B. E. SYLLABUS

INFORMATION SCIENCE & ENGINEERING

V & VI SEMESTER

With
Scheme of Teaching
& Examination
# DEPARTMENT: INFORMATION SCIENCE & ENGG

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Qualification</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dr. Balasubramani R.</td>
<td>Ph.D.</td>
<td>Professor &amp; HOD</td>
</tr>
<tr>
<td>2</td>
<td>Dr. Uday Kumar Shenoy</td>
<td>Ph.D.</td>
<td>Professor</td>
</tr>
<tr>
<td>3</td>
<td>Mr. Karthik Pai B. H.</td>
<td>M.Tech</td>
<td>Associate Professor II</td>
</tr>
<tr>
<td>4</td>
<td>Mrs. Ashwini B.</td>
<td>M.Tech</td>
<td>Associate Professor II</td>
</tr>
<tr>
<td>5</td>
<td>Mr. Vasudeva Pai</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>6</td>
<td>Mr. Pranesh V. K.</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>7</td>
<td>Mrs. Deepa J. Shetty</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>8</td>
<td>Mr. Devidas</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>9</td>
<td>Mrs. Rashmi Naveen</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>10</td>
<td>Mr. Rakesh Joshi U.</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>11</td>
<td>Mrs. Manasa S.</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>12</td>
<td>Mr. Jason Elroy Martis</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>13</td>
<td>Ms. Chinmai Shetty</td>
<td>M.Tech</td>
<td>Assistant Professor II</td>
</tr>
<tr>
<td>14</td>
<td>Mr. Abhir Bhandary</td>
<td>M.Tech</td>
<td>Assistant Professor I</td>
</tr>
<tr>
<td>15</td>
<td>Mr. Srikanth Bhat K.</td>
<td>M.Tech</td>
<td>Assistant Professor I</td>
</tr>
</tbody>
</table>
VISION:

The vision of the Dept. of ISE is to uniquely position the Dept. as a leader in innovation and excellence in information science and engineering through education, research and scholarship in a professional framework by addressing evolving global needs. Also the Dept. aims at creating top quality successful and sustainable programs and curricula for the students to address the emerging educational challenges and market demands.

MISSION:

- To provide outstanding education and research training to the students for their productive careers in industry, academia and government.
- To provide a learning environment that promotes excellence and innovation, ethical practice and responsibility towards society.
- To prepare the students to practice their professions competently to meet the ever-changing needs of society and to continue learning their discipline, allowing them to move into other related fields.
- To promote active learning, critical thinking, and engineering judgment coupled with business and entrepreneurial skills.

Programme Educational Objectives (PEOs):

- Graduates must gain both theoretical and practical knowledge to identify, formulate & solve challenges in Information Science & Engineering problems.
• Graduates must work productively as Information Science Engineers, including supportive and leadership roles on multidisciplinary teams.

• Graduates must communicate effectively, recognize and incorporate societal needs and constraints in their professional endeavors, and practice their profession with high regard to legal and ethical responsibilities.

• Graduates must engage in life-long learning, such as graduate study, to remain current in their profession and be leaders in our technological society.

**Programme Outcomes (POs):**

BE (ISE) Engineering Programme students must attain the following outcomes at the end of the course.

a. An ability to apply knowledge of mathematics, science and engineering

b. An ability to design and conduct experiments, as well as to analyze and interpret data

c. An ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability

d. An ability to function on multidisciplinary teams

e. An ability to identify, formulate and solve engineering problems

f. An understanding of professional and ethical responsibility
g. An ability to communicate effectively
h. The broad education necessary to understand the impact of engineering solutions in global, economic environmental and societal context
i. A recognition of the need for and an ability to engage in lifelong learning
j. A knowledge of contemporary issues
k. An ability to use the techniques, skills and modern engineering tools necessary for engineering practice and
l. An ability to strengthen the knowledge and understanding in the areas of computer networking and software engineering.
## DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
### SCHEME OF TEACHING AND EXAMINATION

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Code</th>
<th>Subject</th>
<th>Theory/Tuto./Prac./Self Study</th>
<th>Total Hrs./Week</th>
<th>C.I.E.</th>
<th>S.E.E.</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13IS501</td>
<td>System Software</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>13IS502</td>
<td>Relational Database Management System</td>
<td>4+0+2+0</td>
<td>6</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>13IS503</td>
<td>Microprocessors and Interfacing</td>
<td>4+2+2+0</td>
<td>8</td>
<td>50</td>
<td>50</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>13IS504</td>
<td>Software Engineering</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>13IS505</td>
<td>Operating System</td>
<td>4+0+0+S</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>13IS51X</td>
<td>Elective -I</td>
<td>3+0+0+0</td>
<td>3</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>13IS506</td>
<td>System Software Lab</td>
<td>0+2+2+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>13IL001#</td>
<td>Employability Skill Development</td>
<td>1+0+0+0</td>
<td>1</td>
<td>50</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>34</strong></td>
<td><strong>350</strong></td>
<td><strong>350</strong></td>
<td><strong>28</strong></td>
</tr>
</tbody>
</table>

# Non credit Mandatory Learning Course (PP/NP)
## DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

### SCHEME OF TEACHING AND EXAMINATION

#### VI Semester

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Code</th>
<th>Subject</th>
<th>Theory/Tuto./Prac./ Self Study</th>
<th>Total Hrs./Week</th>
<th>C.I.E.</th>
<th>S.E.E.</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13IS601</td>
<td>Computer Graphics</td>
<td>4+0+2+0</td>
<td>6</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>13IS602</td>
<td>Computer Networks</td>
<td>4+0+2+0</td>
<td>6</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>13IS603</td>
<td>Java and Internet Technologies</td>
<td>4+0+2+0</td>
<td>6</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>13IS604</td>
<td>Object Oriented Modeling and Design</td>
<td>4+0+0+S</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>13IS61X</td>
<td>Elective – II</td>
<td>3+0+0+0</td>
<td>3</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>13IS62X</td>
<td>Elective –III</td>
<td>3+0+0+0</td>
<td>3</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>13IS605</td>
<td>IGW lab*</td>
<td>0+4**+0+0</td>
<td>4</td>
<td>50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>13IL002</td>
<td>Employability Skill Development #</td>
<td>1+0+0+0</td>
<td>1</td>
<td>50</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td><strong>33</strong></td>
<td><strong>33</strong></td>
<td><strong>350</strong></td>
<td><strong>300</strong></td>
</tr>
</tbody>
</table>
## Electives:

<table>
<thead>
<tr>
<th>Sem</th>
<th>Group</th>
<th>Code</th>
<th>Elective Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Elective 1</td>
<td>13IS511</td>
<td>C# &amp; .NET Technologies Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS512</td>
<td>Advanced Unix System Programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS513</td>
<td>Operations Research</td>
</tr>
<tr>
<td>6</td>
<td>Elective 2</td>
<td>13IS611</td>
<td>Web2.0 Programming and WebServices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS612</td>
<td>Software Architecture</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS613</td>
<td>Advanced DBMS Biomedical</td>
</tr>
<tr>
<td></td>
<td>Elective 3</td>
<td>13IS621</td>
<td>Information Storage Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS622</td>
<td>Free &amp; Open Source Software (FOSS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13IS623</td>
<td>Distributed Systems and Objects</td>
</tr>
</tbody>
</table>
SYSTEM SOFTWARE

Sub Code       : 13IS501       Credits : 04
Hrs/Week    : 4+0+0+0       Total Hours: 52

