AFFILIATED INSTITUTIONS
ANNA UNIVERSITY, CHENNAI
REGULATIONS – 2013
M.E. COMPUTER SCIENCE AND ENGINEERING
I TO IV SEMESTERS CURRICULA AND SYLLABI (FULL TIME)

PROGRAM EDUCATIONAL OBJECTIVES (PEO):
Graduates of this M. E. Computer Science and Engineering will be able to
1. Apply the necessary mathematical tools and fundamental & advanced knowledge of computer science & engineering
2. Develop computer/software/network systems understanding the importance of social, business, technical, environmental, and human context in which the systems would work
3. Articulate fundamental concepts, design underpinnings of computer/software/network systems, and research findings to train professionals or to educate engineering students
4. Contribute effectively as a team member/leader, using common tools and environment, in computer science and engineering projects, research, or education
5. Pursue life-long learning and research in selected fields of computer science & engineering and contribute to the growth of those fields and society at large

PROGRAM OUTCOMES:
- Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the conceptualization of engineering models.
- Identify, formulate, research literature and solve complex engineering problems reaching substantiated conclusions using first principles of mathematics and engineering sciences.
- Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
- Conduct investigations of complex problems including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
- Create, select and apply appropriate techniques, resources, and modern engineering tools, including prediction and modeling, to complex engineering activities, with an understanding of the limitations.
- Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.
- 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.
- Demonstrate understanding of the societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to engineering practice.
- Understand and commit to professional ethics and responsibilities and norms of engineering practice.
- Understand the impact of engineering solutions in a societal context and demonstrate knowledge of and need for sustainable development.
- Demonstrate a knowledge and understanding of management and business practices, such as risk and change management, and understand their limitations.
- Recognize the need for, and have the ability to engage in independent and life-long learning.
# M.E. COMPUTER SCIENCE AND ENGINEERING
## I TO IV SEMESTERS CURRICULA AND SYLLABI (FULL TIME)

### SEMESTER I

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEORY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MA7155</td>
<td>Applied Probability and Statistics</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>CP7101</td>
<td>Design and Management of Computer Networks</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7102</td>
<td>Advanced Data Structures and Algorithms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7103</td>
<td>Multicore Architectures</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Elective I</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Elective II</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>PRACTICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>CP7111</td>
<td>Advanced Data Structures Laboratory</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>CP7112</td>
<td>Case Study - Network Design (Team Work)</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>18</td>
<td>1</td>
<td>6</td>
<td>22</td>
</tr>
</tbody>
</table>

### SEMESTER II

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEORY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>CP7201</td>
<td>Theoretical Foundations of Computer Science</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>CP7202</td>
<td>Advanced Databases</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7203</td>
<td>Principles of Programming Languages</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7204</td>
<td>Advanced Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Elective III</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Elective IV</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>PRACTICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>CP7211</td>
<td>Advanced Databases Laboratory</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>CP7212</td>
<td>Case Study - Operating Systems Design (Team Work)</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>18</td>
<td>1</td>
<td>6</td>
<td>22</td>
</tr>
</tbody>
</table>

### SEMESTER III

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEORY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>CP7301</td>
<td>Software Process and Project Management</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Elective V</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>Elective VI</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>Elective VII</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>PRACTICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>CP7311</td>
<td>Project Work (Phase I)</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>12</td>
<td>1</td>
<td>12</td>
<td>19</td>
</tr>
</tbody>
</table>

2
<table>
<thead>
<tr>
<th>SL. NO</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>PRACTICAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>CP7411</td>
<td>Project Work (Phase II)</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL NO. OF CREDITS: 75</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LIST OF ELECTIVES**

**ELECTIVE I**

<table>
<thead>
<tr>
<th>SL. NO</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>SE7103</td>
<td>Formal models of software systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>CP7001</td>
<td>Performance Evaluation of Computer Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7002</td>
<td>Probabilistic Reasoning Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7003</td>
<td>Data Analysis and Business Intelligence</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>CP7004</td>
<td>Image Processing and Analysis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>NE7001</td>
<td>Sensing Techniques and Sensors</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**ELECTIVE II**

<table>
<thead>
<tr>
<th>SL. NO</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CP7005</td>
<td>Randomized Algorithms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>NE7002</td>
<td>Mobile and Pervasive Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7006</td>
<td>Parallel Programming Paradigms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7007</td>
<td>Software Requirements Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>CP7008</td>
<td>Speech Processing and Synthesis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7009</td>
<td>Machine Learning Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**ELECTIVE III**

<table>
<thead>
<tr>
<th>SL. NO</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CP7010</td>
<td>Concurrency Models</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>CP7011</td>
<td>Real Time Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7012</td>
<td>Computer Vision</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>NE7202</td>
<td>Network and Information Security</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>CP7013</td>
<td>Design and Analysis of Parallel Algorithms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7014</td>
<td>Software Architectures</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
### ELECTIVE IV

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CP7015</td>
<td>Model Checking and Program Verification</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>CP7016</td>
<td>Embedded Software Development</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>IF7202</td>
<td>Cloud Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7017</td>
<td>Data Visualization Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>NE7005</td>
<td>Protocols and Architecture for Wireless Sensor Networks</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7018</td>
<td>Language Technologies</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### ELECTIVE V

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>NE7012</td>
<td>Social Network Analysis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>CP7019</td>
<td>Managing Big Data</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>NE7011</td>
<td>Mobile Application Development</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7020</td>
<td>Bio-inspired Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>CP7021</td>
<td>Medical Image Processing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7022</td>
<td>Software Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### ELECTIVE VI

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CP7023</td>
<td>Reconfigurable Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>IF7013</td>
<td>Energy Aware Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7024</td>
<td>Information Retrieval Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CP7025</td>
<td>Data Mining Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>IF7002</td>
<td>Bio Informatics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7026</td>
<td>Software Quality Assurance</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

### ELECTIVE VII

<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>COURSE CODE</th>
<th>COURSE TITLE</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CP7027</td>
<td>Multi Objective Optimization Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>CP7028</td>
<td>Enterprise Application Integration</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>CP7029</td>
<td>Information Storage Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>CP7030</td>
<td>Robotics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>CP7031</td>
<td>Compiler Optimization Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
MA7155  APPLIED PROBABILITY AND STATISTICS

OBJECTIVES:
• To introduce the basic concepts of one dimensional and two dimensional Random Variables.
• To provide information about Estimation theory, Correlation, Regression and Testing of hypothesis.
• To enable the students to use the concepts of multivariate normal distribution and principle components analysis.

UNIT I  ONE DIMENSIONAL RANDOM VARIABLES  9+3
Random variables - Probability function – Moments – Moment generating functions and their properties – Binomial, Poisson, Geometric, Uniform, Exponential, Gamma and Normal distributions – Functions of a Random Variable.

UNIT II  TWO DIMENSIONAL RANDOM VARIABLES  9+3
Joint distributions – Marginal and Conditional distributions – Functions of two dimensional random variables – Regression Curve – Correlation.

UNIT III  ESTIMATION THEORY  9+3

UNIT IV  TESTING OF HYPOTHETESES  9+3
Sampling distributions - Type I and Type II errors - Tests based on Normal, t, Chi-Square and F distributions for testing of mean, variance and proportions – Tests for Independence of attributes and Goodness of fit.

UNIT V  MULTIVARIATE ANALYSIS  9+3
Random Vectors and Matrices - Mean vectors and Covariance matrices - Multivariate Normal density and its properties - Principal components Population principal components - Principal components from standardized variables.

TOTAL 45+15:60 PERIODS

OUTCOME:
• The student will able to acquire the basic concepts of Probability and Statistical techniques for solving mathematical problems which will be useful in solving Engineering problems

REFERENCES:

5
UNIT I  INTRODUCTION TO NETWORK MANAGEMENT  

UNIT II  REQUIREMENTS ANALYSIS  
Requirement Analysis Process – Gathering and Listing Requirements - Developing service metrics – Characterizing behavior – Developing RMA requirements – Developing delay Requirements - Developing capacity Requirements - Developing supplemental performance Requirements – Requirements mapping – Developing the requirements specification

UNIT III  FLOW ANALYSIS  

UNIT IV  NETWORK ARCHITECTURE  

UNIT V  NETWORK DESIGN  

REFERENCES:


TOTAL: 45 PERIODS
OBJECTIVES:

- To understand the principles of iterative and recursive algorithms.
- To learn the graph search algorithms.
- To study network flow and linear programming problems.
- To learn the hill climbing and dynamic programming design techniques.
- To develop recursive backtracking algorithms.
- To get an awareness of NP completeness and randomized algorithms.
- To learn the principles of shared and concurrent objects.
- To learn concurrent data structures.

UNIT I Iterative and Recursive Algorithms


UNIT II Optimisation Algorithms


UNIT III Dynamic Programming Algorithms


UNIT IV Shared Objects and Concurrent Objects


UNIT V Concurrent Data Structures


TOTAL : 45 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to
- Design and apply iterative and recursive algorithms.
- Design and implement optimisation algorithms in specific applications.
- Design appropriate shared objects and concurrent objects for applications.
- Implement and apply concurrent linked lists, stacks, and queues.

REFERENCES:

CP7103 MULTICORE ARCHITECTURES

OBJECTIVES:
- To understand the recent trends in the field of Computer Architecture and identify performance related parameters
- To appreciate the need for parallel processing
- To expose the students to the problems related to multiprocessing
- To understand the different types of multicore architectures
- To expose the students to warehouse-scale and embedded architectures

UNIT I  FUNDAMENTALS OF QUANTITATIVE DESIGN AND ANALYSIS

UNIT II  DLP IN VECTOR, SIMD AND GPU ARCHITECTURES
Vector Architecture - SIMD Instruction Set Extensions for Multimedia – Graphics Processing Units - Detecting and Enhancing Loop Level Parallelism - Case Studies.

UNIT III  TLP AND MULTIPROCESSORS
Symmetric and Distributed Shared Memory Architectures – Cache Coherence Issues - Performance Issues – Synchronization Issues – Models of Memory Consistency - Interconnection Networks – Buses, Crossbar and Multi-stage Interconnection Networks.

UNIT IV  RLP AND DLP IN WAREHOUSE-SCALE ARCHITECTURES
UNIT V  ARCHITECTURES FOR EMBEDDED SYSTEMS


TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
• Identify the limitations of ILP and the need for multicore architectures
• Discuss the issues related to multiprocessing and suggest solutions
• Point out the salient features of different multicore architectures and how they exploit parallelism
• Critically analyze the different types of inter connection networks
• Discuss the architecture of GPUs, warehouse-scale computers and embedded processors

REFERENCES:

CP7111  ADVANCED DATA STRUCTURES LABORATORY  L T P C
0 0 4 2

OBJECTIVES:
• To learn to implement iterative and recursive algorithms.
• To learn to design and implement algorithms using hill climbing and dynamic programming techniques.
• To learn to implement shared and concurrent objects.
• To learn to implement concurrent data structures.

