Tools for converged network design.
3. Judge the impact and benefits of converged network in exploitation on environment and society.
4. Participate individually and collaboratively to examine the performance of converged-network..
5. Prepare cost effective solutions to fulfil the need of society by convergence technology.
6. Analyse the requirements of the society and propose efficient solutions using convergence technology..

CS42133 :: EMBEDDED SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Understanding of Microprocessors, Peripheral Chips, Conversion, Interfacing Techniques.

Unit 1: Introduction to Embedded Systems (8+1 Hrs)

Part A: Introduction to Embedded Systems, Review of Microprocessors and Micro-controllers, Multiprocessor systems using General Purpose Processor. CISC and RISC Processor architectures. DSP processor. Design Process in Embedded System, Components of Embedded System & its Classification, Characteristic of embedded system. Introduction to embedded processor, Digital Signal Processor, Application Specific System Processor, Design Process in Embedded System, Design metrics, Steps in design process. Challenges in Embedded System design, Design Examples. Advances in Embedded Systems

Part B: Memory Selection Criterion

Unit 2: Bus Design (8+1 Hrs)

Part A: Introduction to the BUS System, Bus design issues, Synchronous Bus, Asynchronous Bus, Bus Allocation, Bus Priority. Protocol Architecture, topology, different Packets, Communication Cycle, Arbitration, Applications. Interfacing Buses: I2C, SPI, CAN .

Part B: Study of RS 232C , Centronics and USB

Unit 3: Microcontroller Support Units (8+1 Hrs)

Part A: Details of Components of Embedded Systems-Management of Power Supply, Clocking Unit, Real Time Clock and Timers, Reset Circuitry and Watchdog Timer. Structural Units of Processor, Processor and Memory Selection, Memory Map Of Embedded System, Interfacing Processors , Memories and I/O. Processor , Memory Map Of Embedded System

Part B: Design Parameters problem solving. PWM ADC Timers

Unit 4: I/O Interfacing (8+1 Hrs)

Part A: I/O interfacing and Communication I/O devices, ADC / DAC, Optical Devices such as LED / LCD Display devices, Opto-Isolator, Relay & stepper motor, Timers/Counters. Parallel v/s serial communication. Parallel ports their uses in device interfacing.

Part B: Design Implementation of OP Amps

Unit 5: RTOS (8+1 Hrs)

Part A: Device drivers, Interrupts, Interrupt Service Routines, Scheduling Algo, Inter Process Communication, Process Syn+chronisation. Multiple Tasks, Threads, Processes. Shared Data / Priority Inversion Char: of RTOS, Real Time Scheduling of RTOS

Part B: Case Study of Embedded Systems in Detail.(H/W + S/W Algo)

Text Books

1. Raj Kamal 'Embedded Systems ', Tata McGraw-Hill . ISBN0-07-049470-3
2. Dr. K. V. K. K. Prasad "Embedded/Real time System : Concepts, Design, & Programming ". Dreamtech Press

Reference Books

1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system " Wiley -Dreamtech India Pvt. Ltd.
2. Sriram Iyer , Pankaj Gupta, "Embedded Real time Systems Programming", Tata Mc-Graw Hill, 2004.

Additional Reading

1. Microcontroller Handbook
2. CAN Specification Version 2.0 Protocol Standard.
3. USB Specification Version 2.0 Protocol Standard.
4. I2C Specification Protocol Standard.
6. Tammy Noergaard, — Embedded Systems Architecture by Elsevier

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Understand Steps in System Design using Computing Devices.
2. Design system interconnects for effective throughput.
3. Create designs using Simulation and RTOS Tools.
4. Deliver Complex Automation Solutions.
5. Cooperate with diverse Teams for delivering automation Solutions.
6. Adapt to ever changing technological Advances.

CS42130 :: IMAGE PROCESSING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: (8+1 Hrs)

Introduction to image processing

Part A: Components of image processing system, Scenes and Images, Vector Algebra, Orthogonal Transform, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, and some basic relationships between pixels, linear and non linear operations.

Part B: Application of different color models in Image processing.

Unit 2: (8+1 Hrs)

Image Formation and Digitization

Part A: Geometric Model, Photometric Model, Sampling, Digitization, Elements of Digital Geometry, Image Properties, Representation

Part B: Overview of application of Image processing.

Unit 3: (8+1 Hrs)

Image Processing

Part A: Basic gray level transformations, histogram processing, enhancement using logical and arithmetic operations Image Enhancement - Contrast Intensification, Smoothing, Sharpening Image Restoration -- Square Error Restoration Techniques, Singular Value Decomposition.

Part B: Image Compressing - Basic - Lossy Compression, Loss-Less Compression.

Unit 4: (8+1 Hrs)

Image transform

Part A: Introduction to two dimensional orthogonal and unitary transforms, properties of unitary transforms one-two dimensional discrete Fourier Transform (DFT), Wavelet transforms. Cosine, sine transforms.

Part B: Slant, KL, affine transforms. Applications of transforms in Image processing.

Unit 5: (8+1 Hrs)

Compression fundamental

Part A: File format (bmp, tiff, pcx, gif, jpeg.) Compression fundamentals, image compression models, error free compression: LZW coding, Bit plane Coding, Lossless Predictive Coding, lossy compression: lossy Predictive Coding, Transform, vector Quantization, image compression standard: Binary Image, Continuous Tone Still Image, Video.

Part B: Elements of information theory, error free compression: VLC, JPEG compression standards Fractal.

Text Books

1. *"Digital Image Processing", Rafael Gonzalez & Richard Woods, Pearson publications, ISBN 0132345633, 3rd Edition.*
2. *"Fundamental of Digital Image Processing", Anil K. Jain, PHI publication, ISBN 13: 9780133361650, 5th Edition.*

Reference Books

1. *"Digital Image Processing", Pratt, Wiley Publication, ISBN 0-471-37407-5, 3rd Edition.*
2. *"Digital Image Processing", K.R. Castleman, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4, 3rd Edition.*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Convert gray scale image into colour image.
2. Describe the components of image processing system.
3. Implement algorithms for digital image processing.
4. Apply lossless and lossy compression techniques for image compression.
5. Design filters for image sharpening and smoothening.
6. Explore various operations on images

CS42220 :: DATA MINING

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Database Management Systems.

