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

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.

MODULE V

FF No. 653 Issue 0, Rev. 1 , Dt. 24/03/2017

T.Y. B. Tech. Structure with effect from Academic Year 2016-17

Module V

Code	Subject	Type	Teaching Scheme			Assessment Scheme				
			Theory	Project	Lab	ISA				Credits
						MSE	HA	L/P	ESE	
CS301THL	Data Base Management System (THL)	S_1	3	-	2	35	-	30	35	4
CS302THP	Web Technology (THP)	S_2	3	2	-	35	-	30	35	4
CS303THL	Software Engineering (THL)	S_3	3	-	2	35	-	30	35	4
CS304TH	Design and Analysis of Algorithms (TH)	S_4	3	-	-	35	30	-	35	3
CS312THP	Advanced Data Structure (THP)	S_5	3	2	-	35	-	30	35	4
HS351TH	Quantitative	HSS	2	-	-	35	30	-	35	2

	Aptitude I (HSS) (TH)									
CS315PD	**Advanced Java**	**PD**			**2-**			30	70	**2**
CS316PD	**Mobile App Development**	**PD**			**2**			30	70	**2**
CS321PD	**Big Data Technology**	**PD**			**2**			30	70	**2**
CS317PD	**PIC microcontroller**	**PD**			**2**			30	70	**2**
CS318PD	**Ethical hacking and Network Defense**	**PD**			**2**			30	70	**2**
CS319PD	**Grail Framework**	**PD**			**2**			30	70	**2**
CS320PD	**Cyber Security and Forensic Tools**	**PD**			**2**			30	70	**2**
CS322INT	**Industrial Training**									**2**

TOTAL	**17**	**4**	**6**					**23**

FF No. : 654

CS301THL : DATABASE MANAGEMENT SYSTEMS

Credits: 04 **Teaching Scheme: Theory 3 Hours / Week**
: Lab 2 Hours / Week

Unit 1: Introduction and Data Models (7 Hours)

Introduction: Need of Database Management System, Evolution, Data Abstraction, Data Independence, System Architecture of DBMS, Life cycle of relational database, Codd's Twelve Rules for Relational 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

Unit 2: Database Design Theory (7 Hours)

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

Unit 3: Query Languages (7 Hours)

Formal Relational Query Languages: Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus; SQL: DDL, DML, Select Queries, Join Queries, Subqueries, Date-Timestamp, String and Numerical Functions, DCL-Security and Authorization; PL/SQL: Procedure, Function, Trigger; Mapping of Relational Algebra to SQL

Unit 4: Storage and Querying (7 Hours)

Storage and File structures, Indexed Files, Single Level and Multi Level Indexes, B+ Trees; Query Processing: Steps, Algorithms for Selection, Join Operation; Query Optimization: Transformation of Relational Expressions, Heuristics in Query Optimization, Selectivity and Cost Estimates in Query Optimization

Unit 5: Transaction Management (7 Hours)

Transaction: ACID Properties, State diagram, Lock based Concurrency Control Protocols, Timestamp based Concurrency Control Protocol, Log based Recovery techniques, ARIES Recovery algorithm

Unit 6: Emerging Trends (7 Hours)

NoSQL: RDBMS vs NoSQL, BASE properties, NoSQL Categories; NewSQL; Distributed Databases, Parallel Databases, Decision support systems, Data Warehouse, Data mining, Information Retrieval

List of Practicals: (For THL, TLP courses)

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. Abraham Silberschatz, Henry F. Korth, S. Sudarshan; "Database System Concepts"; 6th Edition, McGraw-Hill Education
2. Ramez Elmasri, Shamkant B. Navathe; "Fundamentals of Database Systems"; 6th Edition ;Pearson

Reference Books:

1. Thomas M. Connolly, Carolyn E. Begg," Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition ;Pearson
2. Raghu Ramakrishnan, Johannes Gehrke; "Database Management Systems", 3rd Edition ; McGraw Hill Education

Course Outcomes:

The student will be able to –
1. Design data models as per data requirements of an organization
2. Synthesize a relational data model upto a suitable normal form
3. Develop a database system using relational queries and PL/SQL objects
4. Apply indexing techniques and query optimization strategies
5. Understand importance of concurrency control and recovery techniques
6. Adapt to emerging trends considering societal requirements

FF No. : 654

CS302THP: Web Technology

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Project Lab:2 Hours /Week

Unit 1: Web Development Process, Front End Tools (5 Hours)

Introduction to web technology, internet and www, Web site planning and design issues, HTML5: structure of html document, HTML elements: headings, paragraphs, line break, colors & fonts, links, frames, lists, tables, images and forms, , CSS, Bootstrap , XML.

Unit 2: Client Side Technologies (7 Hours)

HTML5 forms Validation, JavaScript: Overview of JavaScript, Data types, Control Structures, Arrays, Functions and Scopes, Objects in JS, DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM, JQuery: Introduction to JQuery, Loading JQuery, Selecting elements, changing styles, creating elements, appending elements, removing elements, handling events. Introduction to JSON

Unit 3: Server Side Technologies -I (9 Hours)

Server Side technology and TOMCAT, Servlet: Introduction to servlet, need and advantages ,Servlet Lifecycle, Creating and testing of sample servlet, session management. JSP: Introduction to JSP, advantages of JSP over Servlet , elements of JSP page: directives, comments, scripting elements, actions and templates, JDBC, MongoDB

Unit 4: Server Side Technologies -II (7 Hours)

PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, File Handling, Exception Handling, E-mail, MySQL with PHP, AJAX

Unit 5: Web Technology Frameworks (7 Hours)

Angular JS : Overview, MVC architecture, directives, expression, controllers, filters, tables, modules, forms, includes, views, scopes, services, dependency injection, custom directives, Internationalization, NodeJS.

Unit 6: Web Services (5 Hours)

Web Services: Overview, types of application web services, SOAP, REST, EJB, JNDI lookup, Content Management System(CMS)

List of Project areas: (THP)

1. Design and deploy web based application using front end technologies HTML5, CSS, Bootstrap and XML. Perform validation using Java script/JQuery/HTML5.
(For Example: Course Registration System, Voter System for Election, e-Shopping System, e-Governance System, On-line Trading System etc)

2. Develop dynamic web application essence as an extension to project 1 using either (JSP/Servlet, Tomcat, MySQL/MongoDB) or (PHP, Apache, MySQL/MongoDB) server side technologies.

Text Books:

1.Achyut Godbole & Atul Kahate," Web Technologies: TCP/IP to Internet Application Architectures" , McGraw Hill Education publications, Third Edition, 2016
2.Ralph Moseley & M. T. Savaliya, "Developing Web Applications", Wiley publications, Second Edition, 2014

Reference Books:

1. *Adam Bretz & Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015*
2. *GiulioZambon, " Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013*
3. *Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, Fifth Edition, 2015*
4. *Black Book, " JDBC 4.2, Servlet 3.1 & JSP 2.3", Dreamtech Press, 2016*
5. *Sandeep Panda, "Angular JS: Novice To Ninja", SPD, First Edition 2014, Indian Reprint 2015*
6. *J2EE Architecture, an illustrative gateway to enterprise solutions: concept to Application Design and deployment by B. V. Kumar, S. Sangeetha, S. V. Subrahmanya, Tata McGraw Hill Publishing Company.*

Course Outcomes:

The student will be able to –

1. Design the front end view of web pages using HTML5, CSS with Bootstrap framework
2. Perform client side web page forms validation.
3. Refine dynamic web pages with JSP, Servlet
4. Deliver realistic and extensible light weight web application using PHP.
5. Practice and utilize web framework paradigms and principles for Web development.
6. Develop reliable, efficient, scalable web services

FF No. : 654

CS303THL: Software Engineering (THL)

Credits: 03 **Teaching Scheme:- Theory 3 Hrs/Week**

Unit 1: Software Engineering Paradigms (7 Hours)

Overview of Software Engineering, Software Process Framework, Process Patterns, Personal and Team Process Models, Process Models: Code-and-Fix, Waterfall Model, Rapid Application Development, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Cleanroom Methodology, Component-Based Software Engineering, CMMI, Impact of Processes and Outcomes, Process Selection and applicability, Software Engineering Principles and Practices Component-Assembly Process Model, Best Practices in Technology Selection, Formal Methods

Unit 2: Requirement Engineering (7 Hours)

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, Requirements Verification and Validation, Requirement Maturity, Technical Reviews, Stakeholder Management

Unit 3: Introduction to Agile Methodology (7 Hours)

Introduction to Agile Project Management, Agile History and the Agile Manifesto, Agile Requirements: Team, Program, Portfolio Level, Scrum Overview, Scrum Framework, Agile Principles, Sprints Design, Time-Boxing, Kanban, and Theory of Constraints, Requirements and User Stories, Stakeholders, User Personas, and User Experiences, Product Backlog, Agile Planning, Estimation and Velocity, Product Backlog, Technical Debt

Unit 4: Agile: Scrum and Sprints (7 Hours)

Agile Roles: Product Owner, Scrum Master, Development Team, Development Team, Managers, Agile Planning: Scrum Planning Principles, Multilevel Planning, Portfolio Planning, Envisioning (Product Planning), Release Planning (Longer-Term Planning) Sprinting: Sprint Planning, Sprint Execution, Sprint Review, Sprint Retrospective, Agile Architecture and Re-architecting with Flow, Agile Methods: Lean Software Development, DSDM, Extreme Programming, TDD

Unit 5: Configuration Management Practices (7 Hours)

Source Code Management, Build Engineering, Environment Configuration, Change Control, Release Management, Deployment, Hardware Configuration, Personality and CM: A Psychologist Looks at the Workplace Management, Establishing IT Controls and Compliance, Industry Standards and Frameworks, CM Values, CM Practices, Agile

Values and CM, CM Practices for Agile, CM Standards and Frameworks to Support Agile

Unit 6: Project Management Principles (7 Hours)

Project Management Activities, Structures and Frameworks, Teamwork, Leadership, Project Planning, Project Scheduling, Risk Analysis, Introduction to Function Points, Empirical Estimation, COCOMO II model, Foundations of Software Testing: Terms, Testing Cycle, Outcomes, Principles,Unit and Acceptance testing, Software Verification and Validation, Classic Mistakes, Complex Systems, Critical Systems, Software Safety

List of Practical:

1. A real-world problem issue is required to be identified with manageable scope. The problem scenarios are required to be identified for target system to be developed. The scenarios are stated in the form of Statement-of-Work template. The SOW document shall address the vision, goals, and objectives of the project.
2. The initial requirements and feature set for the target system is required to be identified. The requirements are required to be synthesized with stakeholder participation. The project roles are assigned to the project team with clear indicator of responsibilities. The initial requirements summary document with adequate and minimal infrastructure is required to be developed using multiple iterations.
3. A concise requirement specification document is required to be prepared using Agile Requirements Practices with the help of user stories narration, user personas and collaborative communication. The Agile tools like Face-to-face communications, Daily standups, and Customer Idea Management shall be practiced.
4. The product backlog for the project aimed at maintaining a prioritized queue of project requirements shall be created.
   - It should be dynamic and should be continuously groomed as the project progresses. Agile projects generally use an iceberg strategy for grooming the product backlog.
   - The items that are near the top of the iceberg and are closest to going into development should get the most attention.
   - There should typically be about two to three sprints worth of stories at the top of the backlog that are well-groomed and ready to go into development in order to avoid a situation where the project team is waiting for work to do.
5. The feasibility of the project shall be prepared and stated in the form of Project Feasibility Study document mentioning finalized requirement set and dropped feature list along with requirement prioritization and traceability matrix.
6. The project plan of the project shall be prepared using Agile Planning Practices indicating level of uncertainty, technology considerations, and related risk nomenclature.