Course Outcomes:
1. Introduction to fundamental software concept Description of Simplified Instructional Computer Architecture.
2. Understand the fundamental operations performed by a typical assembler
3. Describe features of assembler for SIC/XE machine.
4. Describes the structure and logic of one-pass assemblers
5. Introduces the notion of a multi-pass assemblers- an extension to the two pass logic
6. Examples of actual assemblers for a variety of real computers
7. Presents the design of a loader found on a simple sic machine
8. Understanding the first program to be run by the computer – operating system
9. Discuss the issues of relocation and linking from the loaders point of view.
10. Examine linkers and loaders for actual computers
11. To understand the structure of the editor, user interface and the editing process
12. Discuss the functions and the capabilities of the debugging system
13. To understand the basic concepts of macro processors, like macro definition and expansion
14. Discuss the features of macro processor like generation of unique labels within macro expansions, conditional macro expansion and the use of keyword parameters in macros.
15. To describe the internal structure of macro processor
16. To study some of the characteristics of actual macro processors
UNIT-I

Machine Architecture:

Assemblers:
Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation. 11 Hours

UNIT-II

Assemblers:
Machine Independent Assembler Features Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples - MASM Assembler, SPARC Assembler. 10 Hours

UNIT-III

Loaders and Linkers:

UNIT-IV

Editors and Debugging Systems:
Text Editors - Overview of Editing Process, User Interface, Editor Structure, InteractiveDebugging Systems - Debugging Functions and Capabilities, Relationship with Other Parts of TheSystem, User-Interface Criteria. 10 Hours

UNIT-V

Macro Processor:

11 Hours

TEXT BOOKS:

REFERENCE BOOKS:

******************

RELATIONAL DATABASE MANAGEMENT SYSTEMS

Sub Code : 13IS502
Hrs/Week : 4+0+2+0
Credits : 04
Total Hours: 52

Course Outcomes:

1. To learn about the main characteristics and advantages of Database approach
2. To know about various types of data models used in architecture of DBMS and data independence
3. To learn the phases of database design using high-level conceptual models
4. To understand the basic concepts of ER Model and to construct ER diagram for a database using these concepts
5. To learn the characteristics and basic concepts of relations
6. To learn basic constraints of relational model and relational database schemas
7. To learn the syntax and use of basic update operations and corresponding constraint violations
8. To learn the syntax and use of Select and Project operations
9. To learn the syntax and use of Union, Intersection, Set Difference operations
10. To learn the syntax and use of Division and Join operations
11. To learn the syntax and use of Aggregate functions with Grouping, Recursive Closure, Outer Join and Outer Union operations
12. To study SQL data definition and basic constraints in SQL
13. To learn the basic and complex queries in SQL
14. To study the programming issues and techniques and to write programs using Embedded SQL
15. To understand the concept of functional dependency and Normalization
16. To understand the definition and use of various normal forms
17. Study about the various properties of decompositions
18. Learn the concepts of multivalued and join dependencies associated with fourth and fifth normal forms
19. To know the ACID properties and concurrent execution of transactions
20. To learn various locking protocols
21. Describe LOG and Differentiate log based recovery mechanisms
22. To learn about various recovery techniques including shadow paging, immediate update and ARIES recovery algorithm

UNIT-I

Introduction to database systems
Introduction, Characteristics of the Database approach, Actors on the scene, Advantages of using the DBMS approach, Data models, Schemes and Instances, Three Schema Architecture and Data Independence

Entity-Relationship Model
Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design for the COMPANY Database; ER Diagrams, Naming Conventions and Design Issues.

9 Hours
UNIT-II

Relational Model And Relational Algebra
Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations and Dealing with Constraint Violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping

11 Hours

UNIT-III

SQL-The Relational Database Standard
SQL Data Definition and Data Types, Specifying Basic Constraints in SQL, Schema Change Statements in SQL; Basic Queries in SQL; More Complex SQL Queries; Insert, Delete and Update Statements in SQL; Additional Features of SQL; Views (Virtual Tables) in SQL; Database Programming: Issues and Techniques; Embedded SQL

11 Hours

UNIT-IV

Database Design
Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; BoyceCodd Normal Form; Properties of Relational Decompositions; Algorithms for Relational Database Schema Design; Multivalued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form.

10 Hours

UNIT-V

Transaction Management
The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of Locking; Transaction Support in SQL; Introduction to Crash Recovery; 2PL, Serializability and Recoverability; Introduction to Lock Management; Lock Conversions; Dealing with Deadlocks; Specialized Locking Techniques; Concurrency Control without Locking; Introduction
to recovery: Recovery Concepts, Recovery Techniques Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, The Aries Recovery Algorithm

11 Hours

TEXT BOOKS:

REFERENCE BOOK:

*************

LABORATORY COMPONENTS

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0+0+2+0</td>
<td>01</td>
</tr>
</tbody>
</table>

Total Hours : 26(13 weeks)

Course Outcomes:

1. To understand basic concepts of Visual c# and develop simple applications using Visual c#
2. To learn about creating database and tables and insert values using SQL
3. Execution of SQL DDL Statements(drop, Alter) and DML statements (update, delete)
4. To implement queries based on basic structure of SQL queries
5. To implement the complex queries

Student must carry out experiments using some RDBMS package (like MS-SQL/Oracle/DB2..) at the back end and some visual programming
tool at the front end (like VB/VC++/D2K..). Typical experiments must include query processing applications. All SQL statements must be made use of by the student in developing the application. Some input on Query Optimization & some SQL tuning must be given.

***

MICROPROCESSORS AND INTERFACING

Sub Code : 13IS503  
Credits : 04+01
Hrs/Week : 4+2+2+0  
Total Hours: 52

Course Outcomes:
1. In general know about the basics Of a microcomputer structure and history.
2. To understand the internal architecture of Microprocessor 8086.
3. To know how to use the instructions in the program.
4. Understand the coding format of the instruction used in 8086.
5. To know about the relation between high level C program and assembly language.
6. Understand the various programming constructs used in 8086.
7. Brief about the architecture of 80x87 and instruction set used in 80x87.
8. Understand the use of interrupts in 8086 also distinguish between various other types of interrupts.
9. To know about the salient features of advanced processors of the same family.
10. Must be able to interface logic controller stepper motor, keypad etc to 8086.
11. To know about the interfacing of analog to digital converter with 8086.

UNIT-I

Microprocessor and 8086 Architecture
Overview of Microcomputer Structure and Operation, Microprocessor Evolution and Types, 8086 Internal Architecture, Introduction to
Programming the 8086

UNIT-II
Programming the Microprocessor
8086 Family Assembler Language Programming – Instruction Templates, MOV Instruction Coding Format and Examples, MOV Instruction Coding Examples, Writing Programs for use with an Assembler, Assembly Language Program Development Tools, Implementing Standard Program Structures in 8086 Assembly Language: Simple Sequence Programs, Jumps, Flags, and Conditional Jumps, If-Then, If-Then-Else, and Multiple If-Then-Else Programs, While-Do Programs, Repeat-Utill Programs, Instruction Timing and Delay Loops

UNIT-III
Strings, Procedures, and Macros: The 8086 String Instructions, Writing and Using Procedures, Writing and Using Assembler Macros.

Arithmetic Co-Processor
Data formats for the arithmetic co processor, 80 x 87 architecture, and Instruction set.

UNIT-IV
Interrupt Service Routine
8086 Interrupts and Interrupt Applications: 8086 Interrupts and Interrupt Responses, Hardware Interrupt Applications, 8259A Priority Interrupt Controller, Software Interrupt Applications.

Introduction to Advanced Microprocessors
Salient features of 80186, 80286, 80386, 80486, Pentium Family processors, Overview of Protected Mode and Address translation.

UNIT-V
Digital Interfacing
Digital Interfacing: Programmable Parallel Ports and Handshake Input/Output, Methods of Data Transfer, Implementing Handshake Data Transfer, 8255A Internal Block Diagram and System connections, 8255A Operational Modes and Initialization, Constructing and Sending 8255A
Control Words. Interfacing a Microprocessor to Keyboards, Interfacing to Alphanumeric Displays,
**Analog Interfacing** D/A Converter Operation, Interfacing, and Applications, A/D Converter Specifications, Types, and Interfacing.