List of Contents

A TERM-WORK containing the record of the following assignments:

Implementation of following data-mining tasks using a suitable Data Mining Toolkit:

1. Data Preprocessing
2. Association Mining Algorithms
3. Classification Algorithms
4. Linear Regression
5. Clustering Algorithms

Text Books

1. *"Data Mining: Concepts and Techniques"*, Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition.
2. *"Introduction to Data Mining"*, Pang-Ning Tan, Vipin Kumar, Michael Steinbach, Pearson

Reference Books

1. *"Data Mining and Analysis: Fundamental Concepts and Algorithms"*, Mohammed J. Zaki, Wagner Meira Jr., Cambridge University Press
2. *"Mastering Data Mining: The art and science of customer relationship management"*, M Berry and G. Linoff, John Wiley, ISBN 9971-51-369-2, 2001 Edition.
3. *"Data Mining : Theory and Practice"*, Soman K P, Diwakar Shyam, Ajay V, New Delhi, Prentice Hall Of India, ISBN 81-203-2897-3, 2006 Edition.

CS42228 :: MACHINE LEARNING

Credits: 01

Teaching Scheme: -Tutorial: 1 Hr/Week

List of Tutorials

1. Implementation of learning algorithms like Find S algorithm, Version space and the candidate elimination algorithm, list then eliminate algorithm for simple real world problems.
2. Implementation of learning algorithms like Backpropagation algorithm, Support Vector Machines for real time problems.
3. Implementation of algorithms like Evaluating hypothesis accuracy, Sampling theory, Central limit theorem, hypothesis testing, for real time problems.
4. Implementation of learning algorithms like Bayesian Learning for real time problems.
5. Implementation of learning algorithms like weighted majority algorithm, Instance-Based Learning: k-nearest neighbor learning, locally weighted regression for real time problems.
6. Implementation of learning algorithms like Genetic Algorithms for real time problems.
7. Implementation of learning algorithms like unsupervised or reinforcement learning for real time problems.
8. Implementation of Fuzzy logic for real time problems.
9. Presentations on selected research topics.
10. A mini project for a real problem.

Text Books

- 1) *T. Mitchell, "Machine Learning", McGraw-Hill, 1997.*
- 2) *Randy L. Haupt and Sue Ellen Haupt, Practical Genetic Algorithms.*
- 3) *David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning.*

Reference Books

- 1). *Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004.*
- 2). *Jacek M. Zurada, "Introduction to Artificial neural System", JAICO publishing house, 2002,.*
- 3). *Research papers suggested by the faculty.*

CS42204 :: NEURAL NETWORKS

Credits: 01

Teaching Scheme: -Tutorial: 1 Hr/Week

List of Tutorials :

Minimum eight experiments based on the following topics.

1. Supervised Learning rules for a single neuron
2. Unsupervised Learning rules.
3. Simple Perceptron classifier
4. Feedforward neural network.
5. Backpropagation algorithm
6. MAXNET
7. Hamming Distance Classifier
8. Hopfield network
9. SOM
10. SVM
11. Verification of logic gates using NN algorithms
12. Feature extraction for a given real world problem of importance.
13. Design of Fuzzy sets for a given problem statement.
14. Two short assignments based on research papers suggested by staff.

Text Books

1. Jacek M. Zurada, "Introduction to Artificial neural System", JAICO publishing house, 2002,.
2. Haykin, S., *Neural Networks - A Comprehensive Foundation (2nd Edition)*. Macmillan, 1999.
3. Prof. Dr. S. T. Patil, *Neural Networks and applications*, Nirali Prakashan, Pune.

Reference Books

1. Christopher M. Bishop, "Neural Networks for Pattern Recognition", Indian Edition, oxford, 1996
2. B. Yegnarayana, *Artificial Neural Networks*, PHI, 2001.
3. Timothy J. Rosss, "Fuzzy logic with engineering applications", Wiley, 2004.
4. Bart Kosko, "Neural networks and fuzzy systems", prentice hall of India, 1994.
5. John Yen and Reza Langari, "Fuzzy logic, intelligence, control and information", Pearson education.
7. Some basic and modern research papers suggested by staff in charge.

CS42211: Software Testing and Quality Assurance

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites: Software Engineering

List of Contents

A TERM-WORK containing the record of the following:

1. To Prepare Test Plan for the given problem. The Test plan consists of following issues.
 - a. Purpose of the test.
 - b. Location and schedule of the test.
 - c. Test descriptions.
 - d. Pass and Fail Criteria.
2. To identify and narrate Test cases, Test scripts/procedures.
3. To perform Unit testing especially indicating the traced Independent data paths, Control paths and Error handling paths. Prepare control flow graphs for the unit under test and compute the Cyclomatic Complexity of the unit.
4. To perform Data Flow testing for the Program Segments by identifying the Definition-Use chain and type of data flow anomaly.
5. To perform Mutation Analysis of the Program Segments along with mutant history, mutation score and type of mutation by using any Code analysis Tool (JUNIT).
6. To perform Black-Box Testing for all the units contained in the architectural segments using Equivalence Partitioning, Boundary Value Analysis and Orthogonal Array testing methods.
7. To prepare short summary of exploratory Testing, Regression Testing, GUI Testing, Web Based Testing, Performance Testing, Load Testing, Security Testing and Stress Testing.
8. To study difference between Automation Testing and Manual Testing

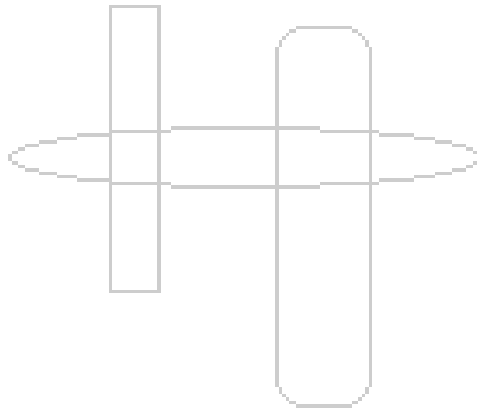
Text Books:

