**CHOICE BASED CREDIT SYSTEM (CBCS)**
**CBCS Pattern**
**Proposed Syllabus Outline of B.Sc.C.S. Second Year**
**Effective from Academic Year (2018-2019)**

| Semester | Course Code | Course Name | Credits / Marks | | | | Total | |
|---|---|---|---|---|---|---|---|---|
| | | | Internal | | External | | | |
| | | | Credit | Marks | Credit | Marks | Credit | Marks |
| Semester - III | **1. Ability Enhancement Courses (AEC) (Compulsory Course)** | | | | | | | |
| | **BScCS-AEC-3** | English- I | -- | 20 | -- | 30 | 2 | 50 |
| | **2. Core Course (CC) (Compulsory Course)** | | | | | | | |
| | **BScCS -CC-A3** | Object Oriented Programming Structure | -- | 20 | -- | 30 | 3 | 50 |
| | **BScCS -CC-B3** | Data Structure | -- | 20 | -- | 30 | 3 | 50 |
| | **BScCS -CC-C3** | Web Programming | -- | 20 | -- | 30 | 3 | 50 |
| | **BScCS -CC-D3** | Operating System | -- | 20 | -- | 30 | 3 | 50 |
| | **3. Skill Enhancement Course** | | | | | | | |
| | **BScCS -SEC-1** | Android O.S. | | 20 | -- | 30 | 2 | 50 |
| | **4. Practical / Lab Course** | | | | | | | |
| | **BScCS -Lab-9** | Lab-Course 1 (C++) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-10** | Lab-Course 2 (DS) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-11** | Lab-Course 3 (CSS, Jscript and jQuery) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-12** | Lab-Course 4 (Operating System) | -- | 20 | -- | 30 | 2 | 50 |
| | | **Total (III)** | | | | | **24** | **500** |

| Semester | Course Code | Course Name | Credits / Marks | | | | Total | |
|---|---|---|---|---|---|---|---|---|
| | | | Internal | | External | | | |
| | | | Credit | Marks | Credit | Marks | Credit | Marks |
| Semester - IV | **1. Ability Enhancement Courses (AEC) (Compulsory Course)** | | | | | | | |
| | **BScCS -AEC-4** | English- II | -- | 20 | -- | 30 | 2 | 50 |
| | **2. Core Course (CC) (Compulsory Course)** | | | | | | | |
| | **BScCS -CC-A4** | Software Engineering. | -- | 20 | -- | 30 | 3 | 50 |
| | **BScCS -CC-B4** | Java Programming | -- | 20 | -- | 30 | 3 | 50 |
| | **BScCS -CC-C4** | Programming with C#.Net | -- | 20 | -- | 20 | 3 | 50 |
| | **BScCS -CC-D4** | Multimedia Systems. | -- | 20 | -- | 30 | 3 | 50 |
| | **3. Skill Enhancement Course** | | | | | | | |
| | **BScCS -SEC-2** | 1)Mobile Application Development OR 2) Hardware Networking | | 20 | -- | 30 | 2 | 50 |
| | **4. Practical / Lab Course** | | | | | | | |
| | **BScCS -Lab-13** | Lab-Course5 (SE) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-14** | Lab-Course6 (Java) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-15** | Lab Course 7 (C#.Net) | -- | 20 | -- | 30 | 2 | 50 |
| | **BScCS -Lab-16** | Lab-Course8 (Multimedia) | -- | 20 | -- | 30 | 2 | 50 |
| | | **Total (IV)** | | | | | **24** | **500** |
| | | **Total (III + IV)** | | | | | **48** | **1000** |

# Semester – III

**Course Title: Object Oriented Programming using C++**
**Course Code:CC-A3**

**Total Teaching Hours: 50** **Total Marks: 50**
**Credit: 3**

---

**Learning Objective:**

1.Understand object oriented programming and advanced C++ concepts

1.1 Be able to explain the difference between object oriented programming and procedural programming.

1.2 Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.

1.3 Be able to build C++ classes using appropriate encapsulation and design principles.

2. Improve your problem solving skills

2.1 Be able to apply object oriented or non-object oriented techniques to solve bigger computing problems .

2.2 Ultimate goal: to make you a good programmer.

**Course Outcomes:**

- Gain the basic knowledge on Object Oriented concepts.
- Ability to develop applications using Object Oriented Programming Concepts.
- Ability to implement features of object oriented programming to solve real world problems.
- Use the characteristics of an object-oriented programming language in a program.
- Use the basic object-oriented design principles in computer problem solving.
- Use the basic principles of software engineering in managing complex software project.
- Program with advanced features of the C++ programming language.

---

# Syllabus

**UNIT- I: Introduction to Object Oriented Programming**

**1. Principles of Object Oriented Programming (OOP):**
Evolution of C++ - Programming Paradigms - Key Concepts of OOP - Advantages of OOP - Usage of OOP and C++ .Input and Output in C++- Streams-Stream classes Unformatted console I/O operations-Member functions of istream class-manipulators-manipulators with parameters

**2. Introduction to C++:**
Tokens, Keywords, Identifiers, Variables, Operators, Expressions and Control Structures: If,If. Else, Switch – Repetitive Statements- for, while, do..while - Pointers and arrays

**UNIT II: Class , Functions and Constructors**

**3. Structures and Unions:** Declaration of structures, Accessing structure members, Structure Initialization, Arrays of structure, nested structures, structure with pointers, functions & structures, Unions, Structure/Union Versus Class in C++.

