SOFTWARE ENGINEERING PROGRAM EDUCATIONAL OBJECTIVES:

1. Apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex, scalable software systems.
2. Design and experiment with software prototypes
3. Select and use software metrics
4. Communicate effectively through oral and written reports, and software documentation
5. Elicit, analyze and specify software requirements through a productive working relationship with project stakeholders
6. Demonstrate professionalism including continued learning and professional activities.
7. Contribute to society by behaving ethically and responsibly.
8. Successfully assume a variety of roles in teams of diverse membership.
9. Apply a systematic, disciplined, quantifiable approach to the cost-effective development, operation and maintenance of software systems to the satisfaction of their beneficiaries.
10. Build solutions using different technologies, architectures and life-cycle approaches in the context of different organizational structures.
11. Insist the development, adoption and sustained use of standards of excellence for software engineering practices.

SOFTWARE ENGINEERING PROGRAM OUTCOMES:

- Upon completion of the course, students would have obtained:
- An ability to apply knowledge of mathematics, science, and engineering.
- An ability to design and conduct experiments, as well as to analyze and interpret data.
- An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, safety, and sustainability.
- Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.
- An ability to identify, formulate, and solve engineering problems.
- An understanding of professional and ethical responsibility.
- An ability to communicate effectively.
- Demonstrate a knowledge and understanding of management and business practices, such as risk and change management, and understand their limitations.
- A recognition of the need for, and an ability to engage in life-long learning.
- An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
- An understanding of real-time, safety-critical, embedded computer systems.
### AFFILIATED INSTITUTIONS
ANNA UNIVERSITY, CHENNAI
REGULATIONS - 2013
M.E. SOFTWARE 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></td>
<td></td>
<td><strong>THEORY</strong></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>SE7101</td>
<td>Software Risk Management and Maintenance</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>SE7102</td>
<td>Advances in Software Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</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>5.</td>
<td>CP7102</td>
<td>Advanced Data Structure and Algorithms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Elective I</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PRACTICAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>SE7111</td>
<td>Software Requirements and Design Laboratory</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>SE7112</td>
<td>Advanced Data Structures Laboratory</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td>18</td>
<td>1</td>
<td>8</td>
<td>23</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></td>
<td></td>
<td><strong>THEORY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>SE7201</td>
<td>Software Project Planning and Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>SE7202</td>
<td>Software Testing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>SE7203</td>
<td>Software Metrics and Quality Assurance</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>IF7203</td>
<td>Data Warehousing and Data Mining</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>SE7204</td>
<td>Big Data Analytics</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></td>
<td></td>
<td><strong>PRACTICAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>SE7211</td>
<td>Software Testing Laboratory</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>SE7212</td>
<td>Socially Relevant Mini Project</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td>18</td>
<td>0</td>
<td>8</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></td>
<td></td>
<td><strong>THEORY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>SE7301</td>
<td>Software Design Patterns</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td>Elective III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td>Elective IV</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td>Elective V</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PRACTICAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>SE7311</td>
<td>Project Work (Phase I)</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>18</td>
</tr>
</tbody>
</table>

### SEMESTER 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></td>
<td></td>
<td><strong>PRACTICAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>SE7411</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>
</tbody>
</table>

**TOTAL NO OF CREDITS: 75**
# 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>IF7013</td>
<td>Energy Aware Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>IF7202</td>
<td>Cloud Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</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>4</td>
<td>SE7001</td>
<td>Distributed System</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>CP7028</td>
<td>Enterprise Application Integration</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>6</td>
<td>SE7010</td>
<td>Software Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>MU7011</td>
<td>Video Compression</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>SE7002</td>
<td>Pattern Classification and Analysis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>CP7012</td>
<td>Computer Vision</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>MU7008</td>
<td>User Interface Design</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>11</td>
<td>IF7301</td>
<td>Soft Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>SE7003</td>
<td>Machine Learning</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>CP7024</td>
<td>Information Retrieval Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>SE7004</td>
<td>Software Agents</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>15</td>
<td>MP7001</td>
<td>XML and Web Services</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>SE7005</td>
<td>Web Engineering and Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>NE7011</td>
<td>Mobile Application Development</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>NE7012</td>
<td>Social Network Analysis</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>19</td>
<td>SE7006</td>
<td>Software Reliability</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>SE7007</td>
<td>Software Documentation</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>SE7008</td>
<td>Software Refactoring</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>22</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>23</td>
<td>CP7006</td>
<td>Parallel Programming Paradigms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>SE7009</td>
<td>Software Process Models</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
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

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

Joint distributions – Marginal and Conditional distributions – Functions of two dimensional random variables – Regression Curve – Correlation.

UNIT III ESTIMATION THEORY


UNIT IV TESTING OF HYPOTHESES

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

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

OUTCOMES:

- 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:

OBJECTIVES:
- To understand the various risk levels in software development
- to gain expertise in discovering risk and usage of risk assessment tools
- to understand the risk plan, implementation and tracking risks
- to realize the software maintenance process, measurement and benchmarking
- to expertise in the SQA maintenance tools

UNIT I RISK CULTURE AND MANAGEMENT PROCESS

UNIT II DISCOVERING RISK AND ASSESSMENT

UNIT III RESPONDING TO RISKS AND TRACKING

UNIT IV MAINTENANCE PROCESS

UNIT V ACTIVITIES FOR MAINTENANCE

TOTAL: 45 PERIODS

OUTCOMES:
- To students will be able to learn about various risk levels in software development
- Students are trained to discover risk and how to use risk assessment tools
- Students will be able to prepare risk plan, implement and track risks
- They learn about measurement, benchmarking and SQA maintenance tools
REFERENCES:

SE7102   ADVANCES IN SOFTWARE ENGINEERING   LTPC
3 0 0 3

OBJECTIVES:
- To have a clear understanding of Software Engineering concepts.
- To gain knowledge of the Analysis and System Design concepts.
- To learn how to manage change during development.
- To learn the SOA and AOP concepts.

UNIT I   INTRODUCTION

UNIT II   ANALYSIS

UNIT III   SYSTEM DESIGN

UNIT IV   IMPLEMENTATION AND MANAGING CHANGE

UNIT V   ASPECT ORIENTED SOFTWARE DEVELOPMENT
AO Design Principles - Separations of Concerns, Subject Oriented Decomposition, Traits, Aspect Oriented Decomposition, Theme Approach, Designing Base and Crosscutting Themes, Aspect-Oriented Programming using Aspect-J.

