**B.C.A Under CBCS with effect from Academic Year 2016-2017 (Revised in April, 2016)**

**Andhra Pradesh State Council of Higher Education**

**BCA Under CBCS with effect from the academic year 2016-2017 course of study**

**Table-2: B.C.A. SEMESTER – II**

| Sno | Course | Total | Mid Sem | Sem End | Teaching | Credits |
|---|---|---|---|---|---|---|
| | | Marks | Exam* | Exam | Hours | |
| | | | | | | |
| 1 | First Language | 100 | 25 | 75 | 4 | 3 |
| | English | | | | | |
| | | | | | | |
| 2 | Foundation course - 3 | 50 | 0 | 50 | 2 | 2 |
| | Environmental Sci | | | | | |
| | | | | | | |
| 3 | Foundation course – 4A | 50 | 0 | 50 | 2 | 2 |
| | Adobe In Design | | | | | |
| | | | | | | |
| 4 | Statistical Methods and | 100 | 25 | 75 | 6 | 5 |
| | their Applications | | | | | |
| | | | | | | |
| 5 | Operating Systems | 100 | 25 | 75 | 4 | 3 |
| | | | | | | |
| 6 | Operating Systems Lab | 50 | 0 | 50 | 2 | 2 |
| | | | | | | |
| 7 | Object Oriented | 100 | 25 | 75 | 4 | 3 |
| | Programming Using | | | | | |
| | “C++” | | | | | |
| | | | | | | |
| 8 | Object Oriented | 50 | 0 | 50 | 2 | 2 |
| | Programming Using | | | | | |
| | “C++” Lab | | | | | |
| | | | | | | |
| 9 | Adobe In Design Lab | 50 | 0 | 50 | 2 | 2 |
| | | | | | | |
| | Total | 650 | | | 28 | 24 |
| | | | | | | |

**B.C.A Under CBCS with effect from Academic Year 2016-2017 (Revised in April, 2016)**

**Andhra Pradesh State Council of Higher Education**

**BCA Under CBCS with effect from the academic year 2016-2017 course of study**

**Table-4: B.C.A. SEMESTER – IV**

| Sno | Course | Total | Mid Sem | Sem End | Teaching | Credits |
|---|---|---|---|---|---|---|
| | | Marks | Exam* | Exam | Hours | |
| | | | | | | |
| 1 | Foundation Course – 2C* | 50 | 0 | 50 | 2 | 2 |
| | Communication & Soft | | | | | |
| | Skills -3 | | | | | |
| 2 | Foundation Course – 6* | 50 | 0 | 50 | 2 | 2 |
| | Analytical Skills | | | | | |
| | | | | | | |
| 3 | Foundation Course - 7 ** | 50 | 0 | 50 | 2 | 2 |
| | CE (Citizenship Education) | | | | | |
| | | | | | | |
| 4 | Foundation course – 4B | 50 | 0 | 50 | 2 | 2 |
| | ICT – 2 Dreamweaver | | | | | |
| | | | | | | |
| 5 | Unix | 100 | 25 | 75 | 4 | 3 |
| | | | | | | |
| 6 | Data Structures Using | 100 | 25 | 75 | 4 | 3 |
| | Java | | | | | |
| | | | | | | |
| 7 | Data Structures Using | 50 | 0 | 50 | 2 | 2 |
| | Java Lab | | | | | |
| | | | | | | |
| 8 | Web Programming | 100 | 25 | 75 | 4 | 3 |
| | | | | | | |
| 9 | Web Programming Lab | 50 | 0 | 50 | 2 | 2 |
| | | | | | | |
| 10 | Unix Lab | 50 | 0 | 50 | 2 | 2 |
| | | | | | | |
| | Total | 650 | | | 28 | 23 |

**BCA I Year II Semester**

# OPERATING SYSTEMS

Course Objectives

1.To understand the services provided by and the design of an operating system.

2.To understand the structure and organization of the file system.