LAB EXERCISES:
Each student has to work individually on assigned lab exercises. Lab sessions could be scheduled as one contiguous four-hour session per week or two two-hour sessions per week. There will be about 15 exercises in a semester. It is recommended that all implementations are carried out in Java. If C or C++ has to be used, then the threads library will be required for concurrency. Exercises should be designed to cover the following topics:

• Implementation of graph search algorithms.
• Implementation and application of network flow and linear programming problems.
• Implementation of algorithms using the hill climbing and dynamic programming design techniques.
• Implementation of recursive backtracking algorithms.
• Implementation of randomized algorithms.
• Implementation of various locking and synchronization mechanisms for concurrent linked lists, concurrent queues, and concurrent stacks.
• Developing applications involving concurrency.

TOTAL : 60 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to
1. Design and apply iterative and recursive algorithms.
2. Design and implement algorithms using the hill climbing and dynamic programming and recursive backtracking techniques.
3. Design and implement optimisation algorithms for specific applications.
4. Design and implement randomized algorithms.
5. Design appropriate shared objects and concurrent objects for applications.
6. Implement and apply concurrent linked lists, stacks, and queues.

REFERENCES:

CP7112 CASE STUDY: NETWORK DESIGN (Team Work) L T P C 0 0 2 1

CASE STUDY: 1
Analyzing the performance of various configurations and protocols in LAN.

1.1. Establishing a Local Area Network (LAN): The main objective is to set up a Local Area Network, concepts involved in this network are IP addressing and the Address Resolution Protocol (ARP). The required equipments are 192.168.1.1, 192.168.1.2, 192.168.1.3, Host A Host B Host C, Switch/HUB, three PC’s equipped with at least one NIC, one HUB or Switch and the necessary cables. Once the physical LAN is set up the hosts need to be configured using the ifconfig command. To verify communication among the machines the ping command is used. Next, to manipulate the routing tables at the hosts to understand how machines know where to send packets. Since the ifconfig command places a default route into the routing tables this route must be deleted to ‘blindfold’ the machine. The ping command is used again to show that communication is no longer available. To re-establish communication the routes are put back into the routing table one host at a time. Communication is once again verified using the ping command.

1.2. Connecting two LANs using multi-router topology with static routes:
The main objective is to extend routing connection by using multiple routers. The concepts include IP addressing and basic network routing principles. Connect two LANs topology. During router configuration attention is paid to the types of interfaces as additional issues are involved with setup. For example, the serial interfaces require docking mechanisms to be set correctly. Once the interfaces are working the ping command is used to check for communication between LANs. The failure of communication illustrates the need for routes to be established inside the routing infrastructure. Static routes are used to show how packets can be transported through any reasonable route. It is run trace route on two different configurations to demonstrate the implementation of different routes.
1.3 Analyzing the performance of various configurations and protocols
Original TCP versus the above modified one: To compare the performance between the operation of TCP with congestion control and the operation of TCP as implemented. The main objective is for students to examine how TCP responds to a congested network. The concepts involved in the lab include network congestion and the host responsibilities for communicating over a network. This lab requires three PC’s connected to a switch. One PC is designated as the target host and the other two PC’s will transfer a file from the target host using FTP. A load is placed on the network to simulate congestion and the file is transferred, first by the host using the normal TCP and then by the host using the modified version. This procedure is performed multiple times to determine average statistics. The students are then asked to summarize the results and draw conclusions about the performance differences and the underlying implications for hosts operating in a network environment.

Case Study 2:
RIP and OSPF Redistribution
This case study addresses the issue of integrating Routing Information Protocol (RIP) networks with Open Shortest Path First (OSPF) networks. Most OSPF networks also use RIP to communicate with hosts or to communicate with portions of the internetwork that do not use OSPF. This case study should provide examples of how to complete the following phases in redistributing information between RIP and OSPF networks, including the following topics:
• Configuring a RIP Network
• Adding OSPF to the Center of a RIP Network
• Adding OSPF Areas
• Setting Up Mutual Redistribution

Case Study 3:
Dial-on-Demand Routing
This case study should describe the use of DDR to connect a worldwide network that consists of a central site located in Mumbai and remote sites located in Chennai, Bangalore, and Hyderabad. The following scenarios should be considered:
• Having the Central Site Dial Out
Describe the central and remote site configurations for three setups: a central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Include examples of the usage of rotary groups and access lists.
• Having the Central and Remote Sites Dial In and Dial Out
Describe the central and remote site configurations for three setups: central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Also describes the usage of Point-to-Point Protocol (PPP) encapsulation and the Challenge Handshake Authentication Protocol (CHAP).
• Having Remote Sites Dial Out
A common configuration is one in which the remote sites place calls to the central site but the central site does not dial out. In a “star” topology, it is possible for all of the remote routers to have their serial interfaces on the same subnet as the central site serial interface.
• Using DDR as a Backup to Leased Lines
Describes the use of DDR as a backup method to leased lines and provides examples of how to use floating static routes on single and shared interfaces.
• Using Leased Lines and Dial Backup
Describes the use of Data Terminal Ready (DTR) dialing and V.25bis dialing with leased lines.

Case Study 4:
Network Security
This case study should provide the specific actions you can take to improve the security of your network. Before going into specifics, however, you should understand the following basic concepts that are essential to any security system:
• Know your enemy
This case study refers to attackers or intruders. Consider who might want to circumvent your security measures and identify their motivations. Determine what they might want to do and the damage that they could cause to your network. Security measures can never make it impossible for a user to perform unauthorized tasks with a computer system. They can only make it harder. The goal is to make sure the network security controls are beyond the attacker’s ability or motivation.

• Count the cost
Security measures almost always reduce convenience, especially for sophisticated users. Security measures can delay work and create expensive administrative and educational overhead. It can use significant computing resources and require dedicated hardware. When you design your security measures, understand their costs and weigh those costs against the potential benefits. To do that, you must understand the costs of the measures themselves and the costs and likelihoods of security breaches. If you incur security costs out of proportion to the actual dangers, you have done yourself a disservice.

• Identify your assumptions
Every security system has underlying assumptions. For example, you might assume that your network is not tapped, or that attackers know less than you do, that they are using standard software, or that a locked room is safe. Be sure to examine and justify your assumptions. Any hidden assumption is a potential security hole.

• Control your secrets
Most security is based on secrets. Passwords and encryption keys, for example, are secrets. Too often, though, the secrets are not really all that secret. The most important part of keeping secrets is knowing the areas you need to protect. What knowledge would enable someone to circumvent your system? You should jealously guard that knowledge and assume that everything else is known to your adversaries. The more secrets you have, the harder it will be to keep all of them. Security systems should be designed so that only a limited number of secrets need to be kept.

• Know your weaknesses
Every security system has vulnerabilities. You should understand your system’s weak points and know how they could be exploited. You should also know the areas that present the largest danger and prevent access to them immediately. Understanding the weak points is the first step toward turning them into secure areas.

• Limit the scope of access
You should create appropriate barriers inside your system so that if intruders access one part of the system, they do not automatically have access to the rest of the system. The security of a system is only as good as the weakest security level of any single host in the system.

Case Study 5:
Controlling Traffic Flow
In this case study, the firewall router allows incoming new connections to one or more communication servers or hosts. Having a designated router act as a firewall is desirable because it clearly identifies the router’s purpose as the external gateway and avoids encumbering other routers with this task. In the event that the internal network needs to isolate itself, the firewall router provides the point of isolation so that the rest of the internal network structure is not affected. Connections to the hosts are restricted to incoming file transfer protocol (FTP) requests and email services. The incoming Telnet, or modem connections to the communication server are screened by the communication server running TACACS username authentication.

Case Study 6:
Defining Access Lists
Access lists define the actual traffic that will be permitted or denied, whereas an access group applies an access list definition to an interface. Access lists can be used to deny connections that
are known to be a security risk and then permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, the latter is the more secure method. In this case study, incoming email and news are permitted for a few hosts, but FTP, Telnet, and rlogin services are permitted only to hosts on the firewall subnet. IP extended access lists (range 100 to 199) and transmission control protocol (TCP) or user datagram protocol (UDP) port numbers are used to filter traffic. When a connection is to be established for email, Telnet, FTP, and so forth, the connection will attempt to open a service on a specified port number. You can, therefore, filter out selected types of connections by denying packets that are attempting to use that service. An access list is invoked after a routing decision has been made but before the packet is sent out on an interface. The best place to define an access list is on a preferred host using your favorite text editor. You can create a file that contains the access-list commands, place the file (marked readable) in the default TFTP directory, and then network load the file onto the router.

Case Study 7:
Configuring a fire wall
Consider a Fire wall communication server with single inbound modem. Configure the modem to ensure security for LAN

Case Study 8:
Integrating EIGRP (Enhanced Interior Gateway Routing Protocol) into Existing Networks:
The case study should provide the benefits and considerations involved in integrating Enhanced IGRP into the following types of internetworks:
• IP—The existing IP network is running IGRP
• Novell IPX—The existing IPX network is running RIP and SAP
• AppleTalk—The existing AppleTalk network is running the Routing Table Maintenance Protocol (RTMP)

When integrating Enhanced IGRP into existing networks, plan a phased implementation. Add Enhanced IGRP at the periphery of the network by configuring Enhanced IGRP on a boundary router on the backbone off the core network. Then integrate Enhanced IGRP into the core network.

TOTAL : 30 PERIODS

CP7201 THEORETICAL FOUNDATIONS OF COMPUTER SCIENCE

OBJECTIVES:
- To review sets, relations, functions, and other foundations
- To understand propositional and predicate logics and their applications
- To understand lambda calculus and functional programming
- To understand graph structures and their applications
- To understand formal models of computation, computability, and decidability

UNIT I FOUNDATIONS

UNIT II LOGIC AND LOGIC PROGRAMMING
UNIT III  LAMBDA CALCULUS AND FUNCTIONAL PROGRAMMING  

UNIT IV  GRAPH STRUCTURES  
Tree Structures – Graph structures – graph representations – regular graph structures – random graphs – Connectivity – Cycles – Graph Coloring – Cliques, Vertex Covers, Independent sets – Spanning Trees – network flows – matching

UNIT V  STATE MACHINES  
Languages and Grammars – Finite State Machines – State machines and languages – Turing Machines – Computational Complexity – computability – Decidability – Church’s Thesis.

OUTCOMES:
Upon Completion of the course, the students will be able
- To explain sets, relations, functions
- To conduct proofs using induction, pigeonhole principle, and logic
- To apply counting, permutations, combinations, and recurrence relations
- To apply recursive functions and lambda calculus
- To explain logic programming and functional programming principles
- To apply sequential structures, tree structures, and graph structures
- To explain computational models, computability, and complexity

TOTAL : 60 PERIODS

REFERENCES:
OBJECTIVES:

- To learn the modeling and design of databases.
- To acquire knowledge on parallel and distributed databases and its applications.
- To study the usage and applications of Object Oriented database.
- To understand the principles of intelligent databases.
- To understand the usage of advanced data models.
- To learn emerging databases such as XML, Cloud and Big Data.
- To acquire inquisitive attitude towards research topics in databases.

