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 ‘A-16’
Effective from Academic Year 2017-18

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

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Program Educational Objectives and Program Outcome of B.Tech (Computer Engineering)</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Course Structure - Module I and II</td>
<td>-</td>
</tr>
<tr>
<td>2.1</td>
<td>CS10102 Computer programming (Theory Course)</td>
<td>-</td>
</tr>
<tr>
<td>2.2</td>
<td>CS10302 Computer Programming (Laboratory)</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Course Structure - Module III</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Course Syllabi for Courses - Module III</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>CS201TH Computer Organization (TH)</td>
<td>12</td>
</tr>
<tr>
<td>4.2</td>
<td>CS202THP Object Oriented Programming (THP)</td>
<td>14</td>
</tr>
<tr>
<td>4.3</td>
<td>CS203TLP Data Structure (TLP)</td>
<td>16</td>
</tr>
<tr>
<td>4.4</td>
<td>CS204THL Digital Electronics and Logic Design (THL)</td>
<td>18</td>
</tr>
<tr>
<td>4.5</td>
<td>CS205TH Discrete Structure and Graph Theory (TH)</td>
<td>21</td>
</tr>
<tr>
<td>4.6</td>
<td>HS201OPE Engineering and Managerial Economics</td>
<td>23</td>
</tr>
<tr>
<td>4.7</td>
<td>CS212SD SD-Core Java</td>
<td>25</td>
</tr>
<tr>
<td>4.8</td>
<td>CS213SD SD-Python</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>CS214SD SD-PHP-MySQL</td>
<td>27</td>
</tr>
<tr>
<td>4.10</td>
<td>CS215SD SD-Problem Solving Programming</td>
<td>28</td>
</tr>
<tr>
<td>4.11</td>
<td>CS216INT Summer Training</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Course Structure - Module IV</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Course Syllabi for Courses - Module IV</td>
<td>32</td>
</tr>
<tr>
<td>6.1</td>
<td>CS206TLP Advanced Data Structure (TLP)</td>
<td>34</td>
</tr>
<tr>
<td>6.2</td>
<td>CS207TH Automata Theory (TH)</td>
<td>36</td>
</tr>
<tr>
<td>6.3</td>
<td>CS208THL Microprocessor and Interfacing/ (THL)</td>
<td>38</td>
</tr>
<tr>
<td>6.4</td>
<td>CS209THL Computer Graphics (THL)</td>
<td>41</td>
</tr>
<tr>
<td>6.5</td>
<td>CS210TH Linear Algebra and Applied Probability (TH)</td>
<td></td>
</tr>
<tr>
<td>6.6</td>
<td>HS307003</td>
<td>Social Sciences (HSS) (TH)</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>6.7</td>
<td>CS211PRJ</td>
<td>Mini Project</td>
</tr>
</tbody>
</table>

## 7 Course Structure - Module V

## 8 Course Syllabi for Courses - Module V

| 8.1 | CS301THL | Data Base Management System (THL) |
| 8.2 | CS302THP | Web Technology (THP) |
| 8.3 | CS303THL | Software Engineering (THL) |
| 8.4 | CS304TH | Design and Analysis of Algorithms (TH) |
| 8.5 | CS305THP | Data Science (THP) |
| 8.6 | HS207002 | COSTING AND COST CONTROL (HSS) (TH) |
| 8.7 | CS315PD | PD- Advanced Java |
| 8.8 | CS316PD | Mobile App Development |
| 8.9 | CS321PD | Big Data Technology |
| 8.10 | CS317PD | PIC microcontroller |
| 8.11 | CS318PD | Ethical hacking and Network Defense |
| 8.12 | CS319PD | Grail Framework |
| 8.13 | CS320PD | Cyber Security and Forensic Tools |
| 8.14 | CS322INT | Industrial Training |

## 9 Course Structure - Module VI

## 10 Course Syllabi for Courses - Module VI

<p>| 10.1 | CS306THP | Operating System (THP) |
| 10.2 | CS307THP | Compiler Design(THP) |
| 10.3 | CS308THL | System Programming (THL) |
| 10.4 | CS309THL | Computer Networks (THL) |
| 10.5 | CS310TH | Signals and Systems (TH) |
| 10.6 | HS307001 | Project Management (HSS) (TH) |
| 10.7 | CS311PRJ | Mini Project |</p>
<table>
<thead>
<tr>
<th>11</th>
<th>Course Structure - Module VII</th>
<th>134</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Course Syllabi for Courses - Module VII</td>
<td>135</td>
</tr>
<tr>
<td>12.1</td>
<td>CS401THL</td>
<td>Artificial Intelligence (THL)</td>
</tr>
<tr>
<td>12.2</td>
<td>CS402THL</td>
<td>Network Security (THL)</td>
</tr>
<tr>
<td>12.3</td>
<td>EL-I (THL)</td>
<td>--</td>
</tr>
<tr>
<td>12.5</td>
<td>EL-II (THP)</td>
<td>--</td>
</tr>
<tr>
<td>12.7</td>
<td>CS405PRJ</td>
<td>Project-I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13</th>
<th>Course Structure - Module VIII (Internship)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Course Syllabi for Courses - Module VIII (Internship)</td>
</tr>
<tr>
<td>14.1</td>
<td>CS406INT</td>
</tr>
<tr>
<td>14.2</td>
<td>CS407INT</td>
</tr>
<tr>
<td></td>
<td>CS408INT</td>
</tr>
</tbody>
</table>

Course Structure - Module VIII (Research Based Project)

<table>
<thead>
<tr>
<th>14</th>
<th>Course Syllabi for Courses - Module VIII (Research Based Project)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.1</td>
<td>CS409PRJ</td>
</tr>
<tr>
<td>14.2</td>
<td>CS410PRJ</td>
</tr>
<tr>
<td>14.3</td>
<td>CS411PRJ</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td>14.1</td>
<td>CS412GIP</td>
</tr>
</tbody>
</table>

Course Structure - Module VIII (Elective Based)

<table>
<thead>
<tr>
<th>14</th>
<th>Course Syllabi for Courses - Module VIII (Elective based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.1</td>
<td>Open EL-I</td>
</tr>
<tr>
<td>14.2</td>
<td>Open EL-II</td>
</tr>
<tr>
<td>14.3</td>
<td>Open EL-III</td>
</tr>
<tr>
<td>14.4</td>
<td>CS413PRJ</td>
</tr>
<tr>
<td>14.5</td>
<td>CS414SEM</td>
</tr>
</tbody>
</table>
### 15 Course Syllabi for EL-I (Lab Based) B.Tech (Computer Engineering)

<table>
<thead>
<tr>
<th>15</th>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1</td>
<td>CS417THL</td>
<td>Embedded Systems</td>
</tr>
<tr>
<td>15.2</td>
<td>CS419THL</td>
<td>Mobile Computing</td>
</tr>
<tr>
<td>15.3</td>
<td>CS421THL</td>
<td>Parallel Computing</td>
</tr>
<tr>
<td>15.4</td>
<td>CS424THL</td>
<td>Image Processing</td>
</tr>
<tr>
<td>15.5</td>
<td>CS426THL</td>
<td>Convergence Technology</td>
</tr>
<tr>
<td>15.6</td>
<td>CS430THL</td>
<td>Neural Network</td>
</tr>
<tr>
<td>15.7</td>
<td>CS431THL</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>15.8</td>
<td>CS432THL</td>
<td>Software Design Methodologies</td>
</tr>
<tr>
<td>15.9</td>
<td>CS437THL</td>
<td>Software Testing and Quality Assurance</td>
</tr>
</tbody>
</table>

### 16 Course Syllabi for EL-II (Project Based) B.Tech (Computer Engineering)

<table>
<thead>
<tr>
<th>16</th>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1</td>
<td>CS420THP</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>16.2</td>
<td>CS422THP</td>
<td>Distributed Computing</td>
</tr>
<tr>
<td>16.3</td>
<td>CS423THP</td>
<td>Ubiquitous Computing</td>
</tr>
<tr>
<td>16.4</td>
<td>CS425THP</td>
<td>Geographical Information System</td>
</tr>
<tr>
<td>16.5</td>
<td>CS427THP</td>
<td>Enterprise System</td>
</tr>
<tr>
<td>16.6</td>
<td>CS428THP</td>
<td>Machine Vision and Pattern Recognition</td>
</tr>
<tr>
<td>16.7</td>
<td>CS429THP</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>16.8</td>
<td>CS433THP</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>16.9</td>
<td>CS434THP</td>
<td>Randomization and Approximation Algorithms</td>
</tr>
<tr>
<td>16.10</td>
<td>CS435THP</td>
<td>Management Information System</td>
</tr>
<tr>
<td>16.11</td>
<td>CS439THP</td>
<td>Modeling and Simulation</td>
</tr>
<tr>
<td>16.12</td>
<td>CS440THP</td>
<td>Data Acquisition System</td>
</tr>
<tr>
<td>16.13</td>
<td>CS442THP</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>16.14</td>
<td>CS436THP</td>
<td>Advanced Computer Architecture</td>
</tr>
<tr>
<td>16.15</td>
<td>CS431THP</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>16.16</td>
<td>CS452INT</td>
<td>Summer Internship</td>
</tr>
<tr>
<td>15.1</td>
<td>CS444TH</td>
<td>Mobile Computing</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>15.2</td>
<td>CS445TH</td>
<td>Software Testing and Quality Assurance</td>
</tr>
<tr>
<td>15.3</td>
<td>CS447TH</td>
<td>Modeling and Simulation</td>
</tr>
<tr>
<td>15.4</td>
<td>CS448TH</td>
<td>Data Acquisition System</td>
</tr>
<tr>
<td>15.5</td>
<td>CS453TH</td>
<td>Cyber Security and Digital Forensics</td>
</tr>
<tr>
<td>15.6</td>
<td>CS454TH</td>
<td>Advanced Computer Architecture</td>
</tr>
<tr>
<td>15.7</td>
<td>CS455TH</td>
<td>Cloud Computing</td>
</tr>
</tbody>
</table>
## Program Educational Objectives (PEO)
### B.Tech (Computer Engineering)

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

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

### List of Programme Outcomes [PO]

Graduates will be able

<table>
<thead>
<tr>
<th>PO</th>
<th>PO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td><strong>Engineering knowledge</strong>: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.</td>
</tr>
<tr>
<td>PO2</td>
<td><strong>Problem analysis</strong>: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.</td>
</tr>
<tr>
<td>PO3</td>
<td><strong>Design/development of solutions</strong>: 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.</td>
</tr>
<tr>
<td>PO4</td>
<td><strong>Conduct investigations of complex problems</strong>: 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.</td>
</tr>
<tr>
<td>PO5</td>
<td><strong>Modern tool usage</strong>: 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.</td>
</tr>
<tr>
<td>PO6</td>
<td><strong>The engineer and society</strong>: 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.</td>
</tr>
<tr>
<td>PO7</td>
<td><strong>Environment and sustainability</strong>: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for</td>
</tr>
<tr>
<td>PO8</td>
<td><strong>Ethics:</strong> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PO9</td>
<td><strong>Individual and team work:</strong> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.</td>
</tr>
<tr>
<td>PO10</td>
<td><strong>Communication:</strong> 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.</td>
</tr>
<tr>
<td>PO11</td>
<td><strong>Project management and finance:</strong> 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.</td>
</tr>
<tr>
<td>PO12</td>
<td><strong>Life-long learning:</strong> 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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSO</th>
<th><strong>PSO Statement</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>Select and incorporate appropriate computing theory principles, data structures and algorithms, programming paradigms to innovatively craft scientific solution addressing complex computing problems.</td>
</tr>
<tr>
<td>PSO2</td>
<td>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.</td>
</tr>
<tr>
<td>PSO3</td>
<td>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.</td>
</tr>
<tr>
<td>PSO4</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project</td>
<td>Test 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lab</td>
<td></td>
</tr>
<tr>
<td>CS201TH</td>
<td>Computer Organization (TH)</td>
<td>S₁</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>CS202THP</td>
<td>Object Oriented Programming (THP)</td>
<td>S₂</td>
<td>3 2</td>
<td>10</td>
</tr>
<tr>
<td>CS203TLP</td>
<td>Data Structure (TLP)</td>
<td>S₃</td>
<td>3 2</td>
<td>10</td>
</tr>
<tr>
<td>CS204THL</td>
<td>Digital Electronics and Logic Design (THL)</td>
<td>S₄</td>
<td>3 - 2</td>
<td>10</td>
</tr>
<tr>
<td>CS205TH</td>
<td>Discrete Structure and Graph Theory (TH)</td>
<td>S₅</td>
<td>3 -</td>
<td>10</td>
</tr>
<tr>
<td>HS201OPE</td>
<td>Engineering Economics (HSS) (TH)</td>
<td>HSS</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Lab</th>
<th>Prac</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS212SD</td>
<td>Core Java</td>
<td>P₁</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>CS213SD</td>
<td>Python</td>
<td>P₂</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>CS214SD</td>
<td>PHP-MySQL</td>
<td>P₃</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>CS215SD</td>
<td>Problem Solving Programming</td>
<td>P₄</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>CS216INT</td>
<td>Summer Training</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>17</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23</td>
</tr>
</tbody>
</table>
CS201TH : Computer Organization

Credits: 3  
Teaching Scheme: 3hrs / Week

Unit 1: Computer Evolution & Arithmetic (7 hours)

Unit 2: Processor Design (6 hours)
CPU Architecture, Register Organization, Instruction types, Types of operands, Instruction formats, addressing modes and address translation. Instruction cycles. RISC Processors: RISC- Features, CISC Features, Comparison of RISC & CISC Superscalar Processors. Case study of Processor.

