ACADEMIC REGULATIONS

COURSE STRUCTURE AND DETAILED SYLLABUS

for

M.TECH (AR-16)

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

(Effective for the students admitted into first year from A.Y 2016-2017 and onwards)

ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AUTONOMOUS)

Approved by AICTE,

Recognized under 2(f) 12(b) of UGC,

Permanently Affiliated to JNTU Kakinada.

K.Kotturu, Tekkali, Srikakulam – 532201, Andhra Pradesh.
Vision of the Institute

To evolve into a premier engineering institute in the country by continuously enhancing the range of our competencies, expanding the gamut of our activities and extending the frontiers of our operations.

Mission of the Institute

Synergizing knowledge, technology and human resource, we impart the best quality education in Technology and Management. In the process, we make education more objective so that the efficiency for employability increases on a continued basis..

Vision of the Department

To become a pioneer in providing high quality education and research in the area of computer science and engineering.

Mission of the Department:

M1: Enrich society and advance computer science and engineering by preparing graduates with the knowledge, ability, and skill to become innovators and leaders who are able to contribute for the aspirations of the country and society.

M2: Benefit humanity through research, creativity, problem solving, and application development.

M3: Share knowledge and expertise to benefit the country, the region, and beyond while inspiring people to engage in computing fields.
Programme Educational Objectives (PEO’S):

PEO1: Use technology and knowledge to analyze, design and evaluate high-end computing systems for solving society problems.
PEO2: Engage in research or become successful entrepreneur.
PEO3: Engage in lifelong learning through collaborative work and ethical practices.

Program Outcomes (PO’s):

After successful completion of the Program, the graduates will be able to

PO1: Apply knowledge of computing, mathematics, science and engineering to real life problem solving.
PO2: Design, implements, and evaluate a computer-based system, process, component, or program to meet desired needs.
PO3: Apply design and development principles of software and/or hardware systems of varying complexity.
PO4: Use current techniques, skills, and tools necessary for computing practice.
PO5: Continuously update their knowledge on contemporary issues and present / express ideas in impressive and professional manner.
PO6: Understanding the impact of computer science and engineering solutions in the Societal and human context.
PO7: Become entrepreneur based on societal needs.
PO8: Understand professional, ethical, legal, security and social issues and responsibilities.
PO9: Function effectively on teams to accomplish a common goal.
PO10: Recognize need for and engage in continuing professional development through Life Long Learning.
PO11: Explore research gaps, analyze and carry out research in the specialized/emerging areas.
Program Specific Outcomes (PSO’s)

After successful completion of the Program, the graduates will be able to

**PSO1:** Model, design and develop robust computer applications by applying tools and techniques.

**PSO2:** Apply knowledge in various domains to scatter the needs of society through research.
Academic Regulations for M.Tech

(Effective for the students admitted into first year from academic year 2016-2017 and onwards)

The M.Tech Degree of the Aditya Institute of Technology and Management (Autonomous), Tekkali shall be conferred on candidates who are admitted to the program and fulfill all the requirements for the award of the Degree.

1.0 ELIGIBILITY FOR ADMISSIONS

Admission to the above program shall be made subject to the eligibility, qualifications and specialization prescribed by the University from time to time. Admissions shall be made on the basis of merit rank obtained by the qualifying candidate in GATE / PGCET, subject to reservations prescribed by the Govt. of AP from time to time.

2.0 AWARD OF M. Tech DEGREE:

2.1 A student shall be declared eligible for award of the M.Tech degree, if he/she pursues a course of study and completes it successfully in not less than two academic years and not more than four academic years and registered for 80 credits and he/she must secure total 80 credits.

2.2 A student, who fails to fulfill all the academic requirements for the award of the degree within four academic years from the year of his/her admission, shall forfeit his/her seat in M.Tech course.

2.3 The minimum instruction for each semester 95 clear instruction days.

3.0 ATTENDANCE

3.1 A candidate shall be deemed to have eligibility to write End Semester examinations if he/she has put in a minimum of 75% of attendance in aggregate of all the subjects.

3.2 Condonation of shortage of attendance up to 10% (65% and above, and below 75%) may be given by the College academic committee.
3.3 Condonation of shortage of attendance shall be granted only on genuine and valid reasons on representations by the candidate with supporting evidence.

3.4 Shortage of attendance below 65% shall in no case be condoned.

3.5 A candidate shall not be promoted to the next semester unless he/she fulfills the attendance requirements of the present semester.

3.6 A stipulated fee shall be payable towards condonation of shortage of attendance.

4.0 COURSE OF STUDY:

The following specializations are offered at present for the M.Tech course of study.

<table>
<thead>
<tr>
<th></th>
<th>Digital Electronics and Communication Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>VLSI System Design</td>
</tr>
<tr>
<td>3</td>
<td>Power Electronics and Drives</td>
</tr>
<tr>
<td>4</td>
<td>Computer Science and Engineering</td>
</tr>
<tr>
<td>5</td>
<td>Structural Engineering</td>
</tr>
<tr>
<td>6</td>
<td>Thermal Engineering</td>
</tr>
</tbody>
</table>

5.0 EVALUATION

The performance of the candidate in each semester shall be evaluated subject-wise, with a Maximum of 100 marks for theory and 100 marks for Laboratory, on the basis of Internal Evaluation and End Semester Examination.

5.1 For the theory subjects 60 marks shall be awarded based on the performance in the End Semester Examination. Out of 40 internal marks 30 marks are assigned for subjective exam, 5 marks for assignments and 5 marks for seminars. The internal evaluation for 30 marks shall be made based on the average of the marks secured in the two Mid Term-Examinations conducted, one in the middle of the Semester and the other immediately after the completion of instruction. Each midterm examination shall be conducted of duration of 120 minutes and question paper shall contain 4 questions. The student should answer all 4 questions.

5.2 For practical subjects, 60 marks shall be awarded based on the performance in the End Semester Examinations. Out of 40 internal marks 20 marks are assigned based on day to day evaluation and 20 marks are assigned based on the internal test.

5.3 There shall be a technical seminar presentation during 3rd semester. For technical seminar, a student under the supervision of a faculty member, shall collect the literature on a topic and critically review the literature and submit it to the Department in a report form and shall make an oral presentation.
before the Departmental Committee. The Departmental Committee consists of Head of the Department, supervisor and two other senior faculty members of the department. For technical seminar there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% to be declared successful.

5.4 A candidate shall be deemed to have secured the academic requirement in a subject if he/she secures a minimum of 40% of marks in the End Examination and a minimum aggregate of 50% of the total marks in the End Semester Examination and Internal Evaluation taken together.

5.5 In case the candidate does not secure the minimum academic requirement in any subject (as specified in 5.4) he has to reappear for the supplementary Examination in that subject.

5.6 The viva-voce examination shall be conducted at the end of the course work after pass in all subjects.

5.7 Laboratory examination for M.Tech courses must be conducted with two Examiners, one of them being Laboratory Class Teacher and second examiner shall be external examiner.

6.0 EVALUATION OF PROJECT/DISSERTATION WORK:

Every candidate shall be required to submit thesis or dissertation after taking up a topic approved by the Project Review Committee.

6.1 A Project Review Committee (PRC) shall be constituted with Principal as chair person, Head of the department and two other senior faculty members of the concerned department (one will be the guide).

6.2 Registration of Project Work: A candidate is permitted to register for the project work after satisfying the attendance requirement of all the subjects (theory and practical subjects).

6.3 After satisfying 6.2, a candidate has to submit, in consultation with his project supervisor, the title, objective and plan of action of his project work (Based on a publication in a Peer Reviewed Journal) to the Project Review Committee for its approval before the second semester end examinations. After obtaining the approval of the Committee the student can initiate the Project work after the second semester end examinations.

6.4 Every candidate shall work on projects approved by the PRC of the college.
6.5 If a candidate wishes to change his supervisor or topic of the project he can do so with approval of the PRC. However, the Project Review Committee (PRC) shall examine whether the change of topic/supervisor leads to a major change of his initial plans of project proposal. If so, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.

6.6 A candidate shall submit status report in two stages at least with a gap of 3 months between them.

6.7 The work on the project shall be initiated in the beginning of the second year/III semester and the duration of the project is for two semesters. The candidate shall identify the problem, Literature survey, design/modeling part of the problem i.e. almost 35% of his dissertation/project work should complete in the III semester itself and it will be evaluated by PRC. If the candidate fails to get the satisfactory report, he has to re-register for the project/dissertation work.

6.8 A candidate shall be allowed to submit the project report only after fulfilling the attendance requirements of all the semesters with approval of PRC and not earlier than 40 weeks from the date of registration of the project work. For the approval of PRC the candidate shall submit the draft copy of thesis to the Principal (through Head of the Department) and shall make an oral presentation before the PRC.

6.9 The Candidate may be permitted to submit the Project Report if only the work is Published/Accepted to be Published in a Journal / International conference of repute and relevance.