1. *Fenton, Pfleeger, "Software Metrics: A Rigorous and practical Approach", Thomson Brooks/Cole, ISBN 981-240-385-X.*
2. *Desikan, Ramesh, "Software Testing: principles and Practices", Pearson Education, ISBN 81-7758-121-X.*

Reference Books :

1. *Burnstein, "Practical Software Testing", Springer International Edition, ISBN 81-8128-089-X*
2. *William E. Perry, "Effective Methods for Software Testing", John Wiley and Sons, ISBN 9971-51-345-5*
3. *Yogesh Singh, "Software Testing", Cambridge University Press, ISBN 978-1-107-01296-7*

4. *Nina Godbole, "Software Quality Assurance: Principles And Practice", Alpha Science International, Ltd (August 1, 2004) ISBN-10: 1842651765 ISBN-13: 978-1842651766*
5. *Ronald Radice, "Software Inspections", Tata McGraw Hill, ISBN 0-07-048340-X*
6. *Capers Jones, "Software Assessments, Benchmarks, and Best Practices", Burlington, Vermont Publisher: Addison-Wesley Professional 2000 ISBN-10: 0201485427 ,ISBN-13: 9780201485424*



CS42224:: ALGORITHMIC NUMBER THEORY AND ALGEBRA

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Basic introduction to linear algebra and abstract algebra (though it is not presumed) is helpful to better appreciate the contents of the course. First course on algorithms.

List of Contents

A TERM-WORK containing the record of the following:

1. Problem solving based on topic like prime numbers, gcd of integers, some simple number theoretic questions
2. Modular arithmetic, Chinese remaindering applications
3. Multiplicative functions, analytical estimates of some number theoretic functions
4. Problem solving (based on topics Groups, subgroups, homomorphism, cyclic groups etc)
5. Ring, ideals, some examples of commutative rings
6. Vector spaces
7. Integral domains, finite fields
8. More on finite fields, automorphisms, Frobenius maps, some algorithmic questions..
9. Ring of univariate polynomials $F[x]$, quotient ring $F[x]/(f)$
10. Berlekamp's algorithm and more
11. Integer lattices
12. Carmichael numbers, Lagrange symbol, Jacobi symbols, Primality testing.

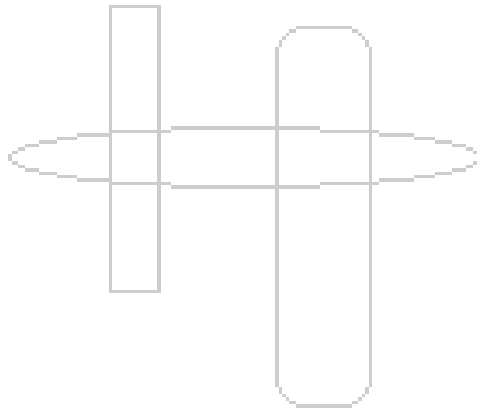
Text Books

1. *Modern Computer Algebra* by Joachim von zur Gathen, Jürgen Gerhard (Cambridge)
2. *A computational introduction to Number Theory and Algebra* by Victor Shoup (Cambridge)

3. *A classical introduction to modern number theory by Ireland and Rosen (Springer)*

Reference Books

1. *Topics in Algebra by I. N. Herstein (Wiley Publishing company)*
2. *Algebra by Michael Artin (Pearson Prentice Hall)*



CS42218::GEOGRAPHICAL INFORMATION SYSTEMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites:

List of Practical

A TERM-WORK containing the record of the following:

1. Prepare a map for the selected geographical area as per topological survey.
2. Design a spatial database for the entities related with a geographical area specified in above map.
3. Study the connectivity of maps with spatial databases.
4. Analyze a case study of any GIS application of your choice.
5. Prepare a presentation on any latest GIS technology / technique / software /hardware.

Text Books

1. *"Remote Sensing and Geographical Information Systems"*, M. Anji Reddy B S Publications, Second Edition, 2006.
2. *"Introduction to Geographic information systems"*, Kang-Tsung Chang, Tata-McGraw-Hill Publications, Third edition, 2006. ISBN: 978-0073101712.

Reference Books

1. *"Principles of Geographical Information Systems"*, Peter A Burroughs and McDonnell, Oxford University Press, 1998. ISBN-978-0198233657.
2. *"The GIS Book"*, George B Korte, Onward press (Thomson Learning), 5th Edition, 2001. ISBN 81-7800-112-8.

CS42229: :MANAGEMENT INFORMATION SYSTEMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

List of Contents

A TERM-WORK containing the record of the following:

1. Consider any organization from any sector. Study its organizational structure and comment about it.
2. By giving examples, differentiate between operational, strategic and tactical level of management process and its effect on design information system.
3. Identify and evaluate the design considerations for the given information system.
4. Study testing and quality assurance strategies.
5. Identify cases of computer crime, hacking, and cyber theft with respect to given information system. Plan about how to avoid and deal with such kind of security threats.

Text Books

1. *"Management Information Systems: Managing Information Technology in the Business Enterprise"*, James O'Brien, Tata McGraw-Hill Publishing Company Limited, 10th Edition, ISBN 0-07-058739-6.
2. *"Management Information Systems"*, Jawadekar Waman S, Tata McGraw-Hill Publishing Company Limited, ISBN 0-07-044575-3, 2nd Edition.

Reference Books

1. *"Management Information Systems: Managing the Digital Firm"*, Kenneth C. Laudon, and Jane P. Laudon, Prentice-Hall of India, ISBN 81-203--2908-2, 9th Edition.
2. *"Management Information Systems"*, Oz Effy, Singapore, Thomson, ISBN 81-315-0174-4, 5th Edition.

Additional Reading

1. *"Management Information Systems"*, Shajahan S, Priyadharshini R, New Age International, ISBN 81-224-1549-0.
2. *"Management Information Systems"*, Arora Ashok, Bhatia Akshaya, Excel Pub, ISBN 81-7446-188-4.
3. *"Information Systems A management Perspective"*, Alter Steven, Addison Wilsey, ISBN 0-201-35109-9, 3rd Edition.
4. *"Information Systems For Modern Management"*, Murdick R G, Ross J E, Claggett J R, Prentice Hall Of India, ISBN 81-203-0397-0, 3rd Edition.