**4. Class Declaration:** Data Members, Member Functions, Private and Public Members, Data Hiding and Encapsulation, Array within a class.

**5. Class Function Definition:**
Member Function definition inside the class and outside the class, Friend Function, Inline Function, Static Members & Functions, Scope Resolution Operator, Private and Public Member Functions, Nesting of Member Functions. Creating Objects, Accessing class data members, Accessing member functions, Arrays of Objects, Objects as function arguments: Pass by value, Pass by reference, Pointers to Objects.

**6. Constructors and Destructors:**
Declaration and Definition, Default Constructors, Parameterized Constructors, Constructor Overloading, Copy Constructors. Destructors: Definition and use.

**UNIT III: Inheritance and Overloading**

**7. Inheritance**
Extending Classes Concept of inheritance, Base class, Derived class, Defining derived classes, Visibility modes : Private, public, protected; Single inheritance : Privately derived, Publicly derived; Making a protected member inheritable, Access Control to private and protected members by member functions of a derived class, Multilevel inheritance, Nesting of classes.

**8. Function Overloading & Operator Overloading**
Binary & Unary.

**UNIT IV: Polymorphism and file operations**

**9. Polymorphism:**
Definition, early Binding, Polymorphism with pointers, Virtual Functions, late binding, pure virtual functions.

**10. Working with files:**
header files,, redirection, Classes for File Stream Operations - Opening and Closing a File - End-of-File Detection - file input and output. File Pointers - Updating a File - Error Handling during File Operations - Command-line Argumentsbuffers &iostreams,

**References / Books:**
1. Object Oriented Programming with C++, E. Balagurusami, Fourth Edition, Tata Mc-Graw Hill
2. Object Oriented Programming in Turbo C++, Robert Lafore, Fourth Edition Galgotia Publications.
3. The C++ Programming Language, BjarnaStroustrup, Third Edition, Addison-Wesley Publishing Company.
4. Object Oriented Programming Using C++, Salaria, R. S, Fourth Edition, Khanna Book Publishing
5. Ashok N.Kamthane, Object Oriented Programming with ANSI & Turbo C ++, Pearson Education, 2006

**Course Code: CC-B3**
**Course Title:- Data Structure**

**Total Teaching Hours: 60** **Total Marks: 50**
**Credits: 3**

---

**Course Objectives**

- To teach efficient storage mechanisms of data for an easy access.
- To design and implementation of various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structures.
- To teach the concept of protection and management of data.
- To improve the logical ability

**Course Outcomes**

- Student will be able to choose appropriate data structure as applied to specified problem definition.
- Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
- Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.
- Students will be able to use linear and non-linear data structures like stacks, queues , linked list etc

---

**UNIT I:**

1. **Introduction to Data structure and Arrays**
    1.1. Definitions –Data types, Data Object, Data structure.
    1.2. Implementations of Data structure.
    1.3. Need of Data Structure
    1.4. Types of Data Structure
    1.5. Array- definition.
    1.6. Types-one, multi dimensional, character string array

2. **Algorithm analysis**
    2.1. Algorithm – definition, characteristics
    2.2. Space complexity, time complexity
    2.3. Asymptotic notation (Big O, Omega _)

**UNIT II:**

3. **Stack**
    3.1. Definition of stack.
    3.2. Operation on stack.
    3.3. Declaration of stack.
    3.4. Application of stack-Recursion infixes, prefixes, and postfixes expression.
4. **Queue**
    4.1. Definition of queue.
    4.2. Operations on queue.

4.3. Types of queue-Linear, Circular.
4.4. Applications of queue.

**UNIT III:**

**5. Linked List**

5.1. Concept of linked list
5.2. Implementation of Linked list
5.3. Operations on linear linked list, on on circular linked list, doubly linked list
5.4. Implementation of stack and queue using linked list.

**6. Tree**

6.1. Trees : definition, terminologies, representation, types ( Only theory)
6.2. Tree Traversal- (Preorder, Inorder, Postorder) (Only theory)
6.3. Expression Trees (infix,prefix ,postfix)

**UNIT IV:**

**7. Searching & Sorting**

7.1. Searching : linear and binary
7.2. Sorting : bubble sort, selection sort, insertion sort,

**8. Graph**

8.1. Concept & terminologies
8.2. Graph Representation
8.3. Traversals – BFS & DFS
8.4. Applications – AOV network – topological sort
8.5. AOE network – critical path
8.6. Shortest path with implementation

**REFERENCE BOOK:**

1. Data Structure using C by A.M. Tanenbaum, Yecidyan lang
2. Data Structure Through C- By Dr. Sahani**.**
3. Data Structures Using C Yashwant Kanitkar – BPB Publication

**Course Title: Web Programming**
**Course Code:CC-C3**

**Total Teaching Hours: 50** **Total Marks: 50**
**Credit: 3**

---

**Learning Objectives:**

The course has been designed to provide the knowledge of Cascading Style Sheet, Java Script and XML for design of the web page / site.

**Course Outcomes:** At the end of this course the successful student will be able to:

- Apply a structured approach to identifying needs, interests, and functionality of a website.
- Design dynamic websites that meet specified needs and interests.
- Write well-structured, easily maintained, standards-compliant CSS code to present HTML pages in different ways.
- Use JavaScript to add dynamic content to pages.
- Select appropriate HTML, CSS, and JavaScript code from public repositories of open-source and free scripts that enhances the experience of site visitors.
- Modify existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice.
- Effectively debug JavaScript code, making use of good practice and debugging tools.
- Use JavaScript libraries (e.g. JQuery) to create dynamic pages.