6.10 Three copies of the Project Thesis certified by the supervisor shall be submitted to the College/School/Institute.

6.11 The thesis shall be adjudicated by external examiner from outside the college.

6.12 The viva-voce examination shall be conducted by a board consisting of the supervisor, Head of the Department and the examiner outside the college who adjudicated the Thesis.

6.13 The student has to clear all the subjects of M.Tech course before submission of the project thesis/dissertation

The Board shall jointly report candidates work as:

A. Excellent
B. Good
C. Satisfactory
D. Unsatisfactory

Head of the Department shall coordinate and make arrangements for the conduct of viva-voce examination. If the report of the viva-voce is unsatisfactory, the candidate will retake the viva-voce examination after three months. If he fails to get a satisfactory report at the second viva-voce examination, the candidate may be asked to submit a new project proposal to PRC starting with 6.5
7. Method of Awarding Letter Grades and Grade Points for a Course

A letter grade and grade points will be awarded to a student in each course based on his/her performance as per the grading system given below.

Table: Grading System for M.Tech. Programme

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade Points</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>95-100%</td>
<td>10</td>
<td>O</td>
</tr>
<tr>
<td>85-&lt;95%</td>
<td>9</td>
<td>A+</td>
</tr>
<tr>
<td>75-&lt;85%</td>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>65-&lt;75%</td>
<td>7</td>
<td>B+</td>
</tr>
<tr>
<td>55-&lt;65%</td>
<td>6</td>
<td>B</td>
</tr>
<tr>
<td>50-&lt;55%</td>
<td>5</td>
<td>P</td>
</tr>
<tr>
<td>&lt; 50%</td>
<td>0</td>
<td>F (Fail)</td>
</tr>
</tbody>
</table>

7.1 Calculation of Semester Grade Points Average (SGPA)* for semester

The performance of each student at the end of each semester is indicated in terms of SGPA. The SGPA is calculated as below:

\[
SGPA = \frac{\sum (CR \times GP)}{\sum CR} \quad \text{(for all courses passed in semester)}
\]

Where CR = Credits of a Course

GP = Grade points awarded for a course

*SGPA is calculated for the candidates who passed all the courses in that semester.
7.2 Calculation of Cumulative Grade Points Average (CGPA*) and Award of Division for Entire Programme.

The CGPA is calculated as below:

\[
\text{CGPA} = \frac{\sum (CR \times GP)}{\sum CR} \quad \text{(for entire programme)}
\]

Where
- \(CR\) = Credits of a course
- \(GP\) = Grade points awarded for a course

*CGPA is calculated for the candidates who passed all the courses till that semester.

Table: Award of Divisions

<table>
<thead>
<tr>
<th>CGPA</th>
<th>DIVISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥ 7.5</td>
<td>First Class with distinction</td>
</tr>
<tr>
<td>≥ 6.5 and &lt; 7.5</td>
<td>First Class</td>
</tr>
<tr>
<td>≥ 5.5 and 6.5</td>
<td>Second Class</td>
</tr>
<tr>
<td>≥ 5.0 and &lt; 5.5</td>
<td>Pass Class</td>
</tr>
<tr>
<td>&lt; 5.0</td>
<td>Fail</td>
</tr>
</tbody>
</table>

After a student has satisfied the requirements prescribed for the completion of the programme and is eligible for receiving the award of M.Tech. Degree, he shall be placed in one of the above divisions.

8.0 WITH-HOLDING OF RESULTS:

If the candidate has not paid any dues to the college or if any case of indiscipline is pending against him / her, the result of the candidate will be withheld and he/she will not be allowed into the next higher semester. The issue of the degree is liable to be withheld in such cases.
9.0 TRANSITORY REGULATIONS:

Candidate who have discontinued or have been detained for want of attendance or who have failed after having undergone the course are eligible for admission to the same or equivalent subjects as and when subjects are offered, subject to 5.5 and 2.0

10.0 GENERAL:

10.1 The academic regulations should be read as a whole for purpose of any Interpretation.

10.2 In case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Principal is final.

10.3 The Institute may change or amend the academic regulations and syllabus at any time and the changes and amendments made shall be applicable to all the students with effect from the date notified by the college.

10.4 Wherever the word he, him or his occur, it will also include she, her and hers.

* * * * *
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI.
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

COURSE STRUCTURE

I Year I Semester

<table>
<thead>
<tr>
<th>S.No.</th>
<th>CODE</th>
<th>COURSE</th>
<th>L</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16MCS1001</td>
<td>Data Structures and Algorithms</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>16MCS1002</td>
<td>Database Management Systems</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>16MCS1003</td>
<td>Computer Networks</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>16MCS1004</td>
<td>Operating Systems</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>16MCS1005</td>
<td>Object Oriented Programming</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Elective – I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1006</td>
<td>Advanced Computer Architecture</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>16MCS1007</td>
<td>Compiler Design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1008</td>
<td>Artificial Intelligence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1009</td>
<td>Bio-Informatics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16MCS1101</td>
<td>Systems Lab</td>
<td>-</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>24</td>
<td>4</td>
<td>26</td>
</tr>
</tbody>
</table>

I Year II Semester

<table>
<thead>
<tr>
<th>S.No.</th>
<th>CODE</th>
<th>COURSE</th>
<th>L</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16MCS1010</td>
<td>Data Mining and Knowledge Discovery</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>16MCS1011</td>
<td>Software Engineering</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>16MCS1012</td>
<td>Object Oriented Analysis &amp; Design</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>16MCS1013</td>
<td>Web Technologies</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Elective – II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1014</td>
<td>Mobile Computing</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>16MCS1015</td>
<td>Cryptography &amp; Network Security</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1016</td>
<td>Machine Learning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1017</td>
<td>Cloud Computing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Elective – III</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1018</td>
<td>Soft Computing</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>16MCS1019</td>
<td>Simulation &amp; Modeling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1020</td>
<td>Parallel Computing &amp; Algorithms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16MCS1021</td>
<td>Digital Image Processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16MCS1102</td>
<td>Object Oriented Analysis &amp; Design Lab</td>
<td>-</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>16MCS1103</td>
<td>Web Technologies Lab</td>
<td>-</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>24</td>
<td>8</td>
<td>28</td>
</tr>
</tbody>
</table>
II Year I Semester

<table>
<thead>
<tr>
<th>S.No.</th>
<th>CODE</th>
<th>COURSE</th>
<th>L</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16MCS2201</td>
<td>Technical Seminar</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>16MCS2202</td>
<td>Project Work Phase-1</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>

II Year II Semester

<table>
<thead>
<tr>
<th>S.No.</th>
<th>CODE</th>
<th>COURSE</th>
<th>L</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16MCS2203</td>
<td>Project Work Phase-2</td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>14</td>
</tr>
</tbody>
</table>

L-Lecture hours/Week;  P-Practical Hours/Week;  C-Credits;
Course Objectives
The objective of this course is to make the student:

- Understand the properties of various data structures.
- Implement the main data structures: Lists, stacks, queues, trees, graphs and hash tables; and use them to solve computational problems.
- Analyze and use the main search, sorting and traversal algorithms in data structures.
- Design and implement data structures to efficiently solve problems.
- Practice methods of data structures and algorithm design.

Course Outcomes
Upon completion of the COURSE, students will be able to:

1. Analyze the efficiency of the designed algorithms, and apply abstractions.
2. Identify the strengths and weaknesses of different search and sorting algorithms;
3. Design and employ appropriate data structures and algorithms for implementing dictionaries;
4. Develop algorithms for non-linear data structures.
5. Analyze and Design algorithms using Greedy and Divide and Conquer strategies to solve problems.
6. Solving searching problems by designing efficient algorithms using Dynamic Programming, Branch and Bound Method and Back tracking strategies;

Unit I:
Introduction to data structures, Algorithm Analysis – performance analysis – time complexity and space complexity. Singly linked lists, doubly linked lists, circular list – Algorithms.
Stacks and queues: algorithms implementation using linked list. The list ADT, Stack ADT, Queue ADT.

Unit II:
Searching – Linear and binary search methods
Internal Sorting: Shell Sort, Heap Sort, Merge Sort, Quick Sort, Radix Sort.
External Sorting: Multi-way Merge, Poly-phase Merge.
Sorting Methods: Order Statistics: Lower Bound on Complexity for Sorting Methods: Lower Bound on Worst Case Complexity, Lower Bound on Average Case Complexity, Heap Sort, Quick Sort, Radix Sorting, Merge Sort.

Unit III:
Dictionaries: Sets, Dictionaries.

Unit IV:
Trees – Binary trees, properties, representation and traversals, expression tree.
Search Trees: Binary Search Trees. AVL Trees: Insertions and Deletions, B- Trees
Graphs – basic concepts, Operations, storage structures and traversals.

Graph Algorithms: Minimum-Cost Spanning Trees – Prim’s Algorithm, Kruskal’s Algorithm.

Shortest Path Algorithms: Dijkstra’s Algorithm.

All Pairs Shortest Paths Problem: Floyd’s Algorithm, Warshall’s Algorithm.

Unit V:

Unit VI:
Dynamic Programming Method - General method, All pairs shortest path problem, Optimal Binary Search Trees, 0/1 Knapsack problem, Traveling salesman problem,
Back tracking Method - General Method, 8-Queens Problem, Sum of subsets, Graph coloring, Hamiltonian cycle,
Branch and Bound Method - General Method, 0/1 Knapsack problem, Traveling salesman.

Text Books :

Reference Books:
Database Management Systems

Credits : 4
Course Code: 16MCS1002

Course Objectives:
• To introduce basic RDBMS Concepts, SQL, Database Design and Query processing. And also to introduce Transaction processing, issues and techniques relating to concurrency and recovery in multi-user Database Environments, Security and various Data Structures for External data storage and efficient retrieval.

Course Outcomes:
Students will be able to:
2. Interpret, Design and Implement an E-R Model.
3. Create/Modify the Structure and write optimized SQL Queries to extract and modify Information from Tables or Views.
4. Apply proper Techniques such as Normalization and analyze the applicability of a Specific Normal form in designing a Database.
6. Explain broad range of Database Management issues including Data integrity, Concurrency And Recovery, security.

Unit I:
Database System Applications; Database Systems versus File Systems; View of Data : Data Abstraction, Instances and Schemas; Data Models: The ER Model, Relational Model, Other Data Models; Database Languages: DDL, DML, Database Access from Application Programs; Database Users and Administrators; Transaction Management; Database System Structure: Storage Manager, the Query Processor. (Korth)

Unit II:
Database Design and ER diagrams: Beyond ER Design; Entities, Attributes and Entity sets; Relationships and Relationship sets; Additional features of ER Model; Conceptual Design with the ER Model; Introduction to the Relational Model; Integrity Constraint Over relations; Enforcing Integrity constraints; Querying relational data; Logical database Design: ER to Relational. Introduction to Views: Destroying/Altering tables and Views. (Ramakrishnan)

Unit III:
SQL: Queries, Constraints, Triggers: Over view; The Form of a Basic SQL Query; Nested Queries: Introduction to Nested Queries, Correlated Nested Queries, Set-Comparison Operators; Aggregative Operators; NULL values: Comparison using Null values; Logical connectives AND, OR, and NOT, Impact on SQL Constructs, Outer Joins, Disallowing NULL values; Complex Integrity Constraints in SQL; Triggers and Active Databases. (Ramakrishnan)

Unit IV:
Schema refinement and Normal forms: Problems Caused by Redundancy, Decompositions, Problem related to Decomposition; Functional Dependencies; Reasoning about FDS; FIRST, SECOND, THIRD Normal Forms, BCNF, Multivalued Dependencies, Fourth Normal form, Join.
Dependencies, Fifth Normal Form; Properties of Decompositions: Lossless join Decomposition, Dependency Preserving Decomposition; Schema refinement in Database Design (Ramakrishnan)

**Unit V:**
Transaction Concept; Transaction State; Implementation of Atomicity and Durability; Concurrent Executions; Serializability; Recoverability; Lock-Based Protocols; Locks, Granting of locks, 2PL, implementation of locking; Timestamp Based Protocols; Validation based Protocols; Multiple Granularity; Recovery System; Failure classification; Lock-Based Recovery; Shadow Paging; Recovery with Concurrent Transactions; Buffer Management; Failure with loss of nonvolatile storage (Korth)

**Unit VI:**
Data on External Storage; File Organization and Indexing: Cluster Indexes, Primary and Secondary Indexes; Index Data Structures: Hash Based Indexing, Tree based Indexing; Comparison of File Organizations; B+ Trees: A Dynamic Index Structure; Introduction to Database Security, Access Control, Discretionary Access Control, Mandatory Access Control (Ramakrishnan)

**Text Books:**

**References Books:**
1. https://www.coursera.org/course/db
4. C.J. Date, Introduction to Database Systems, Pearson Education
Computer Networks

Course Objectives:

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
- Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Course Outcomes:

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

1. Understand the hierarchical, layered structure of typical network architecture.
2. Describe the node-to-node data transfer and list the multiple access protocols.
3. Explain end-to-end routing of packets and splits long messages into smaller Units
4. Express the enhancements made to IPv4 by IPv6.
5. Explain end-to-end data transfer and integrity across the network and techniques to improve QoS
6. Outline the features and operations of the various end user application programs.

Unit I:
Introduction: Data Communications: components, data representation, data flow; Networks: distributed processing, network criteria, physical structures, network models, categories of network, interconnection of networks; The Internet: A brief history, The internet today; Protocols & Standards: Protocols, Standards, Standards organizations, Internet standards; Layered Tasks: sender, receiver, and carrier, Hierarchy; The OSI model: Layered architecture, peer to peer processes, encapsulation; Layers in OSI model: Physical layer, Data link layer, Network layer, Transport layer, Session layer, Presentation layer, Application layer; TCP/IP protocol suite: Physical and Data link layers, Network layer, Transport layer, Application layer, Addressing: Physical address, Logical address, Port address, Specific address (Text Book-1)

Unit II:

Unit III:
Network Layer: Design issues: Store and forward packet switching, Services provided to the transport layer, Implementation of connectionless service, Implementation of connection oriented service and Comparison of virtual circuit and datagram subnets.

Unit IV:
**IPV4 Address:** Address space, Notations, Classful Addressing, Classless Addressing, Network Address Translation (NAT). **IPV6 addresses:** Structure, Address Space. **Internetworking:** Need for network layer, internets as a datagram network, internet as a connectionless network; **IPV4:** Datagram, Fragmentation, Checksum, Options; **IPV6:** Advantages, Packet format, Extension headers, Translation from IPV4 to IPV6. *(Text Book-1)*

Unit V:
**Process to Process delivery:** Client/Server paradigm, Multiplexing and De-multiplexing, connectionless versus connection-oriented services, Reliable versus unreliable; **UDP:** Well known ports for UDP, User datagram, checksum, UDP operation, Uses of UDP; **TCP:** TCP services, TCP features, segment, A TCP connection, Flow control, Error control, Congestion control; **SCTP:** SCTP services, SCTP features, Packet format, An SCTP Association, Flow control, Error control. **Congestion control:** Open-loop congestion control, Closed-loop congestion control, Congestion control in TCP, Congestion control in frame relay; **Quality of Service:** flow characteristics, flow classes; **Techniques to improve QoS:** Scheduling, Traffic shaping, Resource reservation, Admission control. *(Text Book-1)*

Unit VI:
**Domain Name System:** the name space, resource records, name servers; **E-mail:** Architecture and services, The user agent, Message formats, Message transfer, Final delivery; **WWW:** Architecture overview, Static web documents, Dynamic web documents, Hypertext transfer protocol, Performance elements, The wireless web; **Multimedia:** Introduction to audio, Audio compression, Streaming audio, Internet radio, Voice over IP, Introduction to video, Video compression, Video on-demand, The MBone—the multicast backbone. *(Text Book-2)*

Textbooks:


Reference Books:

3. Ne _index1.html
Operating Systems

Credits: 4
Course Code: 16MCS1004

External Marks: 60
Internal Marks: 40

Course Objectives:

- To familiarize students with the UNIX environment
- Learn the fundamentals of shell scripting/programming
- Know memory management including virtual memory
- Summarize the full range of considerations in the design of file systems
- Understand inter-process communication tools.

Course Outcomes:

1. List the Basic Unix commands
2. Familiar with Write shell scripts to automate various tasks
3. Explain UNIX process structure and related system calls.
4. Describe different memory management techniques and signals
5. Discuss file system design tradeoffs
6. Familiarity with Interprocess Communication using pipes, shared memory, semaphores and message Queues.

Unit I:

Unix Utilities: Introduction to Unix file system, VI editor, file handling utilities, security by file permissions, process utilities, disk utilities, Networking commands, cp, mv, in, rm, unlink, mkdir, rmdir, du, df, mount, umount, find, umask, ulimit, ps, who, w, finger, telnet, rlogin. Text processing utilities and backup utilities detailed commands to be covered are: cat, tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, pate, join, tee, more, pg, comm, cmp, diff, tr, awk, tar, cpio.

Unit II:

Shell Programming: What is a shell, shell responsibilities, pipes and input redirection, Output redirection and here documents, the shell as programming Language shell variables, control structures and shell programming.

Unit III:


Unit IV:

Memory Management: contiguous memory allocation, paging, Segmentation, Demand paging, Performance of demand paging, Page Replacement, Page Replacement algorithm, Allocation of frames, Thrashing Signals:- Signal functions, reliable signals, Un reliable signals, kill, raise, alarm, pause abort, system, sleep functions.

Unit-V:

File System Interface and Implementation: the concept of a file, Access Methods, Directory structure, file sharing, protection, File system structure, file system implementation, directory implementation, allocation methods, free-space management.
Unit VI:
**Inter-process Communication:** Pipe, process pipes, the pipe call, parent-child process, named pipes: FIFOs, Semaphores, message queues and shared memory applications of IPC.

**Text Books:**

2. W. Richard Stevens Advanced Programming in Unix Environment, AWL/PHI.
3. W. Richard Stevens, Unix Network Programming, PHI.

**Reference Books:**

1. William Stallings, Operating Systems, 6/e, PHI/Pearson
2. Dhamdhere, Operating Systems, 2/e, TMH
3. N.B. Venkateswarlu, Advanced Unix Programming, BS Publications
Object Oriented Programming

Credits : 4
Course Code: 16MCS1005

Course Objectives
- Be able to explain the difference between object oriented programming and procedural programming
- Its main objective is to teach the basic concepts and techniques which form the object oriented programming paradigm
- Cover issues related to the definition, creation and usage of classes, objects and methods.
- Discuss the principles of inheritance and polymorphism and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces.

Course Outcomes
1. Illustrate the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, and encapsulation.
2. Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
3. Design and develop programs using packages and interfaces.
4. Develop the mechanism of exceptional handling and multithreading
5. Implements the concept of event handling and GUI interface using Java swing.

Unit I
Classes and Objects Concepts of classes and objects, class fundamentals Declaring objects, assigning object reference variables, introducing methods, constructors, usage of static with data and methods, usage of final with data, access control, this key word, garbage collection, overloading methods and constructors, parameter passing - call by value, recursion, nested classes and inner classes, exploring the String class.

Unit II
Inheritance Basic concepts, member access rules, usage of super key word, forms of inheritance, method overriding, abstract classes, dynamic method dispatch, using final with inheritance, the Object class. Packages and Interfaces Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Unit III
Exception Handling and Multithreading Concepts of Exception handling, types of exceptions, usage of try, catch, throw, throws and finally keywords, Built-in exceptions, creating own exception sub classes, Concepts of Multithreading, differences between process and thread, thread life cycle, creating multiple threads using Thread class, Runnable interface, Synchronization, thread priorities, inter thread communication, daemon threads, deadlocks, thread groups.

Unit IV Event Handling Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. AWT : Concepts of components, container, panel, window, frame, canvas, Font class, Color class and Graphics.
Unit V

Unit VI
Networking and Java Library Basics of Networking, Inetaddress, TCP/IP sockets, Datagrams, URL, URL connection, String handling, java.util, java.io and java.net packages.

Text Books:

Reference Books:
4. Iver Horton, Beginning in Java 2, Wrox Publications.
5. Somasundaram, Java, Jaico.
Advanced Computer Architecture
(Elective - I)

Credits : 4
Course Code: 16MCS1006

External Marks:60
Internal Marks:40

Course Objectives
This course helps:

- To get the knowledge on system performance dependence attributes and calculation of system throughput.
- To learn memory hierarchy concepts and how to improve the performance of cache memory.
- Understand how the coherence, inclusion and locality properties are satisfied in memory hierarchy.
- Understand the linear and nonlinear scheduling processes in pipelining.
- Distinguish the design of Shared Memory MIMD and Distributed Memory MIMD machines
- Understand the message passing system to avoiding the inconsistency in multiprocessors.

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

1. Infer knowledge on Hardware and System Design concepts.
2. Learn about memory hierarchy and performance of cache memory.
4. Justify identify permissible latencies and forbidden latencies for the given non-linear pipeline.
5. Distinguish between RISC and CISC architectures.
6. Design the input-output connections in an Omega Network using perfect shuffle method.

Unit – I
Parallel Computer: State of computing, Elements of modern computer, Flynn’s classification of parallel processors, System attributes to performance, Multiprocessors and Multicomputer, Shared memory multiprocessors, Distributed memory multiprocessors.

Unit – II
Memory Hierarchy Design: Basic memory hierarchy, Optimization of cache performance, Small and simple first level cache to reduce hit time and power, Way prediction to reduce hit time, Pipelined cache access to increase cache band width, Non-blocking cache to increase cache band width.

Unit – III
Design Space of Processors, Instruction set architectures, Characteristics of typical CISC and RISC architecture, Hierarchical memory technology, Inclusion, Coherence and locality.

Unit – IV
Linear Pipeline Processors: Asynchronous and synchronous models, Clocking and timing control, Speedup, Efficiency and Throughput. Non-Linear Pipeline Processors: Reservation and latency analysis problems, Collision free scheduling problems.
Unit – V
Multiprocessors and Multivector Computers: Inter connection structure-Crossbar switch and multiport memory, Multistage and combining network routing.
Multivector computers: Vector processing principles, Vector instruction types, Vector access memory schemes.

Unit – VI

Text Books:

2. John L. Hennessy & David A. Patterson Morgan Kufmann“Computer Architecture A quantitative approach”

Reference Books:

Compiler Design
(Elective - I)

Credits :4
Course Code: 16MCS1007

Course Objectives:
After completing this course, the student should be able to:

- Introducing students to the concepts and principles of compiler design.
- Providing students with basic understanding of grammars and language definition.
- Introducing students to the various phases of designing a compiler and CODE optimization techniques, machine CODE generation
- Introducing students to the various programming techniques and structures used in compiler construction.
- Providing students with practical programming skills necessary for constructing a compiler.

Course Outcomes:
After completing this course, the student should be able:

1. To use the knowledge of patterns, tokens & regular expressions for solving a problem.
2. To apply the knowledge of lex tool & yacc tool to develop a scanner & parser.
3. To develop program to solve complex problems in compiler
4. To write the new CODE optimization techniques to improve the performance of a program in terms of speed & space.
5. To employ the knowledge of modern compiler & its features.
6. To experiment the new tools and technologies used for designing a compiler

Unit I:
Overview of Compilation: Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, LEX lexical analyzer generator.

Unit II
Top down Parsing: Context free grammars, Top down parsing – Backtracking, LL (1), recursive descent parsing, Predictive parsing.

Unit III
Bottom up parsing: Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing, handling ambiguous grammar, YACC – automatic parser generator.

Unit IV
Semantic analysis: Intermediate forms of source Programs – abstract syntax tree, polish notation and three address CODEs. Attributed grammars, Syntax directed translation, Type checker.
Symbol Tables: Symbol table format, organization for block structures languages. Block structures and non block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.
Unit V

**CODE optimization:** Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, DAG representation.

**Data flow analysis:** Flow graph, global optimization, redundant sub expression elimination, Induction variable elements, live variable analysis, Copy propagation.

Unit VI

**CODE generation:** Object CODE forms, machine dependent CODE optimization, register allocation and assignment generic CODE generation algorithms, DAG for register allocation.

**Text Books:**

**Reference Books:**
1. Lex&yacc – John R. Levine, Tony Mason, Doug Brown, O’reilly
4. Louden, Thomson, Compiler Construction.
Artificial Intelligence
(Elective - I)

Credits : 4
Course Code: 16MCS1008

Course Objectives:

- To learn the difference between optimal reasoning Vs human like reasoning.
- To understand the notions of state space representation, exhaustive search, heuristic search along with the time and space complexities.
- To learn different knowledge representation techniques.
- To understand the applications of AI: namely Game Playing, Theorem Proving, Expert Systems, Machine Learning and Natural Language Processing.

Course Outcomes:

1. Possess the ability to formulate an efficient problem space for a problem expressed in English.
2. Possess the ability to select a search algorithm for a problem and characterize its time and space complexities.
3. Possess the skill for representing knowledge using the appropriate technique.
4. Possess the ability to apply AI techniques to solve problems of Game Playing, Expert Systems, Machine Learning and Natural Language Processing.

Unit I
Introduction: AI problems, foundation of AI and history of AI intelligent agents:
Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

Unit II

Unit III

Unit IV
Knowledge Representation & Reasons logical Agents, Knowledge–Based Agents, the Wumpus world, logic, propositional logic, Resolution patterns in propositional logic, Resolution, Forward & Backward, Chaining. First order logic. Inference in first order logic, propositional Vs first order inference, unification & lifts forward chaining, Backward chaining, Resolution.
Unit V
Planning–Classical planning problem, Language of planning problems, Expressiveness and extension, planning with state–space search, forward states spare search, backward states space search, Heuristics for stats space search, Planning search, planning with state space search, partial order planning Graphs.

Unit VI
Learning –Forms of learning, Induction learning, Learning Decision Tree, Statistical learning methods, learning with complex data, learning with Hidden variables –The EM Algorithm, Instance Based learning, Neural Networks.

Text Books:


Reference Books:

2. Artificial Intelligence and Expert Systems–Patterson PHI
Course Objectives:
The objective of the program in Bioinformatics is to prepare students for careers in academia and industry, and in particular to deepen their knowledge in both the biosciences and computational sciences to apply this knowledge to manage and analyze data in the life sciences and to train them in research.

In this course, we aim to cover the following:

- The concepts of computer science that relate to problems in biological sciences.
- Commercial and academic perspectives on bioinformatics.
- The impact of bioinformatics on the methodologies used in biological science.
- The influence biological science has on computing science.

Course Outcomes:
Graduates of the Bioinformatics program will demonstrate expertise in the following core competencies essential to success:

1. Extract information from different types of bioinformatics data (gene, protein, disease, etc.), including their biological characteristics and relationships
2. Employ different data representation models and formats used for bioinformatics data representation, including markup languages such as SBML and CellML, and ontologies such as GO ontology
3. Apply the different approaches used for data integration and data management, including data warehouse and wrapper approaches
4. Master computational techniques and diversified bioinformatics tools for processing data, including statistical, machine learning and data mining techniques
5. Analyze processed data with the support of analytical and visualization tools

Unit I

Unit-II
Special Topics In Bioinformatics: DNA mapping and sequencing, Map alignment

Unit III
Databases In Bioinformatics :Databases in Bioinformatics- Genbank, NCBI, EMBL, DDBJ, UniGene, SGD, EMI Genomes, -protein databases-PIR, SWISSPROT, TrEMBL, Prosite, PRINTS -structural databases-PDB, SCOP, CATH, PDB_SELECT, PDBSUM, DSSP, FSSP, DALI, PRODOM, protein families & pattern databases
Unit-IV
**Secondary Databases:** Introduction to Secondary Databases Organization and management of databases Swissprot, PIR, KEGG

Unit V
**Bio Chemical Data Bases:** Introduction to BioChemical databases-organization and Management of databases. KEGG, EXGESCY, BRENDA, WIT.

Unit VI
**File Formats:** File formats-raw/plain format-NCBI, Genbank flat file format-ASN.1, GCG, FASTA, EMBL, NBRF, PIR, swissprot sequence formats, PDB format, etc. - introduction to structure prediction methods.

**Text Books:**

**References:**
Systems Lab

Credits : 2
Course Code: 16MCS1101

Course Objectives:
Upon Completion of this course, students will be able to
- To educate students with fundamental concepts of Database Design, Data Models, Different Database Languages (SQL/PLSQL).
- Understand the scope of Block, Nested Blocks, and Labels
- To learn the fundamentals of shell scripting/programming.
- Introduce the student to Unix/Linux kernel programming techniques.
- Compare and contrast various inter-process communication facilities.

Course Outcomes:
After the completion of this course the student
1. Able to create database with different types of integrity constraints and use of SQL commands such as DDL, DML, DCL, TCL to access data from database objects.
2. Map the model into a relational database system.
3. Implement the given schema on a relational DBMS.
4. Develop advanced packages, stored procedures, triggers and functions using PL/SQL.
5. Classify system calls in UNIX
6. Analyze the concepts of process, threads and file structure
7. Implement IPC using pipes, semaphores, Shared Memory and messages.

List Of DBMS Experiments
1. Introduction to SQL, an exercise on data types in SQL & Data Definition Language Commands
2. Exercise on Data Manipulation Language and transaction control commands
3. Exercise on Types of Data Constraints.
4. Exercise on Joins (single-table or multi-table) and using normalization
5. Exercise on group-by clause and date arithmetic.
6. Exercise on different functions (aggregate, math and string)
7. Exercise on different types of sub queries
8. Introduction to PL/SQL, Control Structures, Procedures and Functions, view
9. Introduction to triggers and cursors.
10. Write a program that creates the function and calculating area of circle.
11. Write a program that uses the concept of user defined exception
List of OS Experiments

1. Simulate the following CPU scheduling algorithms
   a) Round Robin  b) SJF   c) FCFS   d) Priority
2. Simulate all file allocation strategies
   a) sequential   b) indexed c) Linked
3. Simulate bankers algorithm for dead lock prevention
4. Simulate page replacement algorithms
   a) FIFO   b) LRU   c) LFU
5. Write a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two divisions’ options: one returns the quotient and the other returns reminder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract(-s), multiply(-m), quotient(-c) and reminder(-r).
6. Write a C program that counts the number of blanks in a text file.
7. Implement in C the following UNIX commands using system calls.
   a) cat   b) ls   c) mv    d) grep    e) R login
8. Write a C program that illustrates how to execute two commands concurrently with a command pipe.
9. Write a C program that illustrates the creation of child process using fork system call.

Text Books:

1. William Stallings, Operating Systems, 6/e, PHI/Pearson
2. N.B. Venkateswarlu, Advanced Unix Programming, BS Publications
3. Ivan Bayross, SQL, PL/SQL programming language of oracle.

Reference Books:

2. Sharanam Shah, Vaishali Shah, Oracle for professionals.
Data Mining and Knowledge Discovery

Credits : 3
Course Code: 16MCS1010

Course Objectives:

- To introduce basic concepts, principles, major techniques and algorithms in Data Mining. These include concepts and techniques for data preprocessing, OLAP, association rule mining, data classification, and data clustering.
- To discuss Applications, Emerging Areas in Data Mining and the role of it in Society.

Course Outcomes:
Upon completion of this course, students will be able to

1. Recognize types of Data, Data Quality, need of preprocessing and different measures of similarity and dissimilarity.
2. Explain in detail major techniques and algorithms involved in data mining, including techniques and algorithms for data preprocessing, association rule mining, data classification, and data clustering.
3. Evaluate the performance of a classifier.
4. Compare and contrast different classification and clustering algorithms.
5. Apply data mining algorithms and techniques adaptively to real world problem solving.

Unit -I:
Introduction to Data Mining: Data Mining Tasks, Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity: Basics, Similarity and Dissimilarity between simple Attributes, Dissimilarities between Data Objects, Similarities between Data Objects, Examples of Proximity Measures, Issues in Proximity Calculation; Exploring Data: Data Set, Summary Statistics, OLAP and Multidimensional Data Analysis.

Unit -II:
Basic Classification: Preliminaries, General approach to solving a classification problem, Decision Tree induction, Model Overfitting: Due to presence of noise, due to lack of representation samples, Handling Overfitting in Decision Tree Induction; Evaluating the performance of classifier.

Unit -III:
Alternative Classification Techniques: Nearest Neighbor classifiers, Bayesian Classifiers, classification by Backpropagation(ANN), Support Vector Machines: Linear SVM, Separable and Non Separable case, Characteristics of SVM; Ensemble Methods.

Unit -IV:
Unit V:
**Extended Association Analysis:** Handling Categorical Attributes, Continuous Attributes, A Concept Hierarchy, Sequential Patterns, Subgraph Patterns.

Unit VI:
**Cluster Analysis:** Overview, the Basic K–means Algorithm, Additional Issues, Bisecting K-means, K-means and Different Types of Clusters, Strengths and Weaknesses, Basic Agglomerative Hierarchical Clustering Algorithm, specific techniques, Key Issues, Strengths and Weaknesses, DBSCAN Algorithm, Strengths and Weaknesses.

**Text Books:**
1. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, PEA.
2. Data Mining, Concepts and Techniques, 3/e, Jiawei Han, Micheline Kamber, Elsevier.

**Reference Books:**
1. GK Gupta, Introduction to Data Mining with Case Studies, Prentice Hall.
2. H Dunham, 2008 Data Mining: Introductory and Advanced Topics, Margaret, PEA.
Software Engineering

Credits : 4
Course Code: 16MCS1011

Course Objectives:

- Gives the basic knowledge in Software Engineering process, focusing on the different process models.
- Learn how to perform feasibility study of the projects under the requirement engineering process and system models.
- Categorize different design concepts and architecture styles, evaluating the steps for designing a good model.
- Highlight how technology changes in the last two decades has impacted the traditional approach.
- Identifies various testing strategies for software quality

Course Outcomes:

On completion of the course, students should be able to

1. Understand the engineering issues that form the background to develop complex and evolving software-intensive systems.
2. Apply an effective software engineering process, based on knowledge of widely used development lifecycle models.
3. Analyze and translate requirements specification into an implementable design, following a structured and organized process.
4. Formulate a testing strategy for a software system, employing techniques such as black box and white box testing strategies.
5. Evaluate the quality of the requirements, analysis and design work done during the module.

Unit I

Unit II
Software Requirements - Functional and Non-functional requirements, User Requirements, System requirements, Interface specification, the software requirements document.
Requirements Engineering Process (REP) - Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.
System models - Context Models, Behavioral models Data models, Object models, Structured Charts

Unit III
Design Engineering - Design process and Design quality, Design concepts, the design model.
Creating an architectural design - Software architecture, Data design, Architectural styles and patterns, Architectural Design.
Unit IV

Unit V
Testing Strategies - A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.
Testing Object Oriented Applications – Object Oriented Testing methods, testing methods applicable at class level, interclass test case design.

Unit VI
Software Reengineering - A software reengineering process model, software reengineering activities.
Quality Management – Quality Concepts, SQA, Software Reviews, FTR, SSQA, Software Reliability

Text Books:
1. Roger S.Pressman , Software Engineering, 7/e , McGraw Hill Education
2. Somervile, Software Engineering, 8/e, Pearson Education

Reference Books:
1. PankajJalote,Software engineering A precise approach –Wiley
Object Oriented Analysis and Design

Credits : 4  
Course Code: 16MCS1012

Course Objectives:

- Develop the different UML diagrams for a software system based on the given requirements.
- Apply forward engineering to convert diagram to code and reverse engineering to convert code to diagram.
- Analyze & design a s/w system in object oriented approach, using unified modeling language.

Course Outcomes:
At the end of this course the student should be able to

1. Illustrate the use of unified modeling language for object oriented analysis and design
2. Understand the syntax of different UML diagrams.
3. Apply object oriented analysis and design to build a software system.

Unit I:
Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.

Unit II:
Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

Unit III:
Basic Behavioral Modeling: Terms, concepts, common modeling technique for interactions, Use cases, Usecase Diagrams, Interaction diagrams, Activity Diagrams.

Unit IV:
Advanced Behavioral Modeling: Terms, concepts, modeling techniques for Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

UNIT V:
Architectural Modeling: Components, Deployment, Collaborations, Component diagrams and Deployment diagrams.

Unit VI:
Terms, concepts, modeling techniques of System and models, case study of the Unified Library application.
Text Books:


Reference Books:

Aditya Institute Of technology And Management
Declaring Variables and Method, Error Handling and Debugging, Sharing Data between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations.

Unit-V:

Unit-VI

Text Books:
3. Hans Bergsten, SPD O’Reilly, Java Server Pages

Reference Books:
2. Marty Hall and Larry Brown , Core SERVLETS AND JAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES, Pearson
3. Dietel and Nieto, Internet and World Wide Web – How to program, PHI/Pearson Education Asia.
5. Murach, Murach’s beginning JAVA JDK 5, SPD
Mobile Computing
(Elective- II)

Credits : 4
Course Code: 16MCS1014

Course Objectives:
- To present necessary concepts for the mobile communications
- Understanding different mobile devices and system
- Understanding the cellular system design
- Study co channel and non co channel interferences
- Understanding channel assignments and handoff
- Study digital cellular system

Course Outcomes:
At the end of this course the student should be able to

1. Discover the characteristics of pervasive computing applications including the major system components and architectures of the systems
2. Analyze the strengths and limitations of the tools and devices for development of pervasive computing systems
3. Explore the characteristics of different types of mobile networks on the performance of a pervasive computing system
4. Analyze and compare the performance of different data dissemination techniques and algorithms for mobile real-time applications.

Unit I
Introduction to Mobile Communications and Computing
Mobile Computing (MC): Introduction to MC, novel applications, limitations, and architecture
GSM: Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, and New data services

Unit II
(Wireless) Medium Access Control
Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA.

Unit III
Mobile Network Layer and Transport Layer
Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunneling and encapsulation, optimizations), Dynamic Host Configuration Protocol (DHCP). Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission /time-outfreezing, Selective retransmission, Transaction oriented TCP.

Unit IV
Database Issues
Hoarding techniques, caching invalidation mechanisms, client server computing with adaptation, power-aware and context-aware computing, transactional models, query processing, recovery, and quality of service issues, Communications asymmetry, classification of new data delivery
mechanisms, push-based mechanisms, pull-based mechanisms, hybrid mechanisms, selective tuning (indexing) techniques.

Unit V
Mobile Ad hoc Networks (MANETs)
Overview, Properties of a MANET, MANET applications, routing and various routing algorithms, security in MANETs.

Unit VI
Protocols and Tools
Wireless Application Protocol-WAP. (Introduction, protocol architecture, and treatment of protocols of all layers), Bluetooth (User scenarios, physical layer, MAC layer, networking, security, link management)

Text Books:
3. Sivarammurthy, Manoj., 2009 Adhoc Wireless Networks, 2/e, Pearson

Reference Books:
5. Rajkamal, 2008 Mobile Computing, Oxford
Cryptography and Network Security
(Elective – II)

Course Objectives:

- To clearly recognize the different Security Attacks, Security Services and Security Mechanisms.
- To list out the importance and applications of Non-Cryptographic and Software Vulnerabilities.
- To demonstrate the basic categories of Cryptographic Systems.
- To compute different Conventional Encryption Algorithms.
- To describe the important public-key cryptosystems.
- To analyze the authentication by studying different authentication applications.
- To describe the security approaches related to Electronic Mail
- To express the overall structure of IPSec
- To categorize Intrusions and intrusion detection techniques.
- To develop the different firewall principles.

Course Outcomes:
At the end of this course the student will be able to

1. Recognize different Security Attacks, Services and Mechanisms
2. Classify and explain categories of different encryption and decryption techniques
3. Identify the authentication applications such as Kerberos and x.509 directory services
4. Analyze the representation of PGP and S/MIME
5. Familiar with the importance of IP Security and Web Security
6. Exposed to viruses and related threats and design principles of firewalls

Unit I

Unit II
Secret Key Cryptography: DES, Strengths of DES, Block Cipher Design Principles and Modes of Operations, Triple DES, Blowfish, IDEA, CAST-128 and AES

Unit III
Public-Key Cryptography: Public Key Cryptography Principles, RSA, Diffie-Hellman Key Exchange, Introduction to Elliptic Curve Cryptography
Unit IV

Unit V

Unit VI
Intruders: Intrusion Techniques, Password Protection, Intrusion Detection.
Viruses and Related Threats: Malicious Programs, The Nature of Viruses, Types of Viruses.
Firewalls: Design Principles, Characteristics, Types of Firewalls, Firewall Configurations.
Trusted Systems.

Text Books:

Reference Books:
Course Objective:
- The objective of this course is to give students basic knowledge about the key algorithms and theory that form the foundation of machine learning.
- Identify and apply the appropriate Machine learning technique to classification, Pattern Recognition, and Optimization and Decision problems.

Course Outcomes:
At the end of the course, a student will be able to

1. Describe and design the concepts of learning.
2. Describe and apply learning algorithms.
3. Explain the first principles of neural networks.
4. Describe basics of sampling theory and hypothesis testing.
5. Explain Bayesian learning theorem.

Unit-I
Concept Learning And The General-To-Specific Ordering:
Introduction, A Concept Learning Task, Concept Learning as Search, FIND-S: Finding a Maximally Specific Hypothesis, Version Spaces and the Candidate Elimination Algorithm, Remarks on Version spaces and Candidate-Elimination, Inductive Bias

Unit-II
Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate Problems for Decision Tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Learning, Inductive Bias in Decision Tree Learning, Issues in Decision Tree Learning

Unit-III
Artificial Neural Networks: Introduction, Neural Network Representations, Appropriate Problems for Neural Network Learning, Perceptrons, Multilayer Networks and the Back propagation Algorithm, Remarks on Back Propagation Algorithm, An Illustrative Example: Face Recognition, Advanced Topics in Artificial Neural Networks

Unit-IV
Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Maximum Likelihood and Least-Squared Error Hypothesis, Maximum Likelihood Hypothesis for Predicting Probabilities, Minimum Description Length Principle, Bayes Optimal Classifier, Gibbs Algorithm, Naive Bayes Classifier, An Example: Learning to Classify Text, Bayesian Belief Networks, The EM Algorithm
Unit-V

Unit-VI
Instance Based Learning: Introduction, k-Nearest neighbor Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning

Text Books:

Reference Books:

Web References:
https://www.coursera.org/course/ml
Course Objectives:

- To Paraphrase the emerging area of "cloud computing" and how it relates to the corporate world.
- To gain competence in cloud services and search engines. Specifically
- To Paraphrase and be able to cloud environment is collaborating with various webmail services and databases.
- To Paraphrase how virtualization is well-known in cloud computing.
- To gain competence in Cloud Security and Open Cloud delivering highly-interactive Web applications.

Course Outcomes:

1. Articulate the basic concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.
2. Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Identify the Collaborations of Cloud and evaluate webmail services.
4. Provide the appropriate cloud computing solutions and recommendations according to the applications used.
5. Attempt to generate new ideas and innovations based on Virtualization in cloud computing. Provide Security for cloud applications.

Unit I:

Unit II:

Unit III:

Unit IV:
**Introduction to Virtualization** Objectives of virtualization, History of virtualization, Benefits of virtualized technology, the virtual service desk, what can be virtualized, related forms of computing, cloud computing, software as a service – SaaS, grid computing, utility computing, virtualization processes.

Unit V:
**Virtualization Technologies-I** Ubuntu (server edition), altiris, windows, server, software virtualization, vmware, intel virtualization, red hat virtualization, softgrid application, Linux virtualization, desktop,
virtualization, hardware virtualization, resource virtualization, processor virtualization, application virtualization.

**Unit VI:**
**Virtualization Technologies-II**
Storage virtualization, virtualization density, para-virtualization, OS virtualization, virtualization software, data storage virtualization, Intel virtualization technology, thinstall virtualization suite, net framework virtualization, windows virtualization on fedora, storage virtualization technologies, virtualization level, security monitoring and virtualization, oracle virtualization. Case Studies: Amazon S3, Google APP Engine, IBM Clouds, Oracle OBIEE

**Text Books:**
2. George Reese, 2009, Cloud Application Architectures Building Applications and Infrastructure in the Cloud, O'Reilly Media Press.

**Reference Books:**
Course Objective:
By the end of this course, the student should be able to

- Explain the concepts of neural networks, fuzzy logic, and genetic algorithms.
- Solve problems that are appropriately solved by neural networks, fuzzy logic, and Genetic algorithms.

Course Outcome:

1. Explain the learning and adaptation capability of neural and fuzzy systems and genetic algorithm.
2. Describe the learning and retrieval procedures of various neural networks.
3. Comprehend the fuzzy logic control and adaptive fuzzy logic and to design the fuzzy control using genetic algorithm.

Unit I

Unit II
Learning ProcessError Correction learning, Memory based learning, Hebbian learning, Competitive, Boltzmann learning, Credit Assignment Problem, Memory, Adaption, Statistical nature of the learning process.

Unit III
Architecture: perceptron model, solution, single layer artificial neural network, multilayer perception model; back propagation learning methods, effect of learning rule co-efficient; back Propagation algorithm, factors affecting backpropagation training, applications.

Unit IV
Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp relations, Fuzzy to Crisp conversion.

Unit V
Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzyfications&Defuzzificataions, Fuzzy Controller, Industrial applications.

Unit VI
Basic concepts, working principle, procedures of GA, flow chart of GA, Genetic representations, (Encoding) Initialization and selection, Genetic operators, Mutation, Generational Cycle, applications.
Text Books:
2. Simon Haykin, “Neural Networks” Prentice Hall of India

Reference Books:
2. Timothy J. Ross, “Fuzzy Logic with Engineering Applications” Wiley India.
Simulation and Modeling
(Elective – III)

Credits : 4
Course Code: 16MCS1019

Course Objectives:
• Educate students with fundamental knowledge of continuous and discrete system models
• Gain some fundamental knowledge about system simulation techniques
• Gain idea about continuous system simulation and different models of continuous system simulation
• Gain knowledge probability theory and probability functions.
• Acquire knowledge queuing theory with solutions
• Gain knowledge on discrete system simulation and different models of discrete system simulation
• Familiarize with discrete system programming tasks.
• Acquire knowledge about simulation programming techniques.
• Explain some elementary features of SIMSCRIPT and GPSS algorithms.

Course Outcomes:
This course will assist students in their career preparation as computer simulation and modeling designer or managers. Upon successful completion of this course, students should be able to:
1. Classification and study of system modeling.
2. Describe the steps involved in continuous system simulation
3. Analyze System Dynamics
4. Study Probability concepts in Simulation and apply different random number generation Techniques and their applications.
5. Articulate queuing disciplines with mathematical solutions and Outline methods for discrete system simulation
6. Organize SIMSCRIPT and GPSS for analyzing, estimating and processing the problems for deriving the simulation output.

Unit I
Introduction to Modeling and Simulation: Nature of Simulation: Systems, Models and Simulation, Continuous and Discrete Systems, System modeling, concept of simulation, Components of a simulation study, Principles used in modeling, Static and Dynamic physical models, Static and Dynamic mathematical models Introduction to Static and Dynamic System simulation, Advantages, Disadvantages and pitfalls of Simulation.

Unit II
System Simulation: Types of System Simulation: Monte Carlo Method, Comparison of analytical and Simulation methods, Numerical Computation techniques for Continuous and Discrete Models, Distributed Lag Models, Cobweb Model.

Unit III
Continuous System Simulation and System Dynamics Continuous System models, Analog and Hybrid computers, Digital-Analog Simulators, Exponential growth and decay models, logistic curves, Generalization of growth models, System dynamics diagrams, Multi segment models, Representation of Time Delays.
Unit IV

**Probability concepts in Simulation:** Discrete and Continuous probability functions, Continuous Uniformly Distributed Random Numbers, Generation of a Random numbers, Generating Discrete distributions, Non-Uniform Continuously Distributed Random Numbers, Rejection Method.

Unit V

**Simulation of Queuing Systems and Discrete System Simulation** Poisson arrival patterns, Exponential distribution, Service times, Normal Distribution Queuing Disciplines, Simulation of single and two server queue. Application of queuing theory in computersystem. Discrete Events, Generation of arrival patterns, Simulation programming tasks, Gathering statistics, Measuring occupancy and Utilization, Recording Distributions and Transit times.

Unit VI


**Text Books:**

**Reference Books:**
Parallel Computing And Algorithms
(Elective – III)

Credits : 4
Course Code: 16MCS1020

External Marks : 60
Internal Marks : 40

Course Objectives:

• To learn parallel and distributed algorithms development techniques for shared memory and message passing models.
• To study the main classes of parallel algorithms.
• To study the complexity and correctness models for parallel algorithms.

Course Outcomes:

1. Upon successful completion of the course students should be able to
2. Identify the need of Parallel Computing Algorithms.
3. Analyze the performance of the parallel algorithms.
4. Practice Vector matrix multiplication.

Unit I

Unit II

Unit III
Algorithms-1: Elementary Parallel algorithms on SIMD and MIMD machines, Analysis of these algorithms. Matrix Multiplication algorithms on SIMD and MIMD models.

Unit IV

Unit V
Array processors: Array processors, 2D-Mesh processor and Hypercube Processor Array. Sorting: Parallel sorting methods, Odd-even transposition Sorting on processor arrays. Parallel Quick-sort on Multi processors

Unit VI
Graph Algorithms - Connected Graphs, search and traversal, Combinatorial Algorithms-Permutation, Combinations, Derangements.
Textbooks:

1. Michel J.Quinn, Parallel computing theory and practice.

Reference Books:

Digital Image Processing
(Elective – III)

Credits : 4
Course Code: 16MCS1021

External Marks : 60
Internal Marks : 40

Course Objectives:

• Cover the basic theory and algorithms that are widely used in digital image processing.
• Expose students to current technologies and issues that are specific to image processing systems.
• Hands-on experience in using computers to process images.
• Formulate solutions to general image processing problems
• Familiar with image manipulations and analysis

Course Outcomes:

1. Explain basic concepts in image Processing
2. Apply spatial domain techniques for image enhancement
3. List the image compression techniques
4. Discuss various morphological algorithms
5. Classify various image segmentation techniques
6. Describe different color models and color transformations

Unit I
Digital Image Fundamentals: Fundamental steps in Digital image processing, components of an image processing of system. Image sampling and quantization: Basic Concepts in sampling and quantization, Representing Digital images, Spatial and Gray level resolution, Zooming and Shrinking Digital images. some basic Relationship between pixels: Neighbors of a pixel, Adjacency, Connectivity, Regions and Boundaries, Distance Measures, Image operations on a pixel basis.

Unit II

Unit III
Morphology: Preliminaries, Erosion and Dilation, Opening and closing, Hit-or-Miss transform, basic morphological algorithms: Boundary Extraction, Region filling, Extraction of connected components, Convex Hull, Skeletons, Pruning, Gray-Scale Morphology.

Unit IV
Image Segmentation: Detection of Discontinuities: Point detection, line detection, Edge detection, Edge linking and boundary detection: Local processing, Global is processing via Graph theoretic techniques, Thresholding: Basic global thresholding, Basic Adaptive thresholding, optimal global and Adaptive thresholding, thresholding based on several variables, Region Based Segmentation.
Unit V:
**Image Compression:** Image compression Types and requirements, Coding Redundancy, Interpixel Redundancy, Psycho visual redundancy, Fidelity Criteria, Image compression models, Some Basic Compression Methods: Huffman Coding, Arithmetic Coding, LZW Coding, Bit Plane Coding.

Unit VI:
**Color Image Processing:** Color fundamentals, Color models: RGB, CMY, CMYK, HSI models, Color Transformations: Formulation, Color Complements, Color slicing, Tone and color corrections.

**Text Book:**

**Reference Books:**
Object Oriented Analysis & Design Lab

Course Code: 16MCS1102

Course Objectives:

➢ provides a brief overview of the Unified Modeling Language (UML)
➢ Documenting user requirements using the UML
➢ Designing of various diagrams of UML
➢ The usage of different structural and behavioral diagrams
➢ Explain and apply the key facts, concepts, principles of UML
➢ Analyze problems, select and apply appropriate things from UML
➢ Effectively use current techniques, skills, and tools necessary in design
➢ Design, implement, and test software systems that meet the needs of a client

Course Outcomes:

➢ Implementation of the diagrams in Unified Modeling Language
➢ The Difference between Static and Dynamic diagrams designing
➢ Create and analyze use case, sequence and other diagrams
➢ Apply forward and reverse engineering techniques.
➢ make the transition from business model to use cases and even to other models

The student is expected to take up about five mini-projects and model them and produce Use Cases, Analysis Documents - both static & dynamic aspects, Sequence

Diagrams and State-Charts, Database Design using Rational Products A sample collection of ideas is given. Numerous other ideas can be found in the pages from the list of references given below.

Mini-Project - I:
A Point-of-Sale (POS) System: A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store, it includes hardware components such as a computer and bar CODE scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services are temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDAs, touch-screens.

Mini-Project - II:
Online Bookshop Example: Following the model of amazon.com or bn.com, design and implement an online bookstore.

Mini-Project - III:
A Simulated Company: Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company.

Mini-Project - IV:
A Multi-Threaded Airport Simulation: Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/conflicts. Landing: an aircraft uses the runway, lands, and then taxis to the terminal. Take-Off: an aircraft taxies to the runway and then takes off.
Mini-Project -V:
An Automated CommUnity Portal: Business in the 21st Century is above all BUSY. Distractions are everywhere. The current crop of "enterprise intranet portals" are often high noise and low value, despite the large capital expenditures it takes to stand them up. Email takes up 30 - 70% of an employee's time. Chat and Instant Messaging are either in the enterprise or just around the corner. Meanwhile, management is tasked with unforeseen and unfunded leadership and change-agent roles as well as leadership development and succession management. What is needed is a simplified, repeatable process that enhances communications within an enterprise, while allowing management and peers to self-select future leaders and easily recognize high performance team members in a dynamic way. Additionally, the system should function as a general-purpose content management, business intelligence and peer-review application. GlassCODE's goal is to build that system. The software is released under a proprietary license, and will have the following features: Remote, unattended moderation of discussions. However, it will have powerful discovery and business intelligence features, and be infinitely extendable, owing to a powerful API and adherence to Java platform standards. Encourages peer review and indicates for management potential leaders, strong team players and reinforces enterprise and team goals seamlessly and with zero administration.

Mini-Project -VI:
A Content Management System: The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions.

Features

- Robust Permissions System
- Templates for easy custom site designs
- Total control over the content
- Search engine friendly URL’s
- Role based publishing system
- Versioning control
- Visitor profiling

Mini-Project-VII:
An Auction Application:
Several commerce models exist and are the basis for a number of companies like eBay.com, priceline.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

Mini-Project -VIII:
A Notes and File Management System: In the course of one's student years and professional career one produces a lot of personal notes and documents. All these documents are usually kept on papers or individual files on the computer. Either way, the bulk of the information is often erased corrupted and eventually lost. The goal of this project is to build a distributed software application that addresses this problem. The system will provide an interface to create, organize and manage personal notes through the Internet for multiple users. The system will also allow users to collaborate by assigning permissions for multiple users to view and edit notes.
Mini-Project - IX:
A Customizable Program Editor: A programmer's editor which will be focused on an individual programmer's particular needs and style. The editor will act according to the specific language the current source file is in, and will perform numerous features, such as auto-completion or file summarization, on the file. These features will be able to be turned on or off by the programmer, and the programming style of the user will be used to create as efficient an editing environment as possible.

Mini-Project - X:
A Graphics Editor: Design and implement a Java class collection that supports the construction of graph editing applications, i.e., applications that include the ability to draw structured and unstructured diagrams. E.g., The goal of the GEF project is to build a graph editing library that can be used to construct many, high-quality graph editing applications. Some of GEF's features are: A simple, concrete design that makes the framework easy to understand and extend. Node-Port-Edge graph model that is powerful enough for the vast majority of connected graph applications.

Model-View-Controller design based on the Swing Java UI library makes GEF able to act as a UI to existing data structures, and also minimizing learning time for developers familiar with Swing.

High-quality user interactions for moving, resizing, reshaping, etc. GEF also supports several novel interactions such as the broom alignment tool and selection action buttons. Generic properties sheet based on JavaBeans introspection. XML-based file formats based on the PGML standard.

Text Book:

Reference Books:

Various Net Resources and projects:
http://user-mode-linux.sourceforge.net/case-studies.html
http://www.onesmartclick.com/programming/case-studies.html
http://www.tigris.org/sarvlets/ProjectList?type=P rejects
http://hotscripts.com/
http://www.developingwebs.net/
http://sourceforge.net/projects/
http://governing.com/gpp/eponline.htm
http://www.whitehouse.gov/omb/inforeg/egovstrategy.pdf
http://www.andhrapradesh.com/
http://www.ap-lt.com/
Course Objectives:
- Understand the various steps in designing a creative and dynamic website.
- They will able to write html, JavaScript, CSS and XML.
- Design dynamic and interactive web pages by embedding Java Script CODE in HTML. Use Java Script to validate user input.
- Understand the concepts of JDBC, Servlets and JSP and Spring.

Course Outcomes:
After completion of this course, the students would be able to:

1. Understand, analyze and apply the role of languages like HTML, DHTML, CSS, XML, JavaScript in the workings of the web and web applications
2. Analyze a web page and identify its elements and attributes.
3. Build web applications using JSP
4. Build web applications using JDBC, Servlets.
5. Will create a fully functional website (online book store) using MVC architecture.

1. Design the following static web pages required for an online book store website.

1) HOME PAGE:
The static home page must contain three frames.
Top Frame: Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left Frame: At least four links for navigation, which will display the catalogue of respective links. For e.g.: When you click the link “M.Tech.” the catalogue for M.Tech.Books should be displayed in the Right frame.

Right Frame: The pages to the links in the left frame must be loaded here. Initially this page contains description of the web site.

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td>Catalogue</td>
</tr>
<tr>
<td></td>
<td>Cart</td>
</tr>
</tbody>
</table>

CSE  IT  ECE  EEE  MECH  CIVIL  MCA  MBA
2) LOGIN PAGE:

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
<tr>
<td>CSE</td>
<td>Login:</td>
</tr>
<tr>
<td>IT</td>
<td>Password:</td>
</tr>
<tr>
<td>ECE</td>
<td></td>
</tr>
<tr>
<td>EEE</td>
<td></td>
</tr>
<tr>
<td>MECH</td>
<td></td>
</tr>
<tr>
<td>CIVIL</td>
<td></td>
</tr>
<tr>
<td>MCA</td>
<td></td>
</tr>
<tr>
<td>MBA</td>
<td></td>
</tr>
</tbody>
</table>

3) CATALOGUE PAGE:
The catalogue page should contain the details of all the books available in the web site in a table. The details should contain the following:
2. Author Name.
3. Publisher.
5. Add to cart button.

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
</tbody>
</table>
| CSE  | ![XML Bible](image) Book : XML Bible 
Author : Winston 
Publication : Wiley $40.5 |
| IT   | ![AI](image) Book : AI 
Author : S. Russell 
Publication : Princeton Hall $63 |
| ECE  | ![Java 2](image) Book : Java 2 
Author : Watson 
Publication : BPB Publications $35.5 |
| EEE  | ![HTML in 24 hours](image) Book : HTML in 24 hours 
Author : Sam Peter 
Publication : Sam $50 |
| MECH | ![C++](image) Book : C++ 
Author : Jane Doe 
Publication : John Smith $75 |
| CIVIL| ![Database Management](image) Book : Database Management 
Author : John Doe 
Publication : Jane Smith $90 |
| MCA  | ![Computer Science](image) Book : Computer Science 
Author : Jane Doe 
Publication : John Smith $100 |
| MBA  | ![Management](image) Book : Management 
Author : Jane Doe 
Publication : John Smith $110 |

4. REGISTRATION PAGE:
Create a “registration form” with the following fields
1) Name (Text field)
2) Password (password field)
3) E-mail id (text field)
4) Phone number (text field)
5) Sex (radio button)
6) Date of birth (3 select boxes)
7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
8) Address (text area)
5. **VALIDATION:**

Write JavaScript to validate the following fields of the above registration page.

1. Name (Name should contain alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern
   \[(name@domain.com)\]
4. Phone number (Phone number should contain 10 digits only).
   
   Note: You can also validate the login page with these parameters.

6. **Design a web page using CSS (Cascading Style Sheets) which includes the following:**

1) Use different font, styles:
   In the style definition you define how each selector should work (font, color etc.).
   Then, in the body of your pages, you refer to these selectors to activate the styles

7. **Write an XML file which will display the Book information which includes the following:**

   1) Title of the book
   2) Author Name
   3) ISBN number
   4) Publisher name
   5) Edition
   6) Price

   Write a Document Type Definition (DTD) to validate the above XML file.

8. **Write a servlet program to read parameters from web.xml**

9. **Write a servlet program using cookie management**

10. **Write a servlet program to illustrate HttpSession**

11. **Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount)) of each category. Modify your catalogue page

    (experiment- 2)in such a way that you should connect to the database and extract data

    From the tables and display them in the catalogue page using JDBC.

12. **Write a JSP program to connect to the database and retrieve records**

13. **Write a JSP program to Sort List of Names and Search for a Key.**

14. **Write a spring application to display Hello World.**

**Text Books:**

2. Patrick Naughton and Herbert Schildt, The complete Reference Java 2, Fifth Edition. TMH
   (Chapters: 25)
3. Hans Bergsten, SPD O’Reilly, Java Server Pages

**Reference Books:**

2. Marty Hall and Larry Brown, Core SERVLETS AND JAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES, Pearson
3. Dietel and Nieto, Internet and World Wide Web – How to program, PHI/Pearson Education Asia.
5. Murach, Murach’s beginning JAVA JDK 5, SPD