FF No. : 654 C

CS42221 :: ADVANCED COMPUTER ARCHITECTURE

Credits: 01

Teaching Scheme: - Tut 1 Hrs/Week

Prerequisites: Computer Organization

List of Contents:

1. Introduction to Explicitly Parallel Instruction Computing (EPIC) Architecture.
2. To study numerical based on Performance Metrics and Measures, Speedup Performance Laws.
3. To study implementation issues of a program on any pipelined processor and their analysis.
4. To study numerical based on Pipelining.
5. To study implementation issues of Matrix multiplication and sorting on array processor and their analysis.
6. To study implementation issues of a program on multiprocessor system.
7. To study implementation issues of a multithreaded program.
8. To study Architecture of Multithreaded processors

Text Books

1. Kai Hwang, Faye A. Briggs, "Computer Architecture and Parallel Processing" McGraw-Hill international Edition.
2. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill.

Reference Books

1. Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
2. William Stallings, "Computer Organization and Architecture, Designing for performance" Prentice Hall, Sixth edition.
3. Kai Hwang, Scalable Parallel Computing.
4. Harrold Stone, High performance computer Architecture.
5. Richard Y. Kain, Advanced Computer Architecture
6. <http://www.intel.com/products/processor> (for Intel Itanium Processor).

CS42216 :: CONVERGENCE TECHNOLOGIES

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: :

List of Contents

A TERM-WORK containing the record of the following:

Assignments :

1. Study of audio and video data.
2. Signal passing through network for voice data.
3. Study of E/T cables.
4. Study of gatekeepers in convergence technology.
5. Study of ISDN devices.
6. Study of SIP implementation.
7. Study of SS7 implementation.
8. Comparison of ISDN and B-ISDN.
9. Use of frame relay in convergence.
10. Examples on convergence

Text Books

1. *“ISDN and Broadband ISDN with frame relay and ATM”* by William Stallings, Pearson Education, 2003, ISBN 81-7808-422-8, 4th Edition.
2. *“Voice over IP Technologies”* by Mark A. Miller, P.E., Wiley Publications, 2002, ISBN 81-265-0286-X, 1st Edition.

Reference Books

1. *“Computer Communications and Networking Technologies”*, by Michael A. Gallo, William M. Hancock, Cengage Learning, 2002, ISBN 81-315-0364-X, 1st Edition.
2. *“ATM network concepts and protocols”*, by Sumeet Kasera and Pankaj Sethi, Tata McGraw Hill, 2001, ISBN 0-07-463776-2, 1st Edition.

Additional Reading

1. *“VOIP”*, by Ulyess Black, Pearson Education, 2001, ISBN 0130652040, 2nd Edition.
2. *“Multimedia communication system techniques standards and networks”*, by K.R. Rao, Zoran Bojkovic and Dragorad Milovanovic, Pearson Education, 2002, ISBN 0-13031398-X, 1st Edition.

CS42233:: EMBEDDED SYSTEMS

Credits: 01

Teaching Scheme: -Tutorial 1 Hr/Week

Prerequisites: MI MPMC.

List of Contents

A TERM-WORK containing the record of the following:

1. Interface the LCD to Microcontroller
2. Understanding Different Interrupts and Programming The Interrupts
3. Programming the ADC in Microcontroller
4. Program the USART in Microcomputer.
5. Program the I2C Interface
6. Program the SPI interface.
7. Study different kinds of RESET.
8. Program the Timers for Creating the Square Wave.
9. Switching TRIACS and SCR.
10. Mini Project Design and Implementation

Text Books

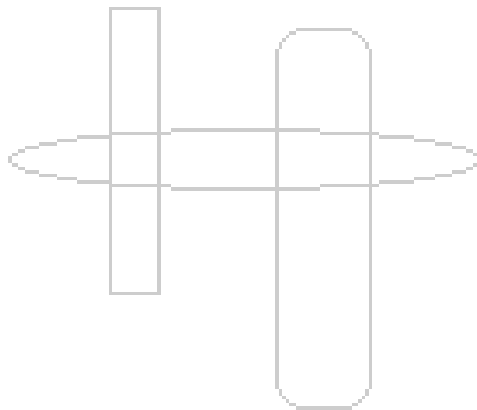
1. Raj Kamal 'Embedded Systems ', Tata McGraw-Hill . ISBN0-07-049470-3
2. Dr. K. V. K. K. Prasad "Embedded/Real time System : Concepts, Design, & Programming ". Dreamtech Press

Reference Books

1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system " Wiley -Dreamtech India Pvt. Ltd.
2. Sriram Iyer , Pankaj Gupta, "Embedded Real time Systems Programming", Tata Mc-Graw Hill, 2004.

Additional Reading

1. *Microcontroller Handbook*
2. *CAN Specification Version 2.0 Protocol Standard.*
3. *USB Specification Version 2.0 Protocol Standard.*
4. *I2C Specification Protocol Standard.*
6. Tammy Noergaard, — *Embedded Systems Architecture* by Elsevier



CS42230 :: IMAGE PROCESSING

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites:

List of Practical

All the assignments should be done using 'MATLAB'.

1. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.
2. Study of statistical properties- mean, standard deviation, profile, variance and Histogram Plotting.
3. Histogram equalization & modification.
4. Gray level transformations such as contrast stretching, negative, power law transformation etc.
5. Spatial Domain filtering- smoothing & sharpening filters.
6. Edge detection using Sobel, Prewitt and Roberts operators.
7. Pseudo coloring.
8. Creating noisy image and filtering using MATLAB.
9. DFT/IDFT of given image.
10. Singular Value Decomposition of given Image.

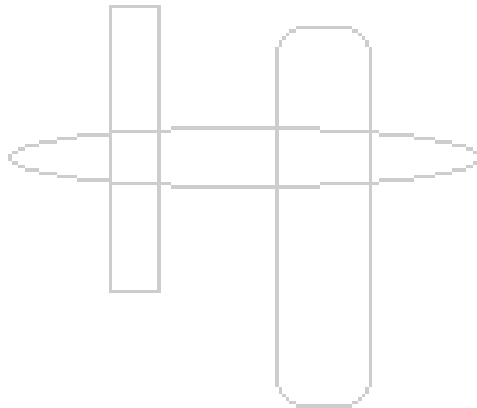
Text Books

1. *"Digital Image Processing", Rafael Gonzalez & Richard Woods, Pearson publications,*

- ISBN 0132345633, 3rd Edition.*
2. *"Fundamental of Digital Image Processing", Anil K. Jain, PHI publication, ISBN 13: 9780133361650, 5th Edition.*

Reference Books

1. *"Digital Image Processing", William Pratt, Wiley Publication, ISBN 0-471-37407-5, 3rd Edition.*
2. *"Introduction to Matlab 7 for Engineering", William J.Palm, McGraw Hill Publication, ISBN 0072922427, 2nd Edition.*



CS40306:: COMPILER DESIGN

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Knowledge of C.