**10 Hours**

**TEXT BOOK:**
1. Microprocessors- Douglas V. Hall, Revised 2\textsuperscript{nd} Edition.

**REFERENCE BOOKS:**
1. Advanced Microprocessors- Barry B. Brey, 4\textsuperscript{th} Edition.

******************

**LABORATORY COMPONENTS**

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Credit: 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0+2+2</td>
<td></td>
</tr>
<tr>
<td>Total Hours</td>
<td>26(13 weeks)</td>
</tr>
</tbody>
</table>

**Course Outcomes:**

1. a) Binary search algorithm.
   b) Bubble Sort algorithm.
   c) Fibonacci numbers.
   d) Read and display current system time in the standard format on the screen.

2. a) Write ALP macros to read and display a string of characters.
   b) Read an alphanumeric character and display its equivalent ASCII code at the center of the screen.

3. a) palindrome
   b) Strings equality
   c) Read and display a text at the specified location on the screen.
   d) Substring present in main string or not.

4. a) Decimal Up-counter to display 00-99
   b) Read a pair of input co-ordinates in BCD and move the cursor to
the specified location on the screen.

5. a) factorial
   b) nCr

6. Program to create a file (input file) and to delete an existing file.
7. Digital Interfacing
8. Analog Interfacing

Student must use MASM/TASM, appropriate linker and debug utility to carry out 8086 assembly language program experiments. Typical programs include

NOTE:
• Develop and execute the following programs using an 8086 Assembly Language. All the programs to be executed using an assembler like MASM, TASM etc.
• Program should have suitable comments.
• The board layout and the circuit diagram of the interface are to be provided to the student during the examination.

1. a) Search a key element in a list of ‘n’ 16-bit numbers using the Binary search algorithm.
   b) Read the status of eight input bits from the Logic Controller Interface and display ‘FF’ if it is even parity bits otherwise display 00. Also display number of 1’s in the input data.

2. a) Write ALP macros:
   i. To read a character from the keyboard in the module (1) (in a different file)
   ii. To display a character in module(2) (from different file)
   iii. Use the above two modules to read a string of characters from the keyboard terminated by the carriage return and print the string on the display in the next line.
   b) Perform the following functions using the Logic Controller Interface.
   i. BCD up-down Counter ii. Ring Counter
3. a) Sort a given set of ‘n’ numbers in ascending and descending orders using the Bubble Sort algorithm.
   b) Read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.

4. a) Read an alphanumeric character and display its equivalent ASCII code at the center of the screen.
   b) Display messages FIRE and HELP alternately with flickering effects on a 7-segment display interface for a suitable period of time.
   Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor it is necessary for the student to compute these values).

5. a) Reverse a given string and check whether it is a palindrome or not.
   b) Assume any suitable message of 12 characters length and display it in the rolling fashion on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages. (Examiner does not specify these delay values nor it is necessary for the student to compute these values).

6. a) Read two strings, store them in locations STR1 and STR2. Check whether they are equal or not and display appropriated messages. Also display the length of the stored strings.
   b) Convert a 16-bit binary value (assumed to be an unsigned Integer) to BCD and display it from left to right and right to left for specified number of times on a 7-segment display interface.

7. a) Read your name from the keyboard and display it at a specified location on the screen in front of the message What is your name? You must clear the entire screen before display.
   b) Drive a Stepper Motor interface to rotate the motor in clockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

8. a) Compute the factorial of a positive integer ‘n’ using recursive procedure.
b) Drive a stepper motor interface to rotate the motor in anticlockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps (Any arbitrary value for the delay may be assumed by the student).

9. a) Compute $nC_r$ using recursive procedure. Assume that ‘n’ and ‘r’ are non-negative integers.
b) Drive a stepper motor interface to rotate the motor by N steps left direction and N steps right direction (N is specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

10. a) Find out whether a given sub-string is present or not in a main string of characters.
b) Scan a 8 x 3 keypad for key closure and to store the code of the key pressed in a memory location or display on screen. Also display row and column numbers of the key pressed.

11. a) Generate the first ‘n’ Fibonacci numbers.
b) Scan a 8 x 3 keypad for key closure and simulate ADD and SUBTRACT operations as in a calculator.

12. a) Read the current time from the system and display it in the standard format on the screen.
b) Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).

13. a) Program to simulate a Decimal Up-counter to display 00-99.
b) Generate a Half Rectified Sine wave form using the DAC interface. (The output of the DAC is to be displayed on the CRO).

14. a) Read a pair of input co-ordinates in BCD and move the cursor to the specified location on the screen.
b) Generate a Fully Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).

15. a) Program to create a file (input file) and to delete an existing file.
b) Drive an elevator interface in the following way:
i. Initially the elevator should be in the ground floor, with all requests in OFF state.
ii. When a request is made from a floor, the elevator should move to that floor, wait there for a couple of seconds, and then come down to ground floor and stop. If some requests occur during going up or coming down they should be ignored.

Note: In the examination each student picks one question from a lot of all 15 questions.

SOFTWARE ENGINEERING

Sub Code : 13IS504
Hrs/Week : 4+0+0+0

Credits : 04
Total Hours: 52

Course Outcomes:
1. To learn about the notable changes in software engineering
2. Study about different types of software process models
3. To learn about requirements gathering and analysis
4. Know about various approaches to software design and their differences
5. To Understand about data flow diagrams and structured design
6. To know the concepts of UML diagrams
7. To study about UI design methodology
8. To learn about Program analysis tools and testing object oriented programs
9. Know about SEI capability maturity model and six sigma
10. To learn about COCOMO and risk management
11. To study about CASE support in software life cycle
12. To learn about reuse approach
UNIT-I

Overview
Introduction: FAQ's about software engineering, Professional and ethical responsibility. Socio-Technical systems: Emergent system properties; Systems engineering; Organizations, people and computer systems; Legacy systems.

Critical Systems, Software Processes

Requirements
Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document. Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management.

12 Hours

UNIT-II

System models, Project Management
System Models: Context models; Behavioral models; Data models; Object models; Structured methods. Project Management: Management activities; Project planning; Project scheduling; Risk management.

8 Hours

UNIT-III

Software Design
Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles. Object-Oriented design: Objects and Object Classes; An Object Oriented design process; Design evolution.

10 Hours

UNIT-IV

Development
Rapid Software Development: Agile methods; Extreme programming; Rapid application development. Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.

Verification and Validation
Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods. Software testing: System testing; Component testing; Test case design; Test automation.

12 Hours
UNIT-V

Management Managing People: Selecting staff; Motivating people; Managing people; The People Capability Maturity Model. Software Cost Estimation: Productivity; Estimation techniques; Algorithmic cost modeling, Project duration and staffing.  

TEXT BOOKS:

REFERENCE BOOKS:

*************

OPERATING SYSTEM

Sub Code : 13IS505
Hrs/Week : 4+0+0+S*
Credits : 04
Total Hours : 52

*Self Study to be exercised under the supervision of course instructor and to be restricted to not more than 10% of the total teaching hours.

Course Outcomes:
1. To understand what operating systems are, what they do, and how different operating systems operates
2. Know about the interface between a process and OS
3. To understand the program execution Environment and various services to programs and to users
4. Understand the program paradigm like process state and process control block
5. To know various Thread Models, advantages of Multi-Threaded programs and its performance
6. Required for Cooperating processes
7. To know about the various schedulers and multiprogramming environment
8. Study about the different types of algorithms to maximize computer resources utilization like CPU
9. To understand the need for Process coordination and usage of Semaphores, Monitors
10. To study the issues of Multithreaded programs competing for shared resources To know about various algorithms to detect, avoid deadlocks
11. To know about various ways to manage main memory
12. Understand the structure of the storage and allocation
13. Study the methods to solve the varying size process storage issues
14. Helps to understand the execution environment of running the process larger than the main memory allocations
15. Study about how to allocate the memory on need basis
16. Study the conflicts of multiprogramming with CPU utilization
17. Study about the file attributes and operations on file
18. Understand the need of sharing and the various protection issues
19. Study the allocation methods of the file so that disk space is utilized effectively
20. Learn how to utilize free spaces in memory
21. Study various algorithms to access the disk efficiently

UNIT-I

Introduction and System structures:
Operating system definition; Operating System operations; Different types of operating system – Mainframe systems, Multi programmed systems, Time sharing systems, Desktop systems, Parallel systems, Distributed systems, Real time systems, Clustered systems, Handheld systems.