UNIT I PARALLEL AND DISTRIBUTED DATABASES 9

UNIT II OBJECT AND OBJECT RELATIONAL DATABASES 9

UNIT III INTELLIGENT DATABASES 9

UNIT IV ADVANCED DATA MODELS 9

UNIT V EMERGING TECHNOLOGIES 9

TOTAL: 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Select the appropriate high performance database like parallel and distributed database
- Model and represent the real world data using object oriented database
- Design a semantic based database to meaningful data access
- Embed the rule set in the database to implement intelligent databases
- Represent the data using XML database for better interoperability
- Handle Big data and store in a transparent manner in the cloud
- To solve the issues related to the data storage and retrieval
REFERENCES:

CP7203 PRINCIPLES OF PROGRAMMING LANGUAGES

OBJECTIVES:
• To understand and describe syntax and semantics of programming languages
• To understand data, data types, and basic statements
• To understand call-return architecture and ways of implementing them
• To understand object-orientation, concurrency, and event handling in programming languages
• To develop programs in non-procedural programming paradigms

UNIT I SYNTAX AND SEMANTICS

UNIT II DATA, DATA TYPES, AND BASIC STATEMENTS

UNIT III SUBPROGRAMS AND IMPLEMENTATIONS

UNIT IV OBJECT-ORIENTATION, CONCURRENCY, AND EVENT HANDLING

UNIT V FUNCTIONAL AND LOGIC PROGRAMMING LANGUAGES
Introduction to lambda calculus – fundamentals of functional programming languages – Programming with Scheme – Programming with ML – Introduction to logic and logic programming – Programming with Prolog – multi-paradigm languages

TOTAL: 45 PERIODS
OUTCOMES:
Upon Completion of the course, the students will be able to
- Describe syntax and semantics of programming languages
- Explain data, data types, and basic statements of programming languages
- Design and implement subprogram constructs
- Apply object-oriented, concurrency, and event handling programming constructs
- Develop programs in Scheme, ML, and Prolog
- Understand and adopt new programming languages

REFERENCES:
3. R. Kent Dybvig

CP7204 ADVANCED OPERATING SYSTEMS L T P C 3 0 0 3

OBJECTIVES:
- To learn the fundamentals of Operating Systems
- To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols
- To know the components and management aspects of Real time, Mobile operating systems

UNIT I FUNDAMENTALS OF OPERATING SYSTEMS 9

UNIT II DISTRIBUTED OPERATING SYSTEMS 9

UNIT III DISTRIBUTED RESOURCE MANAGEMENT 9

UNIT IV REAL TIME AND MOBILE OPERATING SYSTEMS 9
UNIT V  CASE STUDIES

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students should be able to:
• Discuss the various synchronization, scheduling and memory management issues
• Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
• Discuss the various resource management techniques for distributed systems
• Identify the different features of real time and mobile operating systems
• Install and use available open source kernel
• Modify existing open source kernels in terms of functionality or features used

REFERENCES:

CP7211  ADVANCED DATABASE LABORATORY  L T P C
0 0 4 2

OBJECTIVES:
• To learn to work on distributed data bases
• To understand and work on object oriented databases
• To gain knowledge in parallel data base by experimenting it
• To learn to work on active database
• To study and explore deductive database
• To work on the data mining tool weka
• To represent and work with the database using XML

DISTRIBUTED DATABASE:
1. Consider a distributed database for a bookstore with 4 sites called S1, S2, S3 and S4.
   Consider the following relations:
   - Books (ISBN, primary Author, topic, total Stock, price )
   - Book Store (store No, city, state, zip, inventoryValue )
   - Stock (store No, ISBN, Qty)

Total Stock is the total number of books in stock and inventory Value is the total inventory value for the store in dollars.
Consider that Books are fragmented by price amounts into:
F1: Books: price up to $20
F2: Books: price from $20.01 to $50
F3: Books: price from $50.01 to $100
F4: Books: price $100.01 and above

Similarly, Book Stores are divided by ZIP codes into:
S1: Bookstore: Zip up to 25000
S2: Bookstore: Zip 25001 to 50000
S3: Bookstore: Zip 50001 to 75000
S4: Bookstore: Zip 75001 to 99999

Task: Write SQL query for the following
1. Insert and Display details in each table.
2. Find the total number of books in stock where price is between $15 and $55.
3. Update the book price of book No=1234 from $45 to $55 at site S3.
4. Find total number of book at site S2.

2. Implement deadlock detection algorithm for distributed database using wait-for graph and test with the following information.
Consider five transactions T1, T2, T3, T4 and T5 with
T1 initiated at site S1 and spawning an agent at site S2
T2 initiated at site S3 and spawning an agent at site S1
T3 initiated at site S1 and spawning an agent at site S3
T4 initiated at site S2 and spawning an agent at site S3
T5 initiated at site S3

The locking information for these transactions is shown in the following table

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Data items locked by transactions</th>
<th>Data items transaction is waiting for</th>
<th>Site involved in operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>X1</td>
<td>X8</td>
<td>S1</td>
</tr>
<tr>
<td>T1</td>
<td>X6</td>
<td>X2</td>
<td>S2</td>
</tr>
<tr>
<td>T2</td>
<td>X4</td>
<td>X1</td>
<td>S1</td>
</tr>
<tr>
<td>T2</td>
<td>X5</td>
<td>-</td>
<td>S3</td>
</tr>
<tr>
<td>T3</td>
<td>X2</td>
<td>X7</td>
<td>S1</td>
</tr>
<tr>
<td>T3</td>
<td>-</td>
<td>X3</td>
<td>S3</td>
</tr>
<tr>
<td>T4</td>
<td>X7</td>
<td>-</td>
<td>S2</td>
</tr>
<tr>
<td>T4</td>
<td>X8</td>
<td>X5</td>
<td>S3</td>
</tr>
<tr>
<td>T5</td>
<td>X3</td>
<td>X7</td>
<td>S3</td>
</tr>
</tbody>
</table>

Produce local wait for graph for each of the sites and construct global wait for graph and check for dead lock.

OBJECT ORIENTED DATABASE:

3. A University wants to track persons associated with them. A person can be an Employee or Student. Employees are Faculty, Technicians and Project associates. Students are Full time students, Part time students and Teaching Assistants.
a) Design an Enhanced Entity Relationship (EER) Model for university database. Write OQL for the following
i. Insert details in each object.
ii. Display the Employee details.
iii. Display Student Details.
iv. Modify person details.
v. Delete person details.
b) Extend the design by incorporating the following information.
Students are registering for courses which are handled by instructor researchers (graduate students). Faculty are advisors to graduate students. Instructor researchers' class is a category with super class of faculty and graduate students. Faculty are having sponsored research projects with a grant supporting instruction researchers. Grants are sanctioned by different agencies. Faculty belongs to different departments. Department is chaired by a faculty. Implement for the Insertion and Display of details in each class.

PARALLEL DATABASE:

4. Consider the application for University Counselling for Engineering Colleges. The college, department and vacancy details are maintained in 3 sites. Students are allocated colleges in these 3 sites simultaneously. Implement this application using parallel database [State any assumptions you have made].

5. There are 5 processors working in a parallel environment and producing output. The output record contains college details and students mark information. Implement parallel join and parallel sort algorithms to get the marks from different colleges of the university and publish 10 ranks for each discipline.

ACTIVE DATABASE:

6. Create triggers and assertions for Bank database handling deposits and loan and admission database handling seat allocation and vacancy position. Design the above relational database schema and implement the following triggers and assertions.
   a. When a deposit is made by a customer, create a trigger for updating customers account and bank account
   b. When a loan is issued to the customer, create a trigger for updating customer’s loan account and bank account.
   c. Create assertion for bank database so that the total loan amount does not exceed the total balance in the bank.
   d. When an admission is made, create a trigger for updating the seat allocation details and vacancy position.

DEDUCTIVE DATABASE:

7. Construct a knowledge database for kinship domain (family relations) with facts. Extract the following relations using rules.
   Parent, Sibling, Brother, Sister, Child, Daughter, Son, Spouse, Wife, husband, Grandparent, Grandchild, Cousin, Aunt and Uncle.

WEKA TOOL:

8. Work with Weka tool classification and clustering algorithms using the given training data and test with the unknown sample. Also experiment with different scenarios and large data set

<table>
<thead>
<tr>
<th>RID</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Class: buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>middle_aged</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>middle_aged</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>youth</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>youth</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>senior</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>Youth</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>middle_aged</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
</tbody>
</table>
9. Implement Query Optimizer with Relational Algebraic expression construction and execution plan generation for choosing an efficient execution strategy for processing the given query. Also design employee database and test the algorithm with following sample queries.
   a) Select empid, empname from employee where experience > 5
   b) Find all managers working at London Branch

XML
10. Design XML Schema for the given company database
    Department ( deptName, deptNo, deptManagerSSN, deptManagerStartDate, deptLocation )
    Employee ( empName, empSSN, empSex, empSalary, empBirthDate, empDeptNo, empSupervisorSSN, empAddress, empWorksOn )
    Project ( projName, projNo, projLocation, projDeptNo, projWorker )
   a. Implement the following queries using XQuery and XPath
      i. Retrieve the department name, manager name, and manager salary for every department
      ii. Retrieve the employee name, supervisor name and employee salary for each employee who works in the Research Department.
      iii. Retrieve the project name, controlling department name, number of employees and total hours worked per week on the project for each project.
      iv. Retrieve the project name, controlling department name, number of employees and total hours worked per week on the project for each project with more than one employee working on it
   b. Implement a storage structure for storing XML database and test with the above schema.

OUTCOMES:
- Work on distributed databases
- Create and work on object oriented databases
- Create and work with parallel database
- Experiment on active database
- Explore the features of deductive database
- To work on weka tool for clustering and classification
- Represent the database using XML and work on it

CP7212 CASE STUDY – OPERATING SYSTEMS DESIGN (Team Work)

OBJECTIVES:
1. To develop capabilities to work at systems level
2. To learn about issues in designing and implementing modern operating systems
3. To understand team formation, team issues, and allocating roles and responsibilities
4. To make effective presentations on the work done
5. To develop effective written communication skills

LAB EXERCISES:
A team of three or four students will work on assigned case study / mini-project. Case Study / Mini-project can be designed on the following lines:
1. Development of a reasonably sized dynamically loadable kernel module for Linux kernel
2. Study educational operating systems such as Minix (http://www.minix3.org/), Weenix (http://weenix.cs.brown.edu/mediawiki/index.php/Weenix) and develop reasonably sized interesting modules for them
3. Study the Android open source operating system for mobile devices (http://source.android.com/) and develop / modify some modules.
4. Study any embedded and real-time operating system such as eCos (http://ecos.sourceforge.net/) and develop / modify some modules.

TOTAL : 30 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Develop assigned modules of operating systems design carrying out coding, testing, and documentation work involved.
- Describe team issues and apply suitable methods to resolve the same.
- Demonstrate individual competence in building medium size operating system components.
- Demonstrate ethical and professional attributes of a computer engineer.
- Prepare suitable plan with clear statements of deliverables, and track the same.
- Make individual presentation of the work carried out.
- Prepare well-organized written documents to communicate individual work accomplished.