TOTAL: 45 PERIODS

OUTCOMES:
- A clear understanding of Software Engineering concepts.
- Knowledge gained of Analysis and System Design concepts.
- Ability to manage change during development.
- Basic idea of the SOA and AOP concepts.
REFERENCES:
6. Aspect-Oriented Software Development with Use Cases, (The Addison-Wesley Object Technology Series), Ivar Jacobson and Pan-Wei Ng, December 2004

SE7103 FORMAL MODELS OF SOFTWARE SYSTEMS

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

REFERENCES:
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

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:

SE7111  SOFTWARE REQUIREMENTS AND DESIGN LABORATORY  L T P C
0 0 4 2

1. The students should develop all the necessary requirements based on IEEE standards or any other standardized standards and should prepare requirement document and design document after completion.
2. Use any open source software for requirements elicitation, requirements analysis and requirements validation.
3. Use any open source software for performing software design based on the requirements obtained in 2 for each system.
1. ONLINE SHOPPING MALL
PROJECT DESCRIPTION:

The Online Shopping Mall (OSM) application enables vendors to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. Also on the agenda is designing an online shopping site to manage the items in the shop and also help customers purchase them online without having to visit the shop physically.

The online shopping mall will showcase a complete shopping experience in a small package.

This project envisages bridging the gap between the seller, the retailer and the customer. A very high flexibility is being maintained in the design process so that this project can take the following path:

- A multiple merchant venue with each merchant having his/her own window which the customer can visit to browse and subsequently buy the products.
- Maintaining the deliverable goods as well as services through single or multiple windows is also on the agenda.

Target Users:

Mall Administrator: The Mall Administrator is the super user and has complete control over all the activities that can be performed. The application notifies the administrator of all shop creation requests, and the administrator can then approve or reject them. The administrator also manages the list of available product categories. The administrator can also view and delete entries in the guestbook.

Shop Owner: Any user can submit a shop creation request through the application. When the request is approved by the Mall Administrator, the requester is notified, and from there on is given the role of Shop Owner. The Shop Owner is responsible for setting up the shop and maintaining it. The job involves managing the sub-categories of the items in the shop. Also, the shop owner can add or remove items from his shop. The Shop Owner can view different reports that give details of the sales and orders specific to his shop. The Shop Owner can also decide to close shop and remove it from the mall.

Mall Customer/Guests: A Mall Customer can browse through the shops and choose products to place in a virtual shopping cart. The shopping cart details can be viewed and items can be removed from the cart. To proceed with the purchase, the customer is prompted to login. Also, the customer can modify personal profile information (such as phone number and shipping address) stored by the application. The customer can also view the status of any previous orders.

EMPLOYEES:

- Purchase department under a Purchase manager to overlook purchasing activities if warehousing needs arise.
- Sales department under a Sales manager who will look after the sale of products and services.
- Accounts department under an Accounts manager to look after the accounting activities of the enterprise.
2. BANKING SYSTEM

PROJECT DESCRIPTION:
A bank has several automated teller machines (ATMs), which are geographically distributed and connected via a wide area network to a central server. Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer. By using the ATM machine, a customer can withdraw cash from either checking or savings account, query the balance of an account, or transfer funds from one account to another. A transaction is initiated when a customer inserts an ATM card into the card reader. Encoded on the magnetic strip on the back of the ATM card are the card number, the start date, and the expiration date.

Assuming the card is recognized, the system validates the ATM card to determine that the expiration date has not passed, that the user-entered PIN (personal identification number) matches the PIN maintained by the system, and that the card is not lost or stolen. The customer is allowed three attempts to enter the correct PIN; the card is confiscated if the third attempt fails. Cards that have been reported lost or stolen are also confiscated.

If the PIN is validated satisfactorily, the customer is prompted for a withdrawal, query, or transfer transaction. Before withdrawal transaction can be approved, the system determines that sufficient funds exist in the requested account, that the maximum daily limit will not be exceeded, and that there are sufficient funds available at the local cash dispenser.

If the transaction is approved, the requested amount of cash is dispensed, a receipt is printed containing information about the transaction, and the card is ejected. Before a transfer transaction can be approved, the system determines that the customer has at least two accounts and that there are sufficient funds in the account to be debited. For approved query and transfer requests, a receipt is printed and card ejected.

A customer may cancel a transaction at any time; the transaction is terminated and the card is ejected. Customer records, account records, and debit card records are all maintained at the server.

3. CAMPUS MANAGEMENT SYSTEM

PROJECT DESCRIPTION:
The Campus Management System; is fully computerized information organization, storage and retrieval system that could provide us any information about an Institute just at the click of a mouse. The most fascinating asset about a computerized College fee Manager is that it enables us to explore any institute related information at any time on demand and that too in an absolutely user friendly environment that could be accessed even by a layman very easily.

OBJECTIVES AND GOALS:
- To automate the functions at a Higher Education Institute, the main missions of this software are as under
- To provide user-friendly interface to the college administrator
- To minimize the typing errors during data entry
- To search record of a particular object (course, student, faculty etc.)
- To update the record of an object
- To generate various reports for management
- To print various reports
- To reduce the typing work by keeping maximum information available on the screen
- To reduce the expenditure involving stationery items such as paper, ledgers, fee receipt book etc.
- To provide consistent, updated and reliable data at any time on demand.
- To analyse, plan and forecast the inflation or recession graph of the in college in the near future based on the college’s record of revenue sources and expenditure.
- To provide the most important feature of maintaining the valuable back-up of the critical data.
- To be bestowed with the latest security facilities provided by the modern computerized DBMS.

**PROJECT BUILDING BLOCKS:**

- Enrolment Management, Portal management, Admissions/Recruiting, Faculty Information, Student Services, Student Portal, Hostel management, Parking and Security, Student Health, Student Placement, Campus Incidents, Faculty Portal, Forum Portal, Student Billing, Alumni Portal.

4. AIR TRAFFIC CONTROL SYSTEM

Air traffic control is a closed loop activity in which pilots state the intent by filing flight plans. Controllers then plan traffic flow based on the total number of flight plans and, when possible, given clearance to pilots to fly according to their plans. When planning conflicts arise, controllers resolve them by clearing pilots to fly alternatives to their plans to avoid the conflicts. If unpredicted atmospheric conditions (e.g., wind speed or direction) or pilot actions cause deviations from conflict-free planned routings, controllers issue clearances for tactical maneuvers that solve any resultant problem, albeit not necessarily in a way that furthers the pilot’s goal of reaching the planned destination at a certain time.

**PROBLEM FORMULATION:**

Design an air traffic control system (ATCS) that is fault tolerant and scalable, according to the specific requirements listed in the following sections. The primary objective of the ATCS is to provide separation services for aircraft that are flying in controlled air space, or where poor visibility prevents from maintaining visual separation. Aircraft are separated from one another and from terrain hazards.