List of Practical

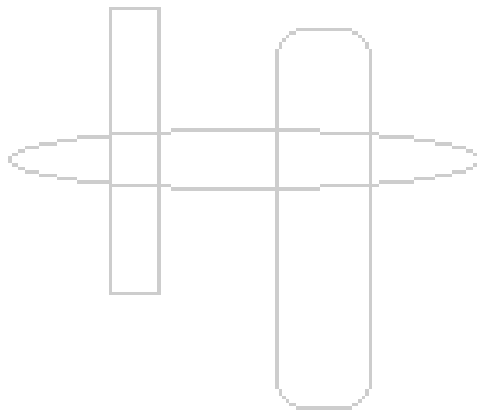
1. Assignment to understand basic syntax of LEX specifications, built-in functions and Variables.
2. Implement a Lexical Analyzer using LEX for a subset of C.
3. Implement a parser for an expression grammar using YACC and LEX .
4. Generate and populate appropriate Symbol Table.
5. Implementation of Semantic Analysis Operations (like type checking, verification of function parameters, variable declarations and coercions) possibly using an Attributed Translation Grammar.
6. Implement the front end of a compiler that generates the three address code for a simple language.
7. Generate an appropriate Target Code from the given intermediate code assuming suitable processor details.
8. A Register Allocation algorithm that translates the given code into one with a fixed number of registers. (Optional)
9. Implementation of Instruction Scheduling Algorithm. (Optional)
10. Implement Local and Global Code Optimizations such as Common Sub-expression Elimination, Copy Propagation, Dead-Code Elimination, Loop and Basic-Block Optimizations. (Optional)

Text Books

1. *“Lex & Yacc”, J. R. Levine, T. Mason, D. Brown, “Lex & Yacc”, J. R. Levine, T Mason, D. Brown, O’Reilly, ISBN 1-56592-000-7, Second Edition, 1992.*
2. *Compilers: Principles, Techniques and Tools”, A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Addison Wesley, ISBN 978-81317-2101-8, Second Edition, 2007.*

Reference Books

1. *“Engineering a Compiler”, K. Cooper, L. Torczon, Morgan Kaufmann, ISBN 1-55860-698-X, First Edition, 2003.*
2. *“Advanced Compiler Design and Implementation”, S. S. Muchnik, Morgan Kaufmann, ISBN 8178672413, First Edition, 1997.*



CS40310:: ARTIFICIAL INTELLIGENCE

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Structure

List of Practical

1. Implement Non-AI and AI Techniques
2. Implement any one Technique from the following
 - a. Best First Search & A* algorithm
 - b. AO* algorithm
 - c. Hill Climbing
3. Implement Constraint Satisfaction Algorithm
4. Expert System in Prolog
5. Implement any two Player game.
Simulate Blocks world problem using goal stack planning

Text Books

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill
2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.

Reference Books

1. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Patterson: "Introduction to AI and Expert Systems", PHI
4. Nilsson : "Principles of Artificial Intelligence", Morgan Kaufmann.
5. Carl Townsend, "Introduction to turbo Prolog", Paperback, 1987

CS47308:: PROJECT STAGE 3**Credits: 06****Teaching Scheme: - Practical 12 Hrs/Week****Aim**

This course addresses the issues associated with the successful management of a software development project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career. The Project Work can lead to:

- a. Transform existing systems into conceptual models.
- b. Transform conceptual models into determinable models.
- c. Use determinable models to obtain system specifications.
- d. Select optimum specifications and create physical models.
- e. Apply the results from physical models to create real target systems.

Overview of the Course:

1. The Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents : (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

Sr.	Project Item
1	Project Cover Front Page
2	Project Completion Certificate [Institute]
3	Project Completion Letter [In case of Sponsored Projects]
4	Acknowledgments
5	Table of Contents
6	List of Figures
7	List of Tables
8	Project Synopsis [Problem Background, Existing System Details, Proposed Solution]
9	Feasibility Study Report
10	Project Plan

11	System Requirement Specification
12	System Analysis Document: UML Use Case Diagrams
13	System Analysis Document: UML Sequence Diagrams
14	System Analysis Document: UML State Diagrams
15	System Design Document with Module Specifications
16	System Implementation
17	System Testing and Experimental Findings
18	Conclusion
19	References

2. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
3. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

Note:

The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

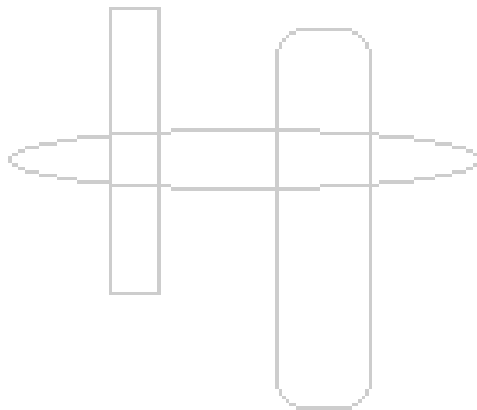
Assessment Scheme

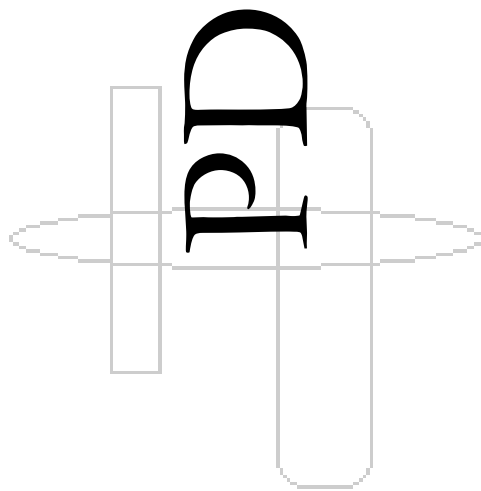
Sr. No.	Content	Marks
1	System Requirement Specification	05
2	Feasibility Study	05
3	System Analysis	10
4	System Design	10
5	System Implementation	25
6	System Testing	25
7	Presentation of the Project Work	20

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Create solution using state of Art Software and Hardware Tools.
2. Determine Impact of Design on Betterment of Society.
3. Guide a Team to Adhere to rigorous Standards laid down by Professional Engineering Bodies
4. Demonstrate the design to various kinds of Stake holders and Develop a Financial Feasibility Report
5. Demonstrate the design Enhancements for future Work
6. Maintain Highest Ethics and Technical Completeness Regarding delivery of Solution





CS33303:: ADVANCED JAVA

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Core Java.