REFERENCES:

CP7301 SOFTWARE PROCESS AND PROJECT MANAGEMENT L T P C
3 1 0 4

OBJECTIVES:
1. To understand overall SDLC and adopt suitable processes
2. To elicit, analyze, prioritize, and manage both functional and quality requirements
3. To estimate efforts required, plan, and track the plans
4. To understand and apply configuration and quality management techniques
5. To evaluate, manage, and design processes

(A mini-project can be chosen by the instructor and use it as a context for the tutorials)

UNIT I DEVELOPMENT LIFE CYCLE PROCESSES
UNIT II REQUIREMENTS MANAGEMENT
9
Functional requirements and quality attributes – elicitation techniques – Quality Attribute Workshops (QAW) – analysis, prioritization, and trade-off – Architecture Centric Development Method (ACDM) – requirements documentation and specification – change management – traceability of requirements

Tutorial: Conduct QAW, elicit, analyze, prioritize, and document requirements using ACDM

UNIT III ESTIMATION, PLANNING, AND TRACKING
9

Tutorial: Estimation, planning, and tracking exercises

UNIT IV CONFIGURATION AND QUALITY MANAGEMENT
9
identifying artifacts to be configured – naming conventions and version control – configuration control – quality assurance techniques – peer reviews – Fegan inspection – unit, integration, system, and acceptance testing – test data and test cases – bug tracking – causal analysis

Tutorial: version control exercises, development of test cases, causal analysis of defects

UNIT V SOFTWARE PROCESS DEFINITION AND MANAGEMENT
9

Tutorial: process measurement exercises, process definition using ETVX

TOTAL 45+15=60 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
1. Explain software development life cycle
2. Adopt a suitable process for software development
3. Elicit functional and quality requirements
4. Analyze, prioritize, and manage requirements
5. Perform trade-off among conflicting requirements
6. Identify and prioritize risks and create mitigation plans
7. Estimate the efforts required for software development
8. Perform planning and tracking activities
9. Control the artifacts during software development
10. Perform various tests to ensure quality
11. Define new processes based on the needs
12. Adopt best practices for process improvement

REFERENCES:
OBJECTIVES:

- To understand the basic elements of Z
- To understand relations, functions, and logical structures in Z
- To understand Z schemas and schema calculus
- To learn selected Z case studies
- To understand Z schema refinement

UNIT I FOUNDATIONS OF Z

UNIT II STRUCTURES IN Z

UNIT III Z SCHEMAS AND SCHEMA CALCULUS

UNIT IV Z CASE STUDIES
Case Study: Text processing system – Case Study: Eight Queens – Case Study: Graphical User Interface – Case Study: Safety critical protection system – Case Study: Concurrency and real time systems

UNIT V Z REFINEMENT

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to

- Apply the basic elements of Z
- Develop relational, functional, and logical Z structures
- Develop Z schema as models of software systems
- Perform verifications and conduct proofs using Z models
- Refine Z models towards implementing software systems

REFERENCES:
OBJECTIVES:
- To understand the mathematical foundations needed for performance evaluation of computer systems
- To understand the metrics used for performance evaluation
- To understand the analytical modeling of computer systems
- To enable the students to develop new queueing analysis for both simple and complex systems
- To appreciate the use of smart scheduling and introduce the students to analytical techniques for evaluating scheduling policies

UNIT I  OVERVIEW OF PERFORMANCE EVALUATION  9

UNIT II  MARKOV CHAINS AND SIMPLE QUEUES  9
Discrete-Time Markov Chains – Ergodicity Theory – Real World Examples – Google, Aloha – Transition to Continuous-Time Markov Chain – M/M/1 and PASTA.

UNIT III  MULTI-SERVER AND MULTI-QUEUE SYSTEMS  9
Server Farms: M/M/k and M/M/k/k – Capacity Provisioning for Server Farms – Time Reversibility and Burke’s Theorem – Networks of Queues and Jackson Product Form – Classed and Closed Networks of Queues.

UNIT IV  REAL-WORLD WORKLOADS  9

UNIT V  SMART SCHEDULING IN THE M/G/1  9

OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the need for performance evaluation and the metrics used for it
- Discuss open and closed queueing networks
- Define Little’s law and other operational laws
- Apply the operational laws to open and closed systems
- Use discrete-time and continuous-time Markov chains to model real world systems
- Develop analytical techniques for evaluating scheduling policies

REFERENCES:

CP7002 PROBABILISTIC REASONING SYSTEMS

OBJECTIVES:
- To construct and reason with Bayesian networks
- To reason with temporal models
- To make exact and approximate inferences with graphical models
- To understand learning of parameters for probabilistic graphical models
- To understand actions and decisions with probabilistic graphical models

UNIT I REPRESENTATION

UNIT II TEMPLATE BASED REPRESENTATION

UNIT III INFERENCE

UNIT IV LEARNING

UNIT V ACTIONS AND DECISIONS
Causality – Utilities and decisions – Structured decision problems

OUTCOMES:
Upon Completion of the course, the students will be able to
- Construct Bayesian networks
- Reason with Bayesian networks
- Reason with Dynamic networks and Hidden Markov Models
- Conduct inferences with Bayesian networks
- Implement algorithms to learn probabilistic graphical models
- Explain actions and decisions with probabilistic graphical models

REFERENCES:

---

**CP7003 DATA ANALYSIS AND BUSINESS INTELLIGENCE**  
**L T P C**  
**3 0 0 3**

**OBJECTIVES:**
- To understand linear regression models
- To understand logistic regression models
- To understand generalized linear models
- To understand simulation using regression models
- To understand causal inference
- To understand multilevel regression
- To understand data collection and model understanding

**UNIT I LINEAR REGRESSION**  
9
- Introduction to data analysis
- Statistical processes
- Statistical models
- Statistical inference
- Review of random variables and probability distributions
- Linear regression
- One predictor
- Multiple predictors
- Prediction and validation
- Linear transformations
- Centering and standardizing
- Correlation
- Logarithmic transformations
- Other transformations
- Building regression models
- Fitting a series of regressions

**UNIT II LOGISTIC AND GENERALIZED LINEAR MODELS**  
9
- Logistic regression
- Logistic regression coefficients
- Latent-data formulation
- Building a logistic regression model
- Logistic regression with interactions
- Evaluating, checking, and comparing fitted logistic regressions
- Identifiability and separation
- Poisson regression
- Logistic-binomial model
- Probit regression
- Multinomial regression
- Robust regression using t model
- Building complex generalized linear models
- Constructive choice models

**UNIT III SIMULATION AND CAUSAL INFERENCE**  
9
- Simulation of probability models
- Summarizing linear regressions
- Simulation of non-linear predictions
- Predictive simulation for generalized linear models
- Fake-data simulation
- Simulating and comparing to actual data
- Predictive simulation to check the fit of a time-series model
- Causal inference
- Randomized experiments
- Observational studies
- Causal inference using advanced models
- Matching
- Instrumental variables

**UNIT IV MULTILEVEL REGRESSION**  
9
- Multilevel structures
- Clustered data
- Multilevel linear models
- Partial pooling
- Group-level predictors
- Model building and statistical significance
- Varying intercepts and slopes
- Scaled inverse-Wishart distribution
- Non-nested models
- Multi-level logistic regression
- Multi-level generalized linear models

**UNIT V DATA COLLECTION AND MODEL UNDERSTANDING**  
9
- Design of data collection
- Classical power calculations
- Multilevel power calculations
- Power calculation using fake-data simulation
- Understanding and summarizing fitted models
- Uncertainty and variability
- Variances
- \( R^2 \) and explained variance
- Multiple comparisons and statistical significance
- Analysis of variance
- ANOVA and multilevel linear and general linear models
- Missing data imputation

**TOTAL: 45 PERIODS**
OUTCOMES:
Upon Completion of the course, the students will be able to
- Build and apply linear regression models
- Build and apply logistic regression models
- Build and apply generalized linear models
- Perform simulation using regression models
- Perform casual inference from data
- Build and apply multilevel regression models
- Perform data collection and variance analysis

REFERENCES:

CP7004                          IMAGE PROCESSING AND ANALYSIS                  L T P C
                                      3 0 0 3

OBJECTIVES:
- To understand the basics of digital images
- To understand noise models
- To understand spatial domain filters
- To understand frequency domain filters
- To learn basic image analysis --- segmentation, edge detection, and corner detection
- To learn morphological operations and texture analysis
- To understand processing of color images
- To understand image compression techniques

UNIT I         SPATIAL DOMAIN PROCESSING

UNIT II       FREQUENCY DOMAIN PROCESSING
UNIT III SEGMENTATION AND EDGE DETECTION

UNIT IV INTEREST POINTS, MORPHOLOGY, AND TEXTURE

UNIT V COLOR IMAGES AND IMAGE COMPRESSION

TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Explain image modalities, sensing, acquisition, sampling, and quantization
- Explain image noise models
- Implement spatial filter operations
- Explain frequency domain transformations
- Implement frequency domain filters
- Apply segmentation algorithms
- Apply edge detection techniques
- Apply corner and interest point detection algorithms
- Apply morphological operations
- Perform texture analysis
- Analyze color images
- Implement image compression algorithms

REFERENCES:
OBJECTIVES:
- To study the sensor characteristics and the fundamental principles of sensing
- To understand the sensor interface electronics
- To study selected motion-related sensors
- To study light and radiation detectors
- To study selected temperature sensors
- To study selected chemical sensors

UNIT I PRINCIPLES OF SENSING

UNIT II OPTICAL COMPONENTS AND INTERFACE ELECTRONICS

UNIT III MOTION RELATED SENSORS

UNIT IV LIGHT AND RADIATION DETECTORS
Light Detectors: Photo diodes – photo transistor – photo resistor – cooled detectors – CCD and CMOS image sensors – thermal detectors – optical design – gas flame detectors
Radiation Detectors: scintillating detectors – ionization detectors – cloud and bubble chambers

UNIT V TEMPERATURE AND CHEMICAL SENSORS

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Explain sensor characteristics
- Explain the physics of sensors
- Explain optical components of sensors
- Apply sensor interface electronics
- Choose and use appropriate motion-related sensors
- Choose and use appropriate light and radiation detectors
- Choose and use appropriate temperature sensors
- Choose and use appropriate chemical sensors

REFERENCE:
OBJECTIVES:
- To understand the mathematical foundations needed for understanding and designing randomized algorithms
- To appreciate the need for randomized algorithms
- To expose the students to probabilistic methods
- To understand the concept of random walk
- To expose the students to different types of applications of randomized algorithms

UNIT I INTRODUCTION TO RANDOMIZED ALGORITHMS

UNIT II PROBABILISTIC METHODS

UNIT III ALGEBRAIC TECHNIQUES AND APPLICATIONS

UNIT IV GEOMETRIC AND GRAPH ALGORITHMS

UNIT V HASHING AND ONLINE ALGORITHMS

TOTAL: 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the need for randomized algorithms
- Discuss the classification of randomized algorithms
- Present the various paradigms for designing randomized algorithms
- Discuss the different probabilistic methods used for designing randomized algorithms
- Apply the techniques studied to design algorithms for different applications like matrix multiplication, hashing, linear programming