**SPECIFIC SOFTWARE REQUIREMENTS:**

The requirements of ATCSs include real-time aspects. The ATCS is a “dynamic” real-time system. Its loading will vary significantly over time, and has no upper bound. Loading scenarios can vary significantly, hence the average loading of the ATCS is not a highly useful metric for schedulability and other analyses. Although an upper bound could possibly be imposed artificially, this may not be a cost-effective solution, since pre-allocation of computing resources for such a worst case would lead to very poor resource utilization. A dynamic resource management policy is thus preferred.

5. CAFETERIA ORDERING SYSTEM

The Cafeteria Ordering System is a new system that replaces the current manual and telephone processes for ordering and picking up lunches in the Process Impact cafeteria.

**Patron:** A Patron is a Process Impact employee at the corporate campus in TidalPark, Chennai, who wishes to order meals to be delivered from the company cafeteria.
There are about 600 potential Patrons, of which an estimated 400 are expected to use the Cafeteria Ordering System. Patrons will sometimes order multiple meals for group events or guests. An estimated 90 percent of orders will be placed using the corporate Intranet, with 10 percent of orders being placed from home. All Patrons have Intranet access from their offices. Some Patrons will wish to set up meal subscriptions, either to have the same meal to be delivered every day or to have the day’s meal special delivered automatically. A Patron must be able to override a subscription for a specific day.

**Cafeteria Staff:** The Process Impact cafeteria currently employs about 20 Cafeteria Staff, who will receive orders from the Cafeteria Ordering System, prepare meals, and package them for delivery, print delivery instructions, and request delivery. Most of the Cafeteria Staff will need to be trained in the use of the computer, the Web browser, and the Cafeteria Ordering System.

**Menu Manager:** The Menu Manager is a cafeteria employee, perhaps the cafeteria manager, who is responsible for establishing and maintaining daily menus of the food items available from the cafeteria and the times of day that each item is available. Some menu items may not be available for delivery. The Menu Manager will also define the cafeteria’s daily specials. The Menu Manager will need to edit the menus periodically to reflect planned food items that are not available or price changes.

**Meal Deliverer:** As the Cafeteria Staff prepare orders for delivery, they will print delivery instructions and issue delivery requests to the Meal Deliverer, who is either another cafeteria employee or a contractor. The Meal Deliverer will pick up food and delivery instructions for each meal and deliver it to the Patron. The Meal Deliverer’s primary interactions with the system will be to reprint the delivery instructions on occasion and to confirm that a meal was (or was not) delivered.

**TOTAL: 60 PERIODS**

---

**SE7112 ADVANCED DATA STRUCTURES LABORATORY**

**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
• Design and apply iterative and recursive algorithms.
• Design and implement algorithms using the hill climbing and dynamic programming and recursive backtracking techniques.
• Design and implement optimisation algorithms for specific applications.
• Design and implement randomized algorithms.
• Design appropriate shared objects and concurrent objects for applications.
• Implement and apply concurrent linked lists, stacks, and queues.

REFERENCES:

SE7201 SOFTWARE PROJECT PLANNING AND MANAGEMENT L T P C 3 0 0 3

OBJECTIVES:
• To understand the various software processes.
• To learn format process models.
• To gain knowledge of the overall project activities
• To analyses the various issues in each phase of project management and people management.

UNIT I BASIC CONCEPTS 9

UNIT II FORMAT PROCESS MODELS AND THEIR USE 9
Definition and format model for a process, The ISO 9001 and CMM models and their relevance to project Management-other emerging models like People CMM.
UNIT III  UMBRELLA ACTIVITIES IN PROJECTS  

UNIT IV  IN STREAM ACTIVITIES IN PROJECTS  
Project Initiation - Project Planning - feasibility study estimation - resource allocation - execution and tracking - root cause analysis - Project Wind-up - Concept of process/project database.

UNIT V  ENGINEERING AND PEOPLE ISSUES IN PROJECT MANAGEMENT  
Phases (Requirements, Design, Development, Testing, maintenance, deployment) - engineering activities and management issues in each phase - Difficulties in people management - Role of Project manager, Special considerations in project management for India and geographic distribution issues.

OUTCOMES:
- Get the basic knowledge about various processes.
- Emphasize the use of format process models.
- Knowledge gained in usage and application of umbrella activities for project management
- Execute the project development in a systematic manner using tools and techniques
- Issues are analysed in various phases of project management and people management

TOTAL: 45 PERIODS

REFERENCES:
4. DeMarco and Lister: "Peopleware".

SE7202  SOFTWARE TESTING  
L T P C  3 0 0 3

OBJECTIVES:
- To know the behavior of the testing techniques to detect the errors in the software
- To understand standard principles to check the occurrence of defects and its removal.
- To learn the functionality of automated testing tools
- To understand the models of software reliability.

UNIT I  TESTING ENVIRONMENT AND TEST PROCESSES  
UNIT II  TESTING TECHNIQUES AND LEVELS OF TESTING

UNIT III  INCORPORATING SPECIALIZED TESTING RESPONSIBILITIES

UNIT IV  TEST AUTOMATION

UNIT V  SOFTWARE TESTING AND QUALITY METRICS

TOTAL: 45 PERIODS

OUTCOMES:
• Test the software by applying testing techniques to deliver a product free from bugs
• Evaluate the web applications using bug tracking tools.
• Investigate the scenario and the able to select the proper testing technique
• Explore the test automation concepts and tools
• Deliver quality product to the clients by way of applying standards such as TQM, Six Sigma
• Evaluate the estimation of cost, schedule based on standard metrics

REFERENCES:

SE7203 SOFTWARE METRICS AND QUALITY ASSURANCE LT P C
3 0 0 3

OBJECTIVES:
- To understand software metrics and measurement.
- To emphasize the use of product and quality metrics.
- To explain quality assurance and various tools used in quality management.
- To learn in detail about various quality assurance models.
- To understand the audit and assessment procedures to achieve quality.

UNIT I INTRODUCTION TO SOFTWARE METRICS
Fundamentals of measurement-Scope of software metrics-Measurement theory-Software measurement validation software metrics data collection – Analysis methods.

UNIT II PRODUCT AND QUALITY METRICS

UNIT III FUNDAMENTALS OF SOFTWARE QUALITY ASSURANCE

UNIT IV QUALITY ASSURANCE MODELS

UNIT V SOFTWARE QUALITY ASSURANCE TRENDS
Software Process- PSP and TSP - OO Methodology, Clean-room software engineering, Defect injection and prevention -Internal Auditing and Assessments-Inspections & Walkthroughs.