7. Sprint-level planning activity accommodating story points, planning poker shall be performed. The Sprint-plan and Sprint-design indicating detailed activity planner shall be developed.
8. The Software Configuration Management Plan (SCMP) shall be prepared to establish and maintain the integrity of the products of the software project throughout the project's software life cycle.The SCM practices identifying specific configuration items/units are contained in the key process areas that describe the development and maintenance of each configuration item/unit.
   - Software configuration management activities are planned.
   - Selected software work products are identified, controlled, and available.
   - Changes to identified software work products are controlled.
   - Affected groups and individuals are informed of the status and content of software baselines.
9. Working software shall be developed by performing Sprint Execution. The software artifacts created shall be verified and validated using unit/module testing.
10. A Sprint Review document shall be prepared using the Summarize, Demonstrate, Discuss, and Adapt approaches indicatingSprint Review Issues and Sign-offs.

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. *SorenLauesen, Software requirements: Styles and techniques, Addison Wesley, ISBN 0201745704, 2002*
2. *Kenneth S. Rubin, Essential SCRUM: A Practical Guide To The Most Popular Agile Process, Addison-Wesley, ISBN-13: 978-0-13-704329-3, 2012*
3. *Dean Leffingwell, Agile Software Requirements, Addison-Wesley, ISBN-13: 978-0-321-63584-6, 2011*
4. *Charles G. Cobb, The Project Manager's Guide To Mastering Agile: Principles and Practices for an Adaptive Approach, Wiley Publications, ISBN: 978-1-118-99104-6 (paperback), ISBN 978-1-118-99177-0 (epdf), 2015*
5. *Bob Aiello and Leslie Sachs, Configuration Management Best Practices, Addison Wesley, ISBN-13: 978-0-321-68586-5, 2010*
6. *Mario E. Moreira, Adapting Configuration Management for Agile Teams, Wiley Publications, ISBN: 9780470746639, 2010*

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 focusing tailored processes that best fit the technical and market demands of a modern software project.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
3. Formulate system specifications by analyzing User-level tasksand compose software artifacts using agile principles, practices and Scrum framework
4. Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
5. Conform to Configuration Management principles and demonstrate cohesive teamwork skills avoiding classic mistakes and emphasizing on software safety adhering to relevant standards.
6. Analyze the target system properties and recommend solution alternatives by practicing project planning, scheduling, estimation and risk management activities.

FF No. : 654

CS304TH: Design and Analysis of Algorithms

Credits: 03 **Teaching Scheme: 3 Hours / Week**

Unit 1: **(9 Hours)**

Basic introduction, time complexity analysis, Divide and Conquer

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. Adversary lower bounds (for comparison based sorting, for finding second minima). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications. Proving correctness of algorithms.

Divide and Conquer: Analyzing Quick sort, Randomized Quick sort, Merge sort, Counting Inversions, Finding majority element, Finding Median, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation), Finding closest pair of points in plane, computing convex hull of points in plane, basic idea of FFT algorithm and applications.

Unit 2: **(7 Hours)**

Dynamic Programming

General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients, Matrix Chain multiplication, Optimal binary search tree (OBST) construction, Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, All pair shortest path algorithm, Longest increasing subsequence problem, Longest common subsequence problem, Largest independent set for trees.

Unit 3: **(4 Hours)**

Greedy

Analysis and correctness proof of minimum spanning tree and shortest path algorithms, Huffman coding, conflict free scheduling, fractional knapsack.

Unit 4: **(6 Hours)**

Backtracking Strategy, Linear Programming

Backtracking: General strategy, n-queen problem, graph coloring, subset sum problem.
Linear Programming: Introduction to linear programming, geometric interpretation, LP duality, Simplex algorithm, Linear optimization problems and their LP formulation.

Unit 5: **(6 Hours)**

Flows and Matchings

Flows: Flows in the network, Max-flow min-cut theorem, Ford Fulkerson's algorithm, LP formulation of flow problem, Applications (e.g. image segmentation, airline scheduling)
Matchings: Perfect matchings in bipartite graphs, LP formulation, Hall's marriage theorem, Konig's theorem, augmenting path algorithm for matchings.

Unit 6: (8 Hours)

Introduction to NP-completeness, Approximation Algorithms

Complexity classes P, NP, coNP, and their interrelation, 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 (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem.)

Introduction to Approximation algorithms, NP-optimization problems, Approximation algorithm for Vertex Cover, Traveling Sales Person Problem(TSP), Set-cover.

Text Books:

1. Cormen, Leiserson, Rivest and Stein "Introduction to Algorithm" ,PHI 3nd edition, 2009. ISBN 81-203-2141-3

2. Jon Kleinberg, Eva Tardos "Algorithm Design", Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6

Reference Books:

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

2. Horowitz, Sahani, "Fundamentals of computer Algorithms", Galgotia. 2nd Edition, 1998.ISBN 81-7515-257-5

Course Outcomes:

The student 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. Formulate computational problem as linear program and apply LP, network flow, based techniques to design efficient algorithms for them.
5. Establish NPcompleteness of some decision problems, grasp the significance of the notion of NPcompleteness and its relation with intractability of the decision problems and design efficient approximation algorithms for standard NP-optimization problems.
6. Incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solution for a complex computing problems.

FF No. : 654

CS312THP: Advanced Data Structures

Credits: 4

Teaching Scheme:
Theory: 3 Hours / Week
Project Based Lab: 2 Hours/Week

Unit 1: Advanced Trees and Applications (7 Hours)
Red-Black Trees, van Emde Boas tree, Fusion tree, Dynamic Finger Search Trees, B Trees and B+ Trees, Splay trees.

Unit 2: Priority Queues and Applications (8 Hours)
Amortized Analysis, Double Ended Priority queues, Leftist Trees, Binomial Heaps, Fibonacci Heaps, skew heaps, pairing heaps.

Unit 3: Data Structures for Strings (5 Hours)
String Searching: preliminaries, the DAWG, the position Heaps, tries and compressed tries, Suffix Trees and suffix arrays , Dictionaries Allowing Errors in Queries.

Unit 4: Randomized Data Structures (5 Hours)
Preliminaries of randomized algorithm and probability theory, Skip Lists: Structural Properties of Skip Lists, Space Complexity of skip list. Treap: A Randomized Binary Search Tree.

Unit 5: Spatial Data Structures (8 Hours)
Multidimensional Spatial Data Structures: introduction, point data, region data, Rectangle data. Quad trees and Octrees: Quad trees for point data, spatial queries with region quad tree. Interval trees, Segment trees, Range trees, and Priority Search Trees. Binary Space Partitioning Trees, R-trees.

Unit 6: Miscellaneous Data Structures (7 Hours)
Google's Big Table, Data Structures for Sets: The Disjoint Set Union-Find Problem, Concurrent Data structures, Succinct Representation of Data Structures: Bit vector, Succinct Dictionaries, Tree Representations. Persistent data structures. Cache-Oblivious Data Structures

List of Project areas:

1. Job Scheduling
2. String processing
3. Dictionary and Search engines
4. Modeling the real world problems using graphs and trees
5. Applications of B trees and B+ trees in Database management system
6. GIS
7. Image processing.

8. Internet routing
9. Computational biology
10. Computational geometry.
11. Data Mining.

Text Books: ***(As per IEEE format)***

1. Sartaj Sahni, Dinesh P. Mehta*;* Handbook of Data Structures and Applications*;* 2nd edition*,* Chapman & Hall/CRC.
2. Fundamentals of Data Structures in C", E. Horwitz, S. Sahani, Anderson-Freed, Second Edition, Universities Press.

Reference Books: ***(As per IEEE format)***

1. T. Cormen, R.Rivest, C. Stein, C. Leiserson, "Introduction to Algorithms", Second Edition, PHI publication.
2. Peter Brass, Advanced Data Structures, First Edition, Cambridge University Press

Course Outcomes:

The student will be able to –

1. Model the real world problem with the help of appropriate tree data structure.
2. Analyze the amortized time complexity by applying suitable priority queue data structure.
3. Comprehend and select the storage pattern for strings processing application.
4. Apply randomized data structures for real world problems.
5. Design suitable Spatial data structure for the geometric problems.
6. Analyze the problem solutions based on state of the art Data structure representation.

FF No. : 654

HS351TH:Quantitative Aptitude I

Credits: 2 **Teaching Scheme:T-2 Hours / Week**

Unit 1: Numbers, Surds and Indices & Logarithms (7 Hours)
Numbers, Average, Decimal fractions, Problem on ages, Simplification, Problems on numbers, Square roots & cube roots, Logarithms, Surds and Indices, HCF and LCM of Numbers.

Unit 2: Time ,distance and work (7 Hours)
Time and distance, Problems on trains, Boats and Streams, Time and Work , Pipes and Cisterns, Alligation or mixture

Unit 3: Measures of Statistical Data (7 Hours)
Percentage, Profit and loss, Ratio and Proportion, Simple interest, Compound interest, Partnership, Chain Rule.

Unit 4: Logical Reasoning (7Hours)
Race and Games , Odd Man Out and Series, Number Series, Analogies, Logical Problems, Letter and Symbol Series, Statement and Conclusion, Artificial Language

Text Books
1. Quantitative Aptitude For Competitive Examinations", Dr. R. S. Aggarwal, S. Chand.
2. "How to Prepare for Quantitative Aptitude", Arun Sharma, Tata Mcgraw-Hill.

Reference Books
1. Quantitative Aptitude Quantum Cat Common Admission Test", K. Sarvesh Verma., Arihant.
2. "Quantitative Aptitude for Competitive Examinations", Abhijit Guha, Fourth Quarter.