3.To understand what a process is and how processes are synchronized and scheduled.

4.To understand different approaches to memory management.

5.Students should be able to use system calls for managing processes, memory and the file system.

6.Students should understand the data structures and algorithms used to implement an OS.

Course Outcomes

1.Analyze the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance.

2.Identify the dead lock situation and provide appropriate solution so that protection and security of the operating system is also maintained.

3.Analyze memory management techniques, concepts of virtual memory and disk scheduling.

4.Understand the implementation of file systems and directories along with the interfacing of IO devices with the operating system.

UNIT - I

Operating System Introduction: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Parallel, Distributed Systems, Real-Time Systems, Operating System services.

UNIT - II

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Case studies: Linux, Windows.

Process Coordination - Process Synchronization, The Critical section Problem, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT - III

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table. Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames.

UNIT - IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Structure,

Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling.

UNIT - V

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

.

**REFERENCES BOOKS:**

1.Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.

2.Principles of Operating Systems by Naresh Chauhan, OXFORD University Press

3.Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

4.Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.

5.Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.

6.Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.

7.Operating Systems, A. S. Godbole, 2nd Edition, TMH

**Student Activity:**

**1.Load any new operating system into your computer.**

**2.Partition the memory in your system**

**3.Create a semaphore for process synchronization**

**Operating Systems Lab**

1. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt

chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)

2.Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)

3.Developing applications using Inter Process Communication (using shared memory, pipes or message queues)

4.Implement the Producer – Consumer problem using semaphores

5.Implement any two memory management schemes

6.Implement any two file allocation techniques (Linked, Indexed or Contiguous)

7.Implement any two Page Replacement Algorithms

8.Implement Deadlock prevention algorithm.

9.Implement any two disk scanning algorithms

**BCA I Year II Semester**

**OBJECT ORIENTED PROGRAMMING USING C++**

Course Objectives

This course covers object-oriented programming principles and techniques using C++. Topics include pointers, classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low- level language features. This course also covers basic concepts for software design and reuse.

Course Outcomes

1.Understand concepts of objects and their significance in real world

2.Investigate software problem in terms of objects and entities

3.Learn to co-relate relationship among different entities involved in a system

4.Find dependency and roles in an environment

5.Develop software in terms of objects, associations, and integrity constraints

6.Generalize and aggregate business entities and transform behavior into functions

7.Identify, understand and analyze various sample development models

UNIT I

Principles of OOP: Software Crisis. Software Evolution- Programming Paradigms. Object Oriented Technology- Basic concepts and benefits of OOP – Application of OOP, OOP languages

Introduction to C++: History of C++, Structure of C++, Application of C++, tokens, keywords, identifiers, basic data types, derived data types, derived data types, symbolic constant, dynamic

initialization, reference variables, scope resolution operator, type modifiers, type casting operators and control statements, input and output statements in C++, Function prototyping and components, Passing parameters: Call by reference, Return by reference, Inline function, Default arguments, Over loaded function.

UNIT II

Classes and Objects: Class specification, Member function definition – nested member function, access qualifiers, static data members and, member functions. Instance creation - Array of objects - Dynamic objects - Static Objects – Objects as arguments -Returning objects Constructors and Destructors: Constructors- Parameterized constructors, Overloaded Constructors, Constructors with default arguments, copy constructors, Destructors.

UNIT III

Operator Overloading: Operator function-overloading unary and binary operators, overloading the operator using Friend function, Stream operator overloading, Data conversion.

Inheritance: Defining derived classes. Single Inheritance - Protected data with private inheritance - Multiple Inheritances - Multi Level Inheritance - Hierarchical Inheritance. Hybrid Inheritance - Multi path Inheritance - Constructors in derived and base Class -Template in Inheritance - Abstract classes - Virtual function and Dynamic polymorphism. -

Virtual destructor - Nested Classes

UNIT- IV

Functions in C++ : Virtual functions- need for Virtual function, , Pure Virtual functions, Generic Programming with Templates.