OUTCOMES:
- Knowledge on how to choose which metrics to collect and use them to make predictions.
- Ken on product and quality metrics.
- Understand how to detect, classify, prevent and remove defects.
- Choose appropriate quality assurance models and develop quality.
- Ability to conduct formal inspections, record and evaluate results of inspections.

TOTAL: 45 PERIODS
REFERENCES:

IF7203 DATA WAREHOUSING AND DATA MINING L T P C 3 0 0 3

OBJECTIVES:
 To expose the students to the concepts of Data warehousing Architecture and Implementation
 To Understand Data mining principles and techniques and Introduce DM as a cutting edge business intelligence
 To learn to use association rule mining for handling large data
 To understand the concept of classification for the retrieval purposes
 To know the clustering techniques in details for better organization and retrieval of data
 To identify Business applications and Trends of Data mining

UNIT I DATA WAREHOUSE 8

UNIT II DATA MINING & DATA PREPROCESSING 9
Introduction to KDD process – Knowledge Discovery from Databases - Need for Data Preprocessing – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization and Concept Hierarchy Generation.

UNIT III ASSOCIATION RULE MINING 8
Introduction - Data Mining Functionalities - Association Rule Mining - Mining Frequent Itemsets with and without Candidate Generation - Mining Various Kinds of Association Rules - Constraint-Based Association Mining.

UNIT IV CLASSIFICATION & PREDICTION 10
UNIT V  CLUSTERING


OUTCOMES:
Upon Completion of the course, the students will be able to
- Store voluminous data for online processing
- Preprocess the data for mining applications
- Apply the association rules for mining the data
- Design and deploy appropriate classification techniques
- Cluster the high dimensional data for better organization of the data
- Discover the knowledge imbibed in the high dimensional system
- Evolve Multidimensional Intelligent model from typical system
- Evaluate various mining techniques on complex data objects

REFERENCES:
4. BERSON, ALEX & SMITH, STEPHEN J, Data Warehousing, Data Mining, and OLAP, TMH Pub. Co. Ltd, New Delhi, 2012
6. PRABHU Data Warehousing, PHI Learning Private Limited, New Delhi, 2012,
7. PONNIAH, PAULRAJ, Data Warehousing Fundamentals, John Wiley & Sons, New Delhi, 2011

SE7204  BIG DATA ANALYTICS  L T P C
3 0 0 3

OBJECTIVES:
- To explore the fundamental concepts of big data analytics
- To learn to analyze the big data using intelligent techniques.
- To understand the various search methods and visualization techniques.
- To learn to use various techniques for mining data stream.
- To understand the applications using Map Reduce Concepts.

UNIT I  INTRODUCTION TO BIG DATA
UNIT II  DATA ANALYSIS

UNIT III  SEARCH METHODS AND VISUALIZATION

UNIT IV  MINING DATA STREAMS

UNIT V  FRAMEWORKS
Map Reduce – Hadoop, Hive, MapR – Sharding – NoSQL Databases - S3 - Hadoop Distributed File Systems– Case Study.

TOTAL: 45 PERIODS

OUTCOMES:
At the end of this course the students will be able to:
- Work with big data platform and its analysis techniques.
- Analyze the big data for useful business applications.
- Select visualization techniques and tools to analyze big data
- Implement search methods and visualization techniques
- Design efficient algorithms for mining the data from large volumes.
- Explore the technologies associated with big data analytics such as NoSQL, Hadoop and Map Reduce.

REFERENCES:
CASE STUDY 1

Cause Effect Graph Testing for a Triangle Program

Perform cause effect graph testing to find a set of test cases for the following program specification: Write a program that takes three positive integers as input and determine if they represent three sides of a triangle, and if they do, indicate what type of triangle it is. To be more specific, it should read three integers and set a flag as follows:

- If they represent a scalene triangle, set it to 1.
- If they represent an isosceles triangle, set it to 2.
- If they represent an equilateral triangle, set it to 3.
- If they do not represent a triangle, set it to 4.

CASE STUDY 2

Boundary Value Analysis for a Software Unit

The following is a specification for a software unit. The unit computes the average of 25 floating point numbers that lie on or between bounding values which are positive values from 1.0 (lowest allowed boundary value) to 5000.0 (highest allowed boundary value). The bounding values and the numbers to average are inputs to the unit. The upper bound must be greater than the lower bound. If an invalid set of values is input for the boundaries an error message appears and the user is reported. If the boundary values are valid the unit computes the sum and the average of the numbers on and within the bounds. The average and sum are output by the unit, as well as the total number of inputs that lie within the boundaries. Derive a set of equivalence classes for the averaging unit using the specification, and complement the classes using boundary value analysis. Be sure to identify valid and invalid classes.

Design a set of test cases for the unit using your equivalence classes and boundary values. For each test case, specify the equivalence classes covered, input values, expected outputs, and test case identifier. Show in tabular form that you have covered all the classes and boundaries. Implement this module in the programming language of your choice. Run the module with your test cases and record the actual outputs. Save an uncorrected version of the program for future use.

CASE STUDY 3

Cyclomatic Complexity for Binary Search

Draw a control flow graph for the given binary search code and clearly label each node so that it is linked to its corresponding statement. Calculate its cyclomatic complexity.
intbinsearch (int x, int v[], int n)
{
    int low, high, mid;
    low = 0;
    high = n - 1;
    while (low <= high) {
        mid = (low + high) / 2
        if (x < v[mid])
            high = mid - 1;
        else if (x > v[mid])
            low = mid + 1;
        else /* found match*/
            return mid;
    }
    return -1; /* no match*/
}

CASE STUDY 4
Data Flow Testing for Gregorian Calendar

A program was written to determine if a given year in the Gregorian calendar is a leap year. The well-known part of the rule, stipulating that it is a leap year if it is divisible by 4, is implemented correctly in the program. The programmer, however, is unaware of the exceptions: A centenary year, although divisible by 4, is not a leap year unless it is also divisible by 400. Thus, while year 2000 was a leap year, the years 1800 and 1900 were not. Determine if the following test-case selection criteria are reliable or valid.