Operating System Services; User - Operating System interface; System calls; Types of system calls; Operating System structure; Virtual machines
**Process Management:** Process concept; Process scheduling; Operations on processes; Inter-process communication.

**Multi-Threaded Programming:** Overview; Multithreading models; Thread Libraries; Threading issues.

**UNIT-II**

**Process Scheduling:** Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; Thread scheduling.

**Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

**UNIT-III**

**Process Synchronization:** The Critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

**Memory Management:** Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

**UNIT-IV**

**File System, Implementation of File System**  
File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection.

Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.
UNIT-V

Secondary Storage Structures, Protection
Mass storage structures;
Disk structure; Disk attachment; Disk scheduling; Disk management;
Swap space management.

Emerging Operating System
Mobile O.S – Symbian, Android, Meebo (Comparative study in terms of
processor scheduling, Memory management, File system and I/O)
Devices) 11 Hours

TEXT BOOK:
1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating
   System Principles, 7 ed Wiley-India, 2006

REFERENCES BOOK:
1. William Stallings,”Operating System Internals and Design
   Gary Nutt,” Operating System”3 ed, Pearson Education,2004
2. D.M Dhamdhere: Operating systems - A concept based Approach,

***********************
C# AND .NET TECHNOLOGIES

Sub Code: 13IS511  
Total Hours: 39

UNIT-I

The Philosophy of .NET

The .NET Solution, The Building Block of the .NET Platform (CLR, CTS, and CLS ), Additional .NET-Aware Programming Languages, An Overview of .NET Binaries (aka Assemblies ), The Role of the Common Intermediate Language, NET Type Metadata and Assembly Manifest, Understanding the CTS, CLS and CLR, NET Namespaces, ildasm.exe, Deploying the .NET Runtime.

Building C# Applications
The Role of the Command Line Compiler (csc.exe), Building C# Application Using csc.exe, Working with csc.exe Response Files, Remaining C# Compiler Options, The Command Line Debugger (cordbg.exe)  
8 Hours

UNIT-II

Introduction to C#: Overview of Writing Applications using C#; Data types, Operators, and Expressions; C# Programming Language Constructs.

Creating Methods, Handling Exceptions: Creating and Invoking Methods; Creating Overloaded Methods and Using Optional and Output Parameters; Handling Exceptions  
8 Hours

UNIT-III

Reviewing the pillars of OOP:
The First Pillars: C#’s Encapsulation Services,
The Second Pillar: C#’s Inheritance, Extending .NET Framework Classes; Creating Generic Types.
The Third Pillar: C#’s Polymorphic Support, Casting rules, Partial types. 
The Role of Application Roots, Understanding Object Generations, The
System.GC Type, Building Finalizable Objects, Building Disposable Objects, Building Finalizable and Disposable Types.  
**File I/O and Object Serialization:** Directory, Directory Info, Files and File Types, Stream Writers and Readers, Object Serialization(Overview)  
8 Hours

**UNIT-IV**

**Advanced C# Features:** Collections and its variants, Delegates and Event Handling, Indexers, Operator Overloading, Anonymous Types, Extension Methods, Pointers and LINQ  
**Accessing a Database:** The ADO.NET Data Provider Factory Model and ADO.NET Disconnected Layer (Overview).  
8 Hours

**UNIT-V**

**Windows Presentation Foundation:** Introduction, Motivation, Flavors, Assemblies, Building WPF Application with and without using XAML,XAML Keywords and Data Binding.  
7 Hours

**TEXT BOOKS:**

**REFERENCES BOOKS:**

******************
ADVANCED UNIX SYSTEM PROGRAMMING

Sub Code : 13IS512  Credits : 03
Hrs/Week : 3+0+0+0  Total Hours: 39

Course Outcomes:

1. To learn about the basic standards for UNIX operating system
2. Know about the different file types available in UNIX
3. To understand how the UNIX OS supports the opening and closing of a file.
4. To apply the General file API’s to handle the files in UNIX operating system.
5. Understand the environment of process which is executing.
6. To understand how the UNIX OS supports creation of a process.
7. To apply the Process related API’s starting from creation of a process to termination.
8. To understand how the UNIX OS supports the signal handling for every process.
9. To apply the signal API’s to control the process.
10. To apply the interprocess communication API’s to communicate between two or more processes.
11. To apply the socket API’s to establish, send and receive messages between client and server process.

UNIT-I

The POSIX standards, File types, General File APIs, File handling programs.  

7 Hours

UNIT-II

Makefile – introduction, creation and execution of make file.
THE PROCESS: Introduction, Mechanism for creating process. The UNIX Kernel support for process.
THE ENVIRONMENT OF A UNIX PROCESS: Introduction, main function, Process Termination, Command line arguments, Environment List, Memory layout Of a C program, Memory allocation, Environment variables, functions.  

8 Hours
UNIT-III

Setjmp and longjmp functions, getrlimit, setrlimit


SIGNALS: The UNIX Kernel Support for signals, Signal.

8 Hours

UNIT-IV

Signal mask, Sigaction, The SIGCHLD Signal and waitpid functions, Thesigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX .1b Timers.

DAEMON PROCESSES: Introduction, Daemon Characteristics, Coding Rules.

8 Hours

UNIT-V

INTERPROCESS COMMUNICATIONS: Overview of IPC Methods, Pipes, popen, Pclose functions, FIFOs, Message Queues, Semaphores, Shared Memory.

SOCKETS: Introduction, functions, Client/Server Message Handling Example.

8 Hours

TEXT BOOKS:

REFERENCE BOOKS:
3. R. Stones, N. Matthew, Beginning Linux Programming, Wrox publication.

*******************

OPERATIONS RESEARCH

Sub Code : 13IS513                  Credits : 03
Hrs/Week  : 3+0+0+0                  Total Hours :39

UNIT-I

Introduction
Introduction to OR, nature and meaning, applications, modeling in OR, phases of OR study

Linear Programming
Introduction to Linear Programming through an example, graphical method, formulation of LP model from practical problems, assumptions and properties of linear programming, simplex method 8 Hours

UNIT-II

Revised simplex method, big M method, 2 phase method, duality theory, primal and dual relationship, dual simplex method 8 Hours

UNIT-III

Transportation assignment
Special types of main programming, transportation problems, methods to find initial feasible solution and modification to obtain optimal solution (Degeneracy in transportation problems, unbalanced transportation problems 8 Hours

UNIT-III

Transportation assignment
Special types of main programming, transportation problems, methods to find initial feasible solution and modification to obtain optimal solution (Degeneracy in transportation problems, unbalanced transportation problems 8 Hours
UNIT-IV

Assignment problem
Mathematical formulation of an assignment problem, unbalanced assignment problem, TSP, Hungarian method

7 Hours

UNIT-V

CPM, PERT
Representation of a project by a network, activities and events, starting times, finishing times, floats, slacks, CPM, Idea of crashing probabilistic times, PERT analysis

8 Hours

TEXTBOOKS:

REFERENCE BOOKS:
1. Operation research, Kantiswaroop, Manmohan and Gupta
2. Introduction to operation research, a computer oriented algorithmic approach, Gillelt B G, McGraw Hill, 1976

********************

SYSTEM SOFTWARE LAB

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS506</td>
<td>01+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Total Hours:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0+2+2+0</td>
<td>26</td>
</tr>
</tbody>
</table>

Course Outcomes:

1. a) Job/process concepts
   b) Scheduling basics : Preemptive vs non Preemptive scheduling
   c) Context switching
   d) Scheduling algorithms: FCFS, SJF, priority scheduling, round robin
2. a) To develop a description of deadlocks, which prevent sets of concurrent processes from completing their tasks. 
   b) Methods for preventing or avoiding deadlocks in a computer system.