---

**Unit I: HTML Basics and UI Designing**

Structure of HTML document, Basic Markup Tags, Working with Table, Working with Images, Creating and Handling Frames. **Forms:** What is Form?, Form Tag, Method, Action, Input Tag, Type Attribute: Check box, Hidden, Image, Radio, Reset, Submit, Text; Other <INPUT> attributes.

**Unit II: Cascading Style Sheet**

Evolution of CSS, Understanding the CSS Syntax, Exploring CSS Selectors (universal, type, class, id, child, descendant, adjacent sibling), Inserting CSS in an HTML document: The Internal Style Sheet The External Style Sheet The Inline Style Sheet Defining Inheritance in CSS Backgrounds and Color Gradients, Fonts and Text Styles, Creating Boxes and Columns, Displaying, Positioning, Floating an Element, List Styles, Table Layouts Pseudo-classes and Pseudo-element

**Unit III: JavaScript**

**Introduction**, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, **Operators:** Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), --(Decrement), -(Unary Negation), Logical Operators, Short-Circuit Evaluation, String Operators, Special Operators, ?: (Conditional operator), , (Comma operator), delete, new, this, void **Statements:** Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while **Core JavaScript (Properties and Methods of Each) :** Array, Boolean, Date, Function, Math, Number, Object, String, regExp **Document and its associated objects:** document, Link, Area, Anchor, Image, Applet, Layer. **Fundamental JavaScript Directives:** In-Line JavaScript, Linking Web Pages to External

JavaScript Files, JavaScript Using <script> Tags and Attributes. **Event Handling:** Events and Event Handlers General Information about Events, Defining Event Handlers, **Events:** onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload

**Unit IV: Jquery and XML**

Fundamentals of jQuery, Loading and using jQuery, jQuery Syntax, jQuery Selectors, Element properties and attributes, Methods to access HTML Attributes, Methods for Traversing, jQuery Events, CSS using jQuery. **Introduction to XML:** Anatomy of an XML document Creating XML Documents, Creating XML DTDs, XML Schemas, XSL

**Reference Books:**


1) HTML Complete Reference
2) JavaScript 2.0: The Complete Reference, Tata McGraw Hill
3) XML Complete Reference


**Recommended Readings:**

1) http://www.w3schools.com/css
2) http://jigsaw.w3.org/css-validator

**Course Code: CC-D3**
**Course Title:- OPERATING SYSTEM**

**Total Teaching Hours: 60** **Theory Total Marks: 50**
**CREDIT: 03**

---

**Learning Objective :**

1. To introduce students with basic concepts of Operating System, its functions and services.
2. To familiarize the students with various views and management policies adopted by O.S. as pertaining with processes , Deadlock , memory , File and I/O operations.
3. To brief the students about functionality of various OS like , Linux and Windows XP as pertaining to resource management.
4. To provide the knowledge of basic concepts towards process synchronization and related issues.

**Course Outcomes :**

1. Master functions, structures and history of operating systems
2. Master understanding of design issues associated with operating systems
3. Master various process management concepts including scheduling, synchronization, deadlocks
4. Be familiar with multithreading
5. Master concepts of memory management including virtual memory
6. Master system resources sharing among the users
7. Master issues related to file system interface and implementation, disk management
8. Be familiar with protection and security mechanisms
9. Be familiar with various types of operating systems including Linux.

---

**SYLLABUS**

**UNIT- I: Introduction to windows O.S.**

**CHAPTER 1. Introduction**
1.1. Definition of O.S.
1.2. Types of O.S.
1.3. O.S. as resource manager
1.4. O.S. Process view
1.5. Hierarchical view

**CHAPTER 2. Introduction to windows O.S.**
2.1. Introduction
2.2. History
2.3. Files and Folders
2.4. Architecture of windows
2.5. Basics of Windows: desktop, my computer, etc

**CHAPTER 3. Features of MS-Windows**
3.1. GUI, Multitasking, Multi-user, network etc.

3.2. Important files of windows

**UNIT II: Memory Management**

**CHAPTER 4. Memory management**
4.1. Single continues allocation
4.2. Introduction to multiprogramming
4.3. Partitioned Memory management
4.4. Paged memory management, demand paged memory management
4.5. Segmented Memory management

**UNIT III: Processor Management**

**CHAPTER 5. Processor Management**
5.1. State model
5.2. Job Scheduling
5.3. Process Scheduling
5.4. Multiprocessor system
5.5. Process synchronization

**UNIT IV: Device and Information Management**

**CHAPTER 6. Device management**
6.1. Techniques for Device management
6.2. Device management characteristics
6.3. Channels and control units
6.4. Device allocation consideration

**CHAPTER 7. Information management**
7.1. A simple file system
7.2. General model of a file system
7.3. Symbolic File System
7.4. Basic File System.

**Reference Books:**
1. Silberschatz A., Galvin P., Gagne G. “Operating Systems Principles”, Willey Eight edition
2. Achyut S. Godbole , Atul Kahate “Operating Systems” McGraw Hill Third Edition
3. “Operating SystemInternal & Design Principles”, William Stallings, Pearson
4. Andrew S. Tanenbaum, “Modern Operating System”, Prentice Hall.

**Class :- B.Sc.(C.S.)- S.Y [III-Sem]**

**Course Title :- Android Operating System** **Course Code:-SEC-I**

**Credit :- 02**

**Learning Objective:-** Learn Basic of Android operating system, Learn basic of XML , and learn basic concepts of java programming like Classes, Packages, Thread, Exception Handling etc.