List of Practical

1. Design a java application to demonstrate the use Java revision, anonymous inner classes, file handling, GUI, event handling, debugging using IDE.
2. Design a java application to demonstrate use of Multithreading, concurrency, synchronous and asynchronous callbacks, ThreadPools using ExecutorService.
3. Design a java application to demonstrate use of Collections and generics.
4. Design a java database application using multithreading and concurrency control.
5. Design a java application to demonstrate use of Servlets and JSP.
6. Design a client-server application demonstrating the use of Java I/O using sockets with GUI for configurations.
7. Design a java RMI application.
8. Designing a java application to demonstrate use of Web Services - REST and SOAP.
9. Design a java application to demonstrate dynamic invocation using reflection.

Reference Books

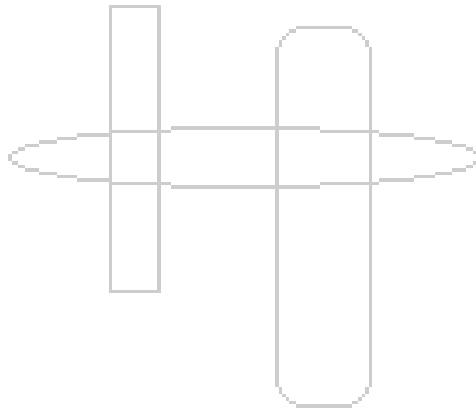
1. *"Java: The Complete Reference"*, , Herbert Schildt, Mc Graw Hill Publication, Seventh Edition, ISBN: 007063677X, 2006.
2. *"Java generics and collections"*, Thomas Powell, O'Reilly Media, ISBN: 0596527756, 2006.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Analyze the nature of a problem to select appropriate advanced feature of Java towards achieving at a problematic solution
2. Develop Java based solution for real world problem.

3. Employ Integrated Development Environment(IDE) for implementing and testing of software solution.
4. Work in well formed teams with proper skill sets to achieve effective solutions.
5. Extend their knowledge in utilizing the appropriate advanced features of Java for using emerging frameworks
6. Construct software solutions by evaluating alternate architectural patterns.



CS33312:: PIC Microcontroller

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Microprocessor, x86 , x86 Interfacing Chips.

List of Practical

1. Assignment on Program Compilation and Burning into Microcontroller.
2. Assignment on Input Output.
3. Assignment on Interrupt.
4. Assignment on LED.
5. Assignment on Timer.
6. Assignment on LCD.
7. Assignment on UART.
8. Assignment on Write and Read from EEPROM.
9. Assignment on ADC.
10. Assignment on PWM.
11. Assignment on Stepper Motor.

Text Books

1. *Data Sheet* www.microchip.com
2. *Hitachi Data Sheet on LCD HD 44780*

Reference Books

1. *Microchip 18F45xx*

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Understand Steps in System Design using Computing Devices
2. Utilize Advanced Features of Advanced Peripherals
3. Use advanced Tools for Program Simulation and implementations
4. Develop Effective Engineering Solutions to minimise Energy Utilisation
5. Cooperate with Diverse Teams to create Solutions
6. Design Effective Automation Solutions using Advanced Microprocessors

FF No. : 654 B

CS33313:: MOBILE APPLICATION DEVELOPMENT

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Java.

List of Practical

1. Download, Install and Configure Eclipse IDE with Android Development Tools (ADT) plug-ins and Android SDK or Android Studio or Net Beans with Android plugin.
2. Building Simple User Interface using UI Widgets such as Buttons, Text Fields and View.
3. Design an android based application using content provider.
4. Develop an android based application to implement the sequential and random file operation.
5. Develop an android based application to create simple embedded database for the student attendance and find defaulters in the class using SQLite.
6. Design an android based application to demonstrate GPS services using Google map.
7. Design an android based application to implement HTTP operations for internet communication.
8. Design an android based application to implement chat application using socket programming.
9. Design an android based application to take a snapshot by using the Camera in your mobile. Save the snapshot in the image or video format. Use Camera Media API provided Android.
10. Mini Project.

Text Books

1. *"Head First Android Development"*, Jonathan Simon, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN: 978-1-449-39330-4, 2011

2. *“Beginning Android™ Application Development”, Published by Wiley Publishing, Inc.10475 Crosspoint BoulevardIndianapolis, IN 46256.ISBN: 978-1-118-01711-1, 2011*

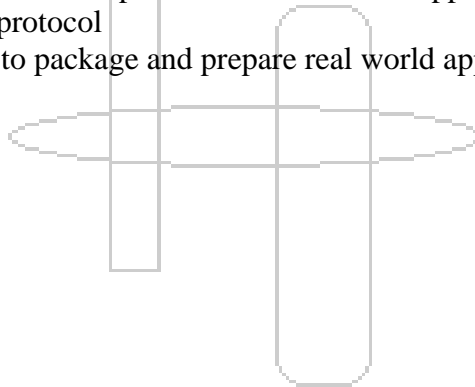
Reference Books

1. *“Professional Android™ Application Development”, Published by Wiley Publishing, Inc.10475 Crosspoint BoulevardIndianapolis, IN 46256, ISBN: 978-0-470-34471-2, 2009*
2. *“Pro Android 4”, Published byApress,Satya Komatineni, Dave MacLean, ISBN 978-1-4302-3930-7,2012*

Course Outcomes:

Upon completion of the course, graduates will be able to

1. Students will be able to simplify the data manipulation using Content Providers, Shared Preferences , embedded database SQLite, Flat files and Multi Media files
2. Students will be able to design UI-rich apps using all the major UI components
3. Students will be able to choose suitable software tools, IDE and APIs for the development of Mobile Application
4. Students will be able to trace and identify the location of specific/ specialized handheld or mobile devices using google map and other alternative techniques
5. Students will be able to develop android social media applications using HTTP and Socket communication protocol
6. Students will be able to package and prepare real world apps for deploying on mobile device



CS33306:: ETHICAL HACKING AND NETWORK DEFENSE

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Fundamentals of IT, Networking, Microsoft OS, LINUX or UNIX operating systems.