3. a) Implement variations of page replacement memory management schemes. 
   b) Replacement strategies: OPT, FIFO, LRU (and approximations)

4. a) Design and Implementation of a SIC assembler. 
   b) Implement a Text Editor.

LAB PROGRAMS

PART A (OS PROGRAMS)
1. Write a Program to show the context switching. 
2. Implement following CPU Scheduling algorithms. Compare the following for Turn around time, waiting time, number of context switches in the CPU. 
   FCFS 
   SJF (non preemptive)  
   Priority Scheduling  
   Round Robin  
3. Implement Banker’s algorithm  
4. Implement following memory replacement algorithms: 
   FIFO  
   LRU  
   Optimal

PART B (MINI PROJECT)
1. Implement a One Pass Assembler for the working model of 8086. 
2. Implement a Two Pass Assembler for the working model of 8086. 
3. Implement a Text Editor.
COMPUTER GRAPHICS

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credits</th>
<th>Hrs / Week</th>
<th>Total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS601</td>
<td>04</td>
<td>4+0+2+0</td>
<td>52</td>
</tr>
</tbody>
</table>

**Course Outcomes:**
1. To understand basics of graphics, its use, its tools.
2. To study different algorithms for scan conversion, clipping.
3. To understand the representation of 2D transformations and its implementation.
4. To understand the representation of 3D transformations and its implementation.
5. Study the types of projections. Understand different Color models.
7. Understand basics of visible surface determination. Study of different algorithms.
8. To learn about the basic functions of various techniques.

**UNIT - I**

Introduction, Raster and random scan displays, video controller, Applications of Computer Graphics.

**RASTER GRAPHICS ALGORITHMS:**

Scan converting lines & circles: Midpoint algorithm, Filling rectangles, Filling Polygons, Clipping lines: Cohen Sutherland, Liang Barsky algorithms, Clipping polygons: Sutherland-Hodgeman algorithm, Antialiasing 12 Hours

**UNIT - II**

**GEOMETRICAL TRANSFORMATIONS:**


The window to view port transformation. Matrix representation of 3D Transformations, Transformations as change in coordinate system.
VIEWING IN 3D:


10 Hours

UNIT – III

CURVES, FRACTALS AND SHADING :
Polygon surfaces, curved lines and surfaces, Quadratic surfaces, Spline Representation, Bezier & B-Spline Curves & Surfaces, Fractal Geometry methods, Illumination models, Shading models for polygons , surface details and shadows. 

9 Hours

UNIT – IV

VISIBLE SURFACE DETERMINATION:

10 Hours

UNIT-V

OpenGL

11Hours

[Only OpenGL functions]

TEXT BOOKS :
REFERENCE BOOKS:
Addison-Wesley Publishing Company

**************************

LABORATORY COMPONENTS

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Credit: 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0+0+2+0</td>
<td></td>
</tr>
<tr>
<td>Total Hours</td>
<td>26(13 weeks)</td>
</tr>
</tbody>
</table>

Course Outcomes:

1. Should be able to use the basic concepts.
2. Be able to use the concept of Conversion algorithm.
3. Implement the concept of line and polygon clipping.
4. Implement the concept of different transformations.
5. Implement different viewing techniques.
6. Implement different hidden surface removal techniques.
7. Implement the different OpenGL concepts and develop a project.

A. Student has to write and execute programs in C/C++ using OPENGL on Windows/Linux platform to implement a few graphics applications like:
   1. Transformations in both 2D and 3D
   2. Clipping
   3. 3D viewing
   4. Hidden line removal
   5. Fractal generation

B. Student may also be asked to implement one or two graphics algorithms like Line drawing or Circle drawing or Filling by using only graphic primitives

C. Graphics Mini project implementation using Open GL.

**************************
COMPUTER NETWORKS

Sub Code : 13IS602
Credits : 04
Hrs/Week : 4+0+2+0
Total Hours: 52

Course Outcomes:

1. Addresses 5 issues: uses of computer networks, business application, home application, mobile users and social users
2. To discuss general aspects of LAN standards, LAN structures and LAN protocols
3. To explain Medium Access Control (MAC) sub layer and Logical Link Control (LLC) sub layer
5. To study the concept of routing and various routing algorithms
6. To discuss the concept of congestion control and different congestion control algorithms
7. To study requirements and different techniques for achieving good QOS
8. To know how networks differ, how networks can be connected, Tunneling concepts.
9. The description about Internet Control Protocols, OSPF, BGP, and IP Addressing
10. To learn the transport services and elements of transport protocols
11. To explain introduction to UDP, RPC, and Real-Time Transport Protocols
12. To explain TCP Service model, TCP Protocol, TCP transmission policy, TCP congestion control, and TCP connection management
13. To study different Performance issues in Computer Networks
14. To learn concept of DNS and WWW

UNIT-I

Introduction to Computer Networks

11 Hours

UNIT-II

Network layer
Network layer design issues: Store and Forward packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual Circuit and Datagram Subnets.

Routing
Routing algorithms: The Optimality Principal, Shortest Path Routing, Flooding, Distance Vector Routing, Link state Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Routing for Mobile Hosts

11 Hours

UNIT-III

Congestion Control

11 Hours

UNIT-IV

Internet Control Protocols: ARP, RARP, ICMP, BOOTP, DHCP
OSPF – The interior gateway routing protocol
BGP – The exterior gateway routing protocol
The Transport Layer
The Transport Service: Services Provided to the Upper Layers, Transport Service Primitives, Berkley Sockets; Elements of Transport Protocols: Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash Recovery

9 Hours

UNIT-V


10 Hours

TEXT BOOKS:

REFERENCE BOOKS:

******************************
LABORATORY COMPONENTS

Hrs/Week : 0+0+2+0
Total Hours : 26 (13 weeks)

Course Outcomes:

1. The study about FIFO, file transfer and message transfer using TCP and UDP
2. The study about Network Simulator to simulate some network scenarios and to calculate performance evaluation
3. The study about Networking tools and router configuration
4. Implement Routing and Congestion control algorithms using C

A. Student must carry out socket programming based experiments for file transfer and message transfer using TCP and UDP
B. Use of any one Network Simulator (Like NS2/NCTUNS 5) to simulate some networking scenario and performance evaluation. Tcl / Tk programs also may be included
C. Study of Router/Firewall configuration, Use of Network Monitoring tool, study of Proxy settings

JAVA AND INTERNET TECHNOLOGIES

Sub Code : 13IS603
Hrs/Week : 4+0+2+0

Course Outcomes:

1. To apply the class concepts to solve problems
2. To apply inheritance in the object oriented programming concepts in java
3. To apply package to class concepts.
4. To apply exception handling methods in catching the exceptions of a class.
5. To apply the thread creation, synchronization, thread priorities and inter thread communication to class.
6. To apply the event handling concepts to interactive java programs
7. To create GUIs using java constructs.
8. To apply the AWT controls, layout managers and menus to enhance the usability of the project.
9. To apply the concepts of reading and writing from the class and to the class.
10. To apply the file handling concepts.
11. To apply the connection procedure to connect database and do the transactions
12. To implement RMI concepts for communication with remote systems.
13. To apply networking concepts in java programming
14. To apply the JavaScript concepts in internet programming in manipulating the client side output.
15. To apply the php concepts to handle the data in the server side

UNIT-I

Introducing Classes – Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, this keyword, Method overloading, Using objects as parameters, Argument passing, Returning objects, Access control, static, final, Using command line arguments, variable length arguments.
Inheritance – Inheritance Basics, Using super, creates a Multilevel Hierarchy, When constructors are called? Method Overriding, Using abstract classes, Using final with Inheritance.
Packages and Interfaces – Packages, Access protection, Importing Packages, Interfaces.
Exception Handling – Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try statements, throw, throws, finally.

10 Hours
UNIT-II

Multithreaded Programming – The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, isAlive and join methods, Thread Priorities, Synchronization, Inter-thread Communication.