REFERENCES:
OBJECTIVES:

- To understand the basics of Mobile Computing and Personal Computing
- To learn the role of cellular networks in Mobile and Pervasive Computing
- To expose to the concept of sensor and mesh networks
- To expose to the context aware and wearable computing
- To learn to develop applications in mobile and pervasive computing environment

UNIT I  INTRODUCTION  9

UNIT II  3G AND 4G CELLULAR NETWORKS  9

UNIT III  SENSOR AND MESH NETWORKS  9

UNIT IV  CONTEXT AWARE COMPUTING & WEARABLE COMPUTING  9
Health BAN – Medical and Technological Requirements-Wearable Sensors-Intra-BAN communications

UNIT V  APPLICATION DEVELOPMENT  9
Three tier architecture – Model View Controller Architecture – Memory Management – Information Access Devices – PDAs and Smart Phones – Smart Cards and Embedded Controls – J2ME – Programming for CLDC – GUI in MIDP – Application Development ON Android and iPhone

TOTAL: 45 PERIODS

OUTCOMES:

At the end of the course the student should be able to

- Design a basic architecture for a pervasive computing environment
- Design and allocate the resources on the 3G-4G wireless networks
- Analyze the role of sensors in Wireless networks
- Work out the routing in mesh network
- Deploy the location and context information for application development
- Develop mobile computing applications based on the paradigm of context aware computing and wearable computing
REFERENCES:

CP7006 PARALLEL PROGRAMMING PARADIGMS

OBJECTIVES:
- To understand models of and issues in concurrency in computing
- To develop message-passing parallel programs using MPI
- To develop shared-memory parallel programs using Pthreads
- To develop shared-memory parallel programs using OpenMP
- To use GPU for parallel programming using OpenCL and CUDA

UNIT I FOUNDATIONS OF PARALLEL PROGRAMMING

UNIT II MESSAGE PASSING PARADIGM

UNIT III SHARED MEMORY PARADIGM: PTHREADS

UNIT IV SHARED MEMORY PARADIGM: OPENMP
UNIT V  GRAPHICAL PROCESSING PARADIGMS: OPENCL AND CUDA

Introduction to CUDA – CUDA programming examples – CUDA execution model – CUDA memory hierarchy – CUDA case study - introduction to OpenCL – OpenCL programming examples – Programs and Kernels – Buffers and Images – Event model – OpenCL case study

TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Explain models of parallel programming
- Explain hardware level support for concurrency
- Explain issues in parallel programming
- Develop message-passing parallel programs using MPI framework
- Develop shared-memory parallel programs using Pthreads
- Develop shared-memory parallel programs using OpenMP
- Develop CUDA programs
- Develop OpenCL programs

REFERENCES:

CP7007  SOFTWARE REQUIREMENTS ENGINEERING  L T P C

OBJECTIVES:
1. Understand system requirements
2. Identify different types of requirement
3. Generate requirements be elicitation
4. Develop requirements documentation
5. Evaluate the requirements

UNIT I  DOMAIN UNDERSTANDING

UNIT II  REQUIREMENTS ELICITATION
Introduction – Understanding stakeholders’ needs – Elicitation techniques – interviews, questionnaire, workshop, brainstorming, prototyping – Documenting stakeholders’ needs
UNIT III  FUNCTIONAL REQUIREMENTS
Introduction – Features and Use cases – Use case scenarios – Documenting use cases – Levels of details – SRS documents.

UNIT IV  QUALITY ATTRIBUTES AND USER EXPERIENCE
Quality of solution – Quality attributes – Eliciting quality attributes – Quality attribute workshop (QAW) – Documenting quality attributes – Six part scenarios – Usability requirements – Eliciting and documenting usability requirements – Modeling user experience – Specifying UI design

UNIT V  MANAGING REQUIREMENTS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Define a process for requirements engineering
- Execute a process for gathering requirements through elicitation techniques.
- Validate requirements according to criteria such as feasibility, clarity, preciseness etc.
- Develop and document functional requirements for different types of systems.
- Develop and document quality attributes of the system to be implemented
- Communicate the requirements to stakeholders
- Negotiate with stakeholders in order to agree on a set of requirements.
- Detect and resolve feature interactions

REFERENCES:

CP7008  SPEECH PROCESSING AND SYNTHESIS  L T P C  3 0 0 3

OBJECTIVES:
- To understand the mathematical foundations needed for speech processing
- To understand the basic concepts and algorithms of speech processing and synthesis
- To familiarize the students with the various speech signal representation, coding and recognition techniques
- To appreciate the use of speech processing in current technologies and to expose the students to real-world applications of speech processing

UNIT I  FUNDAMENTALS OF SPEECH PROCESSING
UNIT II  SPEECH SIGNAL REPRESENTATIONS AND CODING  

UNIT III  SPEECH RECOGNITION  

UNIT IV  TEXT ANALYSIS  

UNIT V  SPEECH SYNTHESIS  

TOTAL: 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the various temporal, spectral and cepstral features required for identifying speech units – phoneme, syllable and word
- Determine and apply Mel-frequency cepstral coefficients for processing all types of signals
- Justify the use of formant and concatenated approaches to speech synthesis
- Identify the apt approach of speech synthesis depending on the language to be processed
- Determine the various encoding techniques for representing speech.

REFERENCES:

CP7009  MACHINE LEARNING TECHNIQUES  
OBJECTIVES:
1. To understand the machine learning theory
2. To implement linear and non-linear learning models
3. To implement distance-based clustering techniques
4. To build tree and rule based models
5. To apply reinforcement learning techniques

UNIT I  FOUNDATIONS OF LEARNING  
UNIT II   LINEAR MODELS

UNIT III   DISTANCE-BASED MODELS

UNIT IV   TREE AND RULE MODELS

UNIT V   REINFORCEMENT LEARNING

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- To explain theory underlying machine learning
- To construct algorithms to learn linear and non-linear models
- To implement data clustering algorithms
- To construct algorithms to learn tree and rule-based models
- To apply reinforcement learning techniques

REFERENCES:

CP7010   CONCURRENCY MODELS
OBJECTIVES:
- To model concurrency in FSP
- To specify and check safety and liveness properties
- To understand concurrency architectures and design
- To apply linear temporal logic to safety and liveness analysis
- To apply Petri nets for concurrency modeling and analysis
UNIT I  FSP AND GRAPH MODELS  9

UNIT II  SAFETY AND LIVENESS PROPERTIES  9

UNIT III  CONCURRENCY ARCHITECTURES AND DESIGN  9

UNIT IV  LINEAR TEMPORAL LOGIC (LTL)  9

UNIT V  PETRI NETS  9

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Develop concurrency models and FSP
- State safety and liveness properties in FSP
- Verify properties using LTSA tool
- Explain concurrency architectures
- Design concurrent Java programs from models
- Apply Linear Temporal Logic to state safety and liveness properties
- Assert LTL properties in FSP and check using LTSA tool
- Model and analyze concurrency using Petri nets

REFERENCES:
OBJECTIVES:
- To provide good understanding of fundamental concepts in real time systems.
- To provide understanding of advanced topics in real time systems.
- To provide understanding on basic multi-task scheduling algorithms for periodic, aperiodic, and sporadic tasks as well as understand the impact of the latter two on scheduling.
- To expose to understand capabilities of commercial off-the-shelf R-T kernel.
- To expose to real time communications and databases.

UNIT I
INTRODUCTION
9

UNIT II
SCHEDULING REAL-TIME TASKS
9

UNIT III
RESOURCE SHARING AMONG RT TASKS & SCHEDULING RT TASKS
9

UNIT IV
COMMERCIAL RT OPERATING SYSTEMS
9

UNIT V
RT COMMUNICATION & DATABASES
9

TOTAL: 45 PERIODS

OUTCOMES:
- Understand the basics and importance of real-time systems
- Generate a high-level analysis document based on requirements specifications
- Generate a high-level design document based on analysis documentation
- Generate a test plan based on requirements specification
- Generate a validation plan based on all documentation
- Understand basic multi-task scheduling algorithms for periodic, aperiodic, and sporadic tasks as well as understand the impact of the latter two on scheduling
- Understand capabilities of at least one commercial off-the-shelf R-T kernel

REFERENCES:
OBJECTIVES:
- To review image processing techniques for computer vision
- To understand shape and region analysis
- To understand Hough Transform and its applications to detect lines, circles, ellipses
- To understand three-dimensional image analysis techniques
- To understand motion analysis
- To study some applications of computer vision algorithms

UNIT I  IMAGE PROCESSING FOUNDATIONS  9
Review of image processing techniques – classical filtering operations – thresholding techniques – edge detection techniques – corner and interest point detection – mathematical morphology – texture

UNIT II  SHAPES AND REGIONS  9

UNIT III  HOUGH TRANSFORM  9

UNIT IV  3D VISION AND MOTION  9

UNIT V  APPLICATIONS  9
Application: Photo album – Face detection – Face recognition – Eigen faces – Active appearance and 3D shape models of faces
Application: In-vehicle vision system: locating roadway – road markings – identifying road signs – locating pedestrians

OUTCOMES:
Upon completion of the course, the students will be able to
- Implement fundamental image processing techniques required for computer vision
- Perform shape analysis
- Implement boundary tracking techniques
- Apply chain codes and other region descriptors
- Apply Hough Transform for line, circle, and ellipse detections
- Apply 3D vision techniques
- Implement motion related techniques
- Develop applications using computer vision techniques

TOTAL : 45 PERIODS
REFERENCES:

NE7202 NETWORK AND INFORMATION SECURITY L T P C 3 0 0 3

OBJECTIVES:
- To understand the fundamentals of Cryptography
- To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
- To understand the various key distribution and management schemes.
- To understand how to deploy encryption techniques to secure data in transit across data networks
- To design security applications in the field of Information technology

UNIT I INTRODUCTION 9

UNIT II CRYPTOSYSTEMS & AUTHENTICATION 9

UNIT III PUBLIC KEY CRYPTOSYSTEMS 9
Introduction to Public key Cryptography- Number theory- The RSA Cryptosystem and Factoring Integer- Attacks on RSA-The ELGamal Cryptosystem- Digital Signature Algorithm-Finite Fields-Elliptic Curves Cryptography- Key management – Session and Interchange keys, Key exchange and generation-PKI

UNIT IV SYSTEM IMPLEMENTATION 9
Design Principles, Representing Identity, Access Control Mechanisms, Information Flow and Confinement Problem
Secure Software Development: Secured Coding - OWASP/SANS Top Vulnerabilities - Buffer Overflows - Incomplete mediation - XSS - Anti Cross Site Scripting Libraries - Canonical Data Format - Command Injection - Redirection - Inference – Application Controls

UNIT V NETWORK SECURITY 9
Secret Sharing Schemes-Kerberos- Pretty Good Privacy (PGP)-Secure Socket Layer (SSL)-Intruders – HIDS- NIDS - Firewalls - Viruses

TOTAL: 45 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to
- Implement basic security algorithms required by any computing system.
- Analyze the vulnerabilities in any computing system and hence be able to design a security solution.
- Analyze the possible security attacks in complex real-time systems and their effective countermeasures.
- Identify the security issues in the network and resolve it.
- Evaluate security mechanisms using rigorous approaches, including theoretical derivation, modeling, and simulations.
- Formulate research problems in the computer security field.