Course Outcomes
The students will be able to:
1 improve their employability skills
2 improve aptitude, problem solving skills and reasoning ability
3 critically evaluate various real life situations by resorting to analysis of key issues and factors.
4 demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions

FF No.: 654

CS315PD Advanced Java

Credits:2 **Teaching Scheme: 2 Hours / Week**

List of Practical's:

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 application to demonstrate use of JSP/Servlet using database.
5. Design a client-server application demonstrating the use of Java I/O using sockets with GUI for configurations.
6. Design an Email Application using Java Mail API.
7. Design a java RMI application
8. Design a java application to demonstrate dynamic invocation using reflection
9. Develop a java application using hibernate
10. Designing a java application to demonstrate use of Web Services - REST and SOAP
11. Mini project

Text Books:

1. Advanced Java 2: development for enterprise applications", Clifford J. Berg, Sun ISBN: 0130848751, Microsystems Press, 2000.
2. "Advanced Java networking", Dick Steflik, PrashantSridharan, ISBN: 0130844667, Prentice Hall PTR, 2000.

Reference Books:

1."Java: The Complete Reference", , Herbert Schildt, McGraw Hill Publication, Seventh Edition, ISBN: 007063677X, 2006.
2. "Java generics and collections", Thomas Powell, O'Reilly Media, ISBN: 0596527756, 2006
3. "Java EE 7" for Beginners, Sharanam shah, Vaishali shah, SPD, ISBN: 13:978-93-5110-349-

Course Outcomes:

The student will be able to –
1. Analyze the nature of a problem to select appropriate advanced featureof 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.

FF No.: 654

CS316PD : Mobile App Development

Credits:2 **Teaching Scheme: 2 Hours / Week**

List of Practical's:

1. Download Install and Configure Android Studio on Windows/ Linux environment.
2. Building Simple User Interface using UI Widgets, Layouts and Adapters use Material Design Pattern.
3. Design an android based application using content provider.
4. Develop an android based application to implement file operations and Shared Preference.
5. Develop an android based application using SQLite/remote database.
6. Develop an application having animation on views.
7. Design an android based application to demonstrate GPS services using Google map.
8. Design an android based application to implement HTTP operations for internet communication.
9. Design an android based application to implement chat application using socket programming.
10. 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.
11. 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 Boulevard Indianapolis, IN 46256. ISBN: 978-1-118-01711-1, 2011*

Reference Books:

1. *"Professional Android™ Application Development", Published by Wiley Publishing, Inc.10475 Crosspoint Boulevard Indianapolis, IN 46256, ISBN: 978-0-470-34471-2, 2009*
2. *"Pro Android 4", Published by Apress,Satya Komatineni, Dave MacLean,ISBN 978-1-4302-3930-7, 2012*

Course Outcomes:

The student will be able to –

1. Simplify the data manipulation using Content Providers, Shared Preferences, embedded database SQLite, Flat files and Multi Media files
2. Design UI-rich apps using all the major UI components
3. Choose suitable software tools, IDE and APIs for the development of Mobile Application
4. Trace and identify the location of specific/ specialized handheld or mobile devices using Google map and other alternative techniques
5. Develop android social media applications using HTTP and Socket communication protocol
6. Package and prepare real world apps for deploying on mobile device

FF No. : 654

CS321PD: Big Data Technology (Hadoop)

Credits: 01 **Teaching Scheme: 02 Hours / Week**

List of Practicals:

1. Study of Hadoop 1 and YARN
2. Study of hadoop distributed file system (HDFS) and its commands
3. Introduction to Map Reduce Programming
4. Introduction to SPARK Programming using Scala
5. Text preprocessing / Result Analysis using Apache Pig
6. Data management through complex queries using Apache Hive

Text Books:

1. "Hadoop: The Definitive Guide", 4th Edition, Tom White, O'Reilly
2. "Programming Pig", Allen Gates, O'Reilly
3. Machine Learning in ActionBook by Peter Harrington

Reference Books:

1. "Programming Hive", Dean Wampler, O'Reilly
2. "HBase: The Definitive Guide",Lars George, O'Reilly

Course Outcomes:

The student will be able to –

1. Illustrate architecture of Hadoop and YARN
2. Break down a computing problem into multiple parallel tasks
3. Explain Hadoop Ecosystem
4. Organise input data to handle it using HDFS
5. Apply map reduce programming technique to address real world problems
6. Adapt to upcoming technologies for management of complex big data problems

FF No. : 654

CS317PD: PIC Microcontroller

Credits:01 **Teaching Scheme: 2 Hours / Week**

List of Practicals:

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
3. Ajay Deshmukh, "Microcontroller and its Applications," 1st Edition, Tata McGraw Hill Publication, ISBN 0-07-058595-4.

Reference Books:

1.Microchip 18F45xx

Course Outcomes:

The student 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 minimize Energy Utilization.

5.Cooperate with Diverse Teams to create Solutions.

6.Design Effective Automation Solutions using Advanced Microcontroller.

FF No. : 654

CS318PD: ETHICAL HACKING AND NETWORK DEFENSE

Credits: 01 **Teaching Scheme: 2 Hours / Week**

List of Practical's:

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:

1. *Michael Simpson, Nicholas Antill, Kent Backman, James Corley; Hands-On Ethical Hacking and Network Defense; 3rd Revised edition,Cengage Learning, Inc;ISBN-10: 1285454618 ISBN-13: 978-1285454610*

Course Outcomes:

The student 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.

FF No. : 654

CS319PD: Grails Framework

Credits:1 **Teaching Scheme: 2 Hours / Week**

List of Laboratory Assignments

Concepts:

1. Getting Started with Grails
2. Understanding Domain Classes
3. Understanding Controllers
4. Understanding Views
5. Mapping URLs
6. GORM
7. Services
8. Integration and Dependency Management

Students will develop web application project in lab in groups. Above concepts should be covered in the project.

Text Books:

1. Bert Beckwith, Programming Grails, First Edition, Orelly Publisher
2. Jason Rudolf, Getting started with Grails, First Edition, InfoQ Publisher

Reference Books:

1. Glen Smith, Grails in Action, Second Edition No., Dreamtech Publisher
2. Shingler, Jim, Beginning groovy and Grails, First Edition, Apress publication

Course Outcomes:

The student will be able to –
1. Develop web applications using grails.
2. Deploy web application in real time.

FF No. : 654

CS320PD: CYBER SECURITY AND FORENSIC TOOLS

Credits: 02 **Teaching Scheme: -** Lab 2 Hrs/Week

Prerequisites: Data structure.

List of Practicals:

1. Study of cyber forensic tools and Techniques
2. Case studies for cyber forensics live demo
3. Study of misuse and abuse of Multimedia
4. Case studies of Audio forensic using tools
5. Case studies of Video forensic using tools
6. Case studies of photographic image forensic using tools
7. Demonstration of speaker identification
8. Demonstration of Personal Identification
9. Study of Cyber crime using social media
10. Study of FTK imager tool (trial version)

Text Books

1. *Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.*
2. *Kevin Mandia, Chris Prosise, Matt Pepe, "Incident Response and Computer Forensics ", Tata McGraw -Hill, New Delhi, 2006.*

Reference Books

1. *Robert M Slade," Software Forensics", Tata McGraw - Hill, New Delhi, 2005.*
2. *Bernadette H Schell, Clemens Martin, "Cybercrime", ABC – CLIO Inc, California, 2004.*
3. *"Understanding Forensics in IT ", NIIT Ltd, 2005.*

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. Analyze cyber attacks
2. Identify the original and modified data/file
3. Use different forensic tools
4. Illustrate the digital forensic involved in the attack
5. Investigate cyber crime involved in the multimedia
6. Understand the cyber laws.

MODULE VI

FF No. 653 Issue 0, Rev. 1 , Dt. 24/03/2017

T.Y. B. Tech. Structure with effect from Academic Year 2016-17

Module VI

Code	Subject	Type	Teaching Scheme			Assessment Scheme				
			Theory	Project	Lab	ISA				Credits
						MSE	HA	L/P	ESE	
CS306THP	Operating System (THP)	S_1	3	2	-	35	-	30	35	4
CS313THL	Microprocessor and Microcontroller(THL)	S_2	3	2	-	35	-	30	35	4
CS308THP	System Programming (THP)	S_3	3	-	2	35	-	30	35	4
CS309THL	Computer Networks (THL)	S_4	3	-	2	35	-	30	35	4
CS314TH	Theory of Computation (TH)	S_5	3	-	-	35	30	-	35	3
HS352TH	Quantitative Aptitude II	HSS	2	-	-	35	30	-	35	2

CS311PRJ	**Mini Project**	**P**			**4**					**2**
		TOTAL	**17**	**4**	**8**					**23**

FF No. : 654

CS306THP: OPERATING SYSTEMS

Credits: 04 **Teaching Scheme: Theory 3 Hours / Week**
: Lab 2 Hours / Week

Unit 1: Introduction to OS (7 Hours)

Introduction to OS: What is OS, Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of system calls.
Types of OS: Batch, Multiprogramming, Time sharing, Parallel, Distributed & Real-time OS.
Structures of OS: Monolithic, Layered, Virtualization-Virtual Machines, Microkernels.
Introduction to Mobile OS: Architecture & Overview of Android OS.

Unit 2: Process Management (8 Hours)

Shell: Linux commands, OS shell, Shell programming.
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 and Monitors.
Classical Problems of Synchronization: Readers-Writers problem, Producer Consumer problem, Dining Philosopher problem

Unit 3: Process Scheduling (6 Hours)

Uniprocessor Scheduling: Scheduling Criteria, Types of Scheduling: Preemptive, Non-preemptive, Long-term, Medium-term, Short-term.
Scheduling Algorithms: FCFS, SJF, RR, Priority.
Multiprocessor Scheduling: Granularity, Design Issues, Process Scheduling.
Thread Scheduling, Real Time Scheduling

Unit 4: Deadlocks (5 Hours)

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

Unit 5: Memory Management (7 Hours)

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

Virtual Memory: Concepts, Swapping, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, Page Size, VM with Segmentation,VM with combined paging and segmentation.
Page Replacement Policies: FIFO, LRU, Optimal, Clock.
Swapping issues: Thrashing

Unit 6: I/O and File Management (7 Hours)
I/O management: I/O Devices - Types, Characteristics of 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.
Case study: Windows 7: Design Principles, Process Management, Scheduling, Memory Management, I/O Management and File Management.

List of Project areas:
1. Linux based application using Shell Scripting and POSIX threads.
2. Design and implementation of a Multiprogramming Operating System: Stage I
    i. CPU/ Machine Simulation
    ii. Supervisor Call through interrupt
3. Design and implementation of a Multiprogramming Operating System: Stage II
    i. Paging
    ii. Error Handling
    iii. Interrupt Generation and Servicing
    iv. Process Data Structure

Text Books:
1. Stalling William; "Operating Systems", 6th Edition, Pearson Education.
2. Silberschatz A., Galvin P., Gagne G.; "Operating System Concepts", 9th Edition, John Wiley and Sons.

Reference Books:
1. Silberschatz A., Galvin P., Gagne G ; "Operating System Principles"; 7th Edition, John Wiley and Sons.
2. Yashavant Kanetkar; "Unix Shell Programming", 2nd Edition, BPB Publications.
3. Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming", 1st Edition, Australia Thomson Brooks Cole.
4. Achyut S. Godbole , Atul Kahate; "Operating Systems", 3rd Edition, McGraw Hill.