11 Hours

UNIT-III


File Handling - Serial Access Files, File Methods, Redirection, Command Line Parameters, Random Access Files.

12 Hours

UNIT-IV

Java Database Connectivity (JDBC) - The Vendor Variation Problem, SQL and Versions of JDBC, Creating an ODBC Data Source, Simple Database Access, Modifying the Database Contents, Transactions, Meta Data, Scrollable ResultSets in JDBC, Modifying Databases via Java Methods, Using the DataSourceInterface

Concurrency Utilities, Synchronization objects. Collection framework, Collection interfaces, Collection classes. Legacy classes.

and IP Addresses, Internet Services, URLs and DNS, TCP, UDP. The InetAddressClass, Using Sockets (TCP and UDP).

8 Hours

UNIT V


PHP

Overview of PHP, General Syntactic Characteristics, Primitives, Output, Arrays, Functions, Pattern Matching, Form Handling, Files. Database Access with PHP and MySQL.

11 Hours

TEXT BOOKS:

REFERENCE BOOKS:

***************************************
LABORATORY COMPONENTS

Hrs/Week : 0+0+2+0
Total Hours : 26 (13 weeks)

Course Outcomes:

1. To implement programs using class and objects.
2. To implement program using various inheritance concept.
3. To implement programs using packages.
4. To implement Programs using exception handling.
5. To implement the thread creation, synchronization, thread priorities and inter thread communication to class.
6. To implement GUI programs that handles the events.
7. To apply the concepts of reading and writing from the class and to the class.
8. To implement file handling programs.
9. To implement Database connectivity to java program.
10. To implement basic java script programs.
11. To implement database connection programs using PHP.

1. Java programs that includes each of the topic discusses in the theory.
2. Programs on Java Script, Simple programs and programs to design simple user interface using HTML.
3. a) Simple programs on PHP to process user data.
   b) Developing a data base application using PHP, Mysql and Apache – HTMLas GUI.

INSTRUCTIONS:

1. In exam two programs may be asked randomly in any combination.
2. Any number of programs can be practiced in lab under each section.
3. Programs should incorporate as many features as possible.
4. Program need may be based on various features of the technology being used.
5. It is mandatory to score minimum marks both in Theory and Lab.

*******************
OBJECT ORIENTED MODELING & DESIGN

Sub Code : 13IS604
Hrs/Week : 4+0+0+0
Credits : 04
Total Hours: 52

Course Outcomes:

1. To learn about the basic functionalities of Object Orientation, OO development, usefulness and history of OO modeling
2. To know about Modeling as Design Technique
3. To study about different types of modeling such as class modeling
4. To learn about advanced techniques in class modeling
5. To know about how to draw the state diagram for different states
6. To learn about advanced techniques in state modeling
7. To study the representation of classes and states to a single diagram
8. To link OO method to SDLC
9. To device and elaborate the system and then prepare the problem statement
10. To study about how the problem statement is converted to a domain model
11. Helps to study the analysis in application model level
12. To study the Estimation the performance and reusability of software
13. To study how to design the algorithm
14. To learn about the implementation of the class, state and interaction models
15. To study about reverse engineering

UNIT-I

Introduction, Modeling concepts, Class modeling

What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history, Modeling as Design Technique: Modeling; abstraction; The three models. Class Modeling: Object and class concepts; Link and associations
concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips.

**Object Oriented Methodologies**
Rambaugh, Booch, Jacobson et al. Methodologies  
10 Hours

**UNIT – II**

**Unified modeling language**
Static and Dynamic models, Modeling, The importance of modeling, Four principles of modeling, Object oriented Modeling, An overview of UML, A conceptual model of UML – Building blocks of UML, Common mechanisms in UML, Software Architecture, Software development life cycle.

**Advanced class modeling, state modeling**
Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips. State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips.  
11 Hours

**UNIT – III**

**Advanced state modeling, Interaction modeling**
Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips. Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models.  
11 Hours

**UNIT – IV**

**Process overview, System conception, Domain analysis**
Process Overview: Development stages; Development life cycle. System Conception: Devising a system concept; Elaborating a concept; Preparing a problem statement. Domain Analysis: Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis.

**Object analysis-classification**
Introduction, Classifications theory, Approaches for identifying classes, Noun phrase approach, Common class patterns approach, Use-case driven approach - identifying classes and their behaviors through Sequence/collaboration modeling, Classes, Responsibilities and collaborators approach  

**UNIT – V**

**Application analysis**
Application Analysis: Application interaction model; Application class model; Application state model; Adding operations.

**Class design, Implementation modeling**
Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example.
Implementation Modeling: Overview of implementation; Fine-tuning classes; Fine-tuning generalizations; Realizing associations; Testing.

**Design pattern**
What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description; Communication Patterns - Publisher-Subscriber.

**TEXTBOOKS:**
3. The unified modeling language user guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Publisher: Addison Wesley, First Edition: (Chapters 1, 2)
REFERENCE BOOKS:

WEB 2.0 PROGRAMMING AND WEB SERVICES
Sub Code : 13IS611
Crédits : 03
Hrs/week : 3+0+0+0
Total Hours: 39

UNIT – I
INTRODUCTION: What is Web 2.0?, Folksonomies and Web 2.0, Software as a Service (SaaS), Data and Web 2.0, Convergence, Iterative development, Rich User experience, Multiple Delivery Channels, Social Networking.
What is JSON?, Array literals, Object literals, Mixing literals, JSON Syntax, JSON Encoding and Decoding, JSON versus XML.3 Hrs
BUILDING RICH INTERNET APPLICATIONS WITH AJAX:
Limitations of Classic Web application model, AJAX principles, Technologies behind AJAX, Examples of usage of AJAX; Asynchronous communication and AJAX application model XMLHTTP Object – properties and methods, handling different browser implementations of XMLHTTP; AJAX Patterns (Only algorithms – examples not required): Predictive fetch pattern, Submission throttling pattern, Periodic refresh, Multi stage download, Fall back patterns. Introduction to JQuery.

UNIT – II
SOAP: The Case for SOAP; What Does SOAP Define? SOAP Message Structure; SOAP Message Elements; SOAP Processing Model; SOAP Encoding;
WSDL: Describing a Web Service; Describing Functional Characteristics of Services; WSDL 1.2; 4 Hours
**UDDI**: Discovering Web Services; Categorizing Services; Identifiers; Business Entity Relationships; UDDI's SOAP Interfaces; UDDI and SOAP/WSDL Relationships; Publishing WSDL Service Interfaces in UDDI; Internationalization and Multiple Languages; Extending a UDDI Registry; Private UDDI Registries;  

**ebXML**: Architectural Overview of ebXML.

**UNIT – III**

**Java Web Service Developer Pack**: Setting up Java WSDP, Java WSDP components.  
**JAXP**: JAXP Architecture; SAX; DOM; When to Use SAX; When to Use DOM; When Not to Use Either; JAXP and XML Schemas; XSLT; XSLTc; JDOM; JAXP RI;  

4 Hours

**JAX-RPC**: JAX-RPC Service Model; Data Types and Serialization; JAX-RPC Development; Advanced JAX-RPC; JAX-RPC Interoperability; JAX-RPC and J2EE;

**UNIT – IV**

**JAXM**: Messaging and MOM; Messaging and Web Services; Messaging in Java; JAXM Architecture; Designing with JAXM; Developing with JAXM;  
**JAXR**: Registries and Repositories; JAXR Architecture; The JAXR Information Model; The JAXR API; JAXR to UDDI Mapping; JAXR and ebXML Registry;  

4 Hours

**JAXB**: The Need for Binding and JAXB; When to Use JAXB; JAXB Architecture; Developing with JAXB; XML-to-Java Mapping; The JAXB API; Validation with JAXB; Customizing JAXB; When to Use Custom Declarations;

**UNIT – V**
**Transaction Management:** Concepts; A Transaction Model for Web Services; New Transaction Specifications; JSRs for Web Service Transaction Support;  

**Security:** Security Considerations for Web Services; Web Services Security Initiatives; Canonical XML; XML Digital Signatures; Apache XML Security; XML Encryption; Security Assertions Markup Language; Web Services Security Assertions; XML Access Control Markup Language; XML Key Management Specification; WS-I Specifications; SOAP and Firewalls;

**TEXT BOOKS:**

**REFERENCE BOOKS:**

********************************************************************************

**SOFTWARE ARCHITECTURE**

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credit</th>
<th>Hrs/Week</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS612</td>
<td>03</td>
<td>3+0+0+0+0</td>
<td>39</td>
</tr>
</tbody>
</table>

**UNIT-I**

**Introduction:** The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle; What makes a “good” architecture? What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views.
UNIT-II

Architectural styles and case studies: Architectural styles; Pipes and filters; Data abstraction and object-oriented organization; Event-based, implicit invocation; Layered systems; Repositories; Interpreters; Process control; Other familiar architectures; Heterogeneous architectures. Case Studies: Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style.