REFERENCES:

CP7013 DESIGN AND ANALYSIS OF PARALLEL ALGORITHMS

OBJECTIVES:
- To understand the need for parallel algorithms.
- To expose the students to different models of parallel computation.
- To expose the students to parallel sorting and searching algorithms.
- To understand the application of the concepts studied to different types of problems.
- To analyze parallel algorithms.

UNIT I INTRODUCTION

UNIT II SORTING AND SEARCHING
Merging on the EREW and CREW Models - Fast Merging on EREW - Sorting Networks – Sorting on a Linear Array – Sorting on CRCW, CREW, EREW Models – Searching a Sorted Sequence – Searching a Random Sequence.
UNIT III  ALGEBRAIC PROBLEMS  9
Generating Permutations and Combinations in Parallel – Matrix Transpositions – Matrix by Matrix Multiplications – Matrix by Vector multiplication.

UNIT IV  GRAPH THEORY AND COMPUTATIONAL GEOMETRY PROBLEMS  9

UNIT V DECISION AND OPTIMIZATION PROBLEMS  9
Computing Prefix Sums – Applications - Job Sequencing with Deadlines – Knapsack Problem-The Bit Complexity of Parallel Computations.

TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the need for parallel algorithms
- Discuss the classification of parallel architectures and identify suitable programming models
- Perform sorting on CRCW, CREW, EREW Models
- Search a sorted as well as random sequence
- Develop and analyze algorithms for different applications like matrix multiplication, shortest path, job sequencing and the knapsack problem.

REFERENCES:

CP7014 SOFTWARE ARCHITECTURES L T P C 3 0 0 3

OBJECTIVES:
- Understand architectural requirements
- Identify architectural structures
- Develop architectural documentation
- Generate architectural alternatives
- Evaluate the architecture against the drivers

UNIT I ARCHITECTURAL DRIVERS  9

UNIT II ARCHITECTURAL VIEWS AND DOCUMENTATION  9
UNIT III ARCHITECTURAL STYLES
Introduction – Data flow styles – Call-return styles – Shared Information styles – Event styles – Case studies for each style

UNIT IV ARCHITECTURAL DESIGN
Approaches for architectural design – System decomposition – Attributes driven design – Architecting for specific quality attributes – Performance, Availability – Security – Architectural conformance

UNIT V ARCHITECTURE EVALUATION AND SOME SPECIAL TOPICS
Need for evaluation – Scenario based evaluation against the drivers – ATAM and its variations – Case studies in architectural evaluations – SOA and Web services – Cloud Computing – Adaptive structures

OUTCOMES:
Upon Completion of the course, the students will be able to
- Explain key architectural drivers
- Explain the influence of architecture on business and technical activities
- Identify key architectural structures
- Adopt good practices for documenting the architecture
- Develop alternative architectures for a given problem
- Explain how to use formal languages to specify architecture
- Evaluate the architecture against the drivers
- Describe the recent trends in software architecture

REFERENCES:

CP7015 MODEL CHECKING AND PROGRAM VERIFICATION L T P C
3 0 0 3

OBJECTIVES:
- To understand automata for model checking
- To understand LTL, CTL, and CTL*
- To understand timed automata, TCTL, and PCTL
- To understand verification of deterministic and recursive programs
- To understand verification of object-oriented programs
- To understand verification of parallel, distributed, and non-deterministic programs
UNIT I AUTOMATA AND TEMPORAL LOGICS
Automata on finite words – model checking regular properties – automata on infinite words – Buchi automata – Linear Temporal Logic (LTL) – automata based LTL model checking – Computational Tree Logic (CTL) – CTL model checking – CTL* model checking

UNIT II TIMED AND PROBABILISTIC TREE LOGICS

UNIT III VERIFYING DETERMINISTIC AND RECURSIVE PROGRAMS
Introduction to program verification – verification of “while” programs – partial and total correctness – verification of recursive programs – case study: binary search – verifying recursive programs with parameters

UNIT IV VERIFYING OBJECT-ORIENTED AND PARALLEL PROGRAMS

UNIT V VERIFYING NON-DETERMINISTIC AND DISTRIBUTED PROGRAMS

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Perform model checking using LTL
- Perform model checking using CTL
- Perform model checking using CTL*
- Perform model checking using TCTL and PCTL
- Verify deterministic and recursive programs
- Verify object-oriented programs
- Verify parallel, distributed, and non-deterministic programs

REFERENCES:
OBJECTIVES:
- To understand processors and their instruction sets for embedded systems
- To understand hardware platform for embedded systems
- To design and analyze programs for embedded systems
- To design multi-tasking embedded systems with RTOS
- To understand overall embedded systems development lifecycle
- To understand distributed and multi-processor embedded systems

UNIT I PROCESSORS AND INSTRUCTION SETS
Introduction to embedded computing – overview of embedded system design process – instruction sets of processors: ARM, PIC, TI C55x, TI C64x – programming I/O – modes and exceptions – coprocessors – memory system – CPU performance – CPU power consumption

UNIT II EMBEDDED COMPUTING PLATFORM

UNIT III PROGRAM DESIGN AND ANALYSIS
Components for embedded programs – models of programs – Assembly, linking, and loading – compiler optimizations – program-level performance analysis – performance optimization – program-level energy optimization – optimizing program size – program validation and testing – design example: Digital Still Camera

UNIT IV PROCESSES AND OPERATING SYSTEMS

UNIT V SYSTEM DESIGN, NETWORKS, AND MULTIPROCESSORS
System design methodologies – requirements analysis – specifications – architecture design – quality assurance – distributed embedded systems – shared-memory multiprocessors – design example: Video accelerator

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Develop assembly code for processors such as ARM, PIC Microcontroller, TI C55x, TI C64x, etc.
- Choose appropriate hardware platform for a given application
- Perform platform-level performance analysis
- Design, develop, and debug embedded programs optimized for size or performance
- Develop embedded applications using an RTOS
- Perform OS-level performance analysis
- Employ best practices in embedded software engineering
- Develop distributed embedded systems and systems with shared-memory concurrency

REFERENCES:
OBJECTIVES:
- To introduce the broad perceptive of cloud architecture and model
- To understand the concept of Virtualization
- To be familiar with the lead players in cloud.
- To understand the features of cloud simulator
- To apply different cloud programming model as per need.
- To be able to set up a private cloud.
- To understand the design of cloud Services.
- To learn to design the trusted cloud Computing system

UNIT I CLOUD ARCHITECTURE AND MODEL

UNIT II VIRTUALIZATION

UNIT III CLOUD INFRASTRUCTURE

UNIT IV PROGRAMMING MODEL
Parallel and Distributed Programming Paradigms – MapReduce, Twister and Iterative MapReduce – Hadoop Library from Apache – Mapping Applications - Programming Support - Google App Engine, Amazon AWS - Cloud Software Environments -Eucalyptus, Open Nebula, OpenStack, Aneka, CloudSim

UNIT V SECURITY IN THE CLOUD

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player
- Choose the appropriate Programming Models and approach.
- Address the core issues of cloud computing such as security, privacy and interoperability
- Design Cloud Services
- Set a private cloud
REFERENCES:
5. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud” O'Reilly

CP7017 DATA VISUALIZATION TECHNIQUES L T P C 3 0 0 3

OBJECTIVES:
- To introduce visual perception and core skills for visual analysis
- To understand visualization for time-series analysis
- To understand visualization for ranking analysis
- To understand visualization for deviation analysis
- To understand visualization for distribution analysis
- To understand visualization for correlation analysis
- To understand visualization for multivariate analysis
- To understand issues and best practices in information dashboard design

UNIT I CORE SKILLS FOR VISUAL ANALYSIS

UNIT II TIME-SERIES, RANKING, AND DEVIATION ANALYSIS

UNIT III DISTRIBUTION, CORRELATION, AND MULTIVARIATE ANALYSIS
UNIT IV INFORMATION DASHBOARD DESIGN I

UNIT V INFORMATION DASHBOARD DESIGN II

TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Explain principles of visual perception
- Apply core skills for visual analysis
- Apply visualization techniques for various data analysis tasks
- Design information dashboard

REFERENCES:
5. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", O'Reilly, 2008.

NE7005 PROTOCOLS AND ARCHITECTURE FOR WIRELESS SENSOR NETWORKS

UNIT I INTRODUCTION AND OVERVIEW OF WIRELESS SENSOR NETWORKS

UNIT II ARCHITECTURES
UNIT III DEPLOYMENT AND CONFIGURATION 9

UNIT IV ROUTING PROTOCOLS AND DATA MANIPULATION 9
Data Centric and Content based Routing, Storage and Retrieval in Network, Compression Technologies for WSN, Data Aggregation Technique.

UNIT V SENSOR NETWORK PLATFORMS AND TOOLS 9

TOTAL : 45 PERIODS

REFERENCES:

CP7018 LANGUAGE TECHNOLOGIES L T P C
3 0 0 3

OBJECTIVES:
- To understand the mathematical foundations needed for language processing
- To understand the representation and processing of Morphology and Part-of-Speech Taggers
- To understand different aspects of natural language syntax and the various methods used for processing syntax
- To understand different methods of disambiguating word senses
- To know about various applications of natural language processing
- To learn the indexing and searching processes of a typical information retrieval system and to study NLP based retrieval systems
- To gain knowledge about typical text categorization and clustering techniques

UNIT I INTRODUCTION 9

UNIT II WORDS 9
UNIT III  GRAMMAR  9

UNIT IV  INFORMATION RETRIEVAL  9

UNIT V  TEXT MINING  9

OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the different linguistic components of given sentences
- Design a morphological analyser for a language of your choice using finite state automata concepts
- Implement a parser by providing suitable grammar and words
- Discuss algorithms for word sense disambiguation
- Build a tagger to semantically tag words using WordNet
- Design an application that uses different aspects of language processing.

REFERENCES:

NE7012  SOCIAL NETWORK ANALYSIS  L T P C
OBJECTIVES:
- To understand the components of the social network
- To model and visualize the social network
- To mine the users in the social network
- To understand the evolution of the social network
- To mine the interest of the user

UNIT I  INTRODUCTION  9
UNIT II  MODELING AND VISUALIZATION

UNIT III  MINING COMMUNITIES
Aggregating and reasoning with social network data, Advanced Representations - Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Evaluating Communities – Core Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Node Classification in Social Networks.

UNIT IV  EVOLUTION

UNIT V  TEXT AND OPINION MINING
Text Mining in Social Networks - Opinion extraction – Sentiment classification and clustering - Temporal sentiment analysis - Irony detection in opinion mining - Wish analysis - Product review mining – Review Classification – Tracking sentiments towards topics over time

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Work on the internals components of the social network
- Model and visualize the social network
- Mine the behaviour of the users in the social network
- Predict the possible next outcome of the social network
- Mine the opinion of the user

REFERENCES:

CP7019  MANAGING BIG DATA

OBJECTIVES:
- Understand big data for business intelligence
- Learn business case studies for big data analytics
- Understand nosql big data management
- Perform map-reduce analytics using Hadoop and related tools

52
UNIT I UNDERSTANDING BIG DATA

UNIT II NOSQL DATA MANAGEMENT

UNIT III BASICS OF HADOOP

UNIT IV MAPREDUCE APPLICATIONS
MapReduce workflows – unit tests with MRUnit – test data and local tests – anatomy of MapReduce job run – classic Map-reduce – YARN – failures in classic Map-reduce and YARN – job scheduling – shuffle and sort – task execution – MapReduce types – input formats – output formats

UNIT V HADOOP RELATED TOOLS

OUTCOMES:
Upon Completion of the course, the students will be able to
• Describe big data and use cases from selected business domains
• Explain NoSQL big data management
• Install, configure, and run Hadoop and HDFS
• Perform map-reduce analytics using Hadoop
• Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics

REFERENCES:
OBJECTIVES:
- Understand system requirements for mobile applications
- Generate suitable design using specific mobile development frameworks
- Generate mobile application design
- Implement the design using specific mobile development frameworks
- Deploy the mobile applications in marketplace for distribution

UNIT I  INTRODUCTION  5
Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications

UNIT II  BASIC DESIGN  8
Introduction – Basics of embedded systems design – Embedded OS - Design constraints for mobile applications, both hardware and software related – Architecting mobile applications – User interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

UNIT III  ADVANCED DESIGN  8
Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications.

UNIT IV  TECHNOLOGY I - ANDROID  12

UNIT V  TECHNOLOGY II - IOS  12
Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wifi - iPhone marketplace.

TOTAL: 45 PERIODS

OUTCOMES:
Upon the students will be able to
1. Describe the requirements for mobile applications
2. Explain the challenges in mobile application design and development
3. Develop design for mobile applications for specific requirements
4. Implement the design using Android SDK
5. Implement the design using Objective C and iOS
6. Deploy mobile applications in Android and iPhone marketplace for distribution

REFERENCES:
OBJECTIVES:
- Learn evolutionary theory and algorithms
- Understand Cellular Automata and artificial life
- Learn artificial neural systems and related learning algorithms
- Learn developmental and artificial immune systems
- Understand behavioral systems especially in the context of Robotics
- Understand collective systems such as ACO, PSO, and swarm robotics

UNIT I EVOLUTIONARY AND CELLULAR SYSTEMS
9

UNIT II NEURAL SYSTEMS
9

UNIT III DEVELOPMENTAL AND IMMUNE SYSTEMS
9
Rewriting systems – synthesis of developmental systems – evolutionary rewriting systems – evolutionary developmental programs
Biological immune systems – lessons for artificial immune systems – algorithms and applications – shape space – negative selection algorithm – clonal selection algorithm - examples

UNIT IV BEHAVIORAL SYSTEMS
9

UNIT V COLLECTIVE SYSTEMS
9
Biological self-organization – Particle Swarm Optimization (PSO) – ant colony optimization (ACO) – swarm robotics – co-evolutionary dynamics – artificial evolution of competing systems – artificial evolution of cooperation – case study

TOTAL: 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Implement and apply evolutionary algorithms
- Explain cellular automata and artificial life
- Implement and apply neural systems
- Explain developmental and artificial immune systems
- Explain behavioral systems
- Implement and apply collective intelligence systems

REFERENCES:
An advanced graduate level course on medical imaging and medical image analysis. The course includes topics in medical image formation, medical imaging techniques, such as X-Ray, Computed Tomography, Magnetic Resonance Imaging, and Nuclear Imaging, image segmentation, registration, statistical modeling, visualization, and applications of computational tools for medicine.

OBJECTIVES:
The course will provide the participants with an up-to-date background in current state-of-the-art in medical imaging and medical image analysis. The aim of the course is to show how to extract, model, and analyze information from medical data and applications in order to help diagnosis, treatment and monitoring of diseases through computer science.

UNIT I  INTRODUCTION
Introduction to medical imaging technology, systems, and modalities. Brief history; importance; applications; trends; challenges. Medical Image Formation Principles: X-Ray physics; X-Ray generation, attenuation, scattering; dose Basic principles of CT; reconstruction methods; artifacts; CT hardware.

UNIT II  STORAGE AND PROCESSING
Medical Image Storage, Archiving and Communication Systems and Formats Picture archiving and communication system (PACS); Formats: DICOM Radiology Information Systems (RIS) and Hospital Information Systems (HIS). Medical Image Processing, Enhancement, Filtering Basic image processing algorithms Thresholding; contrast enhancement; SNR characteristics; filtering; histogram modeling.

UNIT III  VISUALIZATION
Medical Image Visualization Fundamentals of visualization; surface and volume rendering/visualization; animation; interaction. Magnetic Resonance Imaging (MRI) Mathematics of MR; spin physics; NMR spectroscopy; imaging principles and hardware; image artifacts.

UNIT IV  SEGMENTATION AND CLASSIFICATION
Medical Image Segmentation - Histogram-based methods; Region growing and watersheds; Markov Random Field models; active contours; model-based segmentation. Multi-scale segmentation; semi-automated methods; clustering-based methods; classification-based methods; atlas-guided approaches; multi-model segmentation. Medical Image Registration Intensity-based methods; cost functions; optimization techniques.

UNIT V  NUCLEAR IMAGING
PET and SPECT Ultrasound Imaging methods; mathematical principles; resolution; noise effect; 3D imaging; positron emission tomography; single photon emission tomography; ultrasound imaging; applications. Medical Image Search and Retrieval Current technology in medical image search, content-based image retrieval, new trends: ontologies. Applications. Other Applications of Medical Imaging Validation, Image Guided Surgery, Image Guided Therapy, Computer Aided Diagnosis/Diagnostic Support Systems.

TOTAL: 45 PERIODS
REFERENCES:

CP7022 SOFTWARE DESIGN L T P C 3 0 0 3

OBJECTIVES:
- Analyze specifications
- Describe approaches to design
- Develop design documentation
- Evaluate the design

UNIT I SOFTWARE DESIGN PRINCIPLES 9

UNIT II OO DESIGN 9
Object model – Classes and objects – Object oriented analysis – Key abstractions and mechanisms – Object oriented design – Identifying design elements – Detailed design – Case studies.

UNIT III DESIGN PATTERNS 9

UNIT IV FUNCTION AND SERVICE ORIENTED DESIGNS 9
Structural decomposition – Detailed Design – Function oriented design Case study – Services – Service identification – Service design – Service composition – choreography and orchestration – Service oriented design Case study

UNIT V USER CENTERED DESIGN AND DESIGN REVIEW 9

TOTAL : 45 PERIODS

OUTCOMES:
Upon completion of the course, the students will be able to
- Describe different approaches to designing a software application
- Analyze specifications and identify appropriate design strategies.
- Develop an appropriate design for a given set of requirements
- Identify applicable design patterns for the solution
- Abstract and document reusable design patterns
- Evaluate a given design against the specifications
REFERENCES:
6. Eric Gamma et al., "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional, 1994

CP7023                     RECONFIGURABLE COMPUTING                     L T P C
                                                              3 0 0 3

OBJECTIVES:
- To understand the need for reconfigurable computing
- To expose the students to various device architectures
- To examine the various reconfigurable computing systems
- To understand the different types of compute models for programming reconfigurable architectures
- To expose the students to HDL programming and familiarize with the development environment
- To expose the students to the various placement and routing protocols
- To develop applications with FPGAs

UNIT I     DEVICE ARCHITECTURE

UNIT II    RECONFIGURABLE COMPUTING ARCHITECTURES AND SYSTEMS

UNIT III   PROGRAMMING RECONFIGURABLE SYSTEMS

UNIT IV    MAPPING DESIGNS TO RECONFIGURABLE PLATFORMS

UNIT V     APPLICATION DEVELOPMENT WITH FPGAS
Case Studies of FPGA Applications – System on a Programmable Chip (SoPC) Designs.

TOTAL: 45 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to
- Identify the need for reconfigurable architectures
- Discuss the architecture of FPGAs
- Point out the salient features of different reconfigurable architectures
- Build basic modules using any HDL
- Develop applications using any HDL and appropriate tools
- Design and build an SoPC for a particular application

REFERENCES:

IF7013 ENERGY AWARE COMPUTING L T P C 3 0 0 3

OBJECTIVES:
This course examines the design of power efficient architecture, power and performance tradeoffs, restructuring of software and applications and standards for energy aware Hardware and Software. The objective of this course is:
- To know the fundamental principles energy efficient devices
- To study the concepts of Energy efficient storage
- To introduce energy efficient algorithms
- To enable the students to know energy efficient techniques involved to support real-time systems.
- To study Energy aware applications

UNIT I INTRODUCTION
Energy efficient network on chip architecture for multi core system-Energy efficient MIPS CPU core with fine grained run time power gating – Low power design of Emerging memory technologies.

UNIT II ENERGY EFFICIENT STORAGE
Disk Energy Management-Power efficient strategies for storage system-Dynamic thermal management for high performance storage systems-Energy saving technique for Disk storage systems

UNIT III ENERGY EFFICIENT ALGORITHMS

UNIT IV REAL TIME SYSTEMS

UNIT V ENERGY AWARE APPLICATIONS

TOTAL: 45 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to
- Design Power efficient architecture Hardware and Software.
- Analyze power and performance trade off between various energy aware storage devices.
- Implement various energy aware algorithms.
- Restructure the software and Hardware for Energy aware applications.
- Explore the Energy aware applications

REFERENCES:

CP7024 INFORMATION RETRIEVAL TECHNIQUES L T P C
3 0 0 3

OBJECTIVES:
- To understand the basics of Information Retrieval with pertinence to modeling, query operations and indexing
- To get an understanding of machine learning techniques for text classification and clustering
- To understand the various applications of Information Retrieval giving emphasis to Multimedia IR, Web Search
- To understand the concepts of digital libraries

UNIT I INTRODUCTION 8

UNIT II MODELING 10

UNIT III INDEXING 9

UNIT IV CLASSIFICATION AND CLUSTERING 8
Text Classification and Naïve Bayes – Vector Space Classification – Support vector machines and Machine learning on documents. Flat Clustering – Hierarchical Clustering –Matrix decompositions and latent semantic indexing – Fusion and Meta learning

UNIT V SEARCHING AND RANKING 10

TOTAL: 45 PERIODS
OUTCOMES:
Upon completion of the course, the students will be able to

- Build an Information Retrieval system using the available tools
- Identify and design the various components of an Information Retrieval system
- Apply machine learning techniques to text classification and clustering which is used for efficient Information Retrieval
- Analyze the Web content structure
- Design an efficient search engine

REFERENCES:

CP7025 DATA MINING TECHNIQUES L T P C
3 0 0 3
UNIT I INTRODUCTION TO DATA MINING 9
Introduction to Data Mining – Data Mining Tasks – Components of Data Mining Algorithms – Data Mining supporting Techniques – Major Issues in Data Mining – Measurement and Data – Data Preprocessing – Data sets

UNIT II OVERVIEW OF DATA MINING ALGORITHMS 9

UNIT III CLASSIFICATIONS 9

UNIT IV CLUSTER ANALYSIS 9

UNIT V ASSOCIATION RULE MINING AND VISUALIZATION 9

TOTAL: 45 PERIODS
IF7002 BIO INFORMATICS L T P C 3 0 0 3

OBJECTIVES:
- To get exposed to the domain of bioinformatics
- To understand the role of data warehousing and data mining for bioinformatics
- To learn to model bioinformatics based applications
- To understand how to deploy the pattern matching and visualization techniques in bioinformatics
- To study the Microarray technologies for genome expression

UNIT I INTRODUCTION
Need for Bioinformatics technologies – Overview of Bioinformatics technologies – Structural bioinformatics – Data format and processing – secondary resources- Applications – Role of Structural bioinformatics - Biological Data Integration System.