(a) C1(T ) ≡ (T = \{1, 101, 1001, 10001\})
(b) C2(T ) ≡ (T = \{t\mid 1995 \geq t \geq 2005\})
(c) C3(T ) ≡ (T = \{t\mid 1895 \geq t \geq 1905\})
(d) C4(T ) ≡ (T = \{t\} \land t \not\in \{400, 800, 1200, 1600, 2000, 2400\})
(e) C5(T ) ≡ (T = \{t, t + 1, t + 2, t + 3, t + 4\} \land t \not\in \{100, 200, 300, 400, 500\})
(f) C6(T ) ≡ (T = \{t, t + 1, t + 2, \ldots, t + 399\} \land t \not\in \mathbb{D})
(g) C7(T ) ≡ (T = \{t_1, t_2, t_3\} \land t_1, t_2, t_3 \not\in \mathbb{D})

CASE STUDY 5
State based Testing for an Assembler

Suppose you were developing a simple assembler whose syntax can be described as follows :
<statement_ :: = <label field><op code><address>
<label field> :: = “none” | <identifier> :
<op code> :: = MOVE | JUMP
<address> :: = <identifier> | <unsigned integer>
A stream of tokens is input to the assembler. The possible states for such an assembler are: S1, prelabel; S2, label; S3, valid op code; S4, valid address; S5, valid numeric address. Start, Error, and Done. A table that describes the inputs and actions for the assembler is as follows:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>no more tokens</td>
<td>A1: Put the label in the symbol table.</td>
</tr>
<tr>
<td>Identifier</td>
<td>A2: Look up the op code and store its binary value in op code field.</td>
</tr>
<tr>
<td>MOVE, JUMP</td>
<td>A3: Look up symbol in symbol table and store its value in address field.</td>
</tr>
<tr>
<td>colon</td>
<td>A4: Convert number to binary, and store that value in address field.</td>
</tr>
<tr>
<td>Integer</td>
<td>A5: Place instruction in the object module, and print a line in the listing.</td>
</tr>
<tr>
<td></td>
<td>A6: Print error message and put all zeroes in the instruction.</td>
</tr>
</tbody>
</table>

Using this information and any assumptions you need to make, develop a state transition diagram for the assembler. From the state transition diagram develop a set of test cases that will cover all of the state transitions. Be sure to describe the exact sequence of inputs as well as the expected Sequence of state changes and actions.

CASE STUDY 6

Stress Testing of a Map-Aided Vehicle Tracking and Scheduling System

The American package courier and freight business faced the double pressures of consolidation and unstoppable increases in fuel costs. In mid-2008, pump prices were already double those prevailing in early 2007. As well, the recent decision of long-time price leader DHL to “co-locate” dozens of routes with erstwhile competitor UPS revealed just how fragile are market positions built through decades of promotions.

In Omaha, regional freight leader Red Ball Trucking1 was keener than most to maximize operating efficiencies out of its substantial fleet of trucks and vans and thereby maintain margins in the face of low-cost rivals. In March 2008, a brand-new map-based adjunct to the company’s proprietary logistics and routing system neared rollout. Extensive “white box”, line-by-line testing had eliminated most of the gross errors but the Red Ball CEO was concerned about the scalability of the program test bed.

Find out whether the map-enhanced vehicle tracking and scheduling system would remain stable at benchmarks of 50, 100 and 1000 concurrent users. Clean up any remaining bugs not caught by in-house.

CASE STUDY 7

Model Based Testing

Design and develop a scientific calculator program using various GUI components and events. Build the test model for the same. Determine the inputs that can be given to the model. Calculate expected output for the model. Run the test cases. Compare the actual output with the expected output. Any model based technique can be used for building the test model.

CASE STUDY 8

Web Application Testing for Student Grade System

With educational organizations under increasing pressure to improve their performance to secure funding for future provision of programmes, it is vital that they have accurate, up-to-date information. For this reason, they have MIS systems to record and track student enrolment and results on completion of a learning programme. In this way they can monitor achievement statistics. All student assignment work is marked and recorded by individual module tutors using a spreadsheet, or similar, of their own design. In the computing department these results are
input into a master spreadsheet to track a student’s overall progress throughout their programme of study. This is then made available to students through the web portal used in college. Perform web application testing for this scenario.

TOTAL: 60 PERIODS

SE7212  SOCIALLY RELEVANT MINI PROJECT  L T P C

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

- Choose any project of solving social problems
- Team Project with a maximum of three in a team
- Need to concentrate on software development methodologies
- Documentation is based on the standards
- Evaluation pattern is like Lab examination,
- Need to submit a report, presentation with demo.

TOTAL:60 PERIODS

SE7301  SOFTWARE DESIGN PATTERNS  L T P C

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

OBJECTIVES:
- How to add functionality to designs while minimizing complexity.
- What code qualities they need to maintain to keep code flexible.
- Understanding the common design patterns.
- Identifying the appropriate patterns for design problems.
- Refactoring the badly designed program properly using patterns.

UNIT I  INTRODUCTION
Introduction – Design Patterns in Smalltalk MVC – Describing Design patterns –Catalog of Design Patterns- Organizing the Catalog –How Design Patterns Solve Design Problems – How to select a Design Pattern – How to use a Design Pattern – What makes a pattern? – Pattern Categories – Relationship between Patterns – Patterns and Software Architecture

UNIT II  DESIGN PATTERNS FROM POSA1
Whole Part – Master Slave –Command Processor – View Handler – Forward Receiver – Client Dispatcher Server

UNIT III  CREATIONAL AND STRUCTURAL DESIGN PATTERNS
Abstract Factory - Factory Method – Prototype - Singleton – Builder Adapter Pattern – Decorator – Façade – Proxy - Bridge

UNIT IV  BEHAVIORAL DESIGN PATTERNS AND IDIOMS
Chain of Responsibility – Mediator – Observer – Strategy– Memento Idioms – Pattern Systems
UNIT V  CASE STUDY
Case Study Designing a Document Editor - What to expect from Design Patterns – A brief History of Design Patterns – The Pattern Community – Where will Patterns Go? – The Past, Present and the Future of Patterns - Anti Patterns

OUTCOMES:

- Be able to Design and implement codes with higher performance and lower complexity
- Be aware of code qualities needed to keep code flexible
- Understand core design principles and be able to assess the quality of a design with respect to these principles.
- Be capable of applying these principles in the design of object oriented systems.
- Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary.
- Be able to select and apply suitable patterns in specific contexts.
- Understand and apply refactoring techniques in the context of design patterns.

REFERENCES:
2. Frank Bachmann, Regine Meunier, Hans Rohnert “Pattern Oriented Software Architecture” – Volume 1, 1996.