UNIT-III

Quality: Functionality and architecture; Architecture and quality attributes; System quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architectural qualities. Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Architectural patterns and styles. 8 Hours

UNIT-IV

Architectural patterns – 1: Introduction; from mud to structure: Layers, Pipes and Filters, Blackboard.

UNIT-V

Designing and documenting software architecture: Architecture in the life cycle; Designing the architecture; Forming the team structure; Creating a skeletal system. Uses of architectural documentation; Views; Choosing the relevant views; Documenting a view; Documentation across views.

TEXT BOOKS:
2. Pattern-Oriented Software Architecture A System of Patterns, Volume 1 – Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, John Wiley and Sons, 2006

REFERENCE BOOKS:
2. Design Patterns- Elements of Reusable Object-Oriented Software - E. Gamma, R. Helm, R. Johnson, J. Vlissides, Addison-Wesley, 1995.

******************************

ADVANCED DBMS

Sub Code : 13IS613
Credits : 03
Hrs/Week : 3+0+0+0
Total Hours : 39

UNIT-I

Overview of storage and indexing, disks and files:
Data on external storage; File organizations and indexing; Index data structures; Comparison of file organizations; Indexes and performance tuning. Memory hierarchy; RAID; Disk space management; Buffer manager; Files of records; Page formats and record formats.
Tree structured indexing: Intuition for tree indexes; Indexed Sequential Access Method (ISAM); B+trees - Search, Insert, Delete, Duplicates; B+ trees in practice.

UNIT-II

Hash based indexing: Static hashing; Extendible hashing, Linear hashing, comparisons.
External Sorting: When does a DBMS sort data? A simple two-way merge sort; External merge sort, Using B+ trees for sorting.
Evaluating Relational Operators: The Selection operation; General selection conditions; The Join operation; The Projection operation; The Set operations; Aggregate operations; The impact of buffering.
UNIT-III

**Query Optimization:** Using Heuristics in Query Optimization, Using selectivity and cost estimates in Query Optimization, Overview of Query optimization in Oracle, Semantic Query Optimization.

**Physical Database Design and Tuning:** Introduction; Guidelines for index selection, examples; Clustering and indexing; Indexes that enable index-only plans; Overview of database tuning; Choices in tuning the conceptual schema; Choices in tuning queries and views; Impact of concurrency; DBMS benchmarking.

UNIT-IV

**Object Databases:**

**Concepts for Object Databases:** Overview of Object-Oriented Concepts, Object Identity, Object Structure, and Type Constructors, Encapsulation of Operations, Methods, and Persistence, Type and Class Hierarchies and Inheritance, Complex Objects;

**Object Database Standards, Languages, and Design:** Overview of the Object Model of ODMG, The Object Definition Language ODL, The Object Query Language OQL, Overview of the C++ Language Binding, Object Database Conceptual Design;

**Object-Relational and Extended-Relational Systems:** Overview of SQL and its Object-Relational features, Object-Relational Features of Oracle 8.

UNIT-V

**Distributed Databases:**

Distributed Database concepts; Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design; Types of Distributed Database Systems; Query Processing in Distributed Databases; Overview of Concurrency Control and Recovery in Distributed databases; Distributed databases in Oracle.

**Security and Authorization:**

TEXT BOOKS:

REFERENCE BOOKS:

****************

INFORMATION STORAGE MANAGEMENT

Sub Code : 13IS621
Hrs/Week : 3+0+0+0
Credits : 03
Total Hours: 39

UNIT-I
Storage System: Introduction to Information Storage and Management: Information Storage, Evolution of Storage Technology and Architecture, Data Center Infrastructure, Key Challenges in Managing Information, Information Lifecycle.

Data Protection: RAID: Implementation of RAID, RAID Array Components, RAID Levels, RAID Comparison, RAID Impact on Disk Performance, Hot Spares. 7 Hours

UNIT-II

Intelligent Storage System: Components of an Intelligent Storage System, Intelligent Storage Array,

Storage Networking Technologies and Virtualization: Direct-Attached Storage and Introduction to SCSI: Types of DAS, DAS Benefits and
Limitations, Disk Drive Interfaces, Introduction to Parallel SCSI, SCSI Command Model.  

UNIT-III

Storage Area Networks: Fibre Channel: Overview, The SAN and Its Evolution, Components of SAN, FC Connectivity, Fibre Channel Ports, Fibre Channel Architecture, Zoning, Fibre Channel Login Types, FC Topologies.


UNIT-IV

IP SAN: iSCSI, FCIP.

Content-Addressed Storage: Fixed Content and Archives, Types of Archives, Features and Benefits of CAS, CAS Architecture, Object Storage and Retrieval in CAS, CAS Examples.

Storage Virtualization: Forms of Virtualization, SNIA Storage Virtualization Taxonomy, Storage Virtualization Configurations, Storage Virtualization Challenges, Types of Storage Virtualization  

UNIT-V


TEXTBOOK:
1. Information Storage and Management by EMC Education Services.

**REFERENCE BOOKS:**
1. Storage Networks Explained by Ulf Troppen, Rainer Erkens, Wolfgang Muller.
2. Storage Networks by Robert Spalding.

***************

**FREE & OPEN SOURCE SOFTWARE**

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS622</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Total Hours:</th>
</tr>
</thead>
<tbody>
<tr>
<td>3+0+0+0+0</td>
<td>39</td>
</tr>
</tbody>
</table>

**Course Outcomes:**

1. To know What is FOSS and How it Emerged
2. Gives an idea of how FOSS are developed
3. To distinguish the way of which Open software’s are distributes
4. Gives an idea of legal policies and rules of FOSS
5. Know about the standards of FOSS
6. Know how FOSS can be protected from Piracy
7. To analyze and interpret the shortcomings of FOSS
8. To know how to document the FOSS
9. To know the basic level of integration and wrappers required for FOSS
10. To know where FOSS can be applied
11. To know how to create a FOSS License and provide Support
12. To know the different full flexed FOSS Software in action
13. To know how to solve technical issues in FOSS
14. To know the marketing environment of FOSS
15. To interpret the best practices of FOSS
16. To interpret the requirements of a FOSS project
17. To know how to create and release different versions of FOSS
18. To know how to wrap and pack FOSS applications
19. To know the different ways of Packing a Software

**UNIT –I**
Free and Open Source Software (FOSS) philosophy and history
FOSS definition; Free and Open Source Software; GNU project; History of GNU/Linux development; Development process of important FOSS software; Development process of various GNU/Linux distributions

**UNIT-II**

Legal, social aspects of FOSS and parallels in other fields
Various licenses including GPL, LGPL, BSD, etc; Copyleft; Patents, copyrights and trademarks; Concept of free culture with reference to Wikipedia, Creative Commons, Open Street Map; Open Movies; Open Access Journals; Open Standards; Open Hardware

**UNIT- III**

**WHY FOSS?**
Is FOSS free? ,How large are the savings from FOSS? ,Direct Cost Savings - An Example
What are the benefits of using FOSS?,Security ,Reliability/Stability,Open standards and vendor independence ,Reduced reliance on imports,Developing local software capacity ,Piracy, IPR, and the WTO,Localization .
What are the shortcomings of FOSS? ,Lack of business applications ,Interoperability with proprietary systems ,Documentation and “polish”
**Practical aspects of FOSS:** business models .
Sharing the burden; Augmenting services; Supporting Hardware Sales; Undermining a competitor; Dual licencising; Donations; Support service; Study of examples of each type of business model

**UNIT-IV**

FOSS development process and tools
Development environments: Eclipse, Anjuta, Kdevloper, Netbeans; Version control; Bug tracking; Wiki; Mailing lists; Forums; Developer communities; IRC; Non technical issue resolution; Promotion; Communication: You are what you write, structure and content, tone,
face, pitfalls, best practices; Process: Benevolent dictators, do-o-cracy, consensus based democracy. 