**Course Outcome:-**Student should write xml code to design android controls , students also able to write java programs with object oriented features, they should create their own packages and able to access created packages.

**Syllabus**

**Unit –I :- Android History and Scope**

**Chp 1. Introduction to Android**

1.1 Need of Mobile Application
1.2 Introduction to Android
1.3 Types of Mobile Applications
1.4 Android Versions

**Chp 2. Android Architecture**

2.1 Android Architecture
2.2 Linux Kernal
2.3 Dalvik Virtual Machine

**UNIT –II:-IDE's and Java Basic**

**Chp 3. Android IDE's & Components**

3.1 Various IDE For Android
3.2 Installtion of Android
3.3 Android Virtual Device
3.4 Android Components

**Chp 4. Introduction To Java**

4.1 Introduction & History of Java
4.2 Java Applications
4.3 Java Architecture

**Unit – III:- Java Programming**

**Chp 5. Programming Basics**

5.1 Variable, Constants
5.2 Hello World Program
5.3 Classes&Inheritance
5.4 Interface

**Unit –IV:- Packages & UI Designing**

**Chp 6. Java Packages & Threads**

6.1 Packages
6.2 Thread
6.3 Exception Handling
6.4 Method Overloading

**Chp 7. XML &Json**

7.1 Tag ,Attribute
7.2 XML
7.3 Json

**Reference Books:-**


1. The Complete Reference Java2 By Herbert Schildt
2. Java CookBook By Ian Darwin,Publisher O'Reilly

**U-LAC-CC-A3**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**Learning Objective:**

1.Understand object oriented programming and advanced C++ concepts

1.1 Be able to explain the difference between object oriented programming and procedural programming.

1.2 Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.

1.3 Be able to build C++ classes using appropriate encapsulation and design principles.

2. Improve your problem solving skills

2.1 Be able to apply object oriented or non-object oriented techniques to solve bigger computing problems .

2.2 Ultimate goal: to make you a good programmer.

**Course Outcomes:**

- Gain the basic knowledge on Object Oriented concepts.
- Ability to develop applications using Object Oriented Programming Concepts.
- Ability to implement features of object oriented programming to solve real world problems.
- Use the characteristics of an object-oriented programming language in a program.
- Use the basic object-oriented design principles in computer problem solving.
- Use the basic principles of software engineering in managing complex software project.
- Program with advanced features of the C++ programming language.

**Proposed Practical List:**

1. Program to demonstrate encapsulation using of class.
2 Program to demonstrate use of array of objects
3 Program to demonstrate use of pointers
4 Program to demonstrate use of pointer to members of class
5 Program to demonstrate use of function overloading
6. Program to demonstrate inline function.
7 Program to demonstrate use of friend function
8. Program to demonstrate static data members & member functions of class.
9 Program to demonstrate use of different manipulators
10. Program to demonstrate objects as function arguments.
11. Program to demonstrate use of constructor, constructor overloading & destructor
12. Program to demonstrate use of all types of Inheritance.
13. Program to demonstrate use of unary & binary operator overloading
14. Program to demonstrate use of polymorphism (virtual function)
15. Program for reading and writing operations on text file.
16. Program to demonstrate command line arguments.

**U-LAC-CC-B3**

**Practical Total Marks: 50** **CREDIT: [ 02]**

**Learning Objectives**

- To teach efficient storage mechanisms of data for an easy access.
- To design and implementation of various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structures.
- To teach the concept of protection and management of data.
- To improve the logical ability

**Course Outcomes**

- Student will be able to choose appropriate data structure as applied to specified problem definition.
- Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
- Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.
- Students will be able to use linear and non-linear data structures like stacks, queues , linked list etc

**Proposed Practical List:**

1. Introduction to pointers. Call by Value and Call by reference.

2. Introduction to Dynamic Memory Allocation. DMA functions malloc(), calloc(), free() etc.

3. Implement a program for stack that performs following operations using array.
   (a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
4. Implement a program to convert infix notation to postfix notation using stack.
5. Write a program to implement QUEUE using arrays that performs following operations
   (a) INSERT (b) DELETE (c) DISPLAY
6. Write a program to implement Circular Queue using arrays that performs following operations.
   (a) INSERT (b) DELETE (c) DISPLAY
7. Write a menu driven program to implement following operations on the singly linked list.
   (a) Insert a node at the front of the linked list.
   (b) Insert a node at the end of the linked list.
   (c) Insert a node such that linked list is in ascending order.(according to info. Field)
   (d) Delete a first node of the linked list.
   (e) Delete a node before specified position.
   (f) Delete a node after specified position.
8. Write a program to implement stack using linked list.
9. Write a program to implement queue using linked list.
10. Write a program to implement following operations on the doubly linked list.
    (a) Insert a node at the front of the linked list.

(b) Insert a node at the end of the linked list.
(c) Delete a last node of the linked list.
(d) Delete a node before specified position.

11. Write a program to implement following operations on the circular linked list.
    (a) Insert a node at the end of the linked list.
    (b) Insert a node before specified position.
    (c) Delete a first node of the linked list.
    (d) Delete a node after specified position.
12. Write a program which create binary search tree.
13. Implement recursive and non-recursive tree traversing methods inorder, preorder and post-order traversal.

**Course Title: Web Programming**
**Course Code: U-LAC-CC-C3**

**Total Teaching Hours: 50** **Total Marks: 50**
**Credit: 2**

---

**Learning Objectives:**

The course has been designed to provide the knowledge of Cascading Style Sheet, Java Script and XML for design of the web page / site.