Introduction, function templates, overloaded function templates, user defined templates arguments, class templates, Inheritance of class templates.

UNIT-V

Files: file stream, file pointer and manipulation, file open and close, sequential and random access.

Exception Handling: Principle of Exception handling, Exception handling mechanism ,Multiple catch, Nested try, re throwing the Exception.

REFERENCE BOOKS:

1.1 Object Oriented Programming with C++ by Reema Thareja, OXFORD University Press

2.The Complete Reference C++, Herb Schildt, Tata McGraw-Hill, Fourth Edition.

3.Robert Lafore, "Object Oriented Programming in C++", Galgotia Publication Pvt. Ltd,4 th

edition, New Delhi, 2002

4.Ashok N Kamathane, "Object Oriented Programming with ANSI & Turbo C++", Pearson

Education, New Delhi, 2003.

5.Bjarne Stroustrup," C++ Programming language", Pearson Education, New Delhi, 2001.

6.Venugopal K R, Rajkumar Buyya and Ravishankar T," Mastering C++", TMH, ND, 2006

Student Activity:

1.Create a class diagram for academic process in your college

2.Write a program to implement “Vikuntapali”game

**OBJECT ORIENTED PROGRAMMING USING C++ LAB**

1.Write a C++ program to find the sum of individual digits of a positive integer.

2.A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.

3.Write a C++program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.

4.Write a C++program to find the factorial of a given integer

5.Write a C++program to find the GCD of two given integers

6.Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.

7.Write a C++program to implement call by value and call by reference parameters passing

8.Write a C++ program to implement function templates

9.Write a program to implement Overloading and Overriding

10.Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:

a. Reading a matrix. b. Printing a matrix

c. Addition of matrices d. Subtraction of matrices

e. Multiplication of matrices

11.Write C++programs that illustrate how the Single inheritance, Multiple inheritance Multi level inheritance and Hierarchical inheritance forms of inheritance are supported

12.Write a C++program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class

13.Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions

**Adobe In Design LAB**

Implement the following tasks using ADOBE

1.Resume designing

2.Paragraph setting

3.Text column wise designing

4.Text base paper add

5.Create college Logo

6.Table creation

7.Student marks list

8.Book work

9.Picture insertion

10.Application form

11.Text based Visiting card

12.Notice designing

13.Typographic alignment styles

14.Wedding card designing

15.Letter models

**BCA II Year IV Semester**

**UNIX**

Course Objectives

1.To understand Unix Operating System

2.To explore the Basic Shell Commands

Course Outcomes

After this course, the student will be able to

1.Implement and innovate commands using the basic tool kit.

2.Develop shell programs in vi/vim editor

Unit I

UNIX OPERATING SYSTEM

Overview of UNIX Operating System, basic features of Unix operating System, File Structure, CPU Scheduling, Memory Management, File System Implementation of Operating System Functions in UNIX.

Unit II

Starting Of Unix and Text Manipulation and user-to-user communication User Names and Groups, Logging In, Format of Unix Commands, Changing your password, Unix Documentation,

Unit III

Files and Directories: , File permission, Basic Operation on Files, Changing Permission Modes, Standard files , Processes Inspecting Files, Operating On Files, Printing Files, Rearranging Files, Sorting Files, Splitting Files, Translating Characters, On line communication, Off line communication.

Unit IV

VI EDITORS

General characteristics, Adding text and Navigation, changing text, searching for text, copying and Moving text, Features of Ex, Line Editors Ex and Ed, Stream editor SED, changing several file s in SED, AWK.

Unit V

Shell Programming:

Programming in the Bourne and C-Shell, Wild Cards, Simple Shell program, variables, Programming Construct, Interactive Shell scripts, Advanced Features, Unix Compiler, Maintaining program System Administration Define system Administration, Booting the system, Maintaining User Accounts, File System, and special files, Backup and Restoration.