List of Practical

1. Study of different type of attacks
2. Study of Ethical hacking, types of hacking, different phases involved in hacking.
3. Study of skills to become ethical hacker.
4. Study of spoofing techniques
5. Study of password cracking techniques
6. Study of MITM and NetBIOS DOS attack.
7. Study of spyware technology
8. Study of types of viruses, antivirus techniques and virus detection mechanism
9. Study of Sniffing techniques and tools.
10. Study of Flooding attacks like MAC flooding, SYN flooding etc.
11. Study of Session Hijacking and prevention of session hijacking.
12. Web based password cracking techniques
13. Study of Wireless Hacking, WPA Authentication Mechanisms and Cracking Techniques, Wireless Sniffers and Locating SSIDS, MAC spoofing, Wireless hacking Techniques
14. Study of Physical security.
15. Penetration Testing Steps Pen-Test Legal Framework, Automated Penetration Testing Tools

Text Books

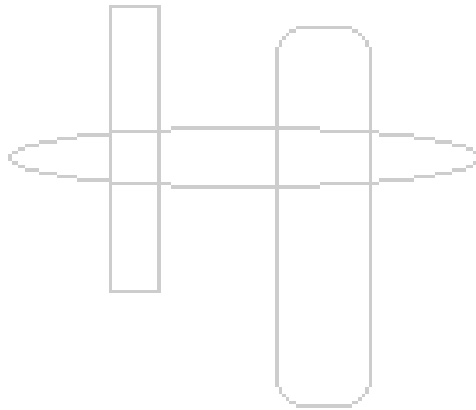
Michael T Simpson – “Ethical Hacking and Network Defense”.

Course Outcomes

Upon completion of the course, the students will be able to:

1. Establish type of attack on a given system.
2. Analyze nature and type of attack.
3. Simulate different types of attacks using tools.

4. Differentiate between the type of communication services used for attack.
5. Design a secure system for protection from the various attacks by determining the need of security from various departments of an organization.
6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.



CS33310: SPRING FRAMEWORK

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Java, JSP, Servlets

List of Practical

1. Assignment on Spring Environment Setup.
2. Assignment on Spring Hello World Example.
3. Assignment on Spring IOC Container.
4. Assignment on Spring Bean Scopes, Spring Bean Life Cycle.
5. Assignment on Spring Bean Post Processors.
6. Assignment on Spring Dependency Injection, Spring Injecting Inner Beans, Spring Injecting Collection, Spring Beans Auto-Wiring.
7. Assignment on Spring Annotation Based Configuration, Spring Java Based Configuration.
8. Assignment on Event Handling in Spring.
9. Assignment on Spring AOP Assignments.
10. Assignment on Spring JDBC assignments.
11. Assignment on Spring Web-MVC Assignments.

Text Books

1. *Spring Recipes – A problem solution approach* by Gary Mak, Josh Long and Daniel Rubio.
2. *Professional Java Development with the Spring Framework*, by Rod Johnson
3. *Spring Live* by Matt Raible.

Reference Books

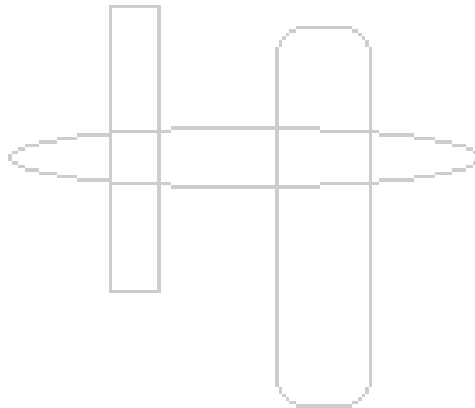
1. *Pro Spring 3.0* by Clarence Ho, Rob Harrop.
2. *Expert Spring MVC and Web Flow* by Seth Ladd, Darren Davison, Steven Devijver, Colin Yates

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Analyze Real world problems using Spring Framework Architecture, MVC model, Aspect Oriented Programming (AOP) and Event Handling in Web

- Architecture.
2. Construct formalized design patterns to effectively implement Java Enterprise Application lifecycle.
 3. Create application using Spring Tool Suite, Software project management and comprehension **tool like** Maven.
 4. Demonstrate that the business rules and validations are implemented in shorter time using this framework.
 5. Acquire skills to work on real time projects in industry.
 6. Use pre-built framework for rapid application development using Spring Framework MVC Applications.



CS33314: STRUTS FRAMEWORK

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Java, JSP, Servlets

List of Practical

1. Building a Simple Struts Application
2. Struts validator framework
3. Setup validator framework in Struts
4. Struts validator Framework
5. Using the validator framework in struts
6. Validator framework work in Struts
7. Sing validator framework work in struts
8. Using the validator Framework
9. Fixed Value check using struts validator framework
10. Struts 2 double validator
11. Struts 2 Date validator
12. Client Side Address Validation in Struts
13. Struts 2 RequiredString validator
14. Struts 2 E-mail Validator
15. XML files used in Validator Framework?
16. struts - Framework
17. Struts 2 Validation (Int Validator)
18. Struts 2 Url Validator
19. Validation using validator-rules.xml – Struts

Text Books

1. *"Jakarta Strus Live"* by Rick Hightower published by SourceBeat.
2. *Struts: The Complete Reference* by James Holmes.