Course Outcomes:

The student will be able to –

1. Examine the functions of a contemporary Operating system with respect to convenience, efficiency and the ability to evolve.
2. Demonstrate knowledge in applying system software and tools available in modern operating system (such as threads, system calls, semaphores, etc.) for software development.
3. Apply various CPU scheduling algorithms to construct solutions to real world problems.
4. Identify the mechanisms to deal with Deadlock.
5. Understand the organization of memory and memory management hardware.
6. Analyze I/O and file management techniques for better utilization of secondary memory.

FF No. : 654

CS313THL:: Microprocessors and Microcontroller

Credits: 04 **Teaching Scheme: Theory 3 Hours / Week**
: Lab 2 Hours / Week

Unit 1: Introduction to Pentium microprocessor (8 Hours)

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.

Unit 2: BUS cycles and Memory organization (6 Hours)

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.

Unit 3: Microcontroller (7 Hours)

Processor - Architecture, Programmer's model, Modes of operation, Interrupt Structure and Applications**.** Management of Power Supply, Clocking Unit, Real Time Clock Reset Circuitry and Watchdog Timer. Processor and Memory Selection, Memory Map, Memories and I/O.

Unit 4: Microcontroller I/O interfacing (6 Hours)

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, Design of Delay Routine using Hardware timers

Unit 5: Protected Mode Architecture of Pentium (7 Hours)

Introduction, segmentation, support registers, related instructions, descriptors, memory management through segmentation, logical to linear address translations, related instructions, , transfer control, Paging-support registers, related data structures ,linear to physical address translation ,TLB

Unit 6: Protection & Multitasking (6 Hours)

Protection by segmentation, privilege-level protection, Inter-privilege level, Page level protection.
Gate Descriptors, Call Gate, Task Structure, Task-State Segment, TSS Descriptor, Task Register, Task switching

List of Practicals: THL

1. Write ALP to Create file / Read File / Write into the File.
2. Write ALP to display system Date/System Time/Directory Path
3. Write ALP to simulate Type command
4. Write ALP to simulate Copy command
5. Write ALP to display mouse pointer position with respect to X and Y cordinates
6. Write ALP to Hide & display mouse pointer
7. Write a program to interface LED & Buzzer with LPC2148.
8. To display the rolling message in left entry mode & right entry mode
9. Write a program to interface Timer to generate 100khz fequncy
10. Write a program to interface ADC

Write an ALP to Capture MSW & Display and Analyse Contents of GDTR IDTR LDTR

Text Books: ***(As per IEEE format)***

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

2. *Deshmukh A., "Microcontrollers - Theory and Applications", Tata McGraw-Hill, 2004.*

Reference Books: ***(As per IEEE format)***

1. ARM data Sheet.

2. Intel data Sheet.

Course Outcomes:

The student will be able to –

1.	Study Application and Development of Microprocessor
2.	Design system interconnects for effective throughput
3.	Utilize the Structures to effectively solve Computing Problems
4.	Design Effective Automation Solutions using Advanced Microprocessors
5.	Study Operating Modes of Advanced Microprocessors
6.	Analyse advanced features of microprocessors.

FF No. : 654

CS308THP:System Programming

Credits: 4 **Teaching Scheme: 3 Hours / Week**
:Project 2 Hours/Week

Unit 1: Introduction to System Software (5 Hours)

Introduction, software types, software hierarchy, components of system software, machine structure, interfaces, address space, levels of system software, recent trends in software development.
Language processors: Programming languages and language processors, fundamentals of language processing, life cycle of a source program, language processing activities, data structures for language processing: search data structures, allocation data structures.

Unit 2: Macro Processor and Assembler (7 Hours)

Macroprocessor: Introduction, macro definition and call, macro expansion, nested macro calls, design of macro processor, design issues of macro processors, two-pass macro processors, one-pass macro processors.
Assembler: Elements of assembly language programming, design of the assembler, assembler design criteria, types of assemblers, two-pass assemblers, one-pass assemblers, assembler algorithms, multi-pass assemblers, variants of assemblers design of two pass assembler, machine dependent and machine independent assembler features.

Unit 3: Compilers, Linkers and Loaders (7 Hours)

Compilers: Introduction to compiler phases, introduction to cross compiler, features of machine dependent and independent compilers, overview of types of compilers. Interpreters: compiler vs. interpreter, phases and working.
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, MSDOS linker.

Unit 4: Systems Programming for Linux as Open Source OS (7 Hours)

Essential concepts of linux system programming, APIs and ABIs, standards, program segments/sections, the elf format, linking and loading, linux dynamic libraries (shared objects), dynamic linking, API compatibility, dynamically linked libraries.
Advanced system programming concepts: Operating system interfaces, stack smashing. Multitasking and paging, address translation, memory protection, comparison with windows.

Unit 5: Encoding, Decoding and Device drivers (7 Hours)

Encoding and decoding schemes for the X-86 processor.
Device Driver: 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.

Unit 6: TSR Programming (7 Hours)

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.

TSR: types, structure, details of TSR loading, examples, writing TSRs.

List of Project areas:

1. Design and implementation of 2 Pass Macroprocessor.
2. Design and implementation of 2 Pass Assembler.
3. Simulation of linker & loader.
4. Implement a Lexical Analyzer using LEX for a subset of C.
5. Design and implementation of DLL on Linux shared library.
6. Design a device driver on Linux system.

Text Books:

1. *D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN - 0074635794*
2. *John J Donovan; "Systems Programming"; Tata Mc-Graw Hill edition , ISBN-13 978-0-07-460482-3*

Reference Books:

1. *Robert Love, "Linux System Programming";O'Reilly, ISBN 978-0-596-00958-8*
2. *Mahesh Jadhav; "Easy Linux Device Driver"; HighTechEasy publishing, Second edition.*
3. *Ray Duncan; "Advanced MSDOS programming"; Microsoft press*

Course Outcomes:

The student will be able to –

1. Discriminate among different System software and their functionalities.
2. Design language translators like Macroprocessor and Assembler.
3. Develop approaches and methods for implementing compiler, linker and loader.
4. Adopt the skills and methods for implementing different system-level software.
5. Interpret the methods and techniques about instructions Encoding-Decoding and implementing device drivers.
6. Design TSR programs for real world applications.

FF No. : 654

CS309THL: Computer Networks

Credits: 4 **Teaching Scheme:Theory:3 Hours / Week**
Lab:2 Hours / Week

Unit 1: Physical Layer (7 Hours)

Introduction of LAN, MAN, WAN, PAN. Ad-hoc Network, Network Architectures: Client-Server; Peer To Peer; Distributed and SDN, OSI Model, TCP/IP Model, Topologies: Star and Hierarchical; Design issues for Layers, Transmission Mediums: CAT5, 5e, 6, OFC and Radio Spectrum, Network Devices: Bridge, Switch, Router, Brouter and Access Point, Manchester and Differential Manchester Encodings; IEEE802.11: Frequency Hopping (FHSS) and Direct Sequence (DSSS)

Unit 2: Logical Link Control (7 Hours)

Design Issues: Services to Network Layer, Framing, Error Control and Flow Control. Error Control: Parity Bits, Hamming Codes (11/12-bits) and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol, Network Performance, WAN Connectivity : PPP and HDLC

Unit 3: Medium Access Control (6 Hours)

Channel allocation: Static and Dynamic, Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA, IEEE 802.3 Standards and Frame Formats, CSMA/CD, Binary Exponential Back -off algorithm, Fast Ethernet, Gigabit Ethernet, IEEE 802.11a/b/g/n and IEEE 802.15 and IEEE 802.16 Standards, Frame formats, CSMA/CA.

Unit 4: Network Layer (8 Hours)

Switching techniques, IP Protocol, IPv4 and IPv6 addressing schemes, Subnetting, NAT, CIDR, ICMP, Routing Protocols: Distance Vector, Link State, Path Vector, Routing in Internet: RIP ,OSPF, BGP, Congestion control and QoS, MPLS, Mobile IP, Routing in MANET : AODV, DSR

Unit 5: Transport Layer (6 Hours)

Services, Berkley Sockets, Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, TCP, TCP Timer management, TCP Congestion Control, Real Time Transport protocol(RTP), Stream Control Transmission Protocol (SCTP), Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless.

Unit 6: Application Layer (6 Hours)

Domain Name System (DNS), Hyper Text Transfer Protocol (HTTP), Email: SMTP, MIME, POP3, Webmail, FTP, TELNET, Dynamic Host Control Protocol (DHCP), Simple Network Management Protocol (SNMP).

List of Practicals: (For THL)

Operating System recommended :- 64-bit Open source Linux or its derivative
Programming tools recommended: - Open Source C,C++, JAVA, PYTHON, Programming tool like G++/GCC, Wireshark, Etheral and Packet Tracer

1. **Lab Assignment on Unit I:**
Part A: Setup a wired LAN using Layer 2 Switch and then IP switch of minimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool.

Part B: Extend the same Assignment for Wireless using Access Point

2. **Lab Assignment on Unit II:**
Write a Program with following four options to transfer-
1. Characters separated by space
2. One Strings at a time
3. One Sentence at a time
4. file

between two RS 232D or USB ports using C/C++. (To demonstrate Framing, Flow control, Error control).

3. **Lab Assignment on Unit II:**
Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.(50% students will perform Hamming Code and others will perform CRC)

4. **Lab Assignment on Unit II:**
Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

5. **Lab Assignment on Unit V:**
Write a program using TCP socket for wired network for following
a. Say Hello to Each other (For all students)
b. File transfer (For all students)
c. Calculator (Arithmetic) (50% students)
d. Calculator (Trigonometry) (50% students)

Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

6. **Lab Assignment on Unit V:**
Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

7. Lab Assignment on Unit V:
Write a program to analyze following packet formats captured through Wireshark for wired network.
1.Ethernet 2. IP 3.TCP 4. UDP

8. Write a program to prepare TCP and UDP packets using header files and send the packets to destination machine in peer to peer mode. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

9. **Lab Assignment on Unit IV:**
Configure RIP/OSPF/BGP using Packet Tracer.

10. **Lab Assignment on Unit IV and Unit V:**
Use network simulator NS2 to implement:
a. Monitoring traffic for the given topology
b. Analysis of CSMA and Ethernet protocols
c. Network Routing: Shortest path routing, AODV.
d. Analysis of congestion control (TCP and UDP).