IF7013  ENERGY AWARE COMPUTING  LT P C
3 0 0 3

OBJECTIVES:
The 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
- 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 9

UNIT V  ENERGY AWARE APPLICATIONS 9

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:

IF7202 CLOUD COMPUTING LT P C
3 0 0 3

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 9

UNIT II VIRTUALIZATION 9
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:
• 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
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

UNIT II  3G AND 4G CELLULAR NETWORKS

UNIT III  SENSOR AND MESH NETWORKS

UNIT IV  CONTEXT AWARE COMPUTING & WEARABLE COMPUTING

UNIT V  APPLICATION DEVELOPMENT
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
- Analyse 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:

SE7001 DISTRIBUTED SYSTEM LT P C
3 0 0 3

OBJECTIVE:
- To explore distributed systems principles associated with communication, naming, synchronization, distributed file systems, system design, distributed scheduling, and several case studies
- To cover both foundational concepts and well as practical deployments.

UNIT I COMMUNICATION IN DISTRIBUTED ENVIRONMENT 8

UNIT II DISTRIBUTED OPERATING SYSTEMS 12

UNIT III DISTRIBUTED RESOURCE MANAGEMENT 10
Distributed Shared Memory – Data-Centric Consistency Models – Client-Centric Consistency Models – Ivy – Munin – Distributed Scheduling – Distributed File Systems – Sun NFS.

UNIT IV FAULT TOLERANCE AND CONSENSUS 7

UNIT V CASE STUDIES 8
Distributed Object-Based System – CORBA – COM+ – Distributed Coordination-Based System – JINI.

TOTAL: 45 PERIODS
OUTCOMES:
The students will understand:
- the concepts underlying distributed systems
- how distributed systems may be constructed using a variety of tools and approaches

Students will be able to design, and implement distributed software systems in Java using:
- sockets
- remote procedure call mechanisms
- JAVA RMI

Students will demonstrate an ability to apply theory and techniques to unseen problems.

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 6
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 6
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 12
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

UNIT V ENTERPRISE SERVICE BUS
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:
UNIT I  INTRODUCTION

UNIT II  SOFTWARE ARCHITECTURAL PATTERNS
Architectural Patterns – Introduction to Styles – Simple Styles - Distributed and Networked Architectures- Architecture for network based applications – Decentralized Architectures

UNIT III  DESIGNING FOR NON FUNCTIONAL PROPERTIES

UNIT IV  ARCHITECTURE DESCRIPTION DOCUMENTATION AND EVALUATION
Early Architecture Description Languages – Domain and Style Specific ADLs – Extensible ADLs - Documenting Software architecture - Architecture Evaluation - ATAM

UNIT V  ARCHITECTURE ADAPTATION AND CASE STUDY

TEXT BOOKS:

REFERENCES:

MU7011  VIDEO COMPRESSION

OBJECTIVES:
- To introduce principles and current technologies of multimedia systems.
- To study the issues in effectively representing, processing and transmitting multimedia data including text, graphics, sound and music, image and video.
- To study the Image, video and audio standards such as JPEG, MPEG, H.26x, Dolby Digital and AAC will be reviewed.
- To study the applications such as video conferencing, multimedia data indexing and retrieval will also be introduced.
UNIT I  INTRODUCTION
Overview of image compression - important information theory concepts - entropy definition and interpretation - Shannon-Fano coding - Huffman coding - Adaptive Huffman coding - Lempel-Ziv codec- QM codec, context-based QM coder - examples of lossless compression

UNIT II  QUANTIZATION
Scalar quantization, optimal scalar quantizer, commander- Vector quantization- Audio and speech compression- JPEG & JPEG-2000 still image compression- Video coding standards (A) MPEG-1, MPEG-2

UNIT III  VIDEO PROCESSING
Video coding standards H.264/AVC and HEVC- Video coding techniques - motion estimation, rate control algorithms, pre & post processing- Video delivery/streaming over wired and wireless networks

UNIT IV  ADVANCED VIDEO CODING TECHNIQUES
Mobile multimedia computing- Multimedia content management and protection- Future directions – Multi-view video coding, depth coding and others

UNIT V  CONTENT MANAGEMENT
Video Compression-Motion Compensation, H.261 standard – FMM-14 Multimedia Applications Content-based retrieval in digital libraries – FMM

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able
- To know principles and current technologies of multimedia systems
- To know issues in effectively representing, processing, and retrieving multimedia data
- To know the areas by implementing some components of a multimedia streaming system
- To know the latest web technologies and some advanced topics in current multimedia research

REFERENCES:
UNIT I  IMAGE FORMATION AND IMAGE PROCESSING  9
Introduction- Geometric primitives and transformations- Photometric image formation- Sampling and aliasing, Compression- Point operators- Linear filtering- More neighbourhood operators- Fourier Transforms- Pyramids and wavelets- Geometric transformations- global optimization.

UNIT II  PATTERN RECOGNITION  9
Linear Discriminant Analysis- Bayes’ classifier – Neural net- Feed forward, unsupervised learning, Hopfield nets- fuzzy system-optimization techniques in Recognition-Genetic algorithm-Simulated annealing, object detection, Face recognition-Category recognition- Context and scene understanding

UNIT III  FEATURE DETECTION AND SEGMENTATION  9
Points and patches-Edges-Lines-Active Contours-Split and merge-Mean shift and mode finding- Normalized cuts-Graph cuts and energy based methods- 2D and 3D feature based alignment- Pose estimation- Geometric intrinsic calibration.

UNIT IV  MOTION ESTIMATION  9

UNIT V  OBJECT DETECTION AND TRACKING  9
Object Detection- Neural Network-Based Face Detection- Instance and Category recognition- Freund & Schapire’s AdaBoost algorithm – Context and scene understanding- Primitive tracking- Gradient Descent Tracking- Object Tracking with RBF Networks- Mean Shift tracking.

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
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

UNIT III  HOUGH TRANSFORM

UNIT IV  3D VISION AND MOTION

UNIT V  APPLICATIONS
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

TOTAL : 45 PERIODS

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

REFERENCES:

MU7008 USER INTERFACE DESIGN L T P C

OBJECTIVES:

- To understand the basics of User Interface Design.
- To design the user interface, design, menu creation and windows creation
- To understand the concept of menus, windows, interfaces, business functions, various problems in windows design with colour, text, Non-anthropomorphic Design.
- To study the design process and evaluations.

UNIT I INTERACTIVE SOFTWARE AND INTERACTION DEVICE

UNIT II HUMAN COMPUTER INTERACTION

UNIT III WINDOWS

UNIT IV MULTIMEDIA

UNIT V DESIGN PROCESS AND EVALUATION
User Interface Design Process - Usability Testing - Usability Requirements and Specification procedures and techniques- User Interface Design Evaluation

TOTAL: 45 PERIODS

OUTCOMES:

- Knowledge on development methodologies, evaluation techniques and user interface building tools
- Explore a representative range of design guidelines
- Gain experience in applying design guidelines to user interface design tasks.
- Ability to design their own Human Computer
REFERENCES:

2. Deborah Mayhew, The Usability Engineering Lifecycle, Morgan Kaufmann, 1999 

IF7301 SOFT COMPUTING L T P C 3 0 0 3

OBJECTIVES:

- To learn the key aspects of Soft computing
- To know about the components and building block hypothesis of Genetic algorithm.
- To understand the features of neural network and its applications
- To study the fuzzy logic components
- To gain insight onto Neuro Fuzzy modeling and control.
- To gain knowledge in machine learning through Support vector machines.

UNIT I INTRODUCTION TO SOFT COMPUTING 9
Evolution of Computing - Soft Computing Constituents – From Conventional AI to Computational Intelligence - Machine Learning Basics

UNIT II GENETIC ALGORITHMS 9
Introduction, Building block hypothesis, working principle, Basic operators and Terminologies like individual, gene, encoding, fitness function and reproduction, Genetic modeling: Significance of Genetic operators, Inheritance operator, cross over, inversion & deletion, mutation operator, Bitwise operator, GA optimization problems, JSPP (Job Shop Scheduling Problem), TSP (Travelling Salesman Problem), Differences & similarities between GA & other traditional methods, Applications of GA.

UNIT III NEURAL NETWORKS 9

UNIT IV FUZZY LOGIC 9

UNIT V NEURO-FUZZY MODELING 9

TOTAL: 45 PERIODS
OUTCOMES:
- Implement machine learning through neural networks.
- Write Genetic Algorithm to solve the optimization problem
- Develop a Fuzzy expert system.
- Model Neuro Fuzzy system for clustering and classification.

REFERENCES:
11. ROSS TIMOTHY J, Fuzzy Logic with Engineering Applications, Wiley India Pvt Ltd, New Delhi, 2010
UNIT V ADVANCED LEARNING
Learning Sets of Rules – Sequential Covering Algorithm – Learning Rule Set – First Order
Analytical Learning – Perfect Domain Theories – Explanation Base Learning – FOCL Algorithm

TOTAL: 45 PERIODS

REFERENCES:
Learning)”, The MIT Press 2004
Verlag, 2001

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
Motivation – Basic Concepts – Practical Issues - Retrieval Process – Architecture - Boolean
Characteristics – The impact of the web on IR — IR Versus Web Search – Components of a
Search engine

UNIT II MODELING
Taxonomy and Characterization of IR Models – Boolean Model – Vector Model - Term
Weighting – Scoring and Ranking – Language Models – Set Theoretic Models - Probabilistic
Models – Algebraic Models – Structured Text Retrieval Models – Models for Browsing

UNIT III INDEXING
Static and Dynamic Inverted Indices – Index Construction and Index Compression. Searching -
Sequential Searching and Pattern Matching. Query Operations - Query Languages – Query
Processing - Relevance Feedback and Query Expansion - Automatic Local and Global Analysis
– Measuring Effectiveness and Efficiency

UNIT IV CLASSIFICATION AND CLUSTERING
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


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:

SE7004 SOFTWARE AGENTS L T P C

OBJECTIVES:
- To learn the principles and fundamentals of designing agents
- To study the architecture design of different agents.
- To learn to do detailed design of the agents
- To understand user interaction with agents
- To explore the role of agents in assisting the users in day to day activities

UNIT I INTRODUCTION

UNIT II ARCHITECTURAL DESIGN
UNIT III  DETAILED DESIGN

UNIT IV  AGENTS AND USER EXPERIENCE

UNIT V  AGENTS FOR INTELLIGENT ASSISTANCE

TOTAL: 45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able to:
• Identify and explore the advantages of agents
• Design the architecture for an agent
• Design the agent in details in a view for the implementation
• Design communicative actions with agents.
• Design typical agents using a tool for different types of applications.

REFERENCES:

MP7001  XML AND WEBSERVICES
OBJECTIVES:
To provide an in-depth knowledge of XML and Web Services.
• To understand the fundamental concepts of Web services.
• To Understand the fundamental concepts of XML Technology.
• To design Web service Architecture.
• To Study Building Blocks of Web services.
• To understand the XML security issues.

UNIT I  WEB FUNDAMENTALS
UNIT II  XML TECHNOLOGY  9
XML-XML DTD-W3C XML Schema-Parsing XML - X path- XML Transformation-Other XML Technologies..

UNIT III  ARCHITECTING WEB SERVICES  9

UNIT IV  WEB SERVICES BUILDING BLOCK  9

UNIT V  XML SECURITY  9

TOTAL:45 PERIODS

OUTCOMES:
Upon Completion of the course, the students will be able
• To Know the fundamental elements in Web Technology and XML services.
• To design the Architecture of Web Services.
• To construct building blocks of Web services.
• To analyze security in XML.

REFERENCES:

SE7005  WEB ENGINEERING AND MANAGEMENT  L T P C  3 0 0 3

OBJECTIVES:
• To understand web page site planning, management and maintenance.
• To know the concept of developing advanced HTML pages with the help of frames, scripting languages, and evolving technology like DHTML.
• To develop web sites which are secure and dynamic in nature and writing scripts which get executed on the servers.
UNIT I  WEB ENGINEERING AND COMMUNICATION  9

UNIT II  PLANNING AND MODELLING ACTIVITY  9
Scope – Refining framework activities – Developing a schedule – Modeling framework-Design goals of Web App and Quality – Types of Model in Web Apps

UNIT III  CLIENT SIDE TECHNOLOGIES  8
Client side scripting: XHTML – DHTML– JavaScript– JSON - jQuery and AJAX, Setting up the environment (LAMP server)

UNIT IV  SERVER SIDE TECHNOLOGIES  10
PHP, PERL, Reading Data in Web Pages - Embedding PHP within HTML - Establishing connectivity with MySQL database, Servlets, JSP, Struts Architecture, - Understanding struts, Struts Validation Framework, Understanding LAMP and Its Effect on Web Development

UNIT V  PRODUCTION, MAINTENANCE AND EVALUATION  9

OUTCOMES:
- Develop project management skills related to web development
- Understand the technical skills required for Web Developers to use W3C standards, HTML, XHTML, Style Sheets, Java
- Understand the concepts in Client and Server-Side Scripting languages such as JavaScript, PERL and PHP.

