INFORMATION SCIENCE & ENGINEERING

V & VI SEMESTER

With
Scheme of Teaching
& Examination
<table>
<thead>
<tr>
<th></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. Udaya Kumar K Shenoy</td>
<td>Ph.D</td>
<td>Professor</td>
</tr>
<tr>
<td>3</td>
<td>Karthik Pai B H</td>
<td>M.Tech</td>
<td>Asst. Prof Gd III</td>
</tr>
<tr>
<td>4</td>
<td>Ashwini B</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>5</td>
<td>Deepa</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>6</td>
<td>Devidas</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>7</td>
<td>Rashmi Naveen</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>8</td>
<td>Vasudev Pai</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>9</td>
<td>Pranesh</td>
<td>M.Tech</td>
<td>Asst Prof Gd-II</td>
</tr>
<tr>
<td>10</td>
<td>Rakesh Joshi U</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>11</td>
<td>Manasa</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>12</td>
<td>Savita Sthawarmath</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>13</td>
<td>Abhir Bhandary</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>14</td>
<td>Srikanth Bhat. K</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>15</td>
<td>Jason Elroy Martis</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>16</td>
<td>Sunil Kumar Aithal</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>17</td>
<td>Shwetha Bhat M</td>
<td>B.E</td>
<td>Asst Prof Gd 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 judgement coupled with business and entrepreneurial skills.

Program: B.E. Information Science & Engineering

Programme Educational Objectives (PEO’s)
- 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 (PO’s)
BE (ISE) Engineering Program 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.
<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Contact Hours/week</th>
<th>CIE</th>
<th>SEE</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory / Tut / Pract.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12IS501</td>
<td>System Software</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>12IS502</td>
<td>Relational Database Management System</td>
<td>4+0+2</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>12IS503</td>
<td>Microprocessors and Interfacing</td>
<td>4+2+2</td>
<td>50</td>
<td>50</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>12IS504</td>
<td>Software Engineering</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>12IS505</td>
<td>Operating System</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>12IS51X</td>
<td>Elective -I</td>
<td>3+0+0</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>12IS506</td>
<td>System Software Lab</td>
<td>0+2+2</td>
<td>50</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>33</strong></td>
<td><strong>350</strong></td>
<td><strong>350</strong></td>
<td><strong>28</strong></td>
</tr>
</tbody>
</table>

**Elective –I**
1. 12IS511 - C# & .NET Technologies
2. 12IS512 – Advanced Unix System Programming
3. 12IS513 – Operations Research
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
SCHEME OF TEACHING

VI Semester

* This will be carried out by the student in a team; This concept may be used in developing mini project in Java and Internet Technologies. Students need to submit a report at the end of the semester.

** These classes will be held only during the first week of the semester

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Contact Hours / Week</th>
<th>CIE</th>
<th>SEE</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory/Tut/Pract.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12IS601</td>
<td>Computer Graphics</td>
<td>4+0+2</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>12IS602</td>
<td>Computer Networks</td>
<td>4+0+2</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>12IS603</td>
<td>Java and Internet Technologies</td>
<td>4+0+2</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>12IS604</td>
<td>Object Oriented Modeling and Design</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>12IS61X</td>
<td>Elective – II</td>
<td>3+0+0</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>12IS62X</td>
<td>Elective – III</td>
<td>3+0+0</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>12IS605</td>
<td>IGW lab*</td>
<td>0+4**+0</td>
<td>50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL</td>
<td>32</td>
<td>350</td>
<td>300</td>
<td>25</td>
</tr>
</tbody>
</table>

**Elective –II**
1. 12IS611 – Web2.0 Programming and WebServices
2. 12IS612 – Software Architecture
3. 12IS613 – Advanced DBMS

**Elective –III**
1. 12IS621 – Information Storage Management
2. 12IS622 – Free & Open Source Software (FOSS)
3. 12IS623 - Distributed Systems and Objects
SYLLABUS OF V & VI SEMESTER B.E./INFORMATION SCIENCE & ENGG.

SYSTEM SOFTWARE

Subject Code: 12IS501
Credits: 04
Hrs/Week: 4
Total Hours: 52

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.

UNIT - II

Assemblers:

UNIT - III

Loaders and Linkers:

UNIT - IV

Editors and Debugging Systems:
Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship with Other Parts of The System, User-Interface Criteria.

UNIT - V

Macro Processor:
Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features - Concatenation of Macro

11 Hrs

TEXT BOOKS:

REFERENCE BOOKS:

RELATIONAL DATABASE MANAGEMENT SYSTEMS

Subject Code :12IS502   Credits : 04
Hrs/Week : 4   Total Hours : 52

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 Hrs

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  

**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

**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; Boyce Codd Normal Form; Properties of Relational Decompositions; Algorithms for Relational Database Schema Design; Multivalued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form.

**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

**TEXT BOOKS :**


**REFERENCE BOOK:**

LABORATORY COMPONENTS

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

Credit : 01

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

Subject Code : 12IS503  
Hrs/Week : 4+2  
Total Hours : 52

Credits : 04+01

UNIT - I

Microprocessor and 8086 Architecture
Overview of Microcomputer Structure and Operation, Microprocessor Evolution and Types, 8086 Internal Architecture, Introduction to Programming the 8086 8086 Instruction Description and Assembler Directives  
11 Hrs

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-Until Programs, Instruction Timing and Delay Loops  
11 Hrs

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.  
10 Hrs
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.

10 Hrs

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,


10 Hrs

TEXT BOOK:

REFERENCE BOOKS:

LABORATORY COMPONENTS

Hrs/Week : 2  Credit : 01
Total Hours : 26(13 weeks)

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 \( \binom{n}{r} \) 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 couples 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

Subject Code : 12IS504
Credits       : 04
Hrs/Week      : 4
Total Hours   : 52

UNIT - 1

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.


**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.  

**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.  

**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.  

**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.  

**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
Subject Code : 12IS505
Credits : 04
Hrs/Week : 4
Total Hours : 52

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. 11 Hrs

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. 10 Hrs

UNIT - III
Process Synchronization: The Critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. 6 Hrs
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. 9 Hrs 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  Cloud O.S – eye O.S, Windows Azure O.S, Ubuntu public cloud. Mobile O.S – Symbian, Android, Meebo (Comparative study in terms of processor scheduling, Memory management, File system and I/O) 11 Hrs

TEXT BOOK:

REFERENCES BOOK:
# C# and NET TECHNOLOGIES

**Subject Code**: 12IS511  
**Exam Hours**: 3  
**Hours/ Week**: 0  
**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 Hrs

## 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 Hrs

## UNIT - III

### Reviewing the pillars of OOP:
The First Pillar: 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 Hrs
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 Hrs

UNIT - V

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

Text books:

1. Pro C# 5.0 and the .NET 4.5 Framework (Professional Apress).  

References:


ADVANCED UNIX SYSTEM PROGRAMMING

Subject Code: 12IS512 
Credits: 03  
Hrs/Week: 3  
Total Hours: 39

UNIT - I

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

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 Hrs

UNIT - III
Setjmp and longjmp functions, getrlimit, setrlimit


SIGNALS: The UNIX Kernel Support for signals, Signal. 8 Hrs

UNIT - IV
Signal mask, Sigaction, The SIGCHLD Signal and waitpid functions, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX .1b Timers.

DAEMON PROCESSES: Introduction, Daemon Characteristics, Coding Rules. 8 Hrs

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 Hrs

TEXT BOOKS:

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

Subject Code : 12IS513  
Credits : 03
Hrs/Week : 3  
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 Hrs

UNIT - II
Revised simplex method, big M method, 2 phase method, duality theory, primal and dual relationship, dual simplex method 8 Hrs

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 Hrs

UNIT - IV
Assignment problem
Mathematical formulation of an assignment problem, unbalanced assignment problem, TSP, Hungarian method 7 Hrs

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

TEXTBOOKS:

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

Subject Code : 12IS506        Credit : 01+01
Hrs/Week     : 2               Total Hours : 26(13 weeks)

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

Subject Code : 12IS601        Credits   : 04
Hrs / week    : 4               Total hours : 52

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 Hrs
UNIT - II

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

VIEWING IN 3D:

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.

UNIT - IV

VISIBLE SURFACE DETERMINATION:

UNIT - V

OpenGL
[Only OpenGL functions]

TEXT BOOKS:

REFERENCE BOOKS:
LABORATORY COMPONENTS

Hrs/Week : 2  Credit : 01
Total Hours : 26 (13 weeks)

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

Subject Code : 12IS602  Credits : 04
Hrs/Week : 4  Total Hours : 52

UNIT - I

Introduction to Computer Networks

11 Hrs
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

UNIT - III

Congestion Control

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

UNIT - V

TEXT BOOKS:

REFERENCE BOOKS:

LABORATORY COMPONENTS
Hrs/Week : 2  Credit : 01
Total Hours : 26 (13 weeks)
A. Student must carryout 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

Subject Code : 12IS603 Credits : 04
Hrs/Week : 4 Total Hours : 52

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 Hrs

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.

UNIT - III
Introducing Swings – component and container, Event handling, Painting,Exploring Swings, Swings UI components.
File Handling - Serial Access Files, File Methods, Redirection, Command Line Parameters, Random Access Files. 12 Hrs
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.


UNIT - V


PHP

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

TEXT BOOKS:


REFERENCE BOOKS:


LABORATORY COMPONENTS

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

1. Java programs that includes Inheritance, Package, Exception handling, multithreading.
2. Programs on Java Script, Simple programs and programs to design simple user interface.
3. a) Simple programs on PHP to process user data (Ex: Addition of two numbers)
   b) Developing a database application using PHP, Mysql and Apache – HTML as GUI.
4. Web Programming (Self Study):- XHTML, CSS.

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

Subject Code : 12IS604
Hrs/Week : 4

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 Hrs
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.
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.
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**

**Subject Code**: 12IS611  
**Credits**: 03  
**Hrs/week**: 3  
**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 XMLHttpRequest Object – properties and methods, handling different browser implementations of XMLHttpRequest; AJAX Patterns (Only algorithms – examples not required): Predictive fetch pattern, Submission throttling pattern, Periodic refresh, Multi stage download, Fall back patterns. Introduction to JQuery.  

6 Hrs

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 Hrs

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.  

4 Hrs

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 Hrs

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

4 Hrs

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 Hrs

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; 4 Hrs

UNIT – V

Transaction Management: Concepts; A Transaction Model for Web Services; New Transaction Specifications; JSRs for Web Service Transaction Support; 3Hrs

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; 3Hrs

TEXT BOOKS:

REFERENCE BOOKS:

SOFTWARE ARCHITECTURE

Subject Code : 12IS612

Credits : 03
Hrs/Week : 3
Total Hours : 39

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. 8 Hrs
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. Case Studies: Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style. 8 Hrs

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 Hrs

UNIT - IV
Architectural patterns – 1: Introduction; from mud to structure: Layers, Pipes and Filters, Blackboard.
Architectural patterns – 3: Adaptable Systems: Microkernel; Reflection. 7 Hrs

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. 8 Hrs

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
Subject Code : 12IS613  
Credits : 03
Hrs/Week : 3  
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.  
8 Hrs

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.  
8 Hrs

UNIT - III
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.  
8 Hrs
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 AND MANAGEMENT

Subject Code : 12IS621   Credits : 03
Hrs/Week    : 3   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.


7 Hrs

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.

8 Hrs

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.


8 Hrs

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

9 Hrs
UNIT - V

Business Continuity: Introduction to Business Continuity:
Information Availability, BC Terminology, BC Planning Lifecycle,
Failure Analysis, Business Impact Analysis, BC Technology
Solutions.

Backup and Recovery: Backup Purpose, Backup Considerations,
Backup Granularity, Recovery Considerations, Backup Methods,
Backup Process, Backup and Restore Operations, Backup Topologies,
Backup in NAS Environments, Backup Technologies.

Securing the Storage Infrastructure: Storage Security Framework,
Risk Triad, Storage Security Domains, Security Implementations in
Storage Networking. 7 Hrs

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

Subject Code: 12IS622  Credits : 03
Hrs/Week : 3  Total Hours : 39

Course Objectives
1. Student knows the foss ecosystem and its philosophical and
   technical aspects
2. Student is capable for leading a foss project or working with a foss
   project
3. Student knows the foss development workflow
4. Student is capable of designing, implementing, deploying,
   maintaining a FOSS based solution for the society for solving a
   problem

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 8 Hrs
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 8 Hrs

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 licencesing; Donations; Support service; Study of examples of each type of business model 7 Hrs

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 8 Hrs

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

8 Hrs

**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

Subject Code : 12IS623
Credits : 03
Hrs/ Week : 3
Total Hours : 39

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 Hrs

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 Hrs

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 Hrs

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.
UNIT - V

Distributed objects

7 Hrs

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)

Subject Code: 12IS605  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 Hrs
- 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 Hrs
- 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 Hrs
- Professional etiquettes at workplace – dressing, telephone, e-mail, meeting and general behaviour
• 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 Hrs
• Establishing trust based relationships in team & organizational environment
• Trust equation – credibility, responsiveness, integrity, self-interest

Module 5: Dealing with Conflicts  3.5 Hrs
• 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.