11. **Lab Assignment on Unit V:**
Write a program using TCP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat

Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

12. **Lab Assignment on Unit V:**
Write a program using UDP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat

Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

Text Books:

1. *Andrew S. Tenenbaum, "Computer Networks",5th Edition, PHI, ISBN 81-203-2175-8.*
2. *Fourauzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006*

Reference Books: (As per IEEE format)

1. *Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204*
2. *Matthew S. Gast "802.11 Wireless Networks", O'Reilly publications; 2nd Edition.*

3. *C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols" Prentice Hall, 2004*
4. *Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", Wiley, ISBN: 0-470-09510-5*

Course Outcomes:

The student will be able to –

1. Select network architecture, topology and essential components to design computer networks.

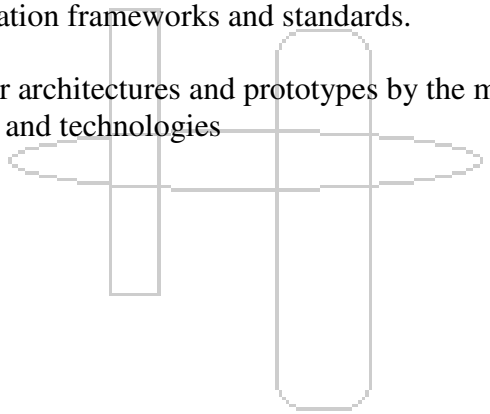
2. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency.

3. Design mechanisms to demonstrate server channel allocation in wired and wireless computer networks

4. Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

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, protocols and technologies



FF No. : 654

CS314TH: Theory of Computation

Credits: 03 **Teaching Scheme: 3 Hours / Week**

Unit 1: **(7 Hours)**

Finite Automata

Introduction to Automata, Computability and Complexity theory, Automaton as a model of computation, Central Concepts of Automata Theory: Alphabets, Strings, Languages. Decision Problems Vs Languages. Finite Automata, Structural Representations, Deterministic Finite Automata(DFA)-Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness, Product construction, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Conversion of NFA with epsilon transitions to DFA, Automata with output. Applications and Limitation of Finite Automata.

Unit 2: **(7 Hours)**

Regular and Non Regular Languages

Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene's Theorem: Equivalence Regular expressions and DFAs, Closure properties of Regular Languages(union, intersection, complementation, concatenation, Kleene clousure), Decision properties of Regular Languages, Applications of Regular expressions Myhill-Nerode theorem and its applications: proving non-regularity, lower bound on number of states of DFA, State Minimization algorithm, Equivalence testing of DFAs. Non Regular Languages, Pumping Lemma for regular Languages.

Unit 3: **(7 Hours)**

Context Free Grammars (CFG)

Context Free Grammars: Definition, Examples, Derivation, Languages of CFG, Constructing CFG, correctness proof using induction. Closure properties of CFLs (Union, Concatenation, Kleene clousure, reversal). Derivation trees, Ambiguity in CFGs, Removing ambiguity, Inherent ambiguity. Simplification of CFGs, Normal forms for CFGs: CNF and GNF. Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

Unit 4: **(7 Hours)**

PDA, Non-Context Free Languages, Context Sensitive Languages

Push Down Automata: Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic, Non-deterministic PDAs, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof). Intersection of CFLs and Regular language.

Pumping lemma for CFLs, non-Context Free Languages, Context Sensitive Languages, Definition and Examples of Context Sensitive Grammars, Linear Bounded Automata. Chomsky hierarchy.

Unit 5: (6 Hours)

Introduction to Turing Machines

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, Universal Turing Machines. TM as enumerator. Recursive and Recursively Enumerable languages and their *closure properties*.

Unit 6: (6 Hours)

Introduction to Undecidability

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 via Cantor's diagonalization. Undecidability of Halting problem. Examples of undecidable problems: Post Correspondence Problem, Hilbert's 10^{th} Problem, Tiling problem (without proof). Example of Turing unrecognizable language. Decision properties of R, RE languages and Rice's theorem.

Text Books: (As per IEEE format)

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

2. Michael Sipser, Course Technology, "Introduction to Theory of Computation", Third Edition, ISBN-10: 053494728X.

Reference Books: (As per IEEE format)

1. J. Martin, "Introduction to Languages and the Theory of Computation", Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003.

Course Outcomes:

The student will be able to –

1. infer the applicability of various automata theoretic models for recognizing formal languages.
2. discriminate the expressive powers of various automata theoretic and formal language theoretic computational models.
3. illustrate significance of non determinism pertaining to expressive powers of various automata theoretic models.
4. comprehend general purpose powers and computability issues related to state machines and grammars.
5. explain the relevance of Church-Turing thesis, and the computational equivalence of Turing machine model with the general purpose computers.
6. grasp the theoretical limit of computation (independent of software orhardware used) via the concept of undecidability

FF No. : 654

HS352TH:: QUANTITATIVE APTITUDE II

Credits: 2 **Teaching Scheme:T-2 Hours / Week**

Unit 1: Area, Volume, Permutation and Combinations (7 Hours)
Area, Volume and Surface Areas, Calendar, Clocks, Permutations and Combinations, Probability, Heights and Distances.

Unit 2: Data Interpretation (7 Hours)
Tabulations: Tabulations of Imports and Exports of Data, Analysis of Tabulated Data, Bar Graphs: Vertical or Horizontal Bars, Pie Charts: Pie Graphs, Central angle, Line Graphs.

(7 Hours)

Unit 3: Probability
Introduction to probability, Structure of probability, Results of probability, Revision of probability: BAYES' RULE, and examples; Random variable and probability distribution: Discrete and Continuous distribution, Expected value and variance of a distribution.

Unit 4: Correlation & Regression Analysis (7 Hours)
Regression analysis (Linear only), Correlation analysis, Karl Pearson's correlation coefficient, Spearman's Rank correlation coefficient

Text Books
1. Quantitative Aptitude For Competitive Examinations", Dr. R. S. Aggarwal, S. Chand.
2. "How to Prepare for Quantitative Aptitude", Arun Sharma, Tata Mcgraw-Hill.
3. Probability & Statistics for Engineers- Richard Johnson – Prentice Hall of India,
4. Statistics for Management- Richard Levin , Rubin - Prentice Hall of India,

Reference Books
1 Quantitative Aptitude Quantum Cat Common Admission Test", K. Sarvesh Verma., Arihant.

2 "Quantitative Aptitude for Competitive Examinations", Abhijit Guha, Fourth Quarter.

Course Outcomes
The students will be able to:
1 improve their employability skills
2 improve aptitude, problem solving skills and reasoning ability
3 critically evaluate various real life situations by resorting to analysis of key issues and factors.
4 demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions

FF No:654

CS311PRJ::Mini Project

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	Project Synopsis	10
2	Project Feasibility	10
3	Concise System Analysis	10
4	System Design Block Diagram Architectural Design Elements Description	20
5	System Implementation	20
6	Test Cases and Result	20
7	Conclusion ,Presentation	10

Course Outcomes:

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

1. Identify Real World Problems
2. Apply Computing Solutions to Real World Problems
3. Construct a Solution Model to Real World Problem
4. Select Design Pattern to Best approach the Solution.
5. Lay Down rules to Minimise Adverse Impact of Design Implementation
6. Adapt to changing Technological and Human resource advances.

Note:

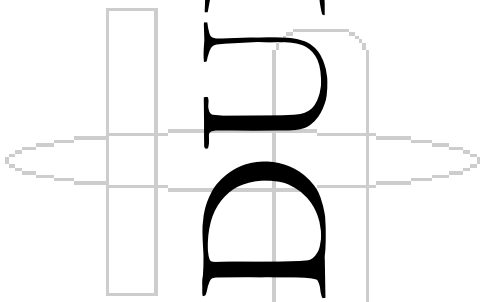
The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students.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



FF No. 653 Issue 0, Rev. 1 , Dt. 24/03/2017

B. Tech. Structure with effect from Academic Year 2016-17

Module VII

Code	Subject	Type	Teaching Scheme			Assessment Scheme				
			Theory	Project	Lab	ISA				Credits
						MSE	HA	L/P	ESE	
CS401THL	Artificial Intelligence (THL)	S_1	3		2	35	-	30	35	4
CS403THL	Compiler Design(THL)	S_2	3		2	35	-	30	35	4
CS417THP	EL-I (Embedded SystemsTHL)	S_3	3	2	2	35	-	30	35	4
CS404THP	EL-II (Data Science THP)	S_4	3	2	-	35	-	30	35	4
CS405PRJ	Project-I	P	-	10	-					5
		TOTAL	12	14	6					21

FF No. : 654 A

CS401THL: Artificial Intelligence

Credits: 04 **Teaching Scheme: - Theory: 3 Hrs/Week**

Lab:2 Hrs/Week

Prerequisites:

Unit 1: Fundamentals of Artificial Intelligence (6 Hours)

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

Unit 2: Uninformed Search Strategies (7Hours)

Formulation of real world problems, Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies, Searching with partial information, Sensorless problems, Contingency problems

Unit 3: Informed Search Strategies (8 Hours)

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence

Unit 4: Knowledge Representation (7 Hours)

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:Case study of Expert System in PROLOG

Unit 5: Planning (5 Hours)

Blocks world, STRIPS, Implementation using goal stack, Partial Order Planning, Hierarchical planning, and least commitment strategy. Conditional Planning, Continuous Planning

Unit 6: Uncertainty (7Hours)

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.

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. Implement real time applications in Prolog.
5. Expert System in Prolog
6. Implement any two Player game.
7. 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*

Course Outcomes:

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

1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents capable of problem formulation.
2. Evaluation of different uninformed search algorithms on well formulated problems along with stating valid conclusions that the evaluation supports.
3. Design and Analysis of informed search algorithms on well formulated problems.
4. Formulate and solve given problem using Propositional and First order logic.
5. Analyze the AI problem using different planning techniques
6. Apply various symbolic knowledge representations to specific multidisciplinary domains and reasoning tasks of a software agent.

FF No. : 654

CS403THL:Compiler Design

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab:2 Hours / Week

Unit 1: Lexical Analysis (4 Hours)

Introduction to Compiler, Phases and Passes, Bootstrapping, Cross Compiler, Role of a Lexical Analyzer, Specification and Recognition of Tokens, Look ahead operation, Lexical Phase errors, LEX tool.

Unit 2: Syntax and Semantic Analysis (8 Hours)

Expressing Syntax: CFG, Top-Down Parsing: Recursive Descent, Predictive Parsers.
Bottom-Up Parsing: LR Parsers: Constructing SLR parsing tables, Constructing Canonical LR parsing tables, Constructing LALR parsing tables, Using ambiguous grammars, Parser generator YACC, Symbol-Table Structure, Error Detection & Recovery strategies.

Unit 3: Syntax-Directed Translation and Intermediate Code Generation (9 Hours)

Syntax-Directed Definitions, Attribute grammar, Dependency graphs, S and L attributed grammar, Bottom-Up Evaluation, Top-Down Translation,
Intermediate Representations – Need, Forms, SDT scheme for Intermediate Code Generation of assignment statement, declaration statement.
Control flow translation of Boolean expression (Back patching), if, if-else statement, while statement, array assignment.
Error Detection & Recovery: semantic errors.

Unit 4: Code Generation (6 Hours)

Issues in Code Generation, Target Language, Basic Blocks and Flow Graphs, Next-use information, Simple Code generator, DAG representation of Basic Blocks, Register allocation and Assignment, Peephole Optimization.