UNIT-V


Packaging applications:
Package; Package management tools; Building a package; Packaging guidelines; Package acceptance criterion; Packaging for .deb and .rpm based distributions
Following projects: Linux kernel, GNU Project, Open office, Mozilla Firefox, Gimp, Inkscape, Scribus, Silpa, LaTeX, VLC, Mplayer, Virtualbox, MySQL, Postgresql, Sugar, Gnome, KDE, Blender, Google Chrome, Vuze, Scilab, Octave, Pidgin, Evolution, Thunderbird

List of assignments:
* Mediawiki: Set up a mediawiki installation with configuration specified by the instructor.
* Version Control: Install, configure and create a project for the course using git version control system.
* Development:
  • Start participating in at least 1 known FOSS project
  • Take up a task as specified by the instructor
  • Test the software and report bugs to the community
  • Improve the software as per the specification and with quality acceptable to the community.

* Packaging
  • Demonstrate your packaging capabilities by maintaining packages in upstream distributions

TEXTBOOKS:
1. Introduction to Free Software, by David Megías Jiménez and David Megías Jiménez (coordinator) et.al., Published by SELF Project (http://www.selfproject.eu/en/Coursebook_Intro_Free_Software)


3. Producing OSS (http://producingoss.com) by Karl Fogel


REFERENCES:

1. The Cathedral and the Bazaar (CatB), by Eric S. Raymond Published by O'Reilly Media

2. GNU Project Website http://www.gnu.org


DISTRIBUTED SYSTEMS AND OBJECTS

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS623</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hrs/ Week</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>3+0+0+0</td>
<td>39</td>
</tr>
</tbody>
</table>

Course Outcomes:

1. To know about the Models used in DOS
2. To understand the main issues of DOS
3. To know the concept of RPC’s
4. To know features of RPC
5. To know how to call RPC’s
6. To know how to handle issues in RPC
7. To interpret the different ways of how parameters are passed
8. To know why Clocks need to be Synched
9. Analyze the algorithms of Clock Synchronization
10. To analyze how deadlock can appear in DOS
11. To know the different ways of handling deadlocks
12. To know the Concept of process migration
13. To know how to create threads and problems related to thread
14. To know what is DOS file system
15. To know what is Distributed Shared Memory
16. To know what is resource management and ways of handling loads
17. To know the concept of CORBA
18. To know about the CORBA architecture
19. To know how to map languages and objects

UNIT-I

Introduction to Distributed Systems:

**Fundamentals:** What is Distributed Computing Systems?, Distributed Computing System Models, What is DOS?, Issues in designing a DOS.

**Remote Procedure Calls:** The RPC model, Transparency of RPC, Implementing RPC mechanism, Stub generation, RPC messages, Marshaling Arguments and results. Server management, Parameter passing semantics, call semantics, communication protocols RPC’s, Complicated RPCs, Client – server binding, Exception handling, Security. 8 Hours

UNIT-II

Synchronization in distributed Systems:

Clock synchronization – logical clocks – physical clocks – vector clocks– clock synchronization algorithms, Mutual exclusion – A centralized algorithm – A distributed algorithm – a token ring algorithm, Comparison of the three algorithms, Election algorithms – the Bully algorithm – ring algorithm, Dead locks in distributed systems – distributed deadlock avoidance algorithms – distributed deadlock prevention algorithms, distributed deadlock detection algorithms: Centralized approach, Hierarchical approach and Fully distributed approach. 8 Hours

UNIT-III

Process & threads in Distributed Systems:

**Distributed File Systems:**
Desirable features of a good distributed file system, file models, file accessing models, file sharing semantics, File Replication.

8 Hours

**UNIT-IV**

**Distributed Shared Memory:** General structure, Design and implementation issues of DSM, Granularity, structure of shared memory

**Resource Management:** Desirable features, task management approach, load balancing approach, load sharing approach

8 Hours

**UNIT -V**

**Distributed objects**


7 Hours

**TEXT BOOKS:**
1. Distributed Operating Systems, Concepts & Design, Pradeep K Sinha, PHI *(Chapters 1,4,5,6 and 8.2,8.3, 9)*
2. Thomas J. Mowbray and William A. Ruh:Inside CORBA, Addison-Wesley, 1997 *(Chapter 1,2 & 3)*

**REFERENCE BOOKS:**
IEL LAB

**Entry Edge: Immersive Group Workshop (IGW)**

Sub Code: 13IS605  
Duration: 5 Days

Timings: 9.00 AM to 12.30 PM, 1.15 PM to 4.45 PM

---

**Syllabus**

**Module 1: Minds-on and hands-on simulation project**  
17.5 Hours

- Understanding Task environment – Goals, responsibilities, Task focus
- Working in Teams towards common goals
- Organizational performance expectations – technical and behavioural competencies

**Module 2: Re-enforcement of critical individual skills and behaviours**  
3.5 Hours

- Application of individual effectiveness skills in team and organizational context – improving self awareness, goal setting, time management, communication and presentation skills

**Module 3: Etiquettes and Ethics**  
7 Hours

- Professional etiquettes at workplace – dressing, telephone, e-mail, meeting and general behavior
- Basic honesty & respect for law / rules
- Conflict of interest
- Use of organizational resources
- Misrepresentation and misappropriation
- Intellectual property
- Whistle blowing
Module 4: Interpersonal Behaviour & relationship skills  
3.5 Hours
- Establishing trust based relationships in team & organizational environment
- Trust equation – credibility, responsiveness, integrity, self-interest

Module 5: Dealing with Conflicts  
3.5 Hours
- Orientation towards conflicts in team and organizational environment
- Understanding sources of conflicts
- Conflict resolution styles and techniques

Pedagogical tools & techniques used in the workshop
- Organizational templates for simulating a organizational context-structures, units, roles and activities
- Metaphoric scenarios for simulating real –life tasks and dynamics in a team/project context
- LEGO™ building blocks for simulating last-mile technical activity in teams
- Case studies, Role play scenarios group learning activities, observation and feedback.

Note: Evaluation is done and a grade of P (pass) or NP (not pass) is awarded.

********************
EMPLYABILITY SKILL DEVELOPMENT

Sub Code    : 13IL001/002  
Credits      : Nil (MLC)  
Hrs/Week     : 1+0+0+0    
Total Hours : 12

UNIT - I
Analytical Aptitude Skill: concept of analytical skill, definition-logical thinking and testing of Analytical Aptitude

UNIT - II
Quantitative Aptitude skill-Concept-definition-Preliminary requirement for development of quantitative skill- testing of quantitative skill.

UNIT - III
Verbal and ability skill – Knowledge and Vocabulary and grammar-comprehension-Verbal Reasoning skill

REFERENCE BOOKS:


**Examination pattern:**
This course is a mandatory learning course without credit. Continuous internal examination (CIE) consists of 2 internal exams (20 marks each) and tasks (10 marks). There is no semester end examination (SEE). The student will be awarded PP or NP grade as per autonomous regulations.

***************