UNIT II DATAWAREHOUSING AND DATAMINING IN BIOINFORMATICS
DNA data analysis – Data warehousing architecture – data quality – Biomedical data analysis – Protein data analysis – Machine learning – Neural network architecture- Applications in bioinformatics

UNIT III MODELING FOR BIOINFORMATICS
Hidden markov modeling for biological data analysis – Sequence identification – Sequence classification – multiple alignment generation – Comparative modeling – Protein modeling – genomic modeling – Probabilistic modeling – Bayesian networks – Boolean networks - Molecular modeling – Computer programs for molecular modeling

UNIT IV PATTERN MATCHING AND VISUALIZATION

UNIT V MICROARRAY ANALYSIS

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Deploy the data warehousing and data mining techniques in Bioinformatics
- Model bioinformatics based applications
- Deploy the pattern matching and visualization techniques in bioinformatics
- Work on the protein sequences
- Use the Microarray technologies for genome expression
REFERENCES:

CP7026 SOFTWARE QUALITY ASSURANCE L T P C
3 0 0 3

OBJECTIVES:
- Describe approaches to quality assurance
- Understand quality models
- Evaluate the system based on the chosen quality model

UNIT I INTRODUCTION 9

UNIT II VERIFICATION 6
Introduction – Verification techniques – Inspections, reviews, walk-throughs – Case studies

UNIT III TEST GENERATION 12
Software testing- Validation – Test plan – Test cases - Test Generation – Equivalence partitioning – Boundary value analysis – Category partition method – Combinatorial generation - Decision tables – Examples and Case studies

UNIT IV STRUCTURAL TESTING 12
Introduction – Test adequacy criteria – Control flow graph – Coverages: block, conditions, multiple conditions, MC/DC, path – Data flow graph – Definition and use coverages – C-use, P-use, Def-clear, Def-use – Finite state machines – Transition coverage – Fault based testing – Mutation analysis – Case studies

UNIT V FUNCTIONAL TESTING 6
Introduction – Test adequacy criteria - Test cases from use cases – Exploratory testing - Integration, system, acceptance, regression testing – Testing for specific attributes: Performance, load and stress testing – Usability testing – Security testing - Test automation – Test oracles

TOTAL : 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Describe different approaches to testing software applications
- Analyze specifications and identify appropriate test generation strategies
- Develop an appropriate test design for a given test object
- Identify applicable measurements for the verification and validation effort
- Execute the test design
- Evaluate the testing effort based on adequate measures
REFERENCES:

CP7027 MULTI OBJECTIVE OPTIMIZATION TECHNIQUES L T P C 3 0 0 3

OBJECTIVES:
• Learn fundamental principles of Multiobjective Optimization (MOP)
• Survey different Multiobjective Optimization algorithms
• Introduce various design issues of MOP
• Develop and Evaluate MOP Algorithms
• Learn Parallel and hybrid MOP Algorithms
• Learn other Metaheuristics

UNIT I INTRODUCTION AND CLASSICAL APPROACHES 9

UNIT II MOP EVOLUTIONARY ALGORITHMS 9

UNIT III THEORETICAL ISSUES 9
Fitness Landscapes - Fitness Functions - Pareto Ranking - Pareto Niching and Fitness Sharing - Recombination Operators - Mating Restriction - Solution Stability and Robustness - MOEA Complexity - MOEA Scalability - Running Time Analysis - MOEA Computational Cost - No Free Lunch Theorem.

UNIT IV MOEA TESTING, ANALYSIS, AND PARALLELIZATION 9

UNIT V APPLICATIONS AND ALTERNATIVE METAHEURISTICS 9

TOTAL: 45 PERIODS
OUTCOMES:
Upon Completion of the course, students will be able to
- Explain MOP principles
- Explain classical methods to solve MOP problems
- Be familiar with and explain structures of different MOP algorithms
- Solve constrained MOP problems
- Explain various design issues of MOP algorithms
- Perform an evaluation and analysis of MOP algorithm results
- Explain parallelization of MOP algorithms
- Develop parallel and hybrid MOP algorithms
- Identify various real-time MOP applications
- Explain other search algorithms

REFERENCES:

CP7028 ENTERPRISE APPLICATION INTEGRATION L T P C
3 0 0 3

OBJECTIVES:
- Describe approaches to enterprise application integration
- Understand the integration middleware
- Evaluate the integration approaches suitable for a given problem

UNIT I INTRODUCTION
Requirements for EAI - Challenges in EAI – Integration with legacy systems – Integration with partners - Heterogeneous environment – Implementation approaches – Web services, messaging, ETL, direct data integration – Middleware requirements – Approaches to integration – services oriented and messaging.

UNIT II INTEGRATION PATTERNS
Introduction to integration patterns – Architecture for application integration – Integration patterns – Point to point, broker, message bus, publish/subscribe, Challenges in performance, security, reliability - Case studies

UNIT III SERVICE ORIENTED INTEGRATION
Business process integration - Composite applications-services – Web services – Service choreography and orchestration - Business process modeling - BPMN, Business process execution - BPEL – Middleware infrastructure - Case studies
UNIT IV  MESSAGING BASED INTEGRATION  9

UNIT V  ENTERPRISE SERVICE BUS  12
Enterprise Service Bus – routing, scalable connectivity, protocol and message transformations, data enrichment, distribution, correlation, monitoring – Deployment configurations – Global ESB, Directly connected, Federated, brokered ESBs – Application server based – Messaging system based – Hardware based ESBs – Support to SOA, message based and event based integrations - Case studies.

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to

- Describe different approaches to integration enterprise applications
- Analyze specifications and identify appropriate integration approaches
- Develop a suitable integration design for a given problem
- Identify appropriate integration middleware for a given problem
- Evaluate the integration approaches against specified requirements

REFERENCES:

CP7029  INFORMATION STORAGE MANAGEMENT  L T P C
3 0 0 3

UNIT I  INTRODUCTION TO STORAGE TECHNOLOGY  9
Review data creation and the amount of data being created and understand the value of data to a business, challenges in data storage and data management, Solutions available for data storage, Core elements of a data center infrastructure, role of each element in supporting business activities

UNIT II  STORAGE SYSTEMS ARCHITECTURE  9
Hardware and software components of the host environment, Key protocols and concepts used by each component , Physical and logical components of a connectivity environment , Major physical components of a disk drive and their function, logical constructs of a physical disk, access characteristics, and performance Implications, Concept of RAID and its components, Different RAID levels and their suitability for different application environments: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5, RAID 0+1, RAID 1+0, RAID 6, Compare and contrast integrated and modular storage systems , high-level architecture and working of an intelligent storage system
UNIT III  INTRODUCTION TO NETWORKED STORAGE  9
Evolution of networked storage, Architecture, components, and topologies of FC-SAN, NAS, and IP-SAN, Benefits of the different networked storage options, understand the need for long-term archiving solutions and describe how CAS fill the need, understand the appropriateness of the different networked storage options for different application environments.

UNIT IV  INFORMATION AVAILABILITY, MONITORING & MANAGING DATACENTER  9
List reasons for planned/unplanned outages and the impact of downtime, Impact of downtime - Differentiate between business continuity (BC) and disaster recovery (DR), RTO and RPO, Identify single points of failure in a storage infrastructure and list solutions to mitigate these failures, Architecture of backup/recovery and the different backup/recovery topologies, replication technologies and their role in ensuring information availability and business continuity, Remote replication technologies and their role in providing disaster recovery and business continuity capabilities. Identify key areas to monitor in a data center, Industry standards for data center monitoring and management, Key metrics to monitor for different components in a storage infrastructure, Key management tasks in a data center.

UNIT V  SECURING STORAGE AND STORAGE VIRTUALIZATION  9
Information security, Critical security attributes for information systems, Storage security domains, List and analyzes the common threats in each domain, Virtualization technologies, block-level and file-level virtualization technologies and processes

TOTAL: 45 PERIODS

REFERENCE BOOKS:
1. EMC Corporation, Information Storage and Management, Wiley, India.
4. Additional resource material on www.emc.com/resource-library/resource-library.esp

CP7030  ROBOTICS  L T P C
3 0 0 3

OBJECTIVES:
• To understand robot locomotion and mobile robot kinematics
• To understand perception in robotics
• To understand mobile robot localization
• To understand mobile robot mapping
• To understand simultaneous localization and mapping (SLAM)
• To understand robot planning and navigation

UNIT I  LOCOMOTION AND KINEMATICS  9

UNIT II  ROBOT PERCEPTION  9
UNIT III MOBILE ROBOT LOCALIZATION

UNIT IV MOBILE ROBOT MAPPING

UNIT V PLANNING AND NAVIGATION
Introduction to planning and navigation – planning and reacting – path planning – obstacle avoidance techniques – navigation architectures – basic exploration algorithms

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to
- Explain robot locomotion
- Apply kinematics models and constraints
- Implement vision algorithms for robotics
- Implement robot localization techniques
- Implement robot mapping techniques
- Implement SLAM algorithms
- Explain planning and navigation in robotics

REFERENCES:

CP7031 COMPILER OPTIMIZATION TECHNIQUES

OBJECTIVES:
- To understand the optimization techniques used in compiler design.
- To be aware of the various computer architectures that support parallelism.
- To become familiar with the theoretical background needed for code optimization.
- To understand the techniques used for identifying parallelism in a sequential program.
- To learn the various optimization algorithms.

UNIT I INTRODUCTION

UNIT II INSTRUCTION-LEVEL PARALLELISM
UNIT III  OPTIMIZING FOR PARALLELISM AND LOCALITY-THEORY  
Basic Concepts – Matrix-Multiply: An Example - Iteration Spaces - Affine Array Indexes – Data Reuse Array data dependence Analysis.

UNIT IV  OPTIMIZING FOR PARALLELISM AND LOCALITY – APPLICATION  

UNIT V  INTERPROCEDURAL ANALYSIS  

TOTAL: 45 PERIODS

REFERENCES: