### Shivaji University, Kolhapur

**REVISED STRUCTURE**

T.E. Computer Science & Engg. (Semester – V & VI)

W.E.F. 2015-16.

#### Semester – V

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Subject</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Total</th>
<th>Theory Marks</th>
<th>TW</th>
<th>POE</th>
<th>Oral</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computer Graphics</td>
<td>3</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>50</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>System Programming</td>
<td>3</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>100</td>
<td>-</td>
<td>50</td>
<td>-</td>
<td>175</td>
</tr>
<tr>
<td>3</td>
<td>Object Oriented Modeling and Design</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>50</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>Computer Algorithms</td>
<td>4</td>
<td>1</td>
<td>-</td>
<td>5</td>
<td>100</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>125</td>
</tr>
<tr>
<td>5</td>
<td>Network Technologies</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>Programming Lab - III</td>
<td>3</td>
<td>-</td>
<td>4</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>Business English</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>2</td>
<td>8</td>
<td>30</td>
<td>350</td>
<td>150</td>
<td>200</td>
<td>50</td>
<td>800</td>
</tr>
</tbody>
</table>

#### Semester – VI

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Subject</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Total</th>
<th>Theory Marks</th>
<th>TW</th>
<th>POE</th>
<th>Oral</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compiler Construction</td>
<td>3</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>50</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>125</td>
</tr>
<tr>
<td>2</td>
<td>Operating System - II</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>100</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>125</td>
</tr>
<tr>
<td>3</td>
<td>Database Engineering</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>6</td>
<td>50</td>
<td>50</td>
<td>25</td>
<td>50</td>
<td>175</td>
</tr>
<tr>
<td>4</td>
<td>Storage Networks</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>Information Security</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>25</td>
<td>-</td>
<td>125</td>
</tr>
<tr>
<td>6</td>
<td>Programming Lab - IV</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>50</td>
<td>75</td>
</tr>
<tr>
<td>7</td>
<td>Domain Specific Mini-Project</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>50</td>
<td>75</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>19</td>
<td>1</td>
<td>10</td>
<td>30</td>
<td>400</td>
<td>100</td>
<td>150</td>
<td>100</td>
<td>800</td>
</tr>
</tbody>
</table>

**Note:**

1. The term work as prescribed in the syllabus is to be periodically and jointly assessed by a team of teachers from the concerned department.
2. In case of tutorials, students of different batches be assigned problems of different types and be guided for the solution of the problem during tutorial session. Problems thus solved be translated into computer programs wherever applicable and executed by respective batches during practical session.
3. The assignments of tutorials and practicals need to be submitted in the form of soft copy and / or written journal.
4. Breakup of term work marks shall be as follows:
   a. For subjects having term work marks 25 -
• Mid-semester test – 5 marks.
• End-semester test – 5 marks.
• Tutorial assignments and / or practical performance – 15 marks.

b. For subjects having term work marks 50 –
• Mid-semester test – 10 marks.
• End-semester test – 10 marks.
• Tutorial assignments and / or practical performance – 30 marks.

5. The theory exam scheme is as under:
• 5.1 : For online exam the scheme to be followed is as under –

  a. As mentioned in the structure above, Three theory papers of TE (CSE) Sem-V and Three theory papers of Sem-VI of 100 marks will be divided into two parts.
  • Part-A: 50 marks theory paper similar to the existing theory paper exam. The nature of the questions will be descriptive, analytical and problem solving.
  • Part-B: 50 marks computer based exam with multiple choice questions (MCQs).

  b. The marks obtained in the individual heads should be added and considered as marks of the respective theory paper out of 100 marks.
  c. The questions of part-A and part-B will be based on the entire syllabus of the respective subjects.
  d. The theory paper for part-A will consist of questions on all the Six Units of the syllabus carrying 50 marks.
  e. The questions in part-B will be of 1 or 2 marks only.
  f. Duration of part-A exam will of 2 hours and that of part-B will be of 1 hour.
  g. No separate passing head for part-A and part-B.
  h. The scheme of moderation / revaluation is not applicable for part-B, however is applicable for part-A

• 5.2 : For theory exam of 100 marks the scheme to be followed is as under :
  a. The theory paper of 100 marks will be based on all Six Units of the syllabus.
  b. The scheme of moderation / revaluation is applicable.

6. Passing scheme is as under -
 a. The passing scheme for the subjects will be similar to existing scheme.
 b. All the existing ordinances will be applicable for passing criteria.
T.E. (Computer Science and Engineering) Semester – V

1. COMPUTER GRAPHICS

Lectures: 3 hrs/week  
Practicals: 2 hrs/week  
Theory: 100 marks  
Termwork: 50 marks

Course Objectives:

1. To expose students to the various transformation techniques and projections.
2. To make students understand different algorithms concerned with scanning, filling, windowing and clipping on graphical objects.
3. To make the students aware of generation of curves and surfaces.
4. To give students with hands on exposure to Open GL and Animation tools.

UNIT 1: Transformations
Basic 2D & 3D transformations - Translation, Scaling, Rotation, Reflection, Shearing, Multiple Transformations, Rotation about an axis parallel to a coordinate axis, rotation about an arbitrary axis in space, Affine and Perspective Geometry, Orthographic projections and Axonometric projections. (8)

UNIT 2: Raster Scan Graphics
Bresenham's line and circle drawing algorithms, Scan Conversion techniques: RLE, Frame Buffer, Scan converting polygons: Edge fill and Seed fill algorithms, Anti-aliasing and Half-toning. (7)

UNIT 3: Viewing and clipping
Introduction, Windowing and View-porting, Introduction to clipping, Point clipping, and line clipping: Sutherland - Cohen line clipping algorithm. (4)

UNIT 4: Curves and Surfaces
Curve Representation, Non-parametric and parametric curves, representation of space curves, Cubic Spline, Parabolic Blended curves, Bezier curves and B-spline curves, Z- buffer, Warnock algorithm. (7)

UNIT 5: Computer Animation
Introduction, Key frame animation, Construction of an animation sequence, Motion control methods, Procedural animation, Key-frame animation vs. Procedural animation, Introduction to Morphing, Wrapping techniques, Three dimensional morphing. (5)
UNIT 6: Illumination models and surface rendering methods.
Light sources, Basic illumination models, Displaying light intensities, Halftone patterns and Dithering Techniques, Polygon Rendering methods, Ray tracing methods, Radiosity lighting model. (6)

Text Books:

1. Mathematical elements for Computer Graphics - David F. Rogers, J. Alan Adams (MGH Int.) (For Units 1, 4)
2. Procedural elements for Computer Graphics - David F. Rogers (MGH International) (For Units 2, 3)
3. Computer Graphics- Rajesh Maurya (WILEY India) (For Unit 5)

References Books:


Term Work:

It should consist of minimum of 10-12 experiments based on the following topics and must be performed using Open-GL.

1. Introduction to computer graphics, OPEN GL, GLUT, GLU.
2. Getting started (Installation of VISUAL STUDIO, Library files of OPEN GL, GLUT).
3. Initial steps in drawing figures (polygon, rectangle etc).
5. Transformations (Scaling, Translation).
7. Lighting objects and displaying a 3-d object.
8. Three Dimensional Objects.
10. Implementation of Filling algorithms.
2. SYSTEM PROGRAMMING

Lectures: 3 hrs/week                Theory: 100 marks
Practicals: 2 hrs/week              Termwork : 50 marks
Orals: 25 marks.

Course Objectives:

1. To expose the students to the fundamentals of languages and processing.
2. To make students to learn design of grammars, assemblers and compilers.
3. To provide hands on experience to the students on simulation of linkers, loaders and software tools for UIs and DLLs.


Unit 2: Assemblers: Elements of assembly language programming, a simple assembly scheme, pass structure of assemblers, design of a two pass assembler. (5)

Unit 3: Macros and Macro Processors: Macro definition and call, Macro expansion, Nested macro calls, Advanced macro facilities, Design of macro preprocessor. (8)

Unit 4: Compilers and Interpreters: Aspects of compilation, memory allocation, compilation of expressions, compilation of control structures, Interpreters. (7)

Unit 5: Linkers: Relocation and linking concepts, design of a linker, Self- relocating programs, linking for overlays, Loaders. (6)

Unit 6: Software tools: Software tools for User interface, software tools for DLLs. (3)

Text books:

1. System Programming and operating systems – 2nd Edition D.M. Dhamdhere (TMGH)

Reference book:


Term Work: Minimum of 10-12 practical assignments should be carried based on following list.
1. First five experiments based on any lex specification
2. Design and implementation of 1 pass assemblers.
3. Design and implementation of 2 pass assemblers.
4. Simulation of linkers and loaders.
5. Using software tools for UIs and DLLs.

3. **OBJECT ORIENTED MODELING AND DESIGN**

**Lectures**: 3 hrs/week  
**Theory**: 100 Marks

**Course Objectives:**

1. To explain how a software design may be represented as a set of interacting objects that manage their own state and operations.
2. To describe the activities in the object-oriented design process.
3. To introduce various models that can be used to describe an object-oriented design.
4. To show how the UML may be used to represent these models.
5. To implement design patterns to provide solutions to real world software design problems.
6. To learn to design flexible and reusable software components.

**Unit 1: Introduction:**
Object oriented themes, modeling as a design technique.  

**Object Modeling:**
Object, classes, Link & association, advanced link & Association concepts, generalization & Inheritance, grouping constructs, aggregation, abstract classes, generalization as extension & restriction, multiple inheritance, metadata, candidate key & constraints.

**Unit 2: Dynamic & Functional Modeling:**
Dynamic modeling: Events & states, operations, nested state diagrams, concurrency, advanced dynamic modeling concepts & simple dynamic model, relation of object dynamic models.  

**Unit 3: Design Methodology:**
OMT methodology, Impact of an object oriented approach, analysis, system design with examples, combining models, design algorithms, design optimization, implementation of controls, design association & physical packaging.

**Unit 4: Introducing the UML:**
An overview of the UML, Conceptual Model of UML, Architecture of UML.
Structure modeling Using UML: (3)
Classes, Relationship, Diagrams, Class Diagrams.

Unit 5: Behavioral Modeling: (8)
Interactions, Use Cases, Use Case Diagram, Interaction diagrams, Activity diagrams, Events & Signals, State Machines, Process & Threads, Time & Space, State chart diagrams.

Unit 6: Architectural Modeling: (8)
Components, Deployment, Collaboration, Patterns & frameworks, component diagrams, Deployment diagrams.

Text Books:

1. Object-orientated Modeling & Design: (Unit 1 to 3) - James Rambaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. (PHI)

Reference Books:

2. Object Oriented analysis & Design – Andrew High(TM)
3. Practical Object Oriented Design with UML – Mark Priestley.
4. Object Oriented Analysis & design – Kahate (TMH)
5. Threat first Object oriented analysis & design - Breet McLaughline, Garry Police & Devide West. (OREILLY)

4. COMPUTER ALGORITHMS

Lectures: 4 hrs/week Theory: 100 marks
Tutorials: 1 hr/week Term work: 25 marks

Course Objectives:

1. To introduce to the students the methods of algorithm designs.
2. To expose students to various searching and sorting techniques.
3. To make students understand the analyses of algorithms.
4. To show how to tackle real time problems.

Unit 1: Divide and Conquer (10)
What is algorithm, Algorithm Specification, Recurrence relations, Performance Analysis,
Randomized Algorithms.
Divide and Conquer-The general method, Binary search, finding the maximum and minimum, Merge sort, Quick sort, Selection sort and analysis of these algorithms.

**Unit 2 : The Greedy method** (6)
The general method, Knapsack problem, Job sequencing with deadlines, minimum-cost spanning trees – Prim’s and Kruskal’s Algorithms, Optimal storage on tapes, Optimal merge patterns, Single source shortest paths.

**Unit 3 : Dynamic Programming** (7)
The general method, Multistage graphs, All pair shortest paths, Optimal binary search trees, 0/1 knapsack, Reliability design, Traveling Sales person problem.

**Unit 4 : Basic Traversal and Search Techniques and Backtracking** (13)
Techniques for Binary Trees, Game Tree; Techniques for Graphs – Breadth First Search & Traversal, Depth First Search & Traversal, AND/OR graphs; Connected components and Spanning Trees; Bi-connected components and depth first search. Backtracking - The general method, 8-queen problem, sum of subsets, Knapsack Problem, Hamiltonian Cycle, and Graph Coloring.

**Unit 5 : NP Hard and NP Complete Problems** (3)
Basic Concepts, Introduction to NP Hard Graph Problems.

**Unit 6 : Introduction to Parallel Algorithm** (6)
Computational Model and Fundamental Techniques and Algorithms – PRAM, MESH and HYPERCUBE.

**Text Book:**


**Reference Books:**


**Term work:** It should consist of 10-12 assignments based on the following guidelines –

1. A batch of students will be assigned different algorithms and expected to analyze the algorithms in terms of time and space complexity.
1. Solve different exercise problems in the text book mentioned in the syllabus.
2. Solve more numerical problems for Greedy and Dynamic Programming methods.

5. NETWORK TECHNOLOGIES

Lectures: 4 hrs/week  
Theory: 100 marks

Course Objectives:

1. To introduce students to the cellular technologies.
2. To expose students to the design issues and standards of wireless networks.
3. To make students understand wireless protocols and security services.

Unit 1:
Introduction: Different generations of wireless cellular Networks, 1G to 4G Cellular systems and beyond, GSM system overview, Introduction to GSM, GSM Network and system Architecture, GSM Channel Concept, GSM Identities, GSM system operations. (Traffic cases).

Unit 2:
Wireless LANs (IEEE 802.11x): Introduction to IEEE 802.11X technologies, Evolution of wireless LANs, IEEE 802.11 Design issues, IEEE 802.11 Services Overview, IEEE 802.11 MAC layer operations, IEEE 802.11 a/b/g standards, IEEE 802.11- Wireless LAN security, Competing wireless Technologies

Unit 3:

Unit 4:
Wireless protocol: medium access control protocol, routing protocol, transfer control protocol

Unit 5:

Unit 6:

Text Books:

2. Wireless Networks By Georgios I. Papadimitriou, Andreas S. Pomportsis, P. Nicopolitidis

References Books:

2. 802.11 Wireless Networks - Mathew S Gast (2nd edition), Publication – SPD O’REILLY.

6. Programming Laboratory –III

Lectures: 3 hrs / Week Termwork: 50 Marks
Practical: 4 hrs / Week POE: 50 Marks

Course Objectives: To expose students to

1. Fundamental and object oriented concepts of Java.
2. Application of Interface, inheritance and packaging in Java.
3. Writing code with Exception handing and I/O programming features.
4. Architecture and components of GUI development in Java.
5. Fundamental concept of multithreading and Network Programming in Java.
6. Collection and database programming in Java.

Unit 1: Fundamental Programming in Java (6)
Objects and Classes: Object-Oriented Programming Concepts, Declaring Classes, Declaring Member Variables, Defining Methods, Constructor, Passing Information to a Method or a Constructor, Creating and using objects, Controlling Access to Class Members, Static Fields and Methods, this keyword, Object Cloning, Class Design Hints.

Unit 2: Interface, Inheritance and Packaging (6)
Interfaces: Defining an Interface, Implementing an Interface, Using an Interface as a Type, Evolving Interfaces, Default Methods.
Inheritance: Definition, Superclasses, and Subclasses, Overriding and Hiding Methods, Polymorphism, Inheritance Hierarchies, Super keyword, Final Classes and Methods, Abstract
Classes and Methods, casting, Design Hints for Inheritance, Nested classes & Inner Classes, finalization and garbage collection.

Packages: Class importing, Creating a Package, Naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File.

**Unit 3: Exception and I/O Streams** (6)

**Unit 4: Graphical User Interfaces using Swing:** (8)
Layout Management: Introduction to Layout Management, APIs for Border Layout, Flow Layout, Grid Layout

Event Handling: Basics of Event Handling, The AWT Event Hierarchy, Semantic and Low-Level Events in the AWT, Low-Level Event Types

User Interface Components: Text Input, Choice Components, Menus, Dialog Boxes

Setting the Look and Feel of UI, Introduction to JApplet

**Unit 5: Networking and Multithreading** (5)
Networking: Overview of Networking, Networking Basics, Working with URLs, Creating a URL, Parsing a URL, Reading Directly from a URL, Connecting to a URL, Reading from and Writing to a URL Connection, Sockets, Reading from and Writing to a Socket, Writing the Server Side of a Socket, Datagrams, Writing a Datagram Client and Server.
Multithreading: Processes and Threads, Runnable Interface and Thread Class, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Thread States, Thread Properties, Joins, Synchronization

**Unit 6: Collection and Database Programming** (5)
Collections: Collection Interfaces, Concrete Collections- List, Queue, Set, Map, the Collections Framework.

Text Books:

1. Core Java- Volume I Fundamentals: Cay Horstmann and Gary Cornell, Pearson, Eight edition (Unit 1 to Unit 4).
2. Core Java- Volume II Advanced Features: Cay Horstmann and Gary Cornell, Pearson, Eight edition (Unit 5 and Unit 6).

Reference Books:

1. The Java Tutorials From ORACLE Java Documentation URL: http://docs.oracle.com/javase/tutorial/ (Refer For All Units)
3. JAVA™ HOW TO PROGRAM, By Deitel Paul, Deitel Harvey. 10th Edition, Publisher: PHI Learning.

Term Work:

Guidelines for Term work marks distribution:

1. 25 marks for performance in practical and experiments
2. 25 marks for Two Objective Tests each of 25 marks.

Guidelines for conducting practical:

Minimum 15 experiments should be conducted based on above topics and covering following list. At least two experiments should be conducted on each unit in the syllabus.

1. Create a class called Employee that includes three pieces of information as instance variables- first name, a last name and a monthly salary. Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test
application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

2. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12; this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value.

Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs 2000.00 and Rs 3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

3. Create Vehicle Interface with name, maxPassanger, and maxSpeed variables. Create LandVehicle and SeaVehicle Interface from Vehicle interface. LandVehicle has numWheels variable and drive method. SeaVehicle has displacement variable and launch method. Create Car class from LandVehicle, HoverCraft from LandVehicle and SeaVehicle interface. Also create Ship from SeaVehicle. Provide additional methods in HoverCraft as enterLand and enterSea. Similarly provide other methods for class Car and Ship. Demonstrate all classes in a application.

4. Create Separate Engine, Tyre, and Door Class. Create a Car class using these classes. And show functionality of each component in the car.

5. Develop a mathematical package for Statistical operations like Mean, Median, Average, Standard deviation. Create a sub package in the math package - convert. In “convert” package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package, and build executable jar file of it.

6. Develop a class Expr to create and evaluate given expression. Constructor accepts the expression as String. For example, Expr("x^2") or Expr("sin(x)+3*x"). If the parameter in the constructor call does not represent a legal expression, then the constructor throws an IllegalArgumentException. The message in the exception describes the error. Provide eval(double num) and eval(int num) method to evaluate given expression and return evaluated answer. For example, if Expr represents the expression 3*x+1, then func.value(5) is 3*5+1, or 16. Finally, getDefinition() returns the definition of the expression. This is just the string that was used in the constructor that created the expression object.

7. Write a class to represent Roman numerals. The class should have two constructors. One constructs a Roman numeral from a string such as "XVII" or "MCMXCIV". It should throw a NumberFormatException if the string is not a legal Roman numeral. The other
constructor constructs a Roman numeral from an int. It should throw a
NumberFormatException if the int is outside the range 1 to 3999. In addition, the class
should have two instance methods. The method toString() returns the string that
represents the Roman numeral. The method toInt() returns the value of the Roman
numeral as an int.

8. Take file name as input to your program. If file is existing the open and display contents
of the file. After displaying contents of file ask user – do you want to add the data at the
end of file. If a user gives yes as response, then accept data from user and append it to
file. If file in not existing then create a fresh new file and store user data into it. User
should type exit on new line to stop the program.

9. Take Student information such as name, age, weight, height, city, phone from user and
store it in the file using DataOutputStream and FileOutputStream and Retrive data using
DataInputStream and FileInputStream and display the result.

10. Write a program to remove whitespaces from a text file. Name of the file is given using
command line.

11. Develop a Swing GUI based standard calculator program.

12. Develop a GUI based application to create Telephone Contacts directory. Store the data
in standard “vcard” format. Also read any standard “vcard” file and display contacts in it.

13. Write a program that bounces a blue ball inside a JPanel. The ball should begin moving
with a mousePressed event. When the ball hits the edge of the JPanel, it should bounce
off the edge and continue in the opposite direction. The ball should be updated using a
Runnable.

14. Create Stop Watch with Swing GUI and Multithreading. Provide Facility for Lap
Counting.

15. Write a Swing GUI based network server program. The program is a simple file server
that makes a collection of files available for transmission to clients. When the server
starts up, it needs to know the name of the directory that contains the collection of files.
Specify this directory name through JFileChooser Dialog. You can assume that the
directory contains only regular files (that is, it does not contain any sub-directories).

When a client connects to the server, the server first reads a one-line command
from the client. The command can be the string "index". In this case, the server responds
by sending a list of names of all the files that are available on the server. Or the command
can be of the form "get <file>", where <file> is a file name. The server checks whether
the requested file actually exists. If so, it first sends the word "ok" as a message to the
client. Then it sends the contents of the file and closes the connection. Otherwise, it sends
the word "error" to the client and closes the connection.
16. Write a GUI based program to create a student registration and Login. Store Registration data in Database and take Login information from Database.
17. Write a GUI based program to store and retrieve, delete and update Student’s information in Database.
18. Fill a HashMap with key-value pairs. Print the results to show ordering by hash code. Extract the pairs, sort by key, and place the result into a LinkedHashMap. Show that the insertion order is maintained.
19. Write a program to read a text file one line at a time. Read each line as a String and place that String object into a LinkedList. Print all of the lines in the LinkedList in reverse order.

7. Business English

Tutorial: 1 hr/week Termwork: 25 Marks
Orals: 25 Marks.

Course Objectives:

1. To improve professional communication skills of the students.
2. To acquire communicative competencies crucial for appropriate workplace behavior.

Unit 1: Getting acquainted with professional culture: First day at work, Induction programme, knowing company hierarchy, 10 things a manager must do on the first day, behavior pruning.

Unit 2: Vocabulary Building and Reading Comprehension: Vocabulary Building –synonyms and antonyms, word roots, one-word substitutes, prefixes and suffixes, study of word origin, analogy, idioms and phrases, Situational Vocabulary.

Reading Comprehension – reading for facts, guessing meanings from context, scanning, skimming and critical reading.

Unit 3: Effective Vocal Communication: Effective telephonic communication skills, Effective Meetings, Breaking Bad news, Video conferencing.

Unit4: Effective Written communication: Email Writing, Business Report writing, Memo & its answering, taking minutes of meeting.

Unit5: Public Speaking and Presentation Skills: Overcoming stage fear, Body language, Best Practices.

Unit 6: Miscellaneous: Issues Escalation- Handling complaints, Practice of Right to Information (RTI), Business Etiquette, Negotiations.
References:

1. Business English- T. Samson (TMGH WE Series) (Units – 1, 3, 5, 6)
2. Technical English- Dr. M. Hemamalini (Wiley Publications) (Unit -4)
3. Communication Sills – Sanjay Kumar and Pushp Lata (Oxford Higher Education) (Unit-2)
4. English Vocabulary in Use series, Cambridge University Press 2008. (Unit -2)

Tutorials: The students are expected to go through the syllabus units and practice accordingly during their tutorial sessions. The faculty member/s dealing with this subject workload and soft-skills subject workload (SE) should undergo training preferably from an IT industry and ensure that all the tutorial sessions are conducted effectively. Expert talks / sessions should be conducted from Industry personnel or professionals. Evaluation of the tutorial work should be done on continuous basis and the record of students’ progress should be maintained.
T.E. (Computer Science and Engineering) Semester – VI

1. COMPILER CONSTRUCTION
Lectures: 3 hrs/week                         Theory: 100 marks
Practicals: 2 hrs/week         Term work: 25 marks

Course Objectives:
1. To introduce the fundamentals of compilers and their phases.
2. To design and implement phases of a compiler.
3. To expose the students to various tools like Lex and Yacc.

UNIT 1- Introduction to Compiling:
Compilers, Phases of a compiler, Compiler construction tools, cousins of the compiler (6)

UNIT 2- Lexical Analysis:
Role of a Lexical analyzer, input buffering, specification and recognition of tokens, finite automata implications, designing a lexical analyzer generator. (5)

UNIT 3- Syntax Analysis:
Role of Parser, Writing grammars for context free environments, Top-down parsing, Recursive descent and predictive parsers (LL), Bottom-Up parsing, Operator precedence parsing, LR, SLR and LALR parsers. (7)

UNIT 4- Syntax Directed Translation and Intermediate Code Generation:
Syntax directed definitions, construction of syntax tree, S-attributed definitions, L-attributed definitions, Intermediate languages, assignment statements, back patching, procedure calls (7)

UNIT 5- Code Optimization:
Sources of optimization, Peephole optimization and basic blocks, loops in flow graphs, Data flow analysis and equations, code improving transformation and aliases (5)

UNIT 6- Code Generation:
Issues in design of a code generator and target machine, Run time storage management, Basic blocks and flow graphs, Next use information and simple code generator, Issues of register allocation, code generation from Dags. (6)

Text Book:
Reference Books:

1. Crafting A Compiler with C - Charles Fischer, Richard LeBlanc (Pearson publication)  
   (For practical use only)
   publication) (For practical use only).
3. Modern Compiler Implementation in Java - Andrew W. Appel (Cambridge University  
5. Unix / Linux manuals.

Term work:
It should consist of minimum 10-12 experiments based on the above topics covering  
the following list of assignments.

1. Design of preprocessor for C program  
2. Design a complete lexical analyzer for C language  
3. Program to create a symbol table generator  
4. Using recursive descent parsing method, design a syntax analyzer for Simple expression  
   in C language.
5. Program to create a syntax tree for simple expression in C language using Recursive  
   descent parsing techniques.
6. Implement intermediate code generator for the Boolean expression in three Address code  
   format.
7. Implement intermediate code generator for the conditional statements in three Address  
   code format.
8. Program to implement bottom up parsing removing shift reduce conflict.
9. Write a program to implement code generator from a labeled tree.
10. Demonstration of compiler and interpreter using Lex and Yacc.

2. Operating System –II

Lectures: 4 Hrs / Week  
Practical: 2 Hrs / Week
Theory: 100 Marks  
Term Work: 25 Marks

Course Objectives: To expose students to

1. Fundamental architecture of UNIX operating system kernel.  
2. Detail algorithms of buffer cache management.
3. Internal File system organizations and related algorithms in UNIX.
4. System calls for UNIX file system.
5. Process structure, creation and management in UNIX.
6. Architecture and algorithms of process scheduling and memory management.
7. I/O subsystem architecture and algorithms.

Unit 1: Introduction and buffer cache: (10)


Buffer Cache: - Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, advantages and disadvantages of cache.

Unit 2: Internal Representation of Files (8)

i-nodes, structure of the regular file, directories, conversion of a pathname to i-node, super block, i-node assignment to a new file, allocation of disk blocks, other file types.

Unit 3: System Calls for file system: (6)


Unit 4: The Structure of process: (8)

Process stages and transitions, layout of system memory, the context of a process, Saving context of a process, manipulation of the process address space.

Unit 5: Process Control and Scheduling: (8)

Process Control: - Process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, the shell, System Boot and the Init process.

Process Scheduling: - Process Scheduling, system call for time, clock.

Unit 6: Memory management and I/O Subsystem: (8)

Swapping, Demand passing, a hybrid system with demand paging and swapping. Driver interfaces, disk drives, terminal drivers, Streams.
Text Book:
1. The design of Unix Operating System - Maurice J. Bach (PHI)

Reference Books:
1. Linux System Programming - Robert Love, Publisher - SPD, O’ REILLY

Term Work:
It should consist of minimum 10-12 experiments based on the above topics and covering the following list of assignments. (Reference book – Linux System Programming by Robert Love may be referred for the assignments listed below.)

1. Fundamentals of Linux system programming and programmers overview of the Linux System (Refer Chapter No 01: Introduction and Essential Concepts)
2. Study & demonstration of how the Linux Kernel implements and Manages files. Ref Chapter No 02 : File I/O.
3. Study & demonstration of User Buffer I/O - Observe practically by writing ‘C’ program. (Refer Chapter No 03: Buffer I/O).
5. Study and demonstration of Unix Process Management – from process creation to process termination (Refer Chapter No 05: Process Management).
7. Study and demonstration of Memory Management (Refer Chapter No 08: Memory Management).
8. Study and Demonstration of Signals (Refer Chapter No 09: Signals).
9. Study and Demonstration of Time, Sleep and Clock Management (Refer Chapter No 10: Time)
10. Study of boot loader like “Grub”
11. Study of compilation of Linux kernel.
12. Implementation of system call for UNIX/Linux.
13. Implement shell for UNIX/Linux operating system.
3. Database Engineering

Lectures- 4/week Theory- 100 Marks
Practical- 2/Week Term work -25 Marks
POE – 50 Marks

Course Objectives:

1. To understand Fundamental Concepts and algorithms related to database.
2. To gain familiarity with SQL & DBMS.
3. To understand basic concepts of Database Design

Unit 1: Introduction to databases [Text Book- 1 & 3 ] [8]

1.1 Introduction
1.2 Traditional File based Systems
1.3 Database Approach
1.4 Roles in Database Environment
1.5 History of Database management systems
1.6 Advantages and Disadvantages of DBMS’s.
1.7 Structure of Relational Databases
1.8 Database Schema
1.9 Keys
1.10 Schema Diagram
1.11 Relational Query Languages.
1.12 Relational Operations

Unit 2: Structured Query Language (SQL) [Text Book -2 & 3] [8]

2.1 Introduction to SQL
2.2 Data Definition Commands
2.3 Data manipulation Commands
2.4 Queries
2.5 Advanced data management commands
2.6 More complex queries and SQL functions

Unit 3: Normalization [Text Book – 1] [6]

3.1 The purposes of Normalization
3.2 Data Redundancies and Update Anomalies
3.3 Functional Dependencies
3.4 The Process of Normalization
3.5 First Normal Form
3.6 Second Normal Form
3.7 Third Normal Form
3.8 Boyce-Codd Normal Form
3.9 Fourth Normal Form
3.10 Fifth Normal Form

Unit 4: Data Storage & Indexing [Text Book -3] [7]
4.1 File Organization
4.2 Organization of records in File
4.3 Data Dictionary Storage
4.4 Database Buffer
4.5 Basic Concepts indexing & hashing
4.6 Ordered Indices
4.7 Multiple-Key Access
4.9 Static Hashing
4.10 Dynamic Hashing
4.11 Bitmap Indices
4.12 Index Definition in SQL

Unit 5: Transaction Management & Concurrency Control [Text Book – 2 & 3] [10]
5.1 What is a Transaction?
5.2 Concurrency Control
5.3 Concurrency Control with Locking Methods
5.4 Concurrency Control with Times tamping Methods
5.5 Concurrency Control with Optimistic Methods

Unit 6: Recovery System [Text Book-3] [6]
6.1 Failure Classification
6.2 Storage
6.3 Recovery & atomicity
6.4 Recovery Algorithm
6.5 Buffer Management
6.6 Failure with loss of non- volatile Storage

Text Books:


Termwork: Minimum 10 -12 Assignments based on the following topics.
1) Draw an E-R Diagram for any organization like Insurance Company, Library systems, College Management systems, Hospital Management systems etc.
2) Convert the Above mentioned E-R Diagram in Relational Tables
3) Installation & Demonstration of DBMS like MySQL, Oracle, IBM-DB2 etc., Draw the architectures of installed DBMS.
4) Write a program of Database connectivity with any object oriented language.
5) Use DDL Queries to create, alter & drop Tables.
6) Use DML Queries to insert, delete, update & display records of the tables.
7) Create tables with using primary key & foreign key with all constraints.
8) Display the records using group by, order by, having and between clauses.
9) Display the records using Aggregate functions
10) Create Indexes & Views for the table.
11) Display the results of union, intersection, set difference, Cartesian product and Join operations of two different tables.
12) Write a program to implement Static Hashing.
13) Write a program to implement to Dense Index.
14) View the contents of data dictionary from the DBMS and write the contents.
15) Find the FC and F+ of relation schema r(A,B,C,G,H,I) and F= {A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H}.

4. STORAGE NETWORKS

Lectures: 3 Hrs / Week Theory: 100 Marks

Course Objectives: To expose students to

1. Finding key challenges in information management
2. Storage system architecture and data protection.
3. Storage Area Network- concepts, components and protocols.
5. Architecture of Storage Virtualization.

Unit 1: Introduction to information storage: (7)

Evolution of storage technology and architecture, Data Center Infrastructure, Key challenges in Managing Information, Information Lifecycle. Components of Storage System Environment, Disk Drive Components, Disk Drive Performance, Laws governing disk Performance, Logical Components of Host, Application requirements and disk performance.

Data Protection (RAID):- Implementation of RAID, RAID array components, RAID levels, Comparison, RAID ,Impact on disk performance, Hot Spares.

**Unit 2: Storage Area Network:**

SAN – Evolution, Components of SAN, Fibre Channel Protocol Stack- Links, ports and topologies, FC-0: Cables, plugs and Signal Encoding, FC-1: 8b/10b encoding, ordered sets and link control protocol, FC-2: data Transfer, FC-3: common Services, FC-4 and ULPs, Fibre Channel SAN – point-to- point topology, Fabric topology, Arbitrated loop topology, Hardware components of Fibre channel SAN. IP SAN – iSCSI – components, connectivity, topology, protocol stack, discovery, names, session, PDU

**Unit 3: Network -Attached Storage:**


Case Study: Direct Access File System, Shared Disk File System

Comparison: NAS, Fibre Channel SAN and iSCSI SAN

**Unit 4: Storage Virtualization:**

Introduction, Virtualization in the I/O path, Limitations and requirements, Definition of Storage Virtualization, Implementation considerations, Storage Virtualization on block, level, File level Virtualization, Storage Virtualization on various levels of the storage, network, Symmetric and Asymmetric Storage Virtualization.

**Unit 5: Business Continuity, Backup and Recovery:**

Introduction, Information Availability, Cause of Information unavailability, Measuring information Availability, Consequences of down time, BC terminology, BC planning life cycle, Failure Analysis, BC Technology Solutions, Backup Purpose, Backup, Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup, Process, Backup and Restore Operations, Backup Topology, Backup in NAS environment, Backup Technologies,
Unit 6: Replication and Storage Security: (6)

Local Replication, Uses of Local Replicas, Data Consistency, Local Replication Technologies, Restore and Restart Considerations.


Text Books:

2. Storage Networks Explained by Ulf Troppen, Rainer Erkens, Wolfgang Müller (Wiley India Edition).

5. INFORMATION SECURITY

Lectures : 3 Hrs/week
Tutorials: 1 hr/week
Theory : 100 Marks
Termwork : 25 Marks

Course Objectives:

1. To introduce Information security services and mechanisms to the students.
2. To make students feel the security services widely used in Internet and Web services.
3. To give hands on exposure to various security tools and security related issues.
4. To practice ethics in using and developing security softwares.


UNIT III. Key Management and Authentication: Key Management; Other Public-Key Cryptosystems- Key Management, Diffie-Hellman Key Exchange, Message Authentication and
HASH Functions- Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions.  (5)


Text Book:

Reference Books:
2. Cryptography and security – Shyalama (Wiley India).

Term work: It should consist of 10-12 assignments based on exercise problems given in the text book and should include study of the following.

1. To study the Viruses, Threads and Advanced Block Cipher Encryption Techniques.
2. To study and analysis of security tools like OpenPuff security Tool, CloudSecurity Readiness Tool, Kismet, John the Ripper.
6. PROGRAMMING LABORATORY – IV

Lectures: 2 hrs/week  
Practicals: 2 hrs/week  
Term work: 25 marks  
POE: 50 marks

Objectives:

1. To make the student familiar with basic .Net framework.
2. To make student understand the OO features and their implementations.

1. .NET Architecture (3)

2. C# Basics (4)
   Variables, Predefined Data Types, Flow Control, Enumerations, Arrays, Namespaces, The Main () Method, More on Compiling C# Files, Console I/O, Using Comments# Programming Guidelines, Dynamic variables, DLL creation & calling.

3. Objects and Types (3)
   Classes and Structs, Class Members, Anonymous Types, Structs, Partial Classes, Static Classes, The Object Class, Extension Methods

4. Inheritance (2)
   Types of Inheritance, Implementation Inheritance, Modifiers, Interfaces

5. Arrays (2)
   Simple Arrays, Multidimensional Arrays, Jagged Arrays, Array Class, Array and Collection Interfaces, Enumerations

6. Operators and Casts (2)
   Operators, Type Safety, Comparing Objects for Equality, Operator Overloading, User-Defined casts

7. Windows Form & Database with ADO.NET (4)
   Introduction to GUI application & components –add data control programmatically, Link data to control, process all control, track the visible forms, Find all MDI child forms, Save configuration setting for form, Force list box to scroll items, Restrict text box, Use of auto complete combo box, Sort a list view, Database with ADO.NET-Overview of Ado.NET, Data components in Visual Studio .NET.

8. Strings (1)
   System. String, Building Strings, String Builder Members, Format Strings, Regular Expressions
9. Threading  (3)
Overview, Asynchronous Delegates, the Thread Class and Thread Pools, Threading Issues, Synchronization, Timers

10. Networking  (2)
Networking-Obtain information about Local network, Detect changes in network, Download data over HTTP or FTP, Download a File & Process using Stream, Respond to HTTP request from your application.

Text books:

1. Professional C# 2012 & .Net 4.5 (For Unit 1 to 6 & Unit 8 to 9) - Christian Nagel, Bill Evjen, Jay Glynn, Morgan Skinner, Karli Watson, Wrox Publication
3. Visual C# 2010 Recipes- A Problem-Solution Approach (For Unit 7 & Unit 10) - By Allen Jones, Adam Freeman, Matthew MacDonald, Rakesh Rajan, Apress Publication.

Term work: It should consist of 10 to 12 experiments based on the above syllabus covering following list of assignments.

(Note: 60% of the experiment should be console based & 40 % experiment should be windows form application.)

1. Language Introduction (Includes console based application, creation of dll, running a program without IDE) calling a method from another program.
2. OOPS concepts in C#-Class, Implementation Inheritance, Extension methods (Use Any application).
3. Develop DLL file and use it in application program. (Use Any application)
5. Implementation of Multidimensional & Jagged array (Use Any application).
6. Use of properties in any application.
8. String manipulation using String & String builder(Any application)
10. Design a Windows Form based application for different controls.(Any application)
11. Design a Windows Form based MDI application with different controls.(Any application)
12. Design a Windows Form based application for field validation.(Any application)
13. Design a any Windows Form based application with Database connectivity with all field validation .(Any application)

14. Develop a Windows Form application that performs SELECT, INSERT, UPDATE & DELETE queries and also displays the List of Books available in a Library System by fetching the details from a database. The C# application must also contain the filter capability.

15. Implement console based networking application to obtain information of network & detect changes in network.


7. **Domain Specific Mini-Project**

    **Practicals: 2 hrs/week**

    **Termwork : 25 Marks**

    **Oral : 50 Marks**

**Course Objectives:**

1. To expose the students to use engineering approach to solve domain specific real time problem.

2. To use the appropriate and newer technologies while developing the project.

3. To learn the skills of team building and team work.

The students should form group of 5 students each and every group is supposed to choose a specific domain in which they would like to carry on their Sem-VII and VIII project work. Further the group should identify the relevant problem and propose the solution, which can be implemented as a mini-project using suitable technology. The domain specific mini-project work should be evaluated by a team of teachers appointed by the department. The evaluation should be done in the mid and end of the semester during which the group should give presentation and demonstration of their work done. **Care should be taken to avoid out-sourcing of the work.** The termwork assessment is to be done as follows.

1. Mid term assessment – 5 marks.

2. End term assessment – 5 marks.

3. Final performance evaluation to be done by guide – 15 marks.
Equivalent subjects at T.E. (CSE) Sem-V & Sem –VI of Pre-revised course to the revised course of T.E. (CSE) Sem-V & Sem-VI

TE (CSE) Sem.-V

<table>
<thead>
<tr>
<th>Sr.no.</th>
<th>TE (CSE) –I (Pre-Revised)</th>
<th>Equivalent / Replacement subject (Revised)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>System Programming</td>
<td>System Programming of TE (CSE) Sem - V</td>
</tr>
<tr>
<td>3</td>
<td>Operating Systems - I</td>
<td>Operating Systems – I of SE (CSE) Sem - IV</td>
</tr>
<tr>
<td>4</td>
<td>Computer Algorithms</td>
<td>Computer Algorithm of TE (CSE) Sem – V</td>
</tr>
<tr>
<td>5</td>
<td>Network Technologies</td>
<td>Network Technologies of TE (CSE) Sem-V</td>
</tr>
<tr>
<td>6</td>
<td>Programming Lab-III</td>
<td>Programming Lab-III of TE (CSE) Sem - V</td>
</tr>
<tr>
<td>7</td>
<td>Mini-Project-II</td>
<td>Domain Specific Mini-Project of TE(CSE) Sem-V</td>
</tr>
</tbody>
</table>

T.E. (CSE) Sem.-VI

<table>
<thead>
<tr>
<th>Sr.no.</th>
<th>TE (CSE) II (Pre-Revised)</th>
<th>Equivalent / Replacement subject (Revised)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compiler Construction</td>
<td>Compiler Construction of TE (CSE) Sem - VI</td>
</tr>
<tr>
<td>2</td>
<td>Operating Systems - II</td>
<td>Operating Systems – II of TE (CSE) Sem - VI</td>
</tr>
<tr>
<td>3</td>
<td>Database Engineering</td>
<td>Database Engineering of TE (CSE) Sem - VI</td>
</tr>
<tr>
<td>4</td>
<td>Object Oriented Modeling &amp; Design</td>
<td>Object Oriented Modeling &amp; Design of TE (CSE) Sem - V</td>
</tr>
<tr>
<td>5</td>
<td>Information Security</td>
<td>Information Security of TE (CSE) Sem-V</td>
</tr>
<tr>
<td>6</td>
<td>Programming Lab-IV</td>
<td>Programming Lab-IV of TE (CSE) Sem - VI</td>
</tr>
<tr>
<td>7</td>
<td>Soft Skills</td>
<td>Soft Skills of SE (CSE) Sem - III</td>
</tr>
</tbody>
</table>