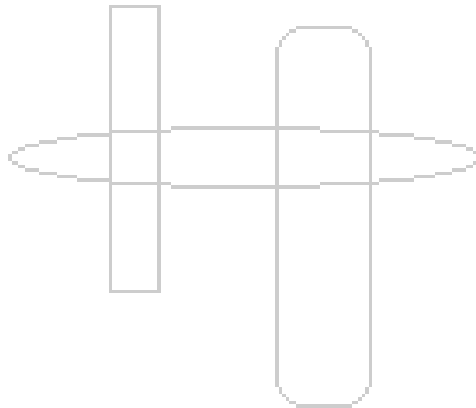
Reference Books

1. *The Struts Framework: Practical Guide for Java Programmers* by Sue Spielman.
2. *Struts 2 Black Book, 2nd Edition*, Kogent Solutions Inc.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Narrate Struts Framework Architecture and validation framework.
2. Select Action Mappings, Forms, JSP Standard Tag Library, Internationalization And Localization, Input Validation and Advanced Configuration
3. Construct datadriven web applications with Struts using Eclipse.
4. Demonstrate business rules and implement validations in shorter time using framework
5. Acquire skills to work on real time projects in industry
6. Incorporate best practices for building applications with Struts.



FF No. : 654 B

CS33311:: PROBLEM SOLVING AND PROGRAMMING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data structures.

List of Practical

7. Data structure review (stack, queue, linked list).
8. Graph searching techniques (DFS, BFS, IDDFS etc.) and applications of graph searching in problems in programming competition..
9. Advanced data structures union-find (including optimized algorithms like path compression), segment trees, interval trees, augmented data structures and their applications.
10. String searching algorithms.
11. Dancing links to speed up backtracking
12. Network flow.

Text Books

1. *"Fundamentals of Data Structures in C"*, E. Horwitz , S. Sahani, Anderson-Freed, Universities Press, Second Edition, 2008, ISBN 978-81-7371-605-8.
2. *"Introduction to Algorithms"*, T. Cormen, R. Rivest, C. Stein, C. Leiserson, PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.
3. Jon Kleinberg, Eva Tardos *"Algorithm Design"*, Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6

Reference Books

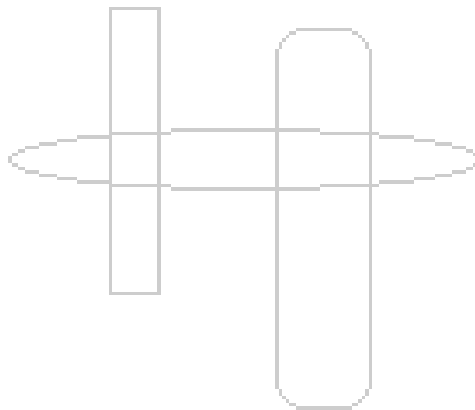
1. *"Advanced Data structures"*, Peter Brass, Cambridge Publication, 1st Edition, 2008 , ISBN – 978-0-521-88037-4.
2. *"Data Structures and algorithms with Object Oriented design Patterns in Java"*, Bruno R. Preiss, Wiley Publication, 1st Edition, 2000. ISBN: 978-0-471-34613-5.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Apply and practice logical ability to solve the problems.

2. Modularize the problems into small modules and then convert them into algorithms
3. Analyze algorithms and determine their time complexity.
4. Trace and code recursive programs.
5. Choose appropriate problem solving technique
6. Verify and validate the correctness of the algorithm.



CS33315:: BIG DATA TECHNOLOGIES

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Programming Skills

List of Practical

1. Study of Hadoop 1 / Hadoop 2 (YARN)
2. Study of hadoop distributed file system (HDFS)
3. Manipulation of data on HDFS
4. Learning Map Reduce Programming
5. Word count problem using Map Reduce Programming
6. Hands-on over Pig
7. Hands-on over Hive
8. Introduction to Hbas

Text Books

1. *"Hadoop: The Definitive Guide", 4th Edition, Tom White, O'Reilly*
2. *"Programming Pig", Allen Gates, O'Reilly*

Reference Books

1. *"Programming Hive", Dean Wampler, O'Reilly*
2. *"HBase: The Definitive Guide", Lars George, O'Reilly*

Additional Reading

1. *"Hadoop In Action", Chuck Lam, Manning Publication*
2. *"Hadoop In Practice", Alex Holmes, Second Edition, Manning Publication*

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Develop solutions to address data intensive problems by applying map reduce programming technique.
2. Analyse a computing problem to organize it into multiple parallel tasks
3. Apply various Hadoop technologies to process big data.
4. Illustrate the role of big data in producing sustainable solutions to address environmental and social issues.
5. Work in multidisciplinary contexts for management of complex big data problems.
6. Adapt to upcoming practices, tools and technologies to solve real world problems involving big data.

CS33307:: MATLAB

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical (Any Ten)

1. Introduction to MATLAB, MATLAB Elements & Simple Programs and debugging concepts.
2. Write a Matlab Program for functions.
3. Write a Matlab Programs by using IF Then Else, Case, Statement, for Loop, While loop.
4. Write a Matlab Program for 2-D graph.
5. Write a Matlab Program for 3-D graph.
6. Write a Matlab Program for various Image operations.
7. Write a Matlab Program for Animations.
8. Study of MATLAB debugging commands.
9. Write a Matlab Program to create GUI.
10. Write a Matlab Program to simulate a simple circuit.
11. Write a Matlab Program to create Movie.
12. Write MATLAB Program to read sound file and adjust its parameters.
13. Write MATLAB Program to read .avi file.

Text Books

1. "Getting started with Matlab 7", RudraPratap, Oxford University Press, ISBN 10 0199731241, 2nd Edition.
2. "Introduction to Matlab 7 for Engineering", William J. Palm, McGraw Hill, ISBN 0072922427, 3rd Edition.

Reference Books

1. "Digital Image Processing", Rafael C. Gonzales & Richard E. Woods, Pearson Education, ISBN -10: 013168728-X, 3rd Edition.
2. "Essentials of MATLAB programming", Stephen J Chapman, Cengage learning, ISBN-10-0-495-29568-X, 2nd Edition.

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Use MATLAB environment for development of solutions
2. Work with large vector and matrices.
3. Create 2D3D graphs in MATLAB.
4. Design a small pattern recognition system
5. Apply MATLAB for future problems and current applications
6. Design solutions to real world problems.