Unit 3: Control Unit (6 hours)

Unit 4: Memory Organization (8 hours)

Unit 5: Pipelining (7 hours)
Basic concepts: role of cache memory, pipeline performance.  
Data hazards: operand forwarding, handling data hazards in software, side effects.  
Instruction hazards: unconditional branches, conditional branches and branch prediction.  
Performance considerations: effect of instruction hazards, number of pipeline stages

Unit 6: Parallel Processing (6 hours)
Text Books

Reference Books

Course Outcomes:
Student will be able to -
1. Understand the structure, function and characteristics of computer systems.
2. Describe the working of Central Processing Unit and RISC and CISC Architecture.
3. Explore the knowledge about Control Unit Design.
4. Design memory with due consideration of tradeoffs and performance issues.
5. Analyze a pipeline for consistent execution of instructions with minimum hazards.
6. Acquaint the advanced concepts of computer architecture.
CS202THP : Object Oriented Programming

Credits: 3

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

Unit 1: Fundamentals of C++

(8 Hours)

**Programming paradigms:** Imperative/Procedural, Object Oriented, Functional Programming, Logic Programming. Need of Object-Oriented Programming (OOP), Basic Concepts of OOP, Benefits/applications of OOP.

**C++ Programming:** Basics, Data Types, Structures, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members.

**Functions:** Function prototype, Constructors, Destructors, Copy Constructor, Objects and Memory requirements, Static Class members, Data abstraction and information hiding, Inline function, Friend Functions.

**Operator Overloading:** Concept, Operator overloading, Overloading Unary Operators, Binary Operators.

Unit 2: Inheritance and Polymorphism

(8 Hours)

**Inheritance:** Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Types of Inheritance, Public and Private Inheritance, Ambiguity in Multiple Inheritance, constructors in derived classes, Aggregation.

**Polymorphism:** Concept, Types of polymorphism, relationship among objects in inheritance hierarchy, Function overloading,

**Virtual Functions:** Pointers- indirection Operators, Memory Management: new and delete, this pointer, Pointers to Objects, Pointer to derived classes, Function pointers, Pure virtual function, Abstract classes.

Unit 3: Fundamentals of Java

(6 Hours)

Java characteristics, Classes and Objects, Methods and Constructors. Information hiding: access modifiers, Static keyword: class variables and instance variables, Class methods and instance methods. Arrays, Strings.

**Inheritance:** Types of inheritance, Constructors in Derived Classes, Overriding & Hiding Fields & Methods, Interfaces.

**Polymorphism:** Static and Dynamic. Abstract classes & methods, Final classes & methods.

Unit 4: Java Exception Handling, Packages and Multithreading

(6 Hours)

**Exception Handling:** Exceptions, checked & unchecked exceptions, User-defined exceptions.

**Packages:** API packages, create, access and use.

**Multithreading:** Thread life Cycle, Thread Priority, Thread Methods, Inter-thread Communication, Producer-Consumer using Java.
Unit 5: Java I/O  
Java I/O: Introduction to Streams, types of streams: iostreams, Readers and Writers, Console class, Print writer, Stream Benefits.  

Unit 6: Java GUI  
Java GUI: Applet, Applet vs Application. AWT, Swing, Components.  
Layout Manager: Flow, Border, Grid and Card.  
Label, Button, Choice, List, Event Handling (mouse, key), Menus, Jtables, Adapter classes, Database connectivity

Text Books:  

Reference Books:  

Course Outcomes:  
The student will be able to –  
1. Analyze the strengths of object oriented programming.  
2. Develop programming application using C++.  
3. Demonstrate the concepts of data encapsulation, reusability and polymorphism.  
4. Create solutions to problems by applying the knowledge of Exception handling, Packages and Threads.  
5. Design an application using Java I/O’s, File handling.  
6. Solve real world problems with the help of event-based GUI handling principles.
CS203TLP: Data Structures

Credits: 5  
Teaching Scheme: Theory: 3 Hours / Week  
Lab: 2 Hours/Week  
Project Based Lab: 2 Hours/Week

Unit 1: Fundamentals of Data Structures, Sorting & Searching  
(6 Hours)

Unit 2: Stack and Queue  
(6 Hours)

Unit 3: Linked Lists  
(6 Hours)

Unit 4: Trees  
(8 Hours)
Basic terminology, representation using array and linked list, Tree Traversals: Recursive And Non recursive, Operations on binary tree: Finding Height, Leaf nodes, counting no of Nodes etc., Construction of binary tree from traversals, Binary Search trees(BST): Insertion, deletion of a node from BST. Threaded Binary tree (TBT): Creation and traversals on TBT, Height Balanced Tree (AVL): Rotations on AVL tree, Expression Trees, Gaming Trees.

Unit 5: Graph  
(8 Hours)
Terminology and representation, Traversals, Connected components and Spanning trees: Prims and Kruskal’s Algorithm, Shortest Paths and Transitive Closures: Single Source All destinations (Dijkstra’s Algorithm), all pair shortest path algorithm, Topological Sort.

Unit 6: Hashing  
(6 Hours)
List of Practicals: (For THL, TLP courses)

1. Assignment based on Sorting and Searching.
2. Assignment based on Stack Application
3. Assignment based on Queue Application
4. Assignment based on different operations on linked list.
5. Assignment based on BST operations
6. Assignment Based on TBT operations
7. Assignment based on AVL tree
8. Assignment based on DFS and BFS
9. Assignment based on MST
10 Assignment Based on Shortest path problem
11 Assignment based on Hashing
12 Assignment based on Bloom Filters

OR

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

1. Job Scheduling
2. String processing
3. Dictionary and Search engines
4. Modeling the real world problems using graphs and trees
5. Database management system using file structures and Hashing techniques

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. To interpret and diagnose the properties of data structures with their memory representations and time complexity analysis.
2. To use linear data structures like stacks, queues etc. with their applications
3. To handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures with the help of dynamic storage representation.
4. To demonstrate the use of binary tree traversals and to perform various operations on non-linear data structures.
5. To handle the operations on Graph data structure and to solve the applications of Graph data structure.
6. To design and analyze the appropriate data structure by applying various hashing techniques.
CS204THL: DIGITAL ELECTRONICS AND LOGIC DESIGN

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

Unit 1: Combinational logic circuits  (7 Hours)

Unit 2: Sequential logic circuits  (7 Hours)

Unit 3: Algorithmic State Machines and RTL  (7 Hours)
Sequence generator. Shift register (modes of operation), 4 bit bi-directional universal shift register, application of shift registers (Ring counter, Sequence generator, Johnson's counter.) ASM charts, notations, design of simple controller, multiplexer controller method. RTL notations and implementation. Examples on ASM, RTL.

Unit 4: Logic Families  (7 Hours)
Characteristics of Digital ICs: Speed, Power dissipation, fan-out, current and voltage parameters, noise margin, operating temperature etc., TTL: Operation of TTL NAND gate, Standard TTL, TTL Characteristics, Active pull-up, Wired-AND, totem pole, open collector, Unconnected Inputs. CMOS Logic: CMOS Inverter, CMOS NAND and NOR, CMOS characteristics. Wired-logic, Unconnected Inputs, Open-Drain Outputs, Comparison of TTL and CMOS, interfacing TTL to CMOS. In terfacing CMOS to TTL, Tri-state logic: tri-state buffers, inverters.
Unit 5: Programmable Logic Devices (6 Hours)
Programmable Logic array: Input, Output Buffers, AND, OR, Invert/Non-Invert Matrix, Programming the PLA, Applications of PLAs to implement combinational and sequential logic circuits Introduction to FPGA, CPLD. Comparison of FPGA and CPLD.

Unit 6: VHDL programming (6 Hours)
Introduction to VHDL: Modeling Digital systems. Different types of modeling: Structural modeling, Behavior modeling and Data Flow modeling. Design and programming of combinational and sequential circuit based on VHDL modeling types.

List of Practical (THL)

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

II Sequential Circuit Design (Any six)
1. Conversion of flip-flops. e.g. JK to D, T.
2. Ripple (asynchronous) mod –N counter using J-K F-F.
3. Ripple (asynchronous) mod –N counter using IC 7490.
5. Sequence generator using JK flip-flop
6. Pseudo random number generator using 74194.(universal shift register)
7. Sequence detector (Moore ckt) using JK flip-flop
8. Sequence detector (Mealy ckt) using JK flip-flop

III ASM (Any one)
2. Design of simple combinational circuit: half adder and substractor using VHDL language.
Text Books

Reference Books

Course Outcomes:

Students will be able to -
1. Construct combinational circuits.
2. Design sequential circuits.
3. Develop the applications of sequential circuits.
4. Analyze internal structure of logic gates.
5. Select an appropriate design and give solution for the Programmable Devices.
6. Make use of advance programming techniques in Digital systems.
CS205TH: Discrete Structure and Graph Theory

Credits: 03  
Teaching Scheme: 3 Hours / Week

Unit 1  
Logic and Proofs  (6 hrs)  
Propositional logic, applications of propositional logic, propositional equivalences, predicates and quantifiers, rules of inference, introduction to proofs: direct, contrapositive, contradiction, counter example, principle of mathematical induction, strong induction. Proving correctness of programs.

Unit 2  
Elementary Discrete Structures & Basic Counting  (7 hrs)  
Elementary set theory, relations, functions, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions, distinguishable/indistinguishable objects), Binomial coefficients and identities.

Unit 3  
Advanced Counting Techniques  (7 hrs)  
Double counting, combinatorial proof technique., Pigeon-Hole Principle, generalized pigeon-hole principle, Some applications from: Ramsey theorem, Mantel’s theorem, Turan’s theorem, Erdos-Szekeres theorem. Inclusion Exclusion Principle: Counting with Venn Diagrams, counting Derangements, number of primes up to n, number of onto functions, Euler’s phi function.

Unit 4  
Recurrence relations and Generating Functions  (6 hrs)  
Recurrence relations, modeling using recurrence relations, some examples from: Fibonacci numbers, Catlan numbers, Derangements, Tower of Hanoi, partitions, solution of linear recurrence relations with constant coefficients (homogeneous and inhomogeneous), generating functions and their application in counting.

Unit 5  
Elementary Number Theory  (7 hrs)  
Integers, divisibility, greatest common divisor, Euclid's algorithm, linear congruences, chinese remaindering theorem, prime numbers: Fundamental Theorem of Arithmetic, infinitely many primes, large gaps in primes, distribution of primes, prime number theorem (without proof).  
Introduction to groups, some natural examples including multiplicative group modulo primes, subgroups, Lagrange's theorem and implications: Fermat’s little theorem, Euler’s theorem.
Unit 6
Graph Theory (7 hrs)
Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, connected components, degree of a vertex, paths, cycles in graph, Euler and Hamiltonian tours/grahps, Trees, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Theorem on bound on number of edges, Graph colorings, matching in bipartite graphs.

Text Books:
1. “Discrete Mathematics and its applications” by Kenneth Rosen (William C Brown Publisher)
2. “Applied Combinatorics” by Alan Tucker (Wiley Publishing company)
3. “Combinatorics: Topics, techniques, algorithms” by Peter J. Cameron (Cambridge University Press)
4. Graph Theory by Reinhard Diestel (Springer Verlag Publishing Company)
5. Introduction to Graph Theory by Douglas B. West (Prentice-Hall publishers)

The student will be able to –
1. Reason mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
2. Describe the elementary properties of modular arithmetic and their applications in Computer Science like cryptography.
3. Summarize graph theory fundamentals and their applications
4. Develop recurrence relations for a wide variety of interesting problems
5. Express mathematical properties via the formal language of propositional and predicate logic
6. Demonstrate use of pigeon-hole and inclusion-exclusion principle in solving elegant and important problems
HS201OPE: ENGINEERING & MANAGERIAL ECONOMICS

Credits: 2

Teaching Scheme: Theory 2 Hours / Week

Unit I
Engineering Economic Analysis
(07 Hrs)
B. Significance of above concept in real life decision making

Unit II
Time Value of Money & Life Cycle Costing
(07 Hrs)
A. Concept of Interest, Time Value of Money – Basis for comparison of alternatives, Discount Rate, Compound Rate, Present Worth, Future Worth, Annual Worth, Annuity, Perpetuity. Life Cycle Costing - Introduction, methodology, applications of LCC in industrial world, differentiation with traditional costing methods, Capital Budgeting: DCF & NDCF Techniques: Payback, Discounted Payback, ARR, IRR, NPV, Annual Worth, Cost Benefit Ratio.
B. Numerical Applications on Time Value of Money

Unit III
Concept of Demand and Supply
(07 Hrs)
B. Exceptions of Law of Demand & Supply

Unit IV
Concept of Utility, Competition
(07 Hrs)
B. Cases related with above concepts
Text Books
1. Theusen H.G., Engineering Economic Analysis, Prentice Hall of India

Reference Books

Course Outcomes:

Our students will be able to:

1. Analyze the effect of inflation, currency fluctuations, and taxes on decision making
2. Compare and select investment alternatives based on costs and time value of money
3. Analyze the impact of demand and supply on pricing of product and competition
4. Understand the concept of utility and competition and its relevance in business environment
CS213SD: Python

Credits: 2

Teaching Scheme: Lab 2 Hours / Week

List of Practical’s:

1. Syntax basics, Arithmetic/String Operations, Input/Output

2. Control Flow constructs: If-else, Relational and Logical Operators

3. Iteration: While loop, For loop

4. Collections: Lists, Tuples

5. Collections: Sets, Dictionary

6. Functions and Modules: sys, math, time

7. File Handling: Data streams, Access modes, Read/Write/Seek

8. Exception handling: hierarchy, raise, assert

9. OOP: Classes, Objects

10. GUI programming: TkInter

Text Books


Reference Books

Course Outcomes:

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

1. Formulate a concrete problem definition for a given problem with consideration for Python specific data structures and capabilities.