Unit 5: Code Optimization (8 Hours)

Introduction: Principal Sources of Optimization, Optimization of basic Blocks (local and global) Data Flow Analysis: data flow equations, reaching definitions, live variable analysis, available expressions, Loops in flow graphs: Dominators, Back edges and Reducibility.

Unit 6: Introduction to compilation for modern architectures (5 Hours)

Automatic Parallelization, Instruction Scheduling, Software Pipelining.
Introduction to advanced topics – Just-In-Time Compilation, Dynamic compilation, Interpreters (JVM/ Dalvik), Cross compilation using XMLVM, Case studies : GCC, g++, LLVM.

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. Extend to handle variables with single letter names.
4. Generate and populate appropriate Symbol Table.
5. Write a YACC specification to check the syntax of following of 'C' language statements with error detection.
   a.) if and if…else
   b.) for
   c.) switch…. case
6. Implementation of Semantic Analysis Operations (like type checking, verification of function parameters, variable declarations and coercions) possibly using an Attributed Translation Grammar.
7. Implement the front end of a compiler that generates the three address code for a simple language.
8. Generate an appropriate Target Code from the given intermediate code assuming suitable processor details.
9. Implement Common Sub-expression Elimination and Copy Propagation optimizations for an input 3-address code.
10. Implement Dead-Code Elimination, Loop optimizations for input 3-address code.

Text Books:

1. *A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman; "Compilers: Principles, Techniques and Tools"; Addison Wesley, ISBN 978-81317-2101-8, Second Edition, 2007.*
2. *K. Cooper, L. Torczon, Morgan Kaufmann; "Engineering a Compiler", ISBN 1-55860-698-X, First Edition, 2003.*

Reference Books:

1. *S. S. Muchnik, Morgan Kaufmann; "Advanced Compiler Design and Implementation", ISBN 8178672413, First Edition, 1997.*
2. *K. Muneeswaran : " Compiler Design", ISBN 978-0-19-806664-4, Oxford University Press,2013*
3. *J. R. Levine, T. Mason, D. Brown ;"Lex & Yacc", O'Reilly, ISBN 1-56592-000-7, Second Edition, 1992.*

Course Outcomes:

The student 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 correctness of a program.
3. Deploy efficient techniques for semantic analysis to generate intermediate code.
4. Propose techniques to generate machine code which conforms to the target machine specifications.
5. Apply code optimization transformations to improve the performance of target code.
6. Acquaint themselves with the knowledge of recent trends in compilation.

FF No. : 654

CS417THP:Embedded Systems

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Project:2 Hours / Week

Unit 1: Introduction to Embedded Systems (7 Hours)

Embedded Systems**:** Components, Classification, Characteristics of ES. Review of Microprocessors and Microcontrollers. Embedded Processor Technology, Application Specific System Processor. CISC, RISC , Power Management of peripherals, Reset Ckt / Watch Dog Timers, Structural Units of Microprocessor , Processor and ,Memory Selection ,Interfacing of Processor Memory, I/O Bus management techniques. Serial / Parallel Bus , Bus Design Issues ArbitrationStudy of Bus Parameter

Unit 2: Microcontroller (ARM) (7 Hours)

Family Architecture, Features, Data / Programme Memory, Reg set, Reg Bank, Special Function Registers, Data Memory, Programme Memory, Interrupt Structure, Timer Prog, Serial Port Prog, Misc Features. Memory I/O Design & Interfacing, Timer Calculation

Unit 3: Title of the Unit PIC Microcontrollers (7 Hours)

Features, Architecture Pin Out, Capture /Compare /Pulse width modulation Mode , Block Dia Prog Model, Rest /Clocking, Mem Org, Prog/Data, Flash Eprom, Add Mode/Inst Set Prog , I/o, Interrupt , Timer, ADC Design Parameters problem solving. PWM ADC Timers

Unit 4: I/O interfacing (6 Hours)

ADC / DAC ,Switch, LED Devices , LCD, High Power Devices Relays , OP Amps : Adder, Subtracter, Gain , Low Pass filter, High Pass filter, Integrator, Differentiator, Current to Voltage Converter, Timing /Counting Devices. Communication using RS232C , Sensors, Magnetic, Temperature, Pressure,Design Implementation of OP Amps

Unit 5: Bus Technologies (6 Hours)

I/O Bus, Bus Bus management techniques. Serial / Parallel Bus , Bus Design Issues, Allocation , Arbitration. Study of Bus Parameter. Detail study of CAN, I2C, SPI.,USB, RS232C

Unit 6: RTOS (7 Hours)

Device drivers, Interrupts, Interrupt Service Routines, Scheduling Algo, Inter Process Communication, Process Synchronisation. Multiple Tasks, Threads, Processes. Shared Data / Priority Inversion Char: of RTOS, Real Time Scheduling of RTOS Case Study of Embedded Systems in Detail.(H/W + S/W Algo)

List of Project areas: (For THP, TLP courses)

1. An Embedded System Project which will use all principles stated in 6 Units.

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:

Reference Books

1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system "Wiley -Dreamtech India Pvt. Ltd.
2. Sriram lyer , Pankaj Gupta,"Embedded Real time Systems Programming", Tata Mc-Graw Hill, 2004.

Additional Reading

1. Microcontroller Handbook

Course Outcomes:

The student will be able to –

1. Learn the Concept of Embedded Systems
2. Learn Fundamentals of Microcontrollers
3. Learn Microcontrollers and IO Interfacing
4. Design Systems using principles of ES.
5. Learn the fundamentals of RTOS.

CO –PO Mapping

CO1	Understand Steps in System Design using Computing Devices	PO1
CO2	Design system interconnects for effective throughput	PO5
CO3	Create designs using Simulation and RTOS Tools	PO9
CO4	Deliver Complex Automation Solutions	PO10
CO5	Cooperate with diverse Teams for delivering automation Solutions	PO16
CO6		

FF No. : 654

CS404THP: Data Science

Credits:04 **Teaching Scheme: 3 Hours / Week**
Project Lab:2 Hours /Week

Unit 1: Data Pre-processing (7 Hours)

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, sub-setting, sorting, transforming scale, determining percentiles, data manipulation, removing noise, removing inconsistencies, transformations, standardizing, normalizing - min-max normalization, zscore standardization, and rules of standardizing data.

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'sInequality, measures of shape (skewness, kurtosis), measures of association (covariance,correlation), outliers

Unit 2: Inferential Analytics (7 Hours)

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.

Unit 3: Predictive Analytics :Regression (7 Hours)

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.

Unit 4: Predictive Analytics: Supervised Method (7 Hours)

Bias-Variance Dichotomy,Model Validation Approaches,Logistic Regression Linear Discriminant Analysis,Quadratic Discriminant Analysis,Regression and Classification Trees,Support Vector Machines, Ensemble Methods: Random Forest Neural Networks,Deep learning.

Unit 5: Predictive Analytics: Unsupervised Method (6 Hours)
Similarity Measures, Design of recommender systems, user based and item based Collaborative filtering, Clustering, Associative Rule Mining

Unit 6: Prescriptive Analytics (6 Hours)
Creating data for analytics through designed experiments, creating data for analytics through Active learning, creating data for analytics through Reinforcement learning

List of Project areas: (For THP courses)

1. Implementing a simple Recommender System based on user buying pattern.
2. Data analysis case study using R for readily available data set using any one machine learning algorithm

Text Books:

1. "Business Analytics" by James R Evans, Pearson
2. Hastie, Trevor, et al.The elements of statistical learning.Vol. 2. No. 1. New York: springer, 2009.

Reference Books:

1.Montgomery, Douglas C., and George C. Runger.Applied statistics and probability for engineers. John Wiley & Sons, 2010
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."Business Intelligence for Dummies"

Course Outcomes:
The student will be able to –

1. Understand the process of converting data into a required format required for particular analysis.
2. Analyze data, test claims, and draw valid conclusions using appropriate statistical methodology.
3. Utilize statistical tools in deriving insights from data .
4. Apply analytic techniques and algorithms (including statistical and data mining approaches) to large data sets to extract meaningful insights.
5. Use appropriate resources to research, develop and contribute to advances and trends within the field of Data Science.
6. Interpret and present visually, orally and in written form, valid conclusions drawn from data analysis.

FF No. : 654

CS405PRJ: PROJECT -I

Credits: 05 **Teaching Scheme: -** Practical 8 Hrs/Week

Aim

This course addresses the issues associated with the successful management 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	Synopsis	5
2	Feasibility Study	5
3	System Requirement Specification	5
4	System Analysis Document	20
5	Use Case Diagram	10
6	Detailed Design Plan	

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.

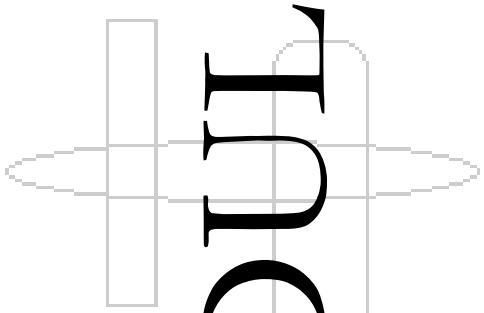
Following is the list of recommended domains for Project Work:

Course Outcomes:

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

1. Model the Real World Problem
2. Identify the Design within Specification and Available Resources
3. Realise the Solution within Defined references
4. Defend his Design with Technical and Ethical reasoning
5. Adapt to changing Technological and Human resource advances
6. Use the gained knowledge for other Real World Problems

MODULE VIII



FF No. 653 Issue 0, Rev. 1 , Dt. 24/03/2017

B. Tech. Structure with effect from Academic Year 2016-17

Module VIII

Code	Subject	Type	Teaching Scheme			Assessment Scheme					
			Theory	Project	Lab	ISA					Credits
						CT	MSE	HA	L/P	ESE	
CS406INT	Open EL-I	S_1	3	-	-	-	-	-	-	-	3
CS407INT	Internship	P	-	16	-	-	-	-	-	-	8
CS408INT	Project Based Viva	P	-	-	-	-	-	-	-	-	4
		TOTAL	3	16	-						15

FF No. : 654

CS407INT: INTERNSHIP

Credits: 08 **Teaching Scheme: -** Practical 8 Hrs/Week

Students pursuing summer internship should undergo a minimum 8 weeks training from a reputed research organization or an IT industry. Students are required to present their work upon successful completion of the internship.

Course Outcomes:
Upon completion of the course, engineering graduates will be able to

CO1	Analyse real life industry problems
CO2	Create solutions to problems with the help of latest tools
CO3	Maintain work ethics in organised sector
CO4	Cooperate with diverse teams and effectively communicate with all the stake holders
CO5	Adapt to changing work environments
CO6	Produce solutions within the technological guidelines and standards

FF No. : 654

CS412GIP: GLOBAL INTERNSHIP

Credits: 15 **Teaching Scheme: -** Hrs/Week

Students going for Global Internship have a great opportunity to work in an international atmosphere for an entire semester. Students are required to present their work upon successful completion of the internship.

Course Outcomes:
Upon completion of the course, engineering graduates will be able to

CO1	Analyse real life industry problems
CO2	Create solutions to problems with the help of latest tools
CO3	Maintain work ethic in organised sector
CO4	Cooperate with diverse teams and effectively communicate with all the stake holders
CO5	Adapt to changing work environments
CO6	Produce solutions within the technological guidelines and standards

FF No. 653 Issue 0, Rev. 1 , Dt. 24/03/2017

B. Tech. Structure with effect from Academic Year 2016-17

Module VIII

Code	Subject	Type	Teaching Scheme			Assessment Scheme				
			Theory	Project	Lab	ISA				Credits
						MSE	HA	L/P	ESE	
	Open EL-I	S_1	3	-	-	35	30	-	35	3
	Open EL-II	P	3	-	-	35	30	-	35	3
	Open EL-III		3	-	-	35	30	-	35	3
CS413PRJ	Project-II	P	-	8	-	-	-	-	-	4
CS414SEM	Seminar									2
		TOTAL	9	8	-					15

FF No. : 654

CS413PRJ:: PROJECT STAGE 2

Credits: 04 **Teaching Scheme: -** Practical 8 Hrs/Week

Aim

This course addresses the issues associated with the successful management 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:

f. Transform existing systems into conceptual models.
g. Transform conceptual models into determinable models.
h. Use determinable models to obtain system specifications.
i. Select optimum specifications and create physical models.
j. Apply the results from physical models to create real target systems.

Overview of the Course:

9. 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).
10. 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.
11. 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).
12. 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.
13. 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.
14. 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.
15. 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.

16. 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	Synopsis	5
2	Feasibility Study	5
3	System Requirement Specification	5
4	System Analysis Document	10
5	System Implementation	25
6	System Testing	25
7	Project Presentation	15

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 -

7. Model the Real World Problem
8. Identify the Design within Specification and Available Resources
9. Realise the Solution within Defined references
10. Defend his Design with Technical and Ethical reasoning
11. Adapt to changing Technological and Human resource advances
12. Use the gained knowledge for other Real World Problems

FF No. : 654

CS414SEM: Seminar

Credits: 2 **Teaching Scheme: 2 Hours / Week**

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:

The student 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.

OPEN ELECTIVE List

FF No. : 654

CS444TH:Mobile Computing

Credits:3 **Teaching Scheme:Theory 3Hours / Week**

Unit 1: Cellular Network (7 Hours)

Personal Communication System (PCS), PCS Architecture, Why cellular networks? Generations (1G,2G,3G,4G), Basic cellular system, Design Considerations: Cell, Cell Clustering, Frequency allocation, System capacity and frequency re-use, Cell splitting , Co-channel interference and its reduction factor. Types of non co-channel interference

Unit 2: GSM Communication (7 Hours)

, Signal and signal propagation, GSM System Architecture: GSM Radio subsystem, GSM Interfaces, GSM Identifiers, Logical Channels: Traffic Channels and Signaling Channels, Network and switching subsystem, Operation subsystem. GSM channels, GSM protocol architecture, Location tracking and call setup, Security, Data services N/W signaling

Unit 3: Cellular Bearer Services (7 Hours)

SMS architecture protocol, Hierarchy, Voice and Video services for mobile networks.. Data Support Services: Paging systems, CDPD GPRS, WLL, DECT, EDGE, UMTS, HSPA, HSPA+, W-CDMA, CDMA-2000, LTE, 1xRTT, EV-DO

Unit 4: Handoff in Cellular Networks (6 Hours)

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

Unit 5: Network & Transport layer in Cellular Networks (7 Hours)

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. TCP over 2.5/3G/4G wireless networks.

Unit 6: Mobile Databases (6 Hours)

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.

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*
3. *"Professional Android™ Application Development", Published by Wiley Publishing, Inc.10475 Crosspoint Boulevard Indianapolis, IN 46256, ISBN: 978-0-470-34471-2, 2009*
4. "*Pro Android 4", Published by Apress,Satya Komatineni, Dave MacLean,ISBN 978-1-4302-3930-7, 2012*

Reference Books: ***(As per IEEE format)***

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:

The student will be able to –

1. Estimate performance parameters for designing the Cellular Network which comply Next Generation Cellular Network Standards.
2. Formulate conceptual Telecommunication system to be deployed to fulfill bandwidth capacity planning
3. Design the mobile network considering futuristic busty data on cellular network.
4. Justify the Mobile Network performance parameters and design decisions while mobile Handoff.
5. Adapt to the requirements of next generation mobile network and mobile applications
6. Simplify the database usage on embedded devices for enterprise applications.

FF No. : 654

CS445TH: Software Testing and Quality Assurance(THL)

Credits: 03 **Teaching Scheme:- Theory 3 Hrs/Week**

Unit 1: Software Measurement (7 Hrs)

Measurement Theory and Why to Measure the Software, Measurement and Models, Measurement Scales, Classification of Software Measures, Measurement Framework, Theory of Program Testing, Discrete Math for Testers, Graph Theory for Testers, Software Complexity, 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, The Rayleigh Model, Exponential Distribution and Reliability Growth Models, SRE process, Reliability Concepts: Hardware and Software, Deploying SRE

Unit 2: Principles of Testing (7 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

Unit 3: Functional Testing (7 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, 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 (7 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, Reliability Testing, Efficiency Testing, Maintainability Testing, Portability Testing, Introduction to Performance Testing, The Fundamentals of Effective Application Performance Testing, The Process of Performance Testing, Interpreting Results: Effective Root-Cause Analysis, Application Technology and Its Impact on Performance, Testing VS Test Automation, Tool evaluation and selection, Automation team roles, Architectures, Planning and implementing test automation process

Unit 5: Introduction to Software Quality Assurance (7 Hrs)

The software quality challenge, Meaning of software quality, Software quality factors , Software Quality Lessons Learned from the Quality Experts, The components of the software quality assurance system – overview, Pre-project software quality components: Contract Review, Development and quality plans, SQA components in the project life cycle: Integrating quality activities in the project life cycle, Assuring the quality of software maintenance components, Assuring the quality of external participants' contributions, CASE tools and their effect on software quality, Software quality infrastructure components: Procedures and work instructions, Supporting quality devices, Staff training and certification, Corrective and preventive actions, Documentation control, Pareto Principles, Total Quality Management, Ishikawa's Seven Basic Tools

Unit 6: Software Quality Assurance Management (7 Hrs)

Management components of software quality: Project progress control, Software quality metrics, Costs of software quality, Standards, certification and assessment: Quality management standards, SQA project process standards – IEEE software engineering standards, Management and its role in software quality assurance, The SQA unit and other actors in the SQA system, Inspection as an Up-Front Quality Technique, Software Audit Methods, Software Safety and Its Relation to Software Quality Assurance, SQA for Small Projects, Development Quality Assurance, Quality Management in IT, Introduction to ITIL, Software Quality Assurance Metrics, Software Benchmarks and Baselines

Text 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. *Daniel Galin, Software Quality Assurance: From theory to implementation, Pearson Education Limited, 2004, ISBN 0201 70945 7*

Reference Books :

1. *KshirasagarNaik, PriyadarshiTripathy, Software Testing and Quality Assurance-Theory and Practice, John Wiley & Sons, Inc., 2008, ISBN 978-0-471-78911-6*
2. *Fenton, Pfleeger, "Software Metrics: A Rigourous and practical Approach", Thomson Brooks/Cole, ISBN 981-240-385-X.*
3. *Desikan, Ramesh, "Software Testing: principles and Practices", Pearson Education, ISBN 81-7758-121-X.*
4. *Anne MetteJonassen Hass, Guide to Advanced Software Testing, ARTECH HOUSE, INC., 2008, ISBN-13: 978-1-59693-285-2*

5. *Ian Molyneaux,The Art of Application Performance Testing, O'Reilly Media, Inc., 2009, ISBN: 978-0-596-52066-3*
6. *Jamie L. Mitchell, Rex Black, Advanced Software Testing—Vol. 3, 2nd Edition, Rocky Nook, 2015, ISBN: 978-1-937538-64-4*
7. *G. Gordon Schulmeyer, Handbook of Software Quality Assurance Fourth Edition, ARTECH HOUSE, INC., 2008, ISBN-13: 978-1-59693-186-2*

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 along with Software Quality best practices usage.
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 by making use of standards and baselines.

FF No. : 654

CS447TH:Modeling and Simulations

Credits:3 **Teaching Scheme: 3Hours / Week**

Unit 1: Process of Modeling and Simulation

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.

Unit 2: Formal models and modeling techniques

Monte Carlo methods, Stochastic processes, Queuing theory: Little's Theoremand applications, M/M/1 Queuing System, Petri nets, Game theory, State spaces and transitions, Graph structures: directed graphs, trees, networks.

Unit 3: Discrete Event Simulation

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.

Unit 4: Agent-based simulation

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.

Unit 5 : M&S Applications and Awareness

Application areas: optimization, decision making support, forecasting, safety considerations, training and education.
ABMS Applications: Social networks, Organizations, Markets, Flows, Epidemiology, Diffusion.

Unit 6: Advanced Topics

Model scalability, Virtual Reality, Virtual Worlds, Intro to Rare Event Simulation, Intro to Parallel Discrete Event Simulation, 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. Demonstrate the effectiveness of modeling and simulation at predicting behavior/performance/problems of systems under development.
2. Develop a model for a given problem using appropriate modeling and simulation technique/formalism.
3. Implement discrete event simulation models using general-purpose programming languages or DES frameworks
4. Design an agent-based simulation model for a complex system.
5. Contribute towards increased utilization of modeling and simulation as a problem solving approach for issues in governance and industry where it could be applied
6. Adapt to the changing needs of the organizations and individuals during the development process.

FF No. : 654

CS448TH::Data Acquisition System

Credits: 03 **Teaching Scheme:**
Theory 3Hrs/ Week

Unit 1 : Systems and Measurements (6 Hours)

Part A:Introduction to data acquisition system(DAS), Block Diagram of DAS, Multichannel DAS, Generalized measurement system, Characteristics of measuring devices, accuracy, precision, errors, linearity, hysteresis, threshold, repeatability, reliability calibration., Basic concepts of feedback control system:- open loop and closed loop.

Unit 2 : Sensors and Signal Conditioning circuits (7 Hours)

Part AIntroduction to the sensors, Types of sensors: Temperature sensor (LM35,RTD, Thermocouple), Light sensor(photodiode, optocoupler), Distance and range sensor (IR,LVDT), Accelerometer sensor, Touch screen sensor. Signal conditioning circuits: Analog and Digital, Opamp in signal conditioning circuits as amplifier.

Unit 3 : ARM Microcontroller (7 Hours)

Part A:Family Architecture, Block diagram, Features, Data / Programme Memory, Reg set, Reg Bank, Special Function Registers, Data Memory, Programme Memory, Interrupt Structure , Timer Prog ,Serial Port Prog , Misc Features.

Unit 4 : I/O interfacing & Bus Systems (6 Hours)

PartA: Introduction to the BUS System, Bus design issues, Synchronous Bus, Asynchronous Bus, Bus Allocation, Bus Priority. Interfacing Buses:I2C, SPI ,CAN , SCADA and RS 232C .

Unit 5 : Relay Sequencer and Ladder diagrams (7Hours)

Introduction to the relay sequencer. Elements of Ladder Diagrams (limit, pressure, level, thermal, mechanical switch) and examples based on ladder diagrams. Data Loggers.

Unit 6: Programmable Logic Controls (7 Hrs)

PartA:PLC : PLC block diagram and its function, PLC Operation: I/O scan mode, execution mode, PLC Software Functions: Timer, accumulated timer, counters. Advantages of PLC over Relay sequencer.

Text Books:

1. *"Process control instrumentation", C.D. Johnson, Pearson education, ISBN-81-7758-*

410-3, seventh edition.

2. *"Opamp and linear integrated circuits", Ramakant Gaikwad, PHI publication, ISBN-9780132808682, 4th edition.*

Reference Books :

1. *"Instrumentation Devices and System", Rangan, G .R. Sarma, V.S.V. Mani, Tata McGraw-Hill, ISBN-0-07-463350-3, 2nd Edition.*

2. *Microcontroller Handbook*

Course Outcomes:

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

1. Define the characteristics of measuring devices for an instrument.
2. Identify the functioning of measuring devices in an industrial process.
3. Design elements of signal conditioning circuit that are necessary for sensors.
4. Describe the structural units of Microcontroller.
5. Interconnect devices using communication buses.
6. Develop programs for the process control systems using sensors.

FF No. : 654 A

CS453TH:: CYBER SECURITY AND DIGITAL FORENSICS

Credits: 03 **Teaching Scheme: -** Theory 3 Hrs/Week

Prerequisites: Computer Networks.

Unit 1: **(7 Hrs)**

INTRODUCTION: Introduction and Overview of Cyber Crime, Nature and Scope of Cyber Crime, Types of Cyber Crime: Social Engineering, Categories of Cyber Crime, Property Cyber Crime.

Unit 2: **(8 Hrs)**

CYBER CRIME ISSUES: Unauthorized Access to Computers, Computer Intrusions, White collar Crimes, Viruses and Malicious Code, Internet Hacking and Cracking, Virus Attacks, Pornography, Software Piracy, Intellectual Property, Mail Bombs, Exploitation ,Stalking and Obscenity in Internet, Digital laws and legislation, Law Enforcement Roles and Responses**.**

Unit 3: **(7 Hrs)**

INVESTIGATION: Introduction to Cyber Crime Investigation, Investigation Tools, eDiscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

Unit 4: **(8 Hrs)**

DIGITAL FORENSICS: Introduction to Digital Forensics, Forensic Software and Hardware, Analysis and Advanced Tools, Forensic Technology and Practices, Forensic Ballistics and Photography, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Network Forensics.

Unit 5: **(6 Hrs)**

LAWS AND ACTS: Laws and Ethics, Digital Evidence Controls, Evidence Handling Procedures, Basics of Indian Evidence ACT IPC and CrPC , Electronic Communication Privacy ACT, Legal Policies.

Unit 6: **(5 Hrs)**

Crimes in Cyber Space-Global Trends & classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digital forensic, Introduction to MQTT and CoAP for IoT.

Text Books:

1. *Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.*
2. *Kevin Mandia, Chris Prosise, Matt Pepe, "Incident Response and Computer Forensics ", Tata McGraw -Hill, New Delhi, 2006.*

Reference Books:

1. *Robert M Slade," Software Forensics", Tata McGraw - Hill, New Delhi, 2005.*
2. *Bernadette H Schell, Clemens Martin, "Cybercrime", ABC – CLIO Inc, California, 2004.*
3. *"Understanding Forensics in IT ", NIIT Ltd, 2005.*

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 threads in cyber security.
2. Use tools for digital forensics.
3. Analyze data of cyber crime.
4. Investigate the problems in cyber security.
5. Understand the laws of cyber security
6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.

FF No. : 654

CS454TH:: ADVANCED COMPUTER ARCHITECTURE

Credits: 03

Teaching Scheme:
Theory: 3 Hours / Week

Unit I **(6Hours)**

Overview of Parallel Processing

Parallelism in Uniprocessor system, Evolution of parallel processors, Architectural Classification, Flynn's, Fengs, handlers classification. Pipeline, Vector, Array, Bernstein's' condition for parallelism, dependence graphs. Multicore architecture- GPU processor and programming model.

Unit II **(8Hours)**

Pipeline Architecture

Principles and implementation of Pipelining, Classification of pipelining processors, Arithmetic and Instruction pipelining, Instruction level parallelism (ILP) Pipelining hazards and resolving techniques, Data forwarding. register renaming pipeline reservation table, sequencing and collision, Branch problem, branch penalty .prediction. Delayed branching Branch target buffer Advanced pipelining techniques, loop unrolling techniques, Superscalar pipelining, speedup, in order, out of order execution, VLIW processor software scheduling, Software pipelining

Unit III **(6 Hours)**

Vector and Array Processor

Basic vector architecture, Issues in Vector Processing, Vector Instructions Register to Register and Memory to Memory Architectures Vector performance, Vector Chaining ,Vector loops and recurrence vectorizers and optimizers, Cray I Cray Y-MP
SIMD Computer Organization **:** Distributed & shared memory model Processing Element (PE) Masking and Data Routing Inter PE Communication
Interconnection networks of SIMD, Static Vs Dynamic network, Permutations, Shuffle exchange, Bit reversal hyper cube ,mesh network, Iliac, omega, blocking, non blocking networks.Compute Unified Device Architecture (CUDA) Hardware , Memory, Thread model Parallel Algorithms For Array Processors**:** Matrix Multiplication. Sorting on mesh, hypercube and CUDA Vectorization and Parallelization methods

Unit IV

Multiprocessor Architecture

UMA, NUMA, COMA MPP Processor. Loosely and Tightly coupled multiprocessors, characteristics of multiprocessors & multiprocessing, Inter Processor communication network, Time shared bus, Crossbar switch, Interleaved memories S access, C access.
Cache coherency and bus snooping and directory based protocols. Massively Parallel Processors (MPP), Inter Processor Communication and Synchronization.

Unit V **(6Hours)**
Multithreaded Architecture

Latency hiding techniques, Principles of multithreading, Issues and solutions. Multithreaded processors model, context switching policies. Blocked multithreading, Interleaved multithreading, simultaneous multithreading.
Software multithreading, thread concept, lightweight process (LWP), bounded, unbounded threads, thread management: thread scheduling & control by thread library & kernel (State transition diagram).

Unit VI **(6 Hours)**
Parallel Software Issues

Operating systems for multiprocessors systems, software issues for multiprocessors Message Passing Interface (MPI) Principles of Parallel Algorithm design: Decomposition, tasks and Dependency graphs. Parallel language constructs. Programming using message passing and shared memory paradigms (Sorting, searching, matrix problems)

Text Books

1. *Kai Hwang, "Advanced Computer Architecture," Tata McGraw-Hill ISBN 0-07-113342-9*
2. *Kai Hwang, Faye A. Briggs, "Computer Architecture and Parallel Processing," McGraw-Hill international Edition ISBN 007-066354-8.*
3. *V. Raja Raman, L Siva ram Murthy, "Parallel Computers Architecture and Programming," PHI ISBN 81-203-1621-5.*

Reference Books:

1. *DezsoSima, Fountain, Kacsuk, "Advanced Computer Architecture," Pearson Education Asia ISBN 981-4053-74-0.*
2. *Culler, Jasvinder Pal Singh , Gupta, "Parallel Computer Architecture," Morgan Kaufman (ELSVIER) ISBN 81-8147-189-X*
3. *3.Grama ,Gupta , Karypis, Vipin Kumar : "Introduction to Parallel Computing ," Pearson learning ISBN 978-81-317-0807-1*

Course Outcomes:

Students will be able to :
1. Understand the architecture of parallel processing
2. Describe the concept of Pipeline Architecture.
3. Acquaint the knowledge of Vector Architecture.

4. Explore the characteristics of multiprocessors & multiprocessing.
5. Study the basic techniques of multithreading.
6. Identify the software issues for multiprocessors.

FF No. : 654

CS455TH: Cloud Computing

Credits: 3 **Teaching Scheme:**
Theory: 3 Hours / Week

Unit 1: Introduction to Cloud Computing (6 Hours)
Introduction to Cloud Computing, Definition, Characteristics, Components, Cloud Service Models: SaaS, PaaS, IaaS, Cloud provider, benefits and limitations, Deploy application over cloud, Cloud computing vs. Cluster computing vs. Grid computing. Open Stack vs Cloud Stack, Role of Open Standards.

Unit 2: Resource Virtualization (6 Hours)
Infrastructure as a Service (IaaS): Virtualization Technology: Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM). Virtualization: Server, Storage, Network. Virtual Machine (resource) provisioning and manageability, storage as a service, Data storage in cloud computing(storage as a service).
Multitenant software: Multi-entity support, Multi-schema approach, Multitenance using cloud data stores, Data access control for enterprise applications.

Unit 3: Data in the cloud (6 Hours)
Cloud file-systems: GFS and HDFS, BigTable, Features and comparisons among GFS, HDFS etc.
Databases on Cloud: NoSQL, MogoDB, HBase, Hive, Dynamo, Graph databases

Unit 4: Map Reduce (8 Hours)
Map-Reduce and extensions: Parallel computing, The map-Reduce model, Parallel efficiency of Map-Reduce, Relational operations using Map-Reduce, Example/Application of Map-reduce.

Unit 5: Cloud Service Models (8 Hours)
PaaS: Introduction to PaaS - What is PaaS, Service Oriented Architecture (SOA). Cloud Platform and Management - computation, storage
SaaS: Introduction to SaaS, Web services, Web 2.0, Web OS, Case Study on SaaS

Unit 6: Cloud Management and Security (6 Hours)
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

Text Books:

1. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley India.
2. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India

Reference Books:

1. *Barrie Sosinsky, "Cloud Computing Bible", Wiley India*
2. Antohy T Velte, et.al, "Cloud Computing : A Practical Approach", McGraw Hill.
3. McGraw Hill, "Cloud Computing", Que Publishing.

Course Outcomes:

The student 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 resource virtualization technique for a given business case
3. Choose the appropriate file system and database for a given business case
4. Develop a algorithm for a given business case using Map-Reduce model
5. Build a SaaS solution for a real world problem with collaborative efforts
6. Identify the challenges in Cloud Management and Cloud Security