References Books:

1.Unix and shell Programming by B.M Harwani, OXFORD University Press

2.Unix Concept and application- Sumitabhadas

3.Unix Shell Programming-Yashwant Kanetkar

4.Unix Programming Environment- RobPike

5.Unix in a Nutshell- Donill Gily

Student Activity:

1.Load unix/linux in your system in a separate drive

2.Create graphics in unix environment

**BCA II Year IV Semester**

**Unix Lab**

1.Execute of various file/directory handling commands.

2.Write a Simple shell script for basic arithmetic and logical calculations.

3.Write Shell scripts to check various attributes of files and directories.

4.Write Shell scripts to perform various operations on give n strings.

5.Write Shell scripts to explore system variables such as PATH, HOME etc.

6.Write Shell scripts to check and list attributes of processes.

7.Execute various system administrative commands

8.Write awk script that uses all of its features. 9.Use seed instruction to process /etc/password file.

10.Write a shell script to display list of users currently logged in.

11.Write a shell script to delete all the temporary files.

12.Write a shell script to search an element from an array using binary searching.

# DATA STRUCUTURES USING JAVA

## Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms. In addition, another objective of the course is to develop effective software engineering practice, emphasizing such principles as decomposition, procedural abstraction, and software reuse.

Course Outcomes

After completing this course satisfactorily, a student will be able to:

1.Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms.

2.Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.

3.Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs

4.Demonstrate different methods for traversing trees

5.Compare alternative implementations of data structures with respect to performance

6.Compare and contrast the benefits of dynamic and static data structures implementations

7.Describe the concept of recursion, give examples of its use, describe how it can be implemented using a stack .

8.Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.

UNIT I

Concept of Abstract Data Types (ADTs)- Data Types, Data Structures, Storage Structures, and File Structures, Primitive and Non-primitive Data Structures, Linear and Non-linear Structures.

Linear Lists - ADT, Array and Linked representations (Single and Double Linked lists), Pointers.

UNIT II

Stacks: Definition, ADT, Array and Linked representations, Implementations and Applications. Queues: Definition, ADT, Array and Linked representations, Circular Queues, Dequeues, Priority Queues and Applications.



UNIT III

Trees: Binary Tree, Definition, Properties, ADT, Array and Linked representations, Implementations and Applications, Heaps Trees and Applications,

Binary Search Trees (BST) - Definition, ADT, Operations and Implementations, BST with Duplicates and Applications.

UNIT IV

Graphs – Graph and its Representation, Graph Traversals, Connected Components, Basic Searching Techniques, Minimal Spanning Trees.

UNIT- V

Sorting and Searching: Selection, Insertion, Bubble, Merge, Quick, Sequential and Binary Searching.

REFERENCE BOOKS

1.SamanthaD, Classic Data Structures, Prentice-Hall of India, 2001

2.Sahani S, Data Structures, Algorithms and Applications in C++, McGraw-Hill, 2002.

3.D S Malik, Data Structures Using C++, Thomson, India Edition 2006

4.Heilman G I,. Data Structures, Algorithms and Object-Oriented Programming, Tata McGraw-l lill. 2002. (Chapters I and 14).

5.Tremblay .1 P, and Sorenson P G, Introduction to Data Structures and Applications, Tata McGraw-Hill,

6.Drozdek A, Data Structures and Algorithms in C++), 2nd edition, Vikas Publishing House, 2002.

7.Kanetkar Y P, Data Structures through C ++, BPB Publications. 2003.

8.Data Structures by Allen Weiss

Student Activity:

1.Create Visual Stack using graphics in JAVA

2.Create Visual Queue using graphics in JAVA

# DATA STRUCTURES USING JAVA LAB

1.Write Programs to implement the Stack operations using an array.

2.Write Programs to implement the Queue operations using an array.

3.Write Programs to implement the Stack operations using Pointers.

4.Write Programs to implement the Queue operations using Pointers.

5.Write a program for arithmetic expression evaluation.

6.Write a program for Binary search Tree Traversals

7.Write a program to implement dequeue using a doubly linked list.

8.Write a program to search an item in a given list using

(i)Linear Search

(ii)Binary Search.

9.Write a program for (i)Bubble Sort

(ii)Quick Sort

(iii)Merge Sort.

10. Write a program for polynomial addition using SLL

**BCA II Year IV Semester**

**WEB PROGRAMMING**

Course Objective

To provide knowledge on web architecture, web services, client side and server side scripting technologies to focus on the development of web-based information systems and web services.

To provide skills to design interactive and dynamic web sites.

Course Outcome

1.To understand the web architecture and web services.

2.To practice latest web technologies and tools by conducting experiments.

3.To design interactive web pages using HTML and Style sheets.

4.To study the framework and building blocks of .NET Integrated Development Environment.

5.To provide solutions by identifying and formulating IT related problems.

UNIT I

DNS – E-mail – FTP – TFTP – History of WWW – Basics of WWW and Browsing - Local information on the internet – HTML – Web Browser Architecture – Web Pages and Multimedia

– Remote Login (TELNET).

UNIT II

Introduction to Web Technology: Web pages – Tiers – Concept of a Tier – Comparison of Microsoft and Java Technologies – Web Pages – Static Web Pages – Plug-ins – Frames – Forms. Dynamic Web Pages: Need – Magic of Dynamic Web Pages – Overview of

Dynamic Web Page Technologies – Overview of DHTML – Common Gateway Interface.

UNIT III

ASP – ASP Technology – ASP Example – Modern Trends in ASP – Java and JVM – Java Servlets – Java Server Pages.

Active Web Pages: Active Web Pages in better solution – Java Applets – Why are Active Web Pages Powerful? – Lifecycle of Java Applets – ActiveX Controls – Java Beans. Middleware and Component-Based E-Commerce Architectures

UNIT IV

CORBA – Java Remote Method Invocation – DCOM. EDI: Overview – Origins of EDI – Understanding of EDI – Data Exchange Standards – EDI Architecture – Significance of EDI – Financial EDI – EDI and internet.

UNIT V

XML: SGML – Basics of XML – XML Parsers – Need for a standard. WAP: Limitations of Mobile devices – Emergence of WAP – WAP Architecture – WAP Stack – Concerns about WAP and its future – Alternatives to WAP.

REFERENCE BOOK

1.WEB TECHNOLOGIES TCP/IP to Internet Applications Architectures – Achyut S Godbole & Atul Kahate, 2007, TMH.

2.Web Technologies by Uttam Kumar Roy, Oxforn University Press

3.INTERNET AND WEB TECHNOLOGIES – Rajkamal, TMH.

4.TCP/IP PROTOCOL SUITE – Behrouz A. Forouzan, 3rd edition, TMH

Student Activity:

1.Design a website for your college

2.Design your personal web site

**BCA II Year IV Semester**

**WEB PROGRAMMING LAB**

1.Create a simple HTML page which demonstrates all types of lists.

2.Create a letter head of your college using following styles

i.image as background

ii.use header tags to format college name and address

3.Create a web page, which contains hyper links like fruits, flowers, animals. When you click on hyper links, it must take you to related web page; these web pages must contain with related images.

4.Create a hyperlink to move around within a single page rather than to load another page.

5.Create a leave letter using different text formatting tags.

6.Create a table format given bellow using row span and colspan.

RNONAME MARKS

M1M2 M3 M4M5

Insert 5 records.

7.Create a table with different formats as given bellow.

i.Give different background and font colors to table header, footer and body.

ii.Use table caption tag.

8.Divide a web page vertically and horizontally with scroll bars, name them as shown bellow decorate it with some items.

F2

F1

F3

9.Create a student Bio-Data, using forms.

10.Create a web page using following style sheets

i.Inline style sheets.

ii.Embedded style sheets.

iii.External style sheets

11.Write a JavaScript program to accept two values from form and apply any 5 mathematical functions

Write student database with XML