2. Develop functional, reliable and user friendly Python programs for given problem statement and constraints.

3. Evaluate various solution alternatives in terms of correctness, space and time complexity and usability.

4. Utilize state-of-the-art tools and techniques for efficient software development.

5. Defend the efficiency of proposed solution in terms of the choice of data structures and algorithms.

6. Apply the problem solving and programming skills learned through the course for tackling relevant pressing issues in public and private sectors.
CS214SD: PHP-MySQL

Credits: 2

Teaching Scheme: Lab 2 Hours / Week

List of Practical’s:

1. Design a web page using HTML5 and CSS.
2. Design a PHP page to demonstrate the use of variables, functions, conditional and lopping constructs.
3. Design a web form using PHP and apply validation.
4. Design a web page to demonstrate the use of session and cookie.
5. Implement user defined exception handling for the web page.
6. Design an application in PHP using MySQL.
7. Design web page using AJAX and PHP
8. Design file upload and download program in PHP
10. Design a client agent to send an email in PHP
11. Mini Project

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. Simplify the data manipulation using MYSQL database and Flat files
2. Design the web pages using HTML, AJAX and apply CSS formatting styles for look and feel
3. Refine dynamic web pages with user defined exception, session, cookies and filters
4. Choose suitable IDE for the development of Desktop/Mobile devices web applications
5. Deliver realistic and extensible web application product for the benefit of individuals and society
6. Practice and utilize PHP programming paradigms and principles of Web development practices
CS215SD : Problem solving and programming.

Credits: 2

Teaching Scheme: Lab: 2 Hours/Week

1. Defining the Problem and Gathering Data, Formulating a Mathematical Model, Deriving Solutions from the Model, Testing the Model, Preparing to Apply the Model, Implementation.
2. Introduction to programming paradigms - Imperative, Declarative, Functional, Object oriented, Structured.
3. Fundamental Data structures: representation, selection of proper data structure, application.
4. Recursive formulation - Tower of Hanoi, Permutations and combinations, 8 queen’s problem.
5. String processing and pattern searching - linear and sub linear pattern search.
7. Statistical computation: Curve fitting and linear regression.
8. Numerical Integration and Differentiation.
10. Solution for the optimization problems using
    - Linear programming,
    - integer programming,
    - Quadratic programming,
    - The Transportation and Assignment Problems.
11. Algorithmic and programming puzzles.
    - Programming Assignments based on above topics needs to be implemented by using suitable programming paradigm.

Text Books:
Course Outcomes:
After completion of this course student will be able to:
1. Select appropriate programming paradigm for solution of given problem.
2. Apply suitable data structure for efficient storage and retrieval of the data.
3. Formulate recursive and dynamic programming solution.
4. Implement programming solution for statistical analysis techniques.
5. Implement programming solution for optimization problems using Linear programming
6. Solve and scale up algorithmic puzzles to realistic case studies.
CS216INT: SUMMER TRAINING

Credits:

Teaching Scheme:

Students pursuing summer training should undergo a minimum 2 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

<table>
<thead>
<tr>
<th>CO</th>
<th>CO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Analyse real life industry problems</td>
</tr>
<tr>
<td>CO2</td>
<td>Create solutions to problems with the help of latest tools</td>
</tr>
<tr>
<td>CO3</td>
<td>Maintain work ethic in organised sector</td>
</tr>
<tr>
<td>CO4</td>
<td>Cooperate with diverse teams and effectively communicate with all the stake holders</td>
</tr>
<tr>
<td>CO5</td>
<td>Adapt to changing work environments</td>
</tr>
<tr>
<td>CO6</td>
<td>Produce solutions within the technological guidelines and standards</td>
</tr>
</tbody>
</table>
MODULE IV
## S.Y. B. Tech. Structure with effect from Academic Year 2015-16

### Module IV

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Projec t</td>
<td>Test 1 Test 2 HA CA ESE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lab</td>
<td>Credits</td>
</tr>
<tr>
<td>CS206TLP</td>
<td>Advanced Data Structure (TLP)</td>
<td>$S_1$</td>
<td>3</td>
<td>10 20 10</td>
</tr>
<tr>
<td>CS207TH</td>
<td>Automata Theory (TH)</td>
<td>$S_2$</td>
<td>3</td>
<td>10 20 10</td>
</tr>
<tr>
<td>CS208THL</td>
<td>Microprocessor and Interfacing/ (THL)</td>
<td>$S_3$</td>
<td>3</td>
<td>10 20 10</td>
</tr>
<tr>
<td>CS209THL</td>
<td>Computer Graphics (THL)</td>
<td>$S_4$</td>
<td>3</td>
<td>10 20 5</td>
</tr>
<tr>
<td>CS210TH</td>
<td>Linear Algebra and Applied Probability</td>
<td>$S_5$</td>
<td>3</td>
<td>10 20 5</td>
</tr>
<tr>
<td>HS307003</td>
<td>Social Sciences (HSS) (TH)</td>
<td>HSS</td>
<td>2</td>
<td>- - -</td>
</tr>
<tr>
<td>CS211PRJ</td>
<td>Mini Project</td>
<td>$P_1$</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>TOTAL</td>
<td>17</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>-----</td>
</tr>
</tbody>
</table>

CS206TLP: Advanced Data Structures

Credits: 5

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

Unit 1: Advanced Trees and Applications (7 Hours)

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)

Unit 5: Spatial Data Structures (8 Hours)

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 Practical’s:

1. Assignment based on operations on RED-Black trees and van Emde Boas trees.
2. Assignment based on B Trees and B+ Trees.
3. Assignment based on Priority Queues Application
4. Assignment based on tries.
5. Assignment based on Suffix Trees.
6. Assignment Based on Randomized Data Structures.
7. Assignment based on Quad trees and Oct trees
8. Assignment based on Interval trees, Segment trees, Range trees
9. Assignment based on R-trees
10 Assignment Based on Disjoint Set data structures.
11 Assignment based on concurrent data structures.
12 Assignment based on Succinct data structures.

OR

**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:**

**Reference Books:**

**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.
CS207TH : Automata Theory

Credits: 03

Teaching Scheme: 3 Hours / Week

Unit 1: Finite Automata

Unit 2: 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 closure), 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: 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 closure, 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: 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

Text Books: (As per IEEE format)

Reference Books: (As per IEEE format)

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 or hardware used) via the concept of undecidability.
Unit 1: Introduction to 8086 Microprocessor (7 Hours)
Introduction to 80x86 microprocessor, Internal Architecture, Generation of physical address, Minimum & Maximum Mode, Ready and Reset pin significance, study of 8086 supporting chips 8282(Latch), 8284(Clock Generator), 8286(Trans receiver), 8288(Bus Controller), Timing Diagram Read Write Machine Cycles, Real Mode.

Unit 2: Interrupt Structure and Programmable Interval (7 Hours)
General Purpose Instructions, Address Translation, Addressing Modes, Introduction to Assembly Language Programming, Examples on Programming, Interrupt Structure, Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR,NMI, Interrupt Response, Execution of an ISR, Priority of Interrupts. 8253/8254–(Programmable Interval timer/counter) block diagram, control word & interfacing, Operating Modes of 8253.

Unit 3: Interfacing with 8086 (7 Hours)
8255 (Programmable Peripheral Interface 8255)-block diagram, control word, Operating Modes of 8255, interfacing ADC (Successive Approximation Method), DAC (R – 2R ladder Network). 8251(USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming.

Unit 4: 80386 Programming Model (7 Hours)
80386– Features and Architecture, 80386 Programming Model: Register Organization, Segmentation-related instructions, descriptors, memory management through Segmentation, logical to linear/physical address translation

Unit 5: 80386 Paging & Protection (6 Hours)
Virtual Memory, Paging – support registers, descriptors, linear to physical address translation, Control Registers. Protection in segmentation, DPL,RPL & CPL, Privileged instructions, page level protection, Translation Look aside Buffer.

Unit 6: Multitasking & Multi core Technologies (6 Hours)
Concept of Multitasking, Task State Segment, TSS Descriptor, Task Register, Task Gate Descriptor, Task Switching, Task Linking, Task Address Space. Limitation of Single Core, Introduction to parallel computers: Instruction Level Parallelism (ILP) vs. Thread Level Parallelism (TLP); Multi core Technologies.
List of Practical (THL)

1. Study of 8086 Architecture and Execution of sample programs.
2. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
3. Write 8086 ALP to perform block transfer operation. (Don’t use string operations) Data bytes in a block stored in one array transfer to another array. Use debugger to show execution of program.
4. Write 8086 ALP to find and count zeros, positive number and negative number from the array of signed number stored in memory and display magnitude of negative numbers.
5. Write 8086 ALP to convert 4-digit HEX number into equivalent 5-digit BCD number.
6. Write 8086 ALP to convert 5-digit BCD number into equivalent 4-digit HEX number.
7. Write 8086 ALP for following operations on the string entered by the user.
   a. String length
   b. Reverse of the String
   c. Palindrome
8. Write 8086 ALP for following operations on the string entered by the user (Use Extern Far Procedure).
   a. Concatenation of two strings
   b. Find number of words, lines.
   c. Find number of occurrence of substring in the given string.
9. Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC.
10. Write 8086 ALP to interface DAC & generate following waveforms on oscilloscope. Comment on types of DAC’s and write detailed specifications of the DAC used
    i) Square wave -- Variable Duty Cycle & frequency.
    ii) Stair case wave
    iii) Triangular wave
    iv) Sine and Cosine wave
11. Write 8086 ALP to program 8253 in Mode 0 to Mode 5. Generate a square wave with a pulse of 10 ms.
Text Books


Reference Books


Course Outcomes:
Student will be able to –

1. Identify different operating modes of Microprocessor
2. Design Microprocessor systems using peripheral components.
3. Learn peripherals and their interfacing with Microprocessor.
4. Analyze different operating modes of Advance Processor
5. Evaluate various memory management & Protection techniques.
CS209THL: COMPUTER GRAPHICS

Credits: 04

Teaching Scheme: - Theory: 3 Hrs/Week
Lab 2 : Hrs/Week

Prerequisites: C programming

Unit 1 : Introduction to Computer Graphics (6 Hours)

Unit 2 : Polygons and Clipping (7 Hours)
Scan Converting Lines, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons, edge data structure, Clipping Lines algorithms– Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.

Unit 3 : 2D Transformation (8 Hours)

Unit 4 : 3D Transformations and Projections (8 Hours)
**Unit 5: Illumination Models and Visible Surface Determination**  
(6 Hours)

Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Atmospheric attenuation, Phong’s model, Gouraud shading, some examples.


**Unit 6: Plane Curves and Surfaces**  
(6 Hours)


**Text Books:**

**Reference Books:**

All the lab assignments has to be implemented using JAVA/C++/Open GL

**List of Practical**

1. Line Drawing and Circle Drawing.
2. Polygon filling algorithms.
3. Line clipping algorithm.
4. Polygon clipping algorithm.
5. 2D transformations
6. Applications of 2D transformations.
7. 3D transformations
8. Applications of 3D transformations

9. Hidden surface removal

10. Curves and fractals

Course Outcomes:

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

1. Understand computer graphics fundamentals
2. Utilize algorithms to draw, fill and perform clipping on various 2D objects.
3. Use mathematics to achieve 2D transformations on different 2D geometrical shapes
4. Systematically identify and solve numerical problems of 3D transformations and projections
5. Detect hidden surfaces and illumination models
6. Interpret the curves and fractals to represent graphics system.
CS210TH: Linear Algebra and Applied Probability

Credits: 03

Teaching Scheme: 3 Hours / Week

Unit 1

Vectors, Linear Equations, Matrices

Vectors and system of linear equations, matrices, representing linear equations using matrices, elimination using matrices, basic operations on matrices, elementary row operations, row reduced echelon form, inverse of matrix, homogeneous system of equations, Cramer's rule, adjoint matrix, complete solution for system of linear equations. Applications in Computer Engineering.

Unit 2

Vector Spaces and subspaces

Vector spaces, notion of linear independence, dimension, basis of vector space, null space, subspaces, homomorphism and isomorphism of vector spaces, matrix as a linear transformation, similar matrices and change of bases, rank of matrix, row and column space of matrix, rank nullity theorem, Applications in Computer Engineering.

Unit 3

Orthogonality, Introduction to Eigen values and Eigen vectors

Vector length, unit vectors, dot product of vectors, orthogonal vectors and dot product, orthogonal projections, orthogonal bases and Gram-Schmidt orthogonalization, Least square approximation and applications. Eigen values of matrix, matrix determinant and its properties, Eigen vectors of matrix, Eigen space, Geometric interpretation, symmetric matrices, Eigen vectors and Principal Component Analysis, Applications in Computer Engineering.

Unit 4

Basic probability theory

Basic counting, definition of probability, examples, independence of events, conditional probability, inclusion exclusion, Bayes’ rule, coupon collector problem, birthday paradox, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, Markov and Chebyshev's inequality, weak law of large numbers, Basic distributions: Bernoulli, Binomial, Geometric, Poisson, Normal, Log-normal, Rayleigh. Applications in Computer Engineering.

Unit 5

Markov Chains and Random Walks


Unit 6
Queuing Theory

(7 Hrs)
Introduction to queuing theory, basic assumptions, arrival rate, serving time, average number of customers in queue, arrival distribution, Poisson arrival rate, some queuing models, utilization, Little's formula, Applications in Computer Engineering.

Text Books: (As per IEEE format)

Reference Books: (As per IEEE format)

Course Outcomes:
The student will be able to –
1. Solve problems based on system of linear equations, matrices.
2. Grasp the concept of vector space as an abstract algebraic structure.
3. Explain the role of orthogonality, eigenvectors, their significance and applications.
4. Solve problems based on basic probability and random variables.
6. Grasp the basic concepts in queuing theory.
HS307003: SOCIAL SCIENCES

Credits: 02

Teaching Scheme: - Theory 2Hrs/Week

Prerequisites:

Unit I
India: Natural Environment
(8 Hrs)
Physiography - location, relief, structure, major physiographic units; Drainage; Role of rivers in the economy; Pollution of rivers and its control. Climate: Factors influencing climate; Monsoon - Its characteristics. Distribution of Rainfall and Temperature; Formation of seasons; Climate and Human Life Vegetation and wild life of India

Unit II
Resources and Development
(8 Hrs)

Unit III
Ethics, Citizen, State and the Constitution
(8 Hrs)

Unit IV
Contemporary India : Issues and Challenges
(8 Hrs)

Text Books

1. R. Rajagopalan, Environmental Studies: From Crisis to Cure, Oxford Press

Reference Books

1. Shukla V.N. - Constitution of India.

Course Outcomes:

Students will be able to:

1. Understand the inter-relationship between environment resources and development
2. appreciate the need for active participation of the citizen in the successful functioning of a democracy
3. Understand the political, social and economic challenges faced within and outside of the country
## T.Y. B. Tech. Structure with effect from Academic Year 2015-16

### Module V

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project</td>
<td>HA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lab</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td>CS301THL</td>
<td>Data Base Management System (THL)</td>
<td>$S_1$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>CS302THP</td>
<td>Web Technology (THP)</td>
<td>$S_2$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS303THL</td>
<td>Software Engineering (THL)</td>
<td>$S_3$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>CS304TH</td>
<td>Design and Analysis of Algorithms (TH)</td>
<td>$S_4$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>CS305THP</td>
<td>Data Science (THP)</td>
<td>$S_5$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>HS207002</td>
<td>COSTING AND COST CONTROL</td>
<td>HSS</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Type</td>
<td>Credits</td>
<td>Theory</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>CS315PD</td>
<td>Advanced Java</td>
<td>PD</td>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>CS316PD</td>
<td>Mobile App Development</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS321PD</td>
<td>Big Data Technology</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS317PD</td>
<td>PIC microcontroller</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS318PD</td>
<td>Ethical hacking and Network Defense</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS319PD</td>
<td>Grail Framework</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS320PD</td>
<td>Cyber Security and Forensic Tools</td>
<td>PD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CS322INT</td>
<td>Industrial Training</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>17</td>
<td>4</td>
</tr>
</tbody>
</table>
CS301THL : DATABASE MANAGEMENT SYSTEMS

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

Unit 1: Introduction and Data Models  
(7 Hours)  

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)  

Unit 5: Transaction Management  
(7 Hours)  

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:

Reference Books:

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 up to 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
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 JavaScript/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:

Reference Books:

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
CS303THL: Software Engineering (THL)

Credits: 03

Teaching Scheme: Theory 3 Hrs/Week

Unit 1: Software Engineering Paradigms (7 Hours)

Unit 2: Requirement Engineering (7 Hours)

Unit 3: Introduction to Agile Methodology (7 Hours)

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

Unit 5: Configuration Management Practices (7 Hours)
Values and CM, CM Practices for Agile, CM Standards and Frameworks to Support Agile

Unit 6: Project Management Principles (7 Hours )

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 indicating Sprint Review Issues and Sign-offs.

Text Books:

Reference Books:
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 tasks and 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.
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.


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

Text Books:

Reference Books:

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 NP-completeness of some decision problems, grasp the significance of the notion of NP-completeness 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.
CS305THP: 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’s Inequality, 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

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
   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.
HS207002 : COSTING AND COST CONTROL

Credits: 02

Teaching Scheme: - Theory 2Hrs/Week

Unit I
Cost

Unit II
Overheads

Unit III
Costing Methods

B. Variance – Variance Analysis. Material variance, Labour Variance, Overhead Variance

Unit IV
Activity Based Costing & Transfer Pricing
A. Concept, Concept of Cost Drivers. Calculation of Costs. Mechanism of Activity Based Costing. Transfer Pricing: Objective, Methods – Cost Based, Market Prices Based, Negotiated Prices. Recommended procedure for Transfer Pricing

B. Limitations of Traditional Costing

Total Contact Hours: 28
Text Books:
1. Prasad N. K.; Cost Accounting; Book Syndicate Pvt. Ltd., Calcutta 700 009.

Reference Books:

Course Outcomes:
1. Classify different types of costs and apply it for ascertainment of costs of a product or a process
2. Understand and apply distribution of overheads to ascertain the cost of any product or service.
3. Apply different types of costing methods and techniques according to the suitability for various production processes and services.
4. Understand and apply the concept of activity based costing for cost ascertainment
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:

Reference Books:

Course Outcomes:
The student will be able to –
1. Analyze the nature of a problem to select appropriate advanced feature of Java towards achieving at a problematic solution.
2. Develop Java based solution for real world problem.
3. Employ Integrated Development Environment (IDE) for implementing and testing of software solution.
4. Work in well-formed teams with proper skill sets to achieve effective solutions.
5. Extend their knowledge in utilizing the appropriate advanced features of Java for using emerging frameworks.
6. Construct software solutions by evaluating alternate architectural patterns.
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:

Reference Books:
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
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:
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
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

Reference Books:
1. Microchip 18F45xx

Course Outcomes:
The student will be able to –
2. Utilize Advanced Features of Advanced Peripherals.
5. Cooperate with Diverse Teams to create Solutions.
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
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
14. Study of Physical security.

Text Books:

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.
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

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


Reference Books


Additional Reading

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.
CS322INT: INDUSTRIAL TRAINING

Credits: 02

Teaching Scheme: ---- Hrs/Week

Information:
Students pursuing industry training should undergo a minimum 4 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

<table>
<thead>
<tr>
<th>CO</th>
<th>CO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Analyse real life industry problems</td>
</tr>
<tr>
<td>CO2</td>
<td>Create solutions to problems with the help of latest tools</td>
</tr>
<tr>
<td>CO3</td>
<td>Maintain work ethic in organised sector</td>
</tr>
<tr>
<td>CO4</td>
<td>Cooperate with diverse teams and effectively communicate with all the stake holders</td>
</tr>
<tr>
<td>CO5</td>
<td>Adapt to changing work environments</td>
</tr>
<tr>
<td>CO6</td>
<td>Produce solutions within the technological guidelines and standards</td>
</tr>
</tbody>
</table>
MODULE VI
### Structure and Syllabus of S.Y. B.Tech Computer Engineering Pattern A

#### Module VI

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Project</td>
</tr>
<tr>
<td>CS306THP</td>
<td>Operating System (THP)</td>
<td>$S_1$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS307THP</td>
<td>Compiler Design (THP)</td>
<td>$S_2$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS308THL</td>
<td>System Programming (THL)</td>
<td>$S_3$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>CS309THL</td>
<td>Computer Networks (THL)</td>
<td>$S_4$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>CS310TH</td>
<td>Signals and Systems (TH)</td>
<td>$S_5$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>HS307001</td>
<td>Project Management (HSS) (TH)</td>
<td>HSS</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>CS311PRJ</td>
<td>Mini Project</td>
<td>P</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>TOTAL</td>
<td>17</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

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)

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)

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.


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:

Reference Books:
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.
CS307THP: Compiler Design

Credits: 04
Teaching Scheme: Theory: 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)


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


Unit 4: Code Generation (6 Hours)


Unit 5: Code Optimization (8 Hours)

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 Projects

1. Assignments to understand basic syntax of LEX specifications, built-in functions and Variables.
2. Assignments to understand usage of YACC parser generator, Symbol Table generation.

Mini projects based on:

3. Design the parser (top-down or bottom-up) for subset of input source language. The students are supposed to write the grammar for subset of language, construct parsing table and handle error detection and recovery.
4. Design the front end of a compiler to analyze the syntax and semantics of input program and generate the three address code.
5. Design code optimizer to perform the optimizing transformations on input 3-address code. Identify basic blocks and perform data flow analysis.
6. Design code generator to generate an appropriate target code for the input intermediate code assuming suitable processor details.

Text Books:


Reference Books:

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.
CS308THL: System Programming

Credits: 4
Teaching Scheme: 3 Hours / Week
: Lab 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 Practicals:**
1. Design and implementation of 2 Pass macroprocessor.
2. Design and implementation of 2 Pass assemblers with generating different data structure for it.
4. Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, white spaces, lines, functions etc.
5. Simulation of linkers.
7. Design and implementation of DLL on Linux shared library.
8. Comparative study of different debugger tools.
9. Implement device driver on Linux system.
10. Implementation of TSR program.

**Text Books:**

**Reference Books:**
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.
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)

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

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:**

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

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
CS310TH : Signals and Systems

Credits: 03
Teaching Scheme: 3 Hours / Week

Unit 1: Basics of Signals and Systems
Classification of Signals: Analog, Discrete-time and Digital
Basic sequences and sequence operations.
Systems: Discrete-time (D.T) systems, Properties of D. T. Systems and Classification,
Linear Time Invariant Systems
Impulse response, linear convolution and its properties
Periodic Sampling, Sampling Theorem, Frequency Domain representation of sampling,
reconstruction of a band limited Signal
A to D conversion Process: Sampling, quantization and encoding

Unit 2: Digital Signals in Frequency Domain
Z-transform, ROC and its properties, Inverse z transform by inspection, partial fraction
and complex inversion
Z transform properties: multiplication by exponential sequence, Linearity, time shifting,
frequency shifting, time reversal: convolution theorem initial value theorem, final value
theorem, System Functions for LTI Systems: Stability and causality, inverse systems,
significance of poles/zeros, Unilateral Z-transform: solution of difference equation with
and without initial condition. Representation of
Sequences by Fourier Transform, F. T. theorems: convolution theorem, and windowing
theorem

Unit 3: Frequency analysis of Signals and Systems
Frequency Response of LTI Systems: Ideal frequency selective filters, magnitude and
phase response, group delay, Frequency Response for Rational System Functions:
Frequency Response of a single zero or pole, Frequency response from pole-zero plot
using simple geometric construction
Sampling the F.T., Fourier representation of finite-duration sequences, The Discrete
Fourier Transform, Properties of DFT: circular shift, duality, symmetry, Circular
Convolution, Linear Convolution using DFT, Effective computation of DFT and FFT,
DIT FFT, Overlap and save algorithm, Inverse DFT using FFT (DIF)

Unit 4: Design of Digital Filters
Concept of filtering, Ideal filters and approximations, specifications, IIR filter design
from continuous time filters: Characteristics of Butterworth, Chebyshev approximations,
impulse invariant and bilinear transformation techniques, Design examples, FIR filter
design using windows: properties of commonly used windows, systems with Linear
phase, Generalized Linear phase systems, Four Types of GLPS (Type I), Design
Examples, Design using Kaiser window, Comparison of IIR and FIR Filters
Unit 5: Realization of Filters  
(6.. Hours)
Block diagrams and Signal flow graph representation of LCCDE
Basic structures for IIR Systems: direct form, cascade form, parallel form, feedback in IIR systems
Basic Structures for FIR Systems: direct form, cascade form, structures for linear phase FIR Systems

Unit 6: Applications of Signal Processing  
(6.. Hours)
Sound and Audio Equalizer
Speech Recognition for Digital Assistants
Instagram Filters in Image Processing

Text Books: (As per IEEE format)
1. Name(s) of author(s); Title of the book; Edition No., Publisher
2.

Reference Books: (As per IEEE format)
1. Name(s) of author(s); Title of the book; Edition No., Publisher
2.
3.

Course Outcomes:
The student will be able to –
1. Convert analog signal into a digital signal without irreversible data loss using mathematical techniques
2. Design faster algorithms for signal’s mathematical domain conversion which ensures expected operation on all architectures
3. Design filters meeting the given specifications
4. Design a stable system which requires minimum components to implement
5. Analyze a signal in different mathematic domains such as Fourier domain to understand the signal characteristics.
6. Implement mathematically optimum realizations technique
HS307001: PROJECT MANAGEMENT

Credits: 02  
Teaching Scheme: - Theory 2Hrs/Week

Prerequisites: Nil

Unit I
Introduction:
(8 Hrs)
Project Life Cycle Phases – Concept/Initiation Phase, Project Definition Phase, Project Planning & Organization Phase, Project Implementation Phase, Cleanup & Shutdown Phase
B. Governmental Framework for Identification of Opportunities, Incentives from state & central govt.; Import-substitution projects.

Unit II
Project Planning & Organization
(8 Hrs)
B. Project Organization & Management. Project Organization Structure, Role of Project Manager

Unit III
Project Monitoring & Contract Management
(8 Hrs)
B. Project Management Information System and Control, Management Pitfalls

Unit IV
Computer Applications in Project Planning & Control
(8 Hrs)
A. Introduction to MS Projects – Understanding the MS Project screen & different views, Defining the project, Working with calendar, Outline the project, Create dependencies between tasks, Creating WBS, Format task list and Gantt chart, Resource planning, leveling and preparing resource graph, Working with baseline, tracking the project.

B. Home Assignment on Exercise with MS Projects Software.

Text Books

1. Narendra Singh; Project Management & Control; Himalaya Publishing House, Mumbai.
2. S.Choudary, Project Management, Tata McGraw Hill
3. Prasanna Chandra; Project: Preparation, Appraisal, Budgeting & Implementation
4. Pinto, Project Management – Achieving Competitive Advantage & MS Projects, Pearson Education

Reference Books

1. Maylor, Project Management, Pearson Education,
2. Gopal & Ramamurthy; Project Management Handbook; Macmilan.
3. Project Management Body of Knowledge

Course Outcomes:

Students will be able to:

1. Learn the basic concepts of project and project management
2. Plan and schedule small and medium projects to achieve the triple constraint of time, cost and quality using software package
3. Understand the concept of earned value management system and critical chain in managing projects
4. Monitor the progress of projects to determine variances and recommend corrective actions using software package
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:

- Transform existing Ideas into conceptual models.
- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- 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 of 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

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Synopsis</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Project Feasibility</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Concise System Analysis</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>System Design Block Diagram</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Architectural Design Elements Description</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>System Implementation</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>Test Cases and Result</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion ,Presentation</td>
<td>10</td>
</tr>
</tbody>
</table>

### 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:
<table>
<thead>
<tr>
<th>Course</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Networks</td>
<td>Image Processing</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td>Network Security</td>
<td>Expert Systems</td>
</tr>
<tr>
<td>Digital Signal Processing</td>
<td>Object Oriented Systems</td>
</tr>
<tr>
<td>Systems Programming</td>
<td>Modeling and Design</td>
</tr>
<tr>
<td>Real Time Systems</td>
<td>System Testing</td>
</tr>
<tr>
<td>Embedded systems</td>
<td>Storage Management</td>
</tr>
<tr>
<td>Cluster Computing</td>
<td>Client-Server Computing</td>
</tr>
<tr>
<td>Mobile &amp; Wireless Communications</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>Multimedia Systems</td>
<td>Protocol Engineering</td>
</tr>
</tbody>
</table>
MODULE VII
B. Tech. Structure with effect from Academic Year 2015-16

Module VII

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Projec</td>
</tr>
<tr>
<td>CS401THL</td>
<td>Artificial Intelligence (THL)</td>
<td>S_1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS402THL</td>
<td>Network Security (THL)</td>
<td>S_2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>EL-I (THL)</td>
<td>S_3</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>EL-II (THP)</td>
<td>S_4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS405PRJ</td>
<td>Project-I</td>
<td>P</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>17</td>
<td>4</td>
</tr>
</tbody>
</table>
CS401THL: Artificial Intelligence

Credits: 04  
Teaching Scheme: - Theory: 3 Hrs/Week 
Lab: 2 Hrs/Week

Prerequisites:

Unit 1: Fundamentals of Artificial Intelligence (6 Hours)

Unit 2: Uninformed Search Strategies (7 Hours)
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)

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 (7 Hours)
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

Reference Books
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley

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.
CS402THL: NETWORK SECURITY

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

Prerequisites: Computer Networks.

Unit 1: (7 Hours)
Introduction
Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services.
Trustworthiness, Ethical issues and practices, Tradeoffs of balancing key security properties - Confidentiality, Integrity, Availability.
Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks, ransomware, SYN-Flooding, SQL-injection, DNS poisoning, Sniffing
Bitcoin and Crypto currency system

Unit 2: (8 Hours)
Private key cryptography
Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).
Role of random numbers in security, Importance of prime numbers
Data Encryption Standard: Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.
Chinese remainder theorem

Unit 3: (7 Hours)
Public key cryptography
RSA: RSA algorithm, Key generation in RSA, attacks on RSA.
Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.
Elliptic Curve Cryptography (ECC): Elliptic Curve over real numbers, Elliptic Curve over Zp, Elliptic Curve arithmetic. Diffie-Hellman key exchange using ECC.

Unit 4: (8 Hours)
Authentication and access control
Authentication Applications: Kerberos, X.509 authentication service, public key infrastructure.
Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control.
Unit 5:  
Security application and design  

Transport layer security: SSL and TLS. 
Application layer security: Security services, S/MIME, PGP, Https, Honey pots. 
Security design: End-to-end security, Security composability, Open design, Cost and tradeoffs 

Unit 6:  
Cyber Security:  

List of Practicals:  
1. Implementation of Caesar and Vigenere Cipher  
2. Implementation of Playfair Cipher  
3. Implementation of Hill Cipher  
4. Implementation of S-RC4  
5. Implementation of S-DES  
6. Implementation of S-AES  
7. Implementation of RSA.  
8. Implementation of Diffie-Hellman key exchange  
10. Study of Nessus tool  

Text Books  

Reference Books  

Additional Reading  
Course Outcomes

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

1. Analyze cryptographic techniques using a mathematical approach by examining nature of attack.
2. Establish type of attack on a given system.
3. Identify different types of attacks.
4. Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
5. Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization.
6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
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

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Synopsis</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Feasibility Study</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>System Requirement Specification</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>System Analysis Document</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Use Case Diagram</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Detailed Design Plan</td>
<td></td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>ISA</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS406INT</td>
<td>Open EL-I</td>
<td>$S_1$</td>
<td>3 - -</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>CS407INT</td>
<td>Internship</td>
<td>P</td>
<td>- 8 -</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>CS408INT</td>
<td>Project Based Viva</td>
<td>P</td>
<td>- 4 -</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>17 4 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The Teaching Scheme includes Theory, Project, and Lab. The Assessment Scheme includes ISA, Test 1, Test 2, HA, CA, and ESE. The credits column represents the total credit hours for each subject.
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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Analyse real life industry problems</td>
</tr>
<tr>
<td>CO2</td>
<td>Create solutions to problems with the help of latest tools</td>
</tr>
<tr>
<td>CO3</td>
<td>Maintain work ethics in organised sector</td>
</tr>
<tr>
<td>CO4</td>
<td>Cooperate with diverse teams and effectively communicate with all the stakeholders</td>
</tr>
<tr>
<td>CO5</td>
<td>Adapt to changing work environments</td>
</tr>
<tr>
<td>CO6</td>
<td>Produce solutions within the technological guidelines and standards</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>CO</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Analyse real life industry problems</td>
</tr>
<tr>
<td>CO2</td>
<td>Create solutions to problems with the help of latest tools</td>
</tr>
<tr>
<td>CO3</td>
<td>Maintain work ethic in organised sector</td>
</tr>
<tr>
<td>CO4</td>
<td>Cooperate with diverse teams and effectively communicate with all the stake holders</td>
</tr>
<tr>
<td>CO5</td>
<td>Adapt to changing work environments</td>
</tr>
<tr>
<td>CO6</td>
<td>Produce solutions within the technological guidelines and standards</td>
</tr>
</tbody>
</table>
ELECTIVE List with Lab
Vishwakarma Institute of Technology  Issue 01 : Rev No. 1 : Dt. 24/03/17

CS417THL: Embedded system

Credits: 4

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

Unit 1: Introduction to Embedded Systems (7 Hours)

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 Program, 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 Diagram, Program Model, Reset / Clocking, Memory Org, Program/Data, Flash Eprom, Add Mode/Inst Set Program, I/O, Interrupt, Timer, ADC Design Parameters problem solving, PWM ADC Timers

Unit 4: I/O interfacing (6 Hours)

Unit 5: Bus Technologies (6 Hours)

Unit 6: RTOS (7 Hours)

Embedded Systems Lab Assignments

1. Study of Microcontroller Development Board
2. Led Flashing
3. 7 Segment LED Display

Page 111 of 186
4. Buzzer / Relay
5. Switch Interface
6. Timer Frequency generation
7. LCD
8. ADC
9. Serial Communications + Graphical GUI on Desktop / Laptop
10. Serial Peripheral Interface EEPROM / RTC
11. IIC Interface EEPROM / RTC
12. Stepper Motor
13. PWM Motor / LED Control
14. Mini Project

Text Books:

Reference Books:
Reference Books

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
<table>
<thead>
<tr>
<th>CO1</th>
<th>Understand Steps in System Design using Computing Devices</th>
<th>PO1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>Design system interconnects for effective throughput</td>
<td>PO5</td>
</tr>
<tr>
<td>CO3</td>
<td>Create designs using Simulation and RTOS Tools</td>
<td>PO9</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>CO4</td>
<td>Deliver Complex Automation Solutions</td>
<td>PO10</td>
</tr>
<tr>
<td>CO5</td>
<td>Cooperate with diverse Teams for delivering automation Solutions</td>
<td>PO16</td>
</tr>
<tr>
<td>CO6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CS419THL: Mobile Computing

Credits: 4

Teaching Scheme:
Theory 3 Hours/Week
Lab: 2 Hours/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)

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

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.

List of Practicals: (For THL)
Assignment 1. Design an android Application for Phone Call
Assignment 2. Design an android application for media player
Assignment 3. Design an android Application for SMS Manager

Assignment 4. Design an android Application using Google Map To Trace The Location of Device

Assignment 5. Design an android Application for Frame Animation

Assignment 6: Mini Project

Assignments should be implemented on android operating systems.

Text Books:

Reference Books: (As per IEEE format)

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 busy 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.
CS421THL: Parallel Computing on GPU

Credits: 4

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

Unit 1: Fundamentals of Parallel computing and architectures (8 Hours)
Parallel programming definition, motivation, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn’s classification of multi-processors, Definition of thread and process, programming parallel computers, Parallel computing architectures (multi-core CPUs, GPUs, traditional multi-processor system, Xeon-Phi, Jetson Kit, Kilocore processor), multiprocessor and multicomputer systems, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability.

Unit 2: Introduction to GPU architecture and parallel algorithms (8 Hours)
Introduction to Modern GPU Tesla architecture, Types of GPU memories: global, shared, texture memory and their properties and uses, Streaming processor (SP), Streaming multiprocessor (SM), Special Functional unit (SFU), SM instruction types Fosters Parallel algorithm design, Designing GPU parallel algorithm for pattern clustering.

Unit 3: Introduction to CUDA (8 Hours)
Introduction to CUDA programming model: threads, blocks, grid, Kernel, Kernel definition and kernel launch configuration, Use of GPU memories: global, shared, texture and constant memories, shared memory: organization, bank conflicts, global memory coalesced accesses, CUDA APIs: for memory allocation, synchronization, Execution of a CUDA kernel on GPU: concept of warp, warp divergence, CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc). Atomic operations in CUDA and their use.

Unit 4: Scientific Computing and problem solving on GPU-Part1 (6 Hours)
Parallel computation of binomial coefficients, Multi-variate polynomials in power form and their GPU parallel evaluation, Polynomials in Bernstein form and parallel computation of Bernstein coefficients: conventional method and using matrix method,

Unit 5: Scientific Computing and problem solving on GPU-Part2 (5 Hours)
Parallel reduction on GPU and its applications. Compute intensive research oriented problems decided by instructor and their GPU parallelization. GPU Parallel implementation of nearest neighbor classifier for large data sets.

Unit 6: CUDA code optimization and Performance improvement (5 Hours)
CUDA code optimization: Memory optimization, Control flow optimization, Execution configuration optimization and Instruction optimization, Concept and application of page locked host memory, Single vs double precision computing on GPU: precision vs speed of computation, choosing correct precision for a real GPU application, memory leaks and associated problems, CUDA tools: cuda-memcheck and profiler.
List of Practical's Parallel Computing on GPU

1. Parallel GPU implementation of vector-vector operations
2. Parallel GPU implementation of vector-Matrix operations
3. Parallel computation of binomial coefficient matrix
4. Parallel GPU implementation of Matrix-Matrix operations
5. Assignment focusing on optimization of data transfer between CPU and GPU: using page locked host memory and to avoid the data transfer
6. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.
7. Parallel GPU implementation involving kernel looping.
8. Working of CUDA tools MemCheck and & Visual Profiler

Text Books: (As per IEEE format)
2. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley

Reference Books: (As per IEEE format)
1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP , Tata McGraw-Hill Edition

Course Outcomes:
The student will be able to –
1. Recognize fundamentals of parallel computing and architectures available
2. Design parallel algorithms that better maps on GPU architecture
3. Write CUDA applications for execution on GPU
4. Apply parallel computing methods to scientific and engineering problems
5. Apply parallel computing methods to research problems
6. Optimize CUDA code using tools for performance improvements
CS424THL: Image Processing

Credits: 4

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

Unit 1: Introduction to Image processing (6 Hours)
Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YCbCr and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF, png, raster image format). Image sampling and quantization.

Unit 2: Image Enhancement (7 Hours)
Threshholding, Spatial domain techniques { Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average, Image Smoothening: low-pass spatial filters, median filtering, Image Sharpening: high-pass spatial filter, derivative filters, Frequency domain techniques- Ideal low-pass filter, Butterworth low-pass filter, High-pass filter, Homo-morphic filters.

Unit 3: Image Analysis (7 Hours)
Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation Feature Extraction- Boundary representation( Chain code, B-spline representation, Fourier descriptor) Region representation (Area, Euler number, Eccentricity, Shape matrix, moment based descriptor), texture based features.

Unit 4: Image Compression and Object recognition (7 Hours)
Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), scalar and vector quantization Object Recognition { Need, Automated object recognition system, pattern and pattern class, relationship between image processing and object recognition, approaches to object recognition.

Unit 5: Image Transform (6 Hours)

Unit 6: Wavelet Transform in image processing (7 Hours)
Sub band coding, Haar Transform – it’s application as a Wavelet, multi resolution expansions, Wavelet Transform in one dimensions; Wavelet transforms in two dimensions, DB4, Fast Wavelet Transform, Other Applications of Wavelet in image processing

List of Assignments: (THL course)
1. Write matlab code to display following binary images
   - Square
   - Triangle
   - Circle
   Write matlab code to perform following operations on images
   - Flip Image along horizontal and vertical direction.
   - Enhance quality of a given image by changing brightness of image.
   - Image negation operation.
   - Change contrast of a given Image.
2. Write Matlab code to implement pseudo colouring operation of a given image.
   Write Matlab Code for Pseudo Colour of Image by using Gray to colour transform.
3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.
4. Write matlab code to find following statistical properties of an image.
   - Mean
   - Median
   - Variance
   - Standard deviation
   - Covariance.
5. Write matlab code to enhance image quality by using following techniques
   - Logarithmic transformation
   - Histogram Equalization
   - Gray level slicing with and without background.
   - Inverse transformation.
6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.
7. Write matlab code to enhance image quality by using following techniques
   - Low pass and weighted low pass filter.
   - Median filter.
   - Laplacian mask.
8. Write matlab code for edge detection using Sobel, Prewitt and Roberts operators.
9. Write C-language code to find out Huffman code for the following word
   COMMITTEE.
10. Write matlab code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3, Y-0.1).
11. Write matlab code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.
Text Books:

Reference Books:

Course Outcomes:
The student will be able to
1. Describe image model
2. Perform spatial filtering on image
3. Identify Image Segmentation techniques.
4. Apply lossless and lossy compression techniques for image compression.
5. Use various image transforms to analyze and modify image.
CS426THL: CONVERGENCE TECHNOLOGY

Credits: 4

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

Unit 1: Introduction to Convergence (6 Hours)
- PSTN: Infrastructure, Working, Services, PSTN versus Internet.
- What is network Convergence, the promise of network convergence, networking issues and convergence, benefits of IP centric network, challenges of converged network, introduction to VOIP, applications of converged networks, VOIP implementation challenges.
- voice and data network growth factor, effects of network convergence on businesses.

Unit 2: Protocols and Standards for Convergence (7 Hours)
- Protocols Supporting VOIP: Multicast IP, RTP, RTCP, RSVP, RTSP, SDP, SAP, SIP.
- Subscriber Lines: T1/T3, DS0, DS1, DS3, E1/E3. Signaling Standards: H.323, SIP.
- Gateways, Gatekeepers. MGCP, Audio and Video Codecs.

Unit 3: Switching networks (7 Hours)
- ISDN: conceptual view of ISDN, transmission structure, user-network interface configuration, ISDN Protocol Architecture, ISDN connection, Addressing, Interworking, PRI, BRI, LAPD.
- Basic Call control, SS7.
- B-ISDN standards, Broadband services, B-ISDN architecture, B-ISDN protocol reference model.
- ISDN standards, SDH.

Unit 4: Frame Relay (7 Hours)
- Frame Relay Circuits, Frame mode protocol architecture, frame mode call control, LAPF, Congestion in frame relay networks, approaches to congestion control, Traffic rate management, Explicit congestion avoidance, implicit congestion control.

Unit 5: ATM technology (7 Hours)
- ATM VPI& VCI, Creation of virtual channel, Definitions of Virtual circuit and permanent virtual circuit, ATM reference model, step-by-step PVC example of how ATM network processes cells, AAL, Adaption layer from voice over ATM perspective AAL1, AAL2, AAL3, Connection admission control (CAC). Cell Loss Priority (CLP), ATM and convergence technology. ATM versus Frame relay, ATM versus SONET.

Unit 6: SMDS and MPLS (6 Hours)
- SMDS: introduction to SMDS, SMDS interface protocol, SMDS addressing. Comparison of SMDS with other LAN technologies.
MPLS: Evolution on MPLS, Architecture, Forwarding Labeled Packets, Label Distribution Protocol, MPLS VPN, IPv6 Over MPLS

Suggested List of Lab Assignments: (For THL)

1. Study of audio and video data.
2. Signal passing through network for voice data
3. Simulation model for transfer of data from DS0, DS1, DS3 lines to E1/T1, E3/T3 lines or vice versa
4. Study of E/T cables
5. Simulation model based on SIP
6. Study of Gateways and Gatekeepers
7. Simulation project for use of Audio and Video
8. Simulation of ATM network
10. Study of SS7 implementation.
11. Comparison of ISDN and B-ISDN.

Text Books: (As per IEEE format)
5. MPLS fundamentals by Luc De Ghein, CCIE No. 189, Cisco Press.

Reference Books: (As per IEEE format)

Course Outcomes:
The student will be able to –
1. Design a network to support voice and data services.
3. Judge the impact and benefits of converged network in exploitation on environment and society.
4. Participate individually and collaboratively to examine the performance of converged- network.
5. Prepare cost effective solutions to fulfill the need of society by convergence technology.
6. Analyze the requirements of the society and propose efficient solutions using convergence technology.
CS430THL:: NEURAL NETWORKS

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

Unit 1 Introduction (7 Hrs)
Introduction and Role of Neural Network (NN), Artificial Neural Networks, Biological Neural Networks, Components of Neural Network, Network topologies, Linear Separability, Learning Rules, Activation Functions, Mc Culloch Pitts NN, Hebb Net, Perceptron, Paradigms of Learning: Supervised, Unsupervised, Reinforced, Examples on Learning.

Unit 2: Perceptron Networks (7 Hrs)

Unit 3: Competitive Neural Network (8 Hrs)
Fixed weight Competitive Neural Network, Winner takes all algorithms, Hebbian Networks, Maxnet, Kohonen Self Organizing Maps, Learning Vector Quantization, Counter propagation: Fully Counter propagation neural network, Forward only Counter propagation Neural network, Self organizing Maps and Applications.

Unit 4: Adaptive Resonance Theory (ART) (6 Hrs)

Unit 5: Pattern Association (6 Hrs)
Training Algorithm for Pattern association, Heteroassociative Associative Memory Neural Network, Auto associative Neural Network, Iterative Auto associative Neural Network, Discrete Hopfield Network, Bidirectional associative Memory (BAM).

Unit 6: Optimization Algorithms (6 Hrs)
Genetic Algorithm, Differential Evolutionary Algorithm, Ant colony Algorithm, and applications of these algorithms. Applications of NN: Signal Processing, Pattern recognition, Medicine, Speech Production/Recognition, Business.

List of Lab Practicals
1. Verification of logic gates using NN algorithms
2. Supervised Learning rules for a single neuron
3. Backpropagation algorithm
4. Fruit Classification Using Backpropagation algorithm.
5. Iris Data Clustering using K means algorithm.
7. Face Recognition
8. KDD data analysis using SVM.
9. Implementation of Hopfield network
10. Implementation of SOM
11. Feature extraction for a given real world problem.
12. Data Optimization using GA.

Text Books

Reference Books

Additional Reading

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

1. To Understand a number of models for supervised, unsupervised, and reinforcement neural networks systematically.
2. student should able to analyze different algorithms according to the properties of their inputs and outputs using different types of big data.
3. Design the most appropriate neural network for classification, Clustering, automatic detection and optimization.
4. Implement the algorithms in a software environment using MATLAB / Neural ware Professional and R-Programming Language.
5. To evaluate the neural network algorithms.
CS431THL: Machine Learning

Credits: 4

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

Unit 1: Introduction to Machine Learning (8 Hours)
Types of Learning: Supervised, Unsupervised, Reinforcement.

Unit 2: Decision Trees (5 Hours)
Decision Tree Learning: representation, Basic decision tree learning algorithm, Hypothesis space, Issues in decision tree learning.

Unit 3: Machine Learning Algorithms (8 Hours)
SVM: Kernel functions, Linear SVM, Nonlinear SVM.
Hidden Markov model, Genetic algorithm, Regression analysis, Multivariable regression.

Unit 4: Clustering Algorithm and recurrent Networks (8 Hours)
k-means algorithm, k-nearest neighbor learning, weighted majority algorithm, Hopefield Net, Hamming net, Maxnet, Kohonen self-organizing map, Principal component Analysis (PCA).

Unit 5: Bayesian Learning (6 Hours)
Bayes theorem, Maximum likelihood hypothesis, minimum description length principle, Gibbs algorithm, Bayesian belief networks, Expectation maximization algorithm.

Unit 6: Evaluating and Validating Hypothesis (5 Hours)
Evaluating hypothesis accuracy, Sampling theory, Central limit theorem, hypothesis testing, comparing learning algorithms.
Validation: Cross validation, Confusion matrix.

List of Practical:

1. Introduction to Matlab

Vishwakarma Institute of Technology  Issue 01 : Rev No. 1 : Dt. 24/03/17

1. Plot trigonometric functions: sin, cos, tan, cosec, sec, cot
2. Perform matrix operations: add, subtract, multiply, inverse
3. Perform following operation on image: change pixel value and display new image
4. Plot exponential function

2. Supervised learning
   a. Implement Mc-Culloch-Pitts Neural Network
   b. Implement Hebbian network

3. Implement Backpropagation algorithm

4. Concept Learning
   a. Implement Find-S algorithm
   b. Implement List-then-eliminate algorithm
   c. Implement Candidate Elimination algorithm

5. Support Vector Machine
   - Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy.

6. Genetic algorithm
   a. Implement Genetic algorithm for the Travelling salesman problem
   b. Implement Genetic algorithm for the 8 queens problem

7. Principal Component Analysis
   - Apply PCA on a data set obtained from UCI ML repository

8. Clustering
   - Implement the K-means algorithm on a data set obtained from UCI ML repository

Text Books

Reference Books

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

1. Demonstrate knowledge learning algorithms and concept learning through implementation for sustainable solutions of applications.

2. Evaluate decision tree learning algorithms.

3. Analyze research based problems using Machine learning techniques.

4. Apply different clustering algorithms used in machine learning to generic datasets and specific multidisciplinary domains.

5. Formulate a given problem within the Bayesian learning framework with focus on building lifelong learning ability.

6. Evaluation of different algorithms on well formulated problems along with stating valid conclusions that the evaluation support.
CS432THL: Software Design Methodologies (THL)

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

Unit 1: Business Process Management (7 Hrs)

Unit 2: System Behavior Specification (7 Hrs)
Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Building Domain Model, Capturing system behavior in use cases
Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization, Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Unit 3: Software Design Engineering Primitives (7 Hrs)

Unit 4: System Design Specification (7 Hrs)
Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes: entity, boundary and control, Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages,
Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors

**Unit 5: Design Patterns**

**Unit 6: Model Driven Development**

**List of Practical:**

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


3. To create Business Process Diagrams for all the scenarios identified using BPMN 2.0 and BPM practices. Process modeling captures the ordered sequence of activities within a process along with supporting information from end to end. In process modeling, the business process is framed in a BPD to reflect the activities, the roles that conduct those activities, conditional branching, and the sequence of the workflow between the activities.

4. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.
   a. The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
   b. Actors (External Users)
c. Transactions (Use Cases)
d. Event responses related to transactions with external agents.
e. Detection of System boundaries indicating scope of system.

5. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
   a. Discrete, distinguishable entities (class).
   b. Events (Individual stimulus from one object to another).
   c. Conditional events and relationship representation.

6. To depict the state transition with the life history of objects of a given class model. The model should depict:
   a. Possible ways the object can respond to events from other objects.
   b. Determine of start, end, and transition states.

7. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges.
   - A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow of control constructs, such as synchronization, decision, and concurrency control.
   - Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.

8. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
   a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
   b. Identification of objects and their purpose.
   c. Roles / responsibilities entities that determine system behavior.

9. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

10. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.

Text Books:

Reference Books:

Course Outcomes:

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

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using business process management practices, object-oriented analysis principles and Model Driven Development practices.
2. Identify and formulate software requirements and behavioral models using static and dynamic behavioral views indicating structured problem partitioning and state-based exploration.
3. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices.
4. Construct and justify the evolutionary system description models expressing high-level architecture accommodating applicable architectural styles compatible to requirements and behavioral models using UML-supported modeling tools.
5. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and relation to other design criteria resulting in well-documented system profiles to the engineering and social community.
6. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices.
CS437THL: Software Testing and Quality Assurance (THL)

Credits: 04

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

Unit 1: Software Measurement (7 Hrs)

Unit 2: Principles of Testing (7 Hrs)

Unit 3: Functional Testing (7 Hrs)

Unit 4: Higher Order Testing (7 Hrs)

Unit 5: Introduction to Software Quality Assurance

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


List of Practical:
1. To Prepare Test Plan for the implemented system under test. The Test Plan shall be based on System Requirement Specification. The Test plan consists of following issues.
   a. Purpose of the test.
   b. Location and schedule of the test.
   c. Test descriptions.
   d. Pass and Fail Criteria.
2. To identify and narrate Test cases, Test scripts/procedures and Test incident Report identifier for the system under test. Refer Use case analysis document to prepare mentioned/identified test documents. The expected specifications/behaviors can be stated with the help of Test Oracle.
3. To perform Unit testing especially indicating the traced Independent data paths, Control paths and Error handling paths. Prepare control flow graphs for the unit under test. Compute the Cyclomatic complexity of the unit.

4. To perform Data Flow testing for the Program Segments by identifying the Definition-Use chain and type of data flow anomaly.

5. To perform Mutation Analysis of the Program Segments along with mutant history, mutation score and type of mutation by using any Code analysis Tool / Mutation Testing Tool (JUNIT, MuJava).

6. To perform Black-Box Testing for all the units contained in the architectural segments using Equivalence Partitioning, Boundary Value Analysis and Orthogonal Array testing methods. To study exploratory Testing for the Module under Test and merits/demerits of this technique.


8. To perform Automated Testing using suitable CASE tool addressing Higher-Order testing strategies.


10. To perform Software Audit (Checklist and Template-based) for the software developed and improve the Code Quality.

Text Books:


Reference Books:


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.
ELECTIVE List
with Project
CS420THP: Cloud Computing

Credits: 3

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

Unit 1: Introduction to Cloud Computing (6 Hours)

Unit 2: Resource Virtualization (6 Hours)

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, MongoDB, 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
List of Project
1. Implement application using NoSQL
2. Implement application using Map-Reduce
3. Implement SaaS application and host it on Cloud Platform

Text Books:

Reference Books:
1. Barrie Sosinsky, “Cloud Computing Bible”, Wiley India

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
CS422THP: Distributed Computing

Credits: 04

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

Unit 1: Introduction (8 Hours)
Introduction to Hadoop/MapReduce.

Unit 2: Communication (8 Hours)
Interprocess Communication: Communication primitives: Blocking/non-blocking, Synchronous/Asynchronous primitives, Message Oriented Communication, Stream Oriented Communication.
Distributed Objects: Remote Method Invocation, Java RMI

Unit 3: Synchronization (6 Hours)
Time and Global States: Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State.
Election Algorithm: Bully Algorithm, Ring Algorithm.
Mutual Exclusion: Requirements, Performance metrics, Centralized algorithm, Lamport’s algorithm, Distributed algorithm, Token Ring algorithm.

Unit 4: Distributed Transaction and Deadlock (6 Hours)
Distributed Transaction: Transaction Model, Classification, Implementation, Concurrency Control: Serializability, 2 Phase Locking, Strict 2 PL, Distributed Commit: 2 Phase Commit, Recovery.
Distributed Deadlock: Avoidance, Prevention, Detection: Classification of distributed deadlock detection algorithms, Centralized Approach, Hierarchical Approach, WFG Based Fully Distributed, Deadlock Recovery

Unit 5: Fault Tolerance (7 Hours)
Introduction to Fault Tolerance, Failure Models, Failure Masking by Redundancy: Triple Modular Redundancy.
Reliable Client Server Communication, Reliable Group Communication.

Unit 6: Distributed Shared Memory (5 Hours)

Case study: Google File System

List of Project areas:

1. Design client-server application using Java RMI/RPC.
2. Develop solution for Clock Synchronization, Election Algorithm and Mutual Exclusion in Distributed system.
3. Implementation of different deadlock detection algorithms in Distributed system.

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. Identify the basic principles, design issues and architectural aspects of distributed systems.
2. Analyze the different techniques used for Communication in distributed system.
3. Develop the solutions for Clock synchronization, Mutual exclusion in distributed system.
4. Construct an optimal and cost-effective solution for Distributed transaction and Deadlock.
5. Use and apply important methods in distributed systems to support Scalability and Fault tolerance.
6. Gain knowledge on Distributed File System and design issues of Distributed Shared Memory.
CS423THP: Ubiquitous Computing

Credits: 4

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

Unit 1: Introduction to Ubiquitous Computing (7 Hours)

Unit 2: Pervasive Computing Devices (7 Hours)
Smart Environment: CPI and CCI Smart Devices: Application and Requirements, Device Technology and Connectivity, Human Computer Interaction.

Unit 3: Human Computer Interaction (6 Hours)
Explicit HCI, Implicit HCI, User Interface and Interaction for four hand-held widely used devices, Hidden UI via basic smart devices, Hidden UI via wearable and Implanted devices, Human centered design, user models.

Unit 4: Middleware for Pervasive Computing (6 Hours)
Adaptive middleware, Context aware middleware, Mobile middleware, Service Discovery, Mobile Agents.

Unit 5: Security in Pervasive Computing (7 Hours)

Unit 6: Challenges and Outlook (7 Hours)
Overview of challenges, smart devices, Smart Interaction, Smart physical environment device interaction, Smart human-device interaction, Human Intelligence versus machine intelligence, social issues.
Case Study- Wearable Computing/ Cyber Physical System.

List of Project areas:
1. Context-aware computing
2. Proactive computing
3. Mobile and real-time data/media management
4. Multimedia data and sensing dissemination
5. Mobility management
6. Location-dependent query processing, and positioning.

Text Books:


Reference Books: (As per IEEE format)

Course Outcomes:
The student will be able to –
1. Describe the characteristics of pervasive computing applications including the basic computing application problems, performance objectives and quality of services, major system components and architectures of the systems.
2. Analyze the strengths, problems and limitations of the current tools, devices and communications for pervasive computing systems.
3. Recognize the different ways that humans will interact with systems in a ubiquitous environment and account for these accordingly.
4. List and exemplify the key technologies involved in the development Ubicomp systems.
5. Develop an attitude to identify and propose solutions for security and privacy issues.
6. Explore the trends and problems of current pervasive computing systems using examples.
CS425THP: GEOGRAPHICAL INFORMATION SYSTEMS

Credits: 4

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

Unit 1: GIS and Maps (6 Hours)

Unit 2: Remote Sensing Fundamentals (7 Hours)

Unit 3: Image Processing (7 Hours)

Unit 4: Spatial Data Modeling and Management (7 Hours)

Unit 5: Data Input, Quality and Analysis (7 Hours)

Unit 6: GIS Applications (6 Hours)

List of Project areas:

1. Prepare a map for the selected geographical area as per topological survey
2. Design a spatial database for the entities related with a geographical area specified in above map.
3. Study the connectivity of maps with spatial databases.
4. Analyze a case study of any GIS application of your choice.
5. Prepare a presentation on any latest GIS technology / technique / software / hardware.

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. Differentiate among map projections for geographical areas
2. Design spatial data model for geographical area encompassing related entities
3. Create database for geographical area encompassing related entities using GIS software
4. Propose effective remote sensing based solutions addressing requirements in domains such as urban planning, weather forecasting, defense and land management
5. Display the impact of data processes such as data input, quality, analysis and output in GIS applications
6. Build skill set required for responsible positions such as GIS Analyst, GIS Technician and GIS Specialist

FF No. : 654

CS427THP: Enterprise Systems(TPL)

Credits: 04

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

Unit 1: Business Process Management
( 7Hrs )
Process Orchestrations: Control Flow Patterns, Workflow Nets, Business Process Modeling Notation
Process Choreographies: Motivation and Terminology, Development Phases, Process Choreography Design and Implementation, Service Interaction Patterns
Properties of Business Processes: Data Dependencies, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness, Soundness Criteria Overview
Petri Nets, Event-driven Process Chains, Yet Another Workflow Language (YAWL), Graph-Based Workflow Language

Unit 2: SOA Fundamentals
( 7Hrs )
Service Contracts (Standardization and Design): Contracts principles, Types of Service Contract Standardization, Contracts and Service Design, Versioning, Technology/Development Tool Dependencies

Unit 3: SOA Design Principles
( 7Hrs )
Service Abstraction (Information Hiding and Meta Abstraction Types): Abstraction principles, Types of Meta Abstraction, Measuring Service Abstraction, Service Abstraction and Service Design, Risks Associated with Service Abstraction
Service Reusability (Commercial and Agnostic Design): Reuse Principle, Service Reuse in SOA, Service Reusability and Service Design
Service Autonomy (Processing Boundaries and Control): Autonomy Principle, Types of Service Autonomy, Measuring Service Autonomy, Service Contract Autonomy (services with normalized contracts) Autonomy and Service Design
Service Statelessness (State Management Deferral and Stateless Design): State Management, Measuring Service Statelessness, Stateless and Service Design

Unit 4: SOA Technology and Implementation (7Hrs)
Service Discoverability (Interpretability and Communication): Discoverability, Types of Discovery and Discoverability, Measuring Service Discoverability, Discoverability and Service
Service Composability (Composition Member Design and Complex Compositions): Composition, Composition Concepts and Terminology, Complex Service Composition, Measuring Service Composability and Composition Effectiveness Potential, Composition and Service Design, Service-Oriented Composition and Object-Orientation, Mapping Service-Oriented Principles to Strategic Goals, SOA Platforms, SOA support in .NET and J2EE platforms

Unit 5: Enterprise Architecture (7Hrs)

Unit 6: Enterprise Architecture Frameworks (7Hrs)

List of Practical:
2. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system along with identification of Business Processes and develop full detail Business Process diagrams.

3. To develop Domain-driven vocabulary of the target system indicating domain lexicon and context-based terminologies.

4. To identify and categorize the target system services with detailed service specifications modeled with component diagram incorporating appropriate architectural style and coupling.

5. To design the service layers and tiers modeled with deployment diagram accommodating abstraction, autonomy, statelessness and reuse.

6. To map the service levels and primitives to appropriate Strategies for data processing using XML / XQuery / JSON / JAXB.

7. To produce, invoke, compose Web Services using SOAP, WSDL and UDDI.

8. To implement and integrate the components of the target system using .NET / J2EE platforms adhering to Service specifications.

9. To create the balanced scorecard for the target system indicating the standards and principles applied.

Text Books:

Reference Books:
Course Outcomes:

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

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

Credits: 4

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

Unit 1: Digital Image Formation and low-level processing (7 Hours)
Overview and State-of-the-art, Fundamentals of Image Formation, Image Acquisition, Sampling, Quantization, Difference in Monochrome and Multichrome imaging, concept of color spaces Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Histogram Processing.

Unit 2: Feature Extraction and Image Segmentation (7 Hours)
Edges - Canny, Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.
Region Growing, Edge Based approaches to segmentation, Segmentation by Graph-Theoretic Clustering, Mean-Shift mode finding, Texture Segmentation; Active contours.

Unit 3: 3D vision. Geometry (6 Hours)
3D vision tasks, Basics of projective geometry, A single perspective camera, Scene reconstruction from multiple views, Two cameras, stereopsis, Three cameras and trifocal tensor, Shape from X, Full 3D objects, 3D model-based vision, 2D view-based representations of a 3D scene.

Unit 4: Motion analysis (6 Hours)
Differential motion analysis methods, Optical flow: Optical flow computation, Global and local optical flow estimation, Combined local-global optical flow estimation. Optical flow in motion analysis, Analysis based on correspondence of interest points, Detection of specific motion patterns, Video tracking: Background modeling, Kernel-based tracking, Object path analysis, Motion models to aid tracking, Kalman filters, Particle filters.

Unit 5: Probability and Bayesian decision theory (7 Hours)
Basics of probability and Distribution, Pattern recognition systems, design cycle, learning and adaptation. Case studies of Pattern recognition, Statistical and syntactic pattern recognition, Classification problem, classification error, Bayes minimum error classifier, Bayes minimum risk classifier, discriminant functions and decision surfaces. multidimensional case for distributions.

Unit 6: Parametric estimation and unsupervised learning (7 Hours)
Parametric estimation of probability density functions, non parametric estimation of probability density functions, Parzen windows, k-nearest neighbor classifier. Properties of linear classifiers, linearly separable training samples, perceptron criterion and algorithm, minimum squared error criterion, Fisher’s linear discriminant function, Unsupervised learning & Clustering, Stages in clustering.
List of Project areas:

1. Object recognition
2. Object Tracking
3. Video Analytics
4. Image Enhancement
5. 3D object modeling

Text Books:

Reference Books: (As per IEEE format)
2. R.O.Duda, P.E. Hart, G.G.Stork ,“Pattern Classification,” John Wiley and sons, 2004

Course Outcomes:
The student will be able to—
1. Perform low level image processing operations
2. Identify feature vectors for object detection purpose.
3. Apply the principles of projective geometry for 3D scene construction
4. Analyze the frame sequence to track the region of interest or event.
5. Apply probability theory to estimate classifier performance.
6. Describe the principles of parametric and non parametric classification methods.
CS429THP: Information Retrieval

Credits: 04

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

Unit 1: Introduction (5 Hours)
Introduction, Definition, Objectives, Search and Browse Capabilities; A Formal Document Representation, Characterization of IR Models; Text operations, Pre-processing, Porter's Stemming Algorithm, term weighting techniques, Zipf's law, Heap's law

Unit 2: Information Retrieval Models (7 Hours)

Unit 3: Query Processing and Retrieval Evaluation (7 Hours)
Digital libraries, Morphological, Lexical Analysis, Thesaurus Construction, Ontology. Retrieval Performance, Evaluation Measures for Ranked and Unranked Results Query Languages, Structural Queries, Relevance Feedback, Query Expansion

Unit 4: Indexing and Searching (7 Hours)
Automatic Indexing, Inverted Files, Fast Inversion (FAST-INV) Algorithm, Signature Files, Partitioning, Tries, Suffix Trees and Suffix Arrays, PAT Tree, Distributed Indexing, Index Compression.

Unit 5: Parallel, Distributed IR and Web Searching (7 Hours)
Parallel IR, Distributed IR, Index Construction, Search Engines, Browsing, Metasearchers, Searching using Hyperlinks, Crawling, Link Analysis, Architectures (Agents, Buses, Wrappers/Mediators), PageRank Algorithm, HillTop Algorithm

Unit 6: Multimedia IR (7 Hours)
Multimedia Data Modeling, Query Languages, A Generic Multimedia Indexing (GEMINI) Approach, One Dimensional Time Series, Two Dimensional Color Images, Automatic Feature Extraction, Operations on images, Motion detection, Object recognition, Automatic image annotation and retrieval, Audio, Graph, Video Retrieval Hashing Algorithms, Image Features and Similarity Functions

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

1. Text preprocessing
2. Building index structures
3. Query processing
4. Text search & ranking score

Text Books:
2. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, “Introduction to Information Retrieval”, Cambridge University Press

Reference Books:
3. C. J. Van Rijsbergen, “Information Retrieval”, Information Retrieval Group, University of Glasgow

Course Outcomes:
The student will be able to –
1. Apply text operations for formal document representation
2. Describe various information retrieval system architectures and models
3. Validate retrieval performance of an information retrieval system
4. Construct indexes using suitable techniques
5. Apply sequential search and pattern matching techniques
6. Illustrate working of parallel, distributed and multimedia information retrieval system
CS433THP:: Internet of Things

Credits: 04  Teaching Scheme:

Theory: 3 Hours / Week

Project Based Lab: 2 Hours/Week

Unit 1: Introduction of Internet of Things  (7 Hours)

Unit 2: Embedded suite for IoT  (7 Hours)
Physical device – Raspberry Pi Interfaces, Hardware requirement of Pi, Connecting remotely to the Raspberry Pi over the network using VNC, Image processing using Raspberry Pi, GPIO Basics, Controlling GPIO Outputs Using a Web Interface, – Programming, APIs / Packages, Arduino Interfaces, Beagle bone Interfaces,

Unit 3: Wireless Technologies For IoT  (7 Hours)

Unit 4: Cloud Analytics  (6 Hours)
Introduction to cloud computing, Role of Cloud Computing in IoT, Cloud-to-Device Connectivity, View of IoT– Ubiquitous IoT Applications, Introduction to Cloud Storage models and communication APIs, Web server – Web server for IoT, Python web application framework, Designing a RESTful web API

Unit 5: Resource Management In The Internet Of Things  (7 Hours)
Clustering, Software Agents, Clustering Principles in an Internet of Things Architecture, Design Guidelines, and Software Agents for Object Representation, Data Synchronization.
Identity portrayal, Identity management, various identity management models: Local, Network, Federated and global web identity, user-centric identity management, device centric identity management and hybrid-identity management

Unit 6: Internet of things Challenges  (6 Hours)
Vulnerabilities of IoT, Security, Privacy & Trust for IoT, Security requirements, Threat analysis, Use cases and misuse cases,
IoT Challenges: Mobility, Reliability, Scalability, Management, Availability, Interoperability, Resource Optimization & cost efficiency, Infrastructure Configuration & reconfiguration, IoT Overarching Challenges, Cloud data management, cloud data monitoring, Cloud data Exchange,

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. Learn the terminology, technology and its applications of IoT
2. Analyze Embedded suite widely used in IoT,
3. Describe the concept of M2M with necessary protocols
4. Understand the cloud storage for IoT applications.
5. Optimize resources for different IoT applications
6. Understand Real world IoT Design constraint
CS434THP: Randomized and Approximation Algorithms

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

Unit 1
Basic probability theory
Introduction to randomization in computation and some simple randomized algorithms.
Basic discrete probability theory: basic counting, definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes’ rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, probabilistic recurrences.
Uniform generation of combinatorial structures. Indicator random variables and their role in algorithm analysis.

Unit 2
Tail Inequalities and applications
Moments and deviation, occupancy problem, Markov and Chebyshev inequalities and some applications, randomized selection, weak law of large numbers, stable marriage problem and principle of deferred decision, coupon collector problem and sharp threshold, Chernoff’s bound and some applications, set balancing.

Unit 3
Randomized Algorithms and Randomized Complexity Classes.
Las Vegas and Monte-Carlo algorithms (with examples: randomized quick sort, Karger’s min-cut algorithm). Basic complexity classes P, NP, RP, Co-RP, ZPP, BPP and their interrelations, probability amplification in RP and BPP, randomness and nonuniformity, Adleman’s theorem. Yao’s min-max principle and lower bound for randomized computations.

Unit 4 Algebraic techniques
Polynomial identity testing, Schwartz-Zippel lemma and applications (with examples verifying matrix multiplication, testing equality of strings, perfect matching problem for bipartite graphs), Mulmuley-Vazirani-Vazirani isolation lemma and application to matching problem. Number theoretic algorithms (finding quadratic non-residues, primality testing), introduction to probabilistic methods.

Unit 5 Markov Chains and Random Walks
Markov chains: definition, representations, randomized algorithm for 2-SAT and 3-SAT, classifying states of Markov chains, Gambler’s ruin, stationary distributions. Random walks on undirected graphs, cover time, hitting time, commute time, graph connectivity, electrical networks, introduction to expander graphs. Expanders and rapidly mixing random walks.

Unit 6 Approximation Algorithms
Introduction to approximation algorithms, NP-hard optimization problems, lower bounding
OPT, Review of approximation algorithm for vertex cover, TSP. Example of set-cover (O(log n) factor approx-algorithm based on greedy strategy, layering), Shortest super-string
problem, Knapsack and FPTAS algorithms.
Linear programming based algorithms, LP relaxation, LP duality. LP rounding strategy and
primal-dual schema, set-cover and some other examples using LP based techniques,
maximum satisfiability.

Text Books: (As per IEEE format)
   9780521613903
   9780521835404

Reference Books: (As per IEEE format)
2. William Feller ”An introduction to probability theory and its applications.” ,Wiley India Pvt
   Ltd ISBN: 9788126518050

Course Outcomes:
The student will be able to –
1. Solve problems based on the basic discrete probability and combinatorics
2. Design Las-Vegas, Monte-Carlo randomized algorithms for various computational
   problems
3. Analyze time complexity and success probability of randomized algorithms using
   random variables.
4. Illustrate application of tail inequalities in tight estimation of the success probability
   and the time complexity of randomized algorithms
5. Explain role of advanced algebraic techniques such as Schwartz-Zippel Lemma,
   Isolation Lemma, Markov chains and random walks on graphs in randomized
   algorithms design
6. Design approximation algorithms for NP-complete problems using suitable paradigm
CS435THP: MANAGEMENT INFORMATION SYSTEMS

Credits: 04

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

Unit 1: Foundations of Information Systems (6 Hours)

Unit 2: Manufacturing and Service Systems (7 Hours)

Unit 3: e-Business (7 Hours)

Unit 4: Information Systems for Decision Support (7 Hours)

Unit 5: Challenges Ahead (7 Hours)
Unit 6: MIS Applications (6 Hours)

List of Project areas:
1. Consider any organization from any sector. Study its organizational structure and comment about it.
2. By giving examples, differentiate between operational, strategic and tactical level of management process and its effect on design information system.
3. Identify and evaluate the design considerations for the given information system.
4. Study testing and quality assurance strategies.
5. Identify cases of computer crime, hacking, and cyber theft with respect to given information system. Plan about how to avoid and deal with such kind of security threats.

Text Books:

Reference Books:

Course Outcomes:
The student will be able to –
1. Estimate the functional complexities in manufacturing and/or service sectors for implementation of Management Information Systems.
2. Design solutions differentiating management information systems based on their features and applicability.
3. Initiate ethically responsible behaviour as a professional.
4. Build a set of professional skills required for responsible positions such as System Analyst, Business Consultant and Information System Manager.
5. Respond positively to cultural, political, economical and organizational challenges during process of project management.
6. Follow required domain specific processes and standards for management information systems.
CS439THP: Modeling and Simulations

Credits: 4

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

Unit 1: Process of Modeling and Simulation (6 hours)
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 (7 hours)
Monte Carlo methods, Stochastic processes, Queuing theory: Little's Theorem and 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 (7 hours)
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 (7 hours)

Unit 5: M&S Applications and Awareness (7 hours)
Application areas: optimization, decision making support, forecasting, safety considerations, training and education.

Unit 6: Advanced Topics (6 hours)
Model scalability, Virtual Reality, Virtual Worlds, Intro to Rare Event Simulation, Intro to Parallel Discrete Event Simulation, PDES Challenges.

Text Books:

**Reference Books:**

**List of Projects:**

1. Develop a simple deterministic simulation to determine the loan tenure for Rs.X principal amount when the customer pays Rs.Y per month. Assume the fixed interest rate of 10% per year.

2. Develop a Monte Carlo simulation model for profit estimation before introducing a new product in the market. Consider the uncertainty in terms of sales, production costs, competitive pricing and other market dynamics.

3. Develop a discrete event simulation of a typical fast-food restaurant. Restaurant configuration, business factors and customer behavior factors should be tunable parameters. (SimPy)


5. Develop a parallel discrete event simulation for a network of routers using conservative event processing.

**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.
CS440THP::Data Acquisition System

Credits: 04
Teaching Scheme:
Theory: 3 Hours / Week
Project Based Lab: 2 Hours/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 A: Introduction 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)

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

Project Domains:

1. Project based on signal conditioning circuit for sensors.
2. Project based on temperature controller using LM35
3. Project based on IR sensor.
4. Project based on ARM controller and any sensor
5. Case study on PLC systems

Text Books:

Reference Books :

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.
CS442THP: HUMAN COMPUTER INTERACTION

Credits: 04

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

Unit 1: Introduction (6 Hours)

Unit 2: Principles and Models (7 Hours)

Unit 3: Design Process and Interaction Styles (7 Hours)

Unit 4: Evaluation Techniques and Interface Categories (7 Hours)

Unit 5: Documentation and Groupware (7 Hours)
Classification of Documents, Printed Manuals, Reading from Displays, Online Help, Tutorial, Error / Warning Messages, Groupware, Goals / Dimensions of Cooperation, Asynchronous Interactions, Synchronous Interactions, Online Communities, Communityware.

Unit 6: Miscellaneous (6 Hours)
List of Project areas:

1. Identify specialized users and related facilities for a selected product / system and make necessary suggestions for its improved accessibility design.

2. Design user persona for the users of selected product / system.

3. Conduct a contextual inquiry for selected product / system.

4. Design an interface prototype for selected product / system.

5. Evaluate an interface using usability evaluation technique.

Text Books:


Reference Books:


Course Outcomes:

The student will be able to –

1. Identify human factors and usability issues related with computing applications

2. Differentiate computing applications into categories based on human factors

3. Design a user interface by applying suitable design principles, models and usability guidelines

4. Integrate ethno-cultural and accessibility computing aspects into the user interface design

5. Display the impact of usability evaluation and testing in computing applications

6. Follow required processes and standards while designing user interfaces
CS436THP:: ADVANCED COMPUTER ARCHITECTURE

Credits: 04

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

Prerequisites: Computer Organization , Operating Systems, Microprocessor and Interfacing

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 IV (6 Hours)
Multiprocessor Architecture
UMA, NUMA, COMA MPP Processor. Loosely and Tightly coupled multiprocessors, characteristics of multiprocessors & multiprocessoring, 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
Unit VI  
Parallel Software Issues

(6 Hours)

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

Reference Books:

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.
CS452INT: SUMMER INTERNSHIP

Credits: 04

Teaching Scheme: - 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

<table>
<thead>
<tr>
<th>CO1</th>
<th>Analyse given problem statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>Create solutions to problems with the help of latest tools</td>
</tr>
<tr>
<td>CO3</td>
<td>Follow and maintain coding practices</td>
</tr>
<tr>
<td>CO4</td>
<td>Cope up with the dynamics of a diverse team</td>
</tr>
<tr>
<td>CO5</td>
<td>Adapt to changing work environments</td>
</tr>
<tr>
<td>CO6</td>
<td>Understand the importance of following technological guidelines and standards</td>
</tr>
</tbody>
</table>
<Course Code xxxx> Machine Learning

Credits: 4

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

Unit 1: Introduction to Machine Learning (8 Hours)
Types of Learning: Supervised, Unsupervised, Reinforcement.

Unit 2: Decision Trees (5 Hours)
Decision Tree Learning: representation, Basic decision tree learning algorithm, Hypothesis space, Issues in decision tree learning.

Unit 3: Machine Learning Algorithms (8 Hours)
SVM: Kernel functions, Linear SVM, Nonlinear SVM.
Hidden Markov model, Genetic algorithm, Regression analysis, Multivariable regression.

Unit 4: Clustering Algorithm and recurrent Networks (8 Hours)
k-means algorithm, k-nearest neighbor learning, weighted majority algorithm, Hopefield Net, Hamming net, Maxnet, Kohonen self organizing map, Principal component Analysis (PCA).

Unit 5: Bayesian Learning (6 Hours)
Bayes theorem, Maximum likelihood hypothesis, minimum description length principle, Gibbs algorithm, Bayesian belief networks, Expectation maximization algorithm.

Unit 6: Evaluating and Validating Hypothesis (5 Hours)
Evaluating hypothesis accuracy, Sampling theory, Central limit theorem, hypothesis testing, comparing learning algorithms.
Validation: Cross validation, Confusion matrix.
List of Project areas:
1. Implement following learning algorithms on a Concept learning task.
   a. Find-S algorithm
   b. List-then-eliminate algorithm
   c. Candidate Elimination algorithm
2. Apply the following machine learning algorithms on a dataset obtained from UCI ML repository.
   a. Support Vector Machine
   b. Genetic algorithm
3. Apply the following clustering algorithms on a dataset obtained from UCI ML repository.
   a. K-means
   b. PCA

Text Books

Reference Books

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

1. Demonstrate knowledge learning algorithms and concept learning through implementation for sustainable solutions of applications.
2. Evaluate decision tree learning algorithms.
3. Analyze research based problems using Machine learning techniques.
4. Apply different clustering algorithms used in machine learning to generic datasets and specific multidisciplinary domains.
5. Formulate a given problem within the Bayesian learning framework with focus on building lifelong learning ability.
6. Evaluation of different algorithms on well formulated problems along with stating valid conclusions that the evaluation support.
CSI44TH: Mobile Computing

Credits: 3
Teaching Scheme: Theory 3 Hours / 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)

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

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:


Reference Books: (As per IEEE format)

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.
CS445TH: Software Testing and Quality Assurance(THL)

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

Unit 1: Software Measurement  
( 7 Hrs )

Unit 2: Principles of Testing  
( 7 Hrs )

Unit 3: Functional Testing  
( 7 Hrs )

Unit 4: Higher Order Testing  
( 7 Hrs )


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)

Text Books:


Reference Books:

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.
CS447TH: Modeling and Simulations

Credits: 3
Teaching Scheme: 3 Hours / 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 Theorem and
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

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:**

**Reference Books:**

**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.

---


CS448TH::Data Acquisition System

Credits: 03

Teaching Scheme:
Theory 3Hrs/ Week

Unit 1 : Systems and Measurements
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
Part A:Introduction 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
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

Unit 5 : Relay Sequencer and Ladder diagrams
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
Text Books:

Reference Books:

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.
CS453TH:: CYBER SECURITY AND DIGITAL FORENSICS

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

Prerequisites: Computer Networks.

Unit 1:  

Unit 2:  

Unit 3:  

Unit 4:  

Unit 5:  

Unit 6:  
Text Books:


Reference Books:


Additional Reading


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.
CS454TH:: ADVANCED COMPUTER ARCHITECTURE

Credits: 03

Teaching Scheme:
Theory: 3 Hours / Week

Prerequisites: Computer Organization, Operating Systems, Microprocessor and Interfacing

Unit I
Overview of Parallel Processing
(6 Hours)
Parallelism in Uniprocessor system, Evolution of parallel processors, Architectural Classification, Flynn’s, Feng’s, handlers classification. Pipeline, Vector, Array, Bernstein’s’ condition for parallelism, dependence graphs. Multicore architecture- GPU processor and programming model.

Unit II
Pipeline Architecture
(8 Hours)
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 IV
Multiprocessor Architecture
(6 Hours)
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
Multithreaded Architecture
(6 Hours)
Unit VI  
Parallel Software Issues  
(6 Hours)

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

Reference Books:

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.
CS455TH: Cloud Computing

Credits: 3

Teaching Scheme:
Theory: 3 Hours / Week

Unit 1: Introduction to Cloud Computing (6 Hours)

Unit 2: Resource Virtualization (6 Hours)

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, MongoDB, 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:

Reference Books:
1. Barrie Sosinsky, “Cloud Computing Bible”, Wiley India

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