**Course Outcomes:** At the end of this course the successful student will be able to:

- Apply a structured approach to identifying needs, interests, and functionality of a website.
- Design dynamic websites that meet specified needs and interests.
- Write well-structured, easily maintained, standards-compliant CSS code to present HTML pages in different ways.
- Use JavaScript to add dynamic content to pages.
- Select appropriate HTML, CSS, and JavaScript code from public repositories of open-source and free scripts that enhances the experience of site visitors.
- Modify existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice.
- Effectively debug JavaScript code, making use of good practice and debugging tools.
- Use JavaScript libraries (e.g. JQuery) to create dynamic pages.

---

1. Design a web page using different text formatting tags.
2. Design a web page with links to different pages and allow navigation between web pages.
3. Design a web page with different tables.
4. Design a webpage using table so that the content appears well placed.
5. Design a web page with a form that uses all types of controls.
6. Creating and Using Internal Style Sheet
7. Creating and Using External cascading style sheet
8. Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series.
9. Design a form and validate all the controls placed on the form using Java Script.
10. Write a JavaScript program to display all the prime numbers between 1 and 100.
11. Write a JavaScript program to accept a number from the user and display the sum of its digits.
12. Write a java script program to design simple calculator.
13. Design a web page demonstrating different conditional statements.
14. Design a web page demonstrating different looping statements.

15. Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).
16. Web pages using jQuery

**U-LAC-CC-D3**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**Learning Objective :**

1. To introduce students with basic concepts of Operating System, its functions and services.
2. To familiarize the students with various views and management policies adopted by O.S. as pertaining with processes , Deadlock , memory , File and I/O operations.
3. To brief the students about functionality of various OS like , Linux and Windows XP as pertaining to resource management.
4. To provide the knowledge of basic concepts towards process synchronization and related issues.

**Course Outcomes :**

1. Master functions, structures and history of operating systems
2. Master understanding of design issues associated with operating systems
3. Master various process management concepts including scheduling, synchronization, deadlocks
4. Be familiar with multithreading
5. Master concepts of memory management including virtual memory
6. Master system resources sharing among the users
7. Master issues related to file system interface and implementation, disk management
8. Be familiar with protection and security mechanisms
9. Be familiar with various types of operating systems including Linux.

---

**Proposed Practical List:**

1. Study booting process of Windows XP, Linux.
2. To describe the basic function & services of operating systems
3. To explain the block level description of functional units of Operating system
4. To Differentiate Between working of various types of operating systems
5. To identify different types of services and system calls.
6. To describe the fundamentals and technological aspects of File Management.
7. To list and explain different types of disk scheduling algorithms.
8. To describethe File structure.
9. Comparevarious file allocation techniques.
10. Summarize the features of LINUX. (U) 2.Analyze directory system in LINUX.
11. Summarize the features of WINDOWS 7.
12.Analyze directory system, Process management in windows 7.

# Semester – IV

**Course code: CC-A4**
**Course Title: SOFTWARE ENGINEERING**

**Credit:03**

**TOTAL TEACHING HOUR : 60** **TOTAL MARKS :50**

## LEARNING OBJECTIVE:

The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high-quality software at low cost and with a small cycle of time. In software engineering you develop your skills for developing new and useful software's. Main objectives are: Understanding user conceptual manual and develop better specifications. Improvement in design languages. Reusable codes. Interactive debugging. Mockup to conform specifications.

A. Be employed in industry, government, or entrepreneurial endeavors to demonstrate professional advancement through significant technical achievements and expanded leadership responsibility;

B. Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment.

C. Progress through advanced degree or certificate programs in computing, science, engineering, business, and other professionally related fields.

## COURSE OUTCOME:

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. An ability to function on multi-disciplinary teams. An ability to identify, formulate, and solve engineering problems. An understanding of professional and ethical responsibility. An ability to communicate effectively.

---

## SYLLABUS

### UNIT I: ROLE OF SOFTWARE

**1.INTRODUCTION TO SOFTWARE ENGINEERING:**

The evolving role of software, changing nature of software, software myths. The software problem: Cost, Schedule and quality, scale and change.

**2. SOFTWARE PROCESS:**

Process and project, component software process.

Software development process models- waterfall model, prototyping , iterative development , relational unified process, time boxing model, Extreme programming and agile process, using process models in a project. project management process.

### UNIT II: REQUIREMENT ANALYSIS

**3.SOFTWARE REQUIREMENT ANALYSIS AND SPECIFICATION:**

Value of good SRS, requirement process, requirement specification, functional specifications with use-cases, other approaches for analysis, validation.

**4. PLANNING A SOFTWARE PROJECT:**
Effort estimation, project schedule and staffing, quality planning, risk management plans, project monitoring plan, detailed scheduling.

**UNIT III: ARCHITECTURE**

**5. SOFTWARE ARCHITECTURE:**
Role of software architecture, architecture view, components and connector view, Architecture styles for C and C view , documenting architecture design, evaluating architectures.

**6.DESIGN:**
Design concepts, function-oriented design, object oriented design, detailed design, verification, and metrics.

**UNIT IV: TESTINGS**

**7. CODING AND UNIT TESTING:**
Programming principles and guidelines, incrementally developing code, managing evolving code, unit testing ,code inspection, and metrics.
Testing: Testing concepts, testing process, black-box testing, white-box testing and metrics.

**REFERENCE BOOKS:**
1. R.PRESSMAN: Software Engineering- Mc Graw -Hill
2. R.K. Agrawal and Y.Sing: Software Engineering- New Age International.
3. P. Jalote : Software Project Management in practice- Pearson.

**Course Code:CC-B4**
**Course Title:- Java Programming**

**Total Teaching Hours: 55** **Max. Marks: 50**
**Total Credit:- 03**

---

**Learning Objectives:**

- Learn the basic concepts of Object-Orientation and how they are handled in Java
- Covers techniques for better class construction
- Understand Exceptions. How and when they should be handled
- An overview of database access and details for managing information using the JDBC API
- Examines the use of Object Serialization
- Learn how to use Servlet and JSP and XML with JSP
- Be able to create and use custom JSP tags
- A presentation of Enterprise JavaBeans and how to use it

**Course Outcome:** After Completion of this course students are able to :

- Use the methods of the Applet and Component classes required for a basic applet
- Describe the classes in the AWT package that relate to the Applet class
- Describe the AWT graphics explain controls and how to apply them in the container
- Develop programs using Event class and Event Listener Interface
- Develop a program for steps to connect a database
- Describe the use of JDBC
- Develop program to use JDBC to query a database and modify
- Describe life cycle of servlet
- Develop program using javax.servlet package
- Explain JSP Architecture and its Life cycle

---

**Syllabus**

**Unit I**

**1.Introduction to AWT: Working with windows, Graphics Text**

1.1AWT Classes
1.2Windows Fundamentals
1.3Working with Frame window
1.4Working with Graphics
1.5Working with Colors & Fonts

**2. Swing Components**

2.1 Icons & Labels Button & Label, TextField & Buttons,
2.2 CheckBoxes, Radio buttons
2.3 Combo Box & Lists
2.4 Scroll panes
2.5 Trees
2.6 Tables
2.7 Menu Bars & Menus

**Unit-II**

**3. Networking**
3.1 The java.net package
3.2 Connection oriented transmission – Stream Socket Class
3.3 Creating a Socket to a remote host on a port (creating TCP client and server)
3.4 Simple Socket Program Example
3.5 Programs on chatting 1-1 .
**4. JDBC**
4.1 The design of JDBC
4.2 Basic JDBC Concept
4.3 Drivers
4.4 Making the Connection, Statement
4.5 Executing SQL commands
4.6 Executing queries
4.7Scrollable and updatable result sets
4.8Metadata, transactions

**Unit-III**
**5. Servlet**
5.1 Introduction
5.2 Life cycle of servlet
5.3 Handling HTTP Get Request
5.4 Handling HTTP Post Request
**6. Introduction to JSP**
6.1 Getting Familiar with JSP Server
6.2 First JSP
6.3 Adding Dynamic contents via expressions
6.4 Scriptlets, Mixing Scriptlets and HTML
6.5 Directives, Declaration, Tags and Session

**Unit-IV**
**7. Introduction to Java Beans & Hibernate**
7.1 What is bean
7.2 Advantages
7.3 The bean-writing process
7.4 Introduction to jar and manifest files
7.5 The java beans API
7.6 Overview Of hibernate
7.7 Hibernate Architecture

**Reference Books:**
1) Complete reference Java by Herbert Schildt(5th edition)
2) Java 2 programming black books, Steven Horlzner
3) Programming with Java , A primer ,Forth edition , By E. Balagurusamy
4) Java servlet Programming by Jason Hunter, O'Reilly
5) Core Java Volume-II-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press.

**Course Title: Programming in C#.Net**
**Course code: CC-C4**

**Total Teaching Hours: 50** **Total Marks: 50**
**Credits: 03**

---

**Learning Objectives**:

- To understand the DOTNET framework andC# language features
- To develop object oriented programs on C#.
- To develop windows based applications on .NET framework.

**Course Outcome**:

After the successful completion of the course students will be able to:

- Describe basic concepts and develop programs in C# using object oriented features, delegates, events, errors and exceptions
- Interpret data access and develop windows applications
- Explain Common language runtime (CLR) as a platform for managed code
- Describe the features of Common language runtime (CLR) and develop efficient code with C# on .NET framework
- Develop windows based applications & services on .NET framework
- Describe overview of .NET framework
- Apply an understanding of the .NET technology and C#.net components to develop a windows based application which solves specified problem domain
- Use of ADO.NET technology for developing database oriented applications
- Understand the professional responsibility
- Apply an understanding of the need for high ethical standards in the practice of engineering towards people and the environment

---

## SYLLABUS

**UNIT I: Introduction to .net, Arrays and operators**

What is .net?,.Net Framework, CLR, Visual Studio.net & .net Languages, Integrated Development Environment, Project types, c#.net History & design Goals, How c# differs from c++, Characterstics of c#.net, I/O Statement with C#.net, Boxing & Unboxing, Short Circuiting Operators, Array & ArrayList class, Jagged Array, String Class

**UNIT II: Properties, Events, Delegates and C# namespaces**

Properties & its type, Event, Delegate & Multicast Delegate, Creating & Starting thread, Exception handling, using keyword, creating and using namespaces, interface, Method overloading & method overriding, Partial Class

**UNIT III: Windows Application**

Event Driven Programming, Building windows application with visual studio, TextBox, Label & Button Control, ComboBox, ListBox ,CheckBox & GroupBoxControl, DateTimePicker , Timer control,Building Menu,. MDI Form, PictureBox , ProgressBarControl, Common Dialog boxes, Introduction to WPF

**UNIT IV: Ado.Net and Database Oriented Applications**

How Ado.net differs from Ado, Advantages of Ado.net, Connected & Disconnected Architecture, Dataset, DataReader & DataAdapter, Managed Data Providers, DataGridViewControl, Developing Ado.net Based Application, Insert, Update & Delete operation on table, Filling the Dataset

**Reference books:**

1. Programming in C# A Primer - Second Edition By - E Balagurusamy
2. C#.Net Programming Wrox Publication
3. .Net 4.0 programming black book by KOGENT LEARNINGSOLUTIONS INC.
4. C# 2010 programming black book by KOGENT LEARNINGSOLUTIONS INC.

**Course Title: Multimedia Systems**
**Course Code: CC-D4**

**Total Teaching Hours: 50** **Total Marks: 50**
**Total Credit :-03**

**LEARNING OBJECTIVES:**

- To learn the basic concepts of animation as an art.
- To produce exercises as well as a final project in animation using Flash.
- To understand the basic animation techniques and concepts covered in the films and projects viewed in class.

**COURSE OUTCOMES:**

- To develop an understanding of the tools used for creating two dimensional (2D) Graphics and animation.
- To facilitate the students to design 2D Graphics, 2D character modelling and Animation.
- To make the students create a CBT presentation.

**UNIT – I**

1. **Exploring The Flash Interface**
1.1. The Flash stage
1.2. Stage Settings
1.3. Creating a new Flash file
1.4. The various import formats
1.5. Timeline- Play head/Frames/Key Frames/ Blank frames
1.6. Menus, Toolbox and Properties
1.7. Color Swatches and Color Mixer
1.8. Rulers, Guides, Grids and Snappings

**UNIT - II**

2. **Introduction**
2.1. CDROM and Multimedia Highway
2.2. Applications of Multimedia
2.3. Stages of Multimedia Project
3. **Macintosh and Windows Productions Platforms**
3.1. Macintosh Platform
3.2. Windows Platform
3.3. Connections- SCSI and IDE
3.4. Memory and Storage devices
3.5. Input and Output Devices

**UNIT - III**

4. **Basic Software Tools**
4.1. Text editing and word Processing tools
4.2. Painting and drawing tools
4.3. Image Editing Tools

4.4. Sound Editing Tools
4.5. Font Editing and designing tools
4.6. Hypermedia and Hypertext
4.7. Making Still Images : BITMAPS , Vector Drawing
4.8. Colors, Image file formats

**Unit - IV**

5. **Animation and Video**
5.1. Principal of Animation
5.2. Making animation that work: Rolling Ball, Bouncing ball
5.3. Using Video
5.4. Broadcast Video Standards
5.5. Recording Formats

**References Books:**

1. Macromedia Flash MX 2004: The Complete Reference by Brian Underdahl
2. Action Script for Flash MX: The Definitive Guide, 2nd Edition By Colin Moock
3. Macromedia Flash MX 2004 Bible by Robert Reinhardt and Snow Dowd
4. Multimedia : Making it work (5th Editions) By Tay Vaughan (Tataamc)
5. Multimedia : Computing Communications and Applications By Ralf Steinmetz ,Klara Nahrstedt

**Recommended Web sites:**

1. http://www.webdevelopersnotes.com/tutorials/flash/
2. http://www.adobe.com/devnet/flash/
3. http://www.adobe.com/support/flash/tutorial_index.html
4. http://www.thefreecountry.com/webmaster/flash.shtml

**Class B.Sc.C.S. S.Y. (IV Sem)**
**Course Title:- Mobile Application Development Using Android**
**Course Code:- SEC-II**

**Total Credit :-02** **Marks: 50**

**Learning Objective :-**Learn Designing of android application, writing java code, joining xml with java, testing application on real mobile device or virtual device, database connectivity etc.
**Course Outcome:-** Student should design xml controls, join xml controls with java object, run app on real mobile device, create services, works with different java android classes like LocationManager, SensorManager, SQLiteOpenHelper etc.
Students can get job of Android developer or xml-android app UI designer.

## SYLLABUS

**Unit-I :- Android Basic**

**Chp1. Android Basic**
- 1.1 What is Android?
- 1.2 Activity Life Cycle
- 1.3 Call Back Methods
- 1.4 Logcat usage

**Chp2. Android Application Structure**
- 2.1 Android Application Structure
- 2.2 AndroidManifest.xml, <Uses-permission>, < uses-sdk>
- 2.3 First Sample Android Application
- 2.4 Activity Registration
- 2.5 Activity & Intent

**Unit-II :- Android Widgets**

**Chp3.UI Widget – I**
- 3.1 LinearLayout, RelativeLayout
- 3.2 Button, EditText,TextView
- 3.3 Event Handling
- 3.4 RadioButton , CheckBox
- 3.5 ImageView

**Chp 4.UI Widget – II**
- 4.1 SeekBar, ProgressBar
- 4.2 Switch
- 4.3 Analog Clock , Digital Clock
- 4.4 ListView
- 4.5 WebView

**Unit-III :- Android Menus &Database Connectivity**

**Chp 5.Menus & Notifications**
- 5.1 Alert Dialog
- 5.2 Prompt Dialog

5.3 Android Menus
5.4 Toast Notification
5.5 Custom Toast

**Chp6. Services & SQLite**

6.1 Android Services
6.2 SQL Commands
6.3 SQLiteOpenHelper Class
6.4 SQLite Based Application

**Unit- IV :- Telephony & Mini Projects**

**Chp7. Telephony & Sensor API**

7.1 TelephonyManager
7.2 PhoneCall
7.3 Send SMS
7.4 SensorManager

**Chp8. Mini Projects**

8.1 Torch App
8.2 Media Player
8.3 Voice To Text Conversion

**Reference Books:-**


3. Learning Android , OREILLY By:- Marko Gargenta
4. Android Black Book

**U-LAC-CC-A4**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**LEARNING OBJECTIVE:**

The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high-quality software at low cost and with a small cycle of time. In software engineering you develop your skills for developing new and useful software's. Main objectives are: Understanding user conceptual manual and develop better specifications. Improvement in design languages. Reusable codes. Interactive debugging. Mockup to conform specifications.

A. Be employed in industry, government, or entrepreneurial endeavors to demonstrate professional advancement through significant technical achievements and expanded leadership responsibility;

B. Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment.

C. Progress through advanced degree or certificate programs in computing, science, engineering, business, and other professionally related fields.

**COURSE OUTCOME:**

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. An ability to function on multi-disciplinary teams. An ability to identify, formulate, and solve engineering problems. An understanding of professional and ethical responsibility. An ability to communicate effectively.

---

**Proposed Practical List:**

1. Perform the practical on Requirement gathering.
2. Perform the practical on Requirement Specification.
3. Perform the practical on Requirement Analysis.
4. Practical on Designing the software.
5. Creating the Login form.
6. Testing the all forms.
7. Practical on designing the Tables.
8. Testing the all the tables.
9. Perform the practical on Database connectivity.
10. Perform the practical on Testing's
11. Perform the practical on Error testing.

**Java Programming**
**U-LAC-CC-B4**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**Learning Objectives:**

- Learn the basic concepts of Object-Orientation and how they are handled in Java
- Covers techniques for better class construction
- Understand Exceptions. How and when they should be handled
- An overview of database access and details for managing information using the JDBC API
- Examines the use of Object Serialization
- Learn how to use Servlet and JSP and XML with JSP
- Be able to create and use custom JSP tags
- A presentation of Enterprise JavaBeans and how to use it

**Course Outcome:** After Completion of this course students are able to :

- Write programs based upon java concepts.
- Create animation & events based upon advanced java concepts.
- Connect an application with database.
- Develop programs using java collection API as well as java Standard Library.
- Write, debug & document well structured java application

**Proposed Practical List:**

1. Write a program to demonstrate life cycle of applet and display Hello world on applet UI.
2. Write a program to create a frame window.
3. Write a program to make use various methods of graphics class object.
4. Write a program to display buttons, labels, and Image icons using swing.
5. Write a program to display check boxes and radio buttons using swing.
6. Write a program to display combo box and scroll pane using swing.
7. Write a program to display a tree using swing.
8. Write a program to display a table using swing.
9. Write a program to display Menu bar and menus using swing.
10. Write a program to demonstrate socket programming. E.g. send hello world to server from client.
11. Write a program to chat between client and server.
12. Write a program to connect to db and to execute the queries.
13. Write a Servlet code to demonstrate GET & POST methods.
14. Write a program to demonstrate JSP use.

**U-LAC-387**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**Learning Objectives**:

- To understand the DOTNET framework andC# language features
- To develop object oriented programs on C#.
- To develop windows based applications on .NET framework.

**Course Outcome**:

After the successful completion of the course students will be able to:

- Describe basic concepts and develop programs in C# using object oriented features, delegates, events, errors and exceptions
- Interpret data access and develop windows applications
- Explain Common language runtime (CLR) as a platform for managed code
- Describe the features of Common language runtime (CLR) and develop efficient code with C# on .NET framework
- Develop windows based applications & services on .NET framework
- Describe overview of .NET framework
- Apply an understanding of the .NET technology and C#.net components to develop a windows based application which solves specified problem domain
- Use of ADO.NET technology for developing database oriented applications
- Understand the professional responsibility
- Apply an understanding of the need for high ethical standards in the practice of engineering towards people and the environment

**Proposed Practical List:**

1. Program to demonstrate jagged array
2. Program to demonstrate String class and its methods
3. Program to demonstrate properties concept
4. Program to demonstrate delegate concept
5. Program to demonstrate creation of C# namespaces
6. Program to demonstrate interface concept
7. Program to demonstrate common windows controls
8. Program to demonstrate advance windows controls
9. Program to demonstrate simple database connectivity application
10. Program to perform insert, update and delete operation on database

**U-LAC-CC-D4**

**Practical Total Marks: 50** **CREDIT: [ PR-2]**

**LEARNING OBJECTIVES:**

- To learn the basic concepts of animation as an art.
- To produce exercises as well as a final project in animation using Flash.
- To understand the basic animation techniques and concepts covered in the films and projects viewed in class.

**COURSE OUTCOMES:**

- To develop an understanding of the tools used for creating two dimensional (2D) Graphics and animation.
- To facilitate the students to design 2D Graphics, 2D character modelling and Animation.
- To make the students create a CBT presentation.

---

**Proposed Practical List:**

1. Study of Flash environment.
2. Working with shape tween by designing A to Z characters animation.
3. Working with shape motion by designing A to Z characters animation.
4. Designing Bouncing Ball animation.
5. Creating a commercial Advertisement.
6. Creating a Jumping Man animation.
7. Creating a Flying Bird animation.
8. Creating a Walking Man animation.
9. Creating a Riding Bike animation.
10. Creating a Flying Honey Bee animation.
11. Creating a growing Plant animation.
12. Working with shape tween with blending options.