REFERENCES:
5. Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP, “JamesLee, Brent Ware (Unit -4)
7. Hibernate Quickly, “Patrick Peak and Nick Heudecker, Patrick Peak, Nick Heudecker” (Unit 4

NE7011  MOBILE APPLICATION DEVELOPMENT  L T P C
3 0 0 3

OBJECTIVES:
1. Understand system requirements for mobile applications
2. Generate suitable design using specific mobile development frameworks
3. Generate mobile application design
4. Implement the design using specific mobile development frameworks
5. Deploy the mobile applications in marketplace for distribution
UNIT I  INTRODUCTION
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
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
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

UNIT V  TECHNOLOGY II - IOS
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  Completion of the course,

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:
- 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

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.

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:


SE7006 SOFTWARE RELIABILITY LT P C 3 0 0 3

UNIT I INTRODUCTION TO SOFTWARE RELIABILITY 7

UNIT II SOFTWARE RELIABILITY MODELING 12
Concepts – General Model Characteristic – Historical Development of models – Model Classification scheme – Markovian models – General concepts – General Poisson Type Models – Binomial Type Models – Poisson Type models – Fault reduction factor for Poisson Type models.

UNIT III COMPARISON OF SOFTWARE RELIABILITY MODELS 10

UNIT IV ARCHITECTURE AND DESIGN TECHNIQUES 8

UNIT V MEASUREMENT AND TOOLS 8

TOTAL : 45 PERIODS

REFERENCES:

OBJECTIVES:

- To understand task orientation and process of software documentation
- to gain expertise in designing tutorials, various view types
- to understand document planning, conduct review and usability test
- to realize the software architecture and document interfaces
- to expertise in layout pages and screens graphics usage

UNIT I UNDERSTANDING TASK ORIENTATION


UNIT II FORMS AND MODULE OF SOFTWARE DOCUMENTATION


UNIT III PROCESS OF SOFTWARE DOCUMENTATION

Analyzing your Users – Planning and Writing your Documents – Getting Useful Reviews – Conducting Usability Tests – Editing and Fine Tuning.

UNIT IV SOFTWARE ARCHITECTURE DOCUMENTATION IN PRACTICE


UNIT V TOOLS OF SOFTWARE DOCUMENTATION


TOTAL: 45 PERIODS

OUTCOMES:

- Students gain knowledge of task orientation and process of software documentation
- Students know-how to design tutorials, types of view etc.
- Students are trained to document planning and conduct review and usability test
- Students get expertise in documenting interfaces, designing layout pages, screens and graphics usage

REFERENCES:

OBJECTIVES:

- To learn about Refactoring, need for refactoring and the problems with refactoring
- To gain expertise in building test, refactoring, composing methods, removing parameters
- To understand organizing data, replace, remove, preserve objects etc
- To apprehend generalization, extract and replace field, class, object, perform big refactoring
- To skill in refactoring reuse and tools for refactoring

UNIT I

UNIT II

UNIT III
Organizing Data - Self Encapsulate Field - Replace Data Value with Object - Change Value to Reference - Change Reference to Value - Replace Array with Object - Duplicate Observed Data - Change Unidirectional Association to Bidirectional - Change Bidirectional Association to Unidirectional - Replace Magic Number with Symbolic Constant - Encapsulate Field - Encapsulate Collection - Replace Record with Data Class - Replace Type Code with Class - Replace Type Code with Subclasses - Replace Type Code with State/Strategy - Replace Subclass with Fields - Simplifying Conditional Expressions - Decompose Conditional - Consolidate Conditional Expression - Consolidate Duplicate Conditional Fragments - Remove Control Flag - Replace Nested Conditional with Guard Clauses - Replace Conditional with Polymorphism - Introduce Null Object - Introduce Assertion - Making Method Calls Simpler - Rename Method - Add Parameter - Remove Parameter - Separate Query from Modifier - Parameterize Method - Replace Parameter with Explicit Methods - Preserve Whole Object - Replace Parameter with Method - Introduce Parameter Object - Remove Setting Method - Hide Method - Replace Constructor with Factory Method - Encapsulate Downcast - Replace Error Code with Exception - Replace Exception with Test
UNIT IV

UNIT V
Refactoring, Reuse, and Reality - Why Are Developers Reluctant to Refactor Their Programs? - Resources and References for Refactoring - Implications Regarding Software Reuse and Technology Transfer - Refactoring Tools - Refactoring with a Tool - Technical Criteria for a Refactoring Tool - Practical Criteria for a Refactoring Tool - Putting It All Together.

TOTAL : 45 PERIODS

OUTCOMES:
• Students be trained about Refactoring, need and problems with refactoring
• Students are capable of building test, composing methods and removing parameters
• Students identify and know how to organizing data, replace, remove, preserve objects
• Students able to capture about generalization, able to replace field, class, object and perform big refactoring
• Students gain proficiency in refactoring reuse and tools for refactoring

REFERENCES:
5. Joshua Kerievsky," Refactoring to Patterns" ,Addison-Wesley, 2005

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

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

TOTAL: 45 PERIODS

REFERENCES:
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

Basic MPI programming - MPI_Init and MPI_Finalize - MPI communicators - SPMD programs - message passing - MPI_Send and MPI_Recv - message matching - MPI I/O - parallel I/O - collective communication - MPI_Reduce - MPI_Allreduce - broadcast - scatter - gather - allgather - derived types - remote memory access - dynamic process management - MPI for grids - performance evaluation of MPI programs

UNIT III SHARED MEMORY PARADIGM: PTHREADS

Basics of Pthreads - thread synchronization - critical sections - busy-waiting - mutexes - semaphores - barriers and condition variables - read-write locks - Caches, cache coherence and false sharing - thread safety - Pthreads case study

UNIT IV SHARED MEMORY PARADIGM: OPENMP

Basic OpenMP constructs - scope of variables - reduction clause - parallel for directive - loops in OpenMP - scheduling loops - synchronization in OpenMP - Case Study: Producer-Consumer problem - cache issues - threads safety in OpenMP - OpenMP best practices

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.

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

TOTAL : 45 PERIODS
REFERENCES:

SE7009 SOFTWARE PROCESS MODELS

UNIT I PROCESS AND BASIC PROCESS MODELS 9

UNIT II ADVANCED PROCESS MODELS 9

UNIT III REUSE ORIENTED SOFTWARE ENGINEERING 9

UNIT IV PROCESS IMPROVEMENT MODELS 9

UNIT V EMERGING TRENDS AND NEW DIRECTIONS 9
Software process simulation- Web-based software process models and process engineering- Software process and business process reengineering- Understanding, capturing, and operationalizing process models.

TOTAL: 45 PERIODS
REFERENCES: