# Rajarshi Shahu Mahavidyalaya, Latur
# (Autonomous)

## Department of Computer Science and Information Technology
## Syllabus for under graduation course (BCA)

### (With Effect from Academic Year:2017-18 Under CBCS)

### Programme Name: Bachelor of Computer Application (First Year Semester I + II)

**Semester : I (CBCS)**

| Sr. No. | Subject | Title of the course | Course code | Credit Points | |
|---|---|---|---|---|---|
| 1 | English | Communicative English - II | U-COE-101 | 2 | |
| 2 | Foundation of IT | Foundation of IT | U-OPS-179 | 3 | |
| 3 | Introduction to C | Introduction to C | U-FDE-180 | 3 | |
| 4 | Web Page Designing | Web Page Designing | U-ADC-181 | 3 | |
| 5 | Fundamentals of Statistics | Fundamentals of Statistics | U-DAS-182 | 3 | |
| 6 | Lab. Course | Lab. Course – I | U-LAC-183 | 2 | |
| 7 | Lab. Course | Lab. Course - II | U-LAC-184 | 2 | |
| 8 | Lab. Course | Lab. Course - III | U-LAC-185 | 2 | |
| 9 | Lab. Course | Lab. Course - IV | U-LAC-186 | 2 | |

**Semester - II (CBCS)**

| Sr. No. | Subject | Title of the course | Course code | Credit Points | |
|---|---|---|---|---|---|
| 1 | English | Communicative English - II | U-COE-201 | 2 | |
| 2 | Operating System | Operating System | U-OPS-279 | 3 | |
| 3 | Fundamentals of Digital Electronics | Fundamentals of Digital Electronics | U-FDE-280 | 3 | |
| 4 | Advance C | Advance C | U-ADC-281 | 3 | |
| 5 | Data Structure | Data Structure | U-DAS-282 | 3 | |
| 6 | Lab. Course | Lab. Course - V | U-LAC-283 | 2 | |
| 7 | Lab. Course | Lab. Course - VI | U-LAC-284 | 2 | |
| 8 | Lab. Course | Lab. Course - VII | U-LAC-285 | 2 | |
| 9 | Lab. Course | Lab. Course - VIII | U-LAC-286 | 2 | |
| 10 | Moral Education | Moral Education | U-MOE-235 | - | |

## Course code:-U-FIT-179(cbcs)

**Course Title:-Foundation of IT** **Credit 3**

---

## Learning Objectives:

- Techniques for solving problems
- Basic computational concepts and elementary data structures
- The edit-compile-link-run cycle from a user point of view
- Testing strategies
- The main activities of software development and their interactions, and some of the major problems of software development

## Course Outcome:

The curriculum leading to a baccalaureate degree in Computer Science prepares students for positions as computer scientists in business, industry and government, or for graduate study in computer science. The curriculum's main objectives are to impart students with an understanding of the basics of computer science

### SYLLABUS

**UNIT I: Computer System AND Data Representation within Computer**

**1. Introduction to Computer System**

1.1. Introduction
1.2. Basic structure , ALU memory , CPU , I/O devices
1.3. Generations of computer
1.4. Evolution of computer
1.5. Classification of computers : Notebook computers, personal computers, workstation, micro, mini, mainframe, super computer

**2. Data Representation within Computer**

2.1. Bit , Byte, Word
2.2. ASCII, EBCDIC, BCD code
2.3. Introduction to number system
2.4. Decimal , Binary, Octal , Hexadecimal

**UNIT II : . Input Output Devices AND Memory**

**3. Input Output Devices**

3.1. Input Devices: Keyboard, Point & Draw Devices,
3.2. Data Scanning devices, Digitizer, Electronic Card Reader, Voice Recognition

devices
3.3. Output Devices Monitor, Printer, Plotter, Screen Image projector, voice response system.

**4. Memory**

4.1. RAM, ROM, PROM, EPROM, EEPROM
4.2. Base Memory , Extended memory, Expanded memory, cache memory
4.3. Storage devices : Tape, FDD, HDD, CD ROM

**UNIT III: Computer Software**

**5. Computer Software**

5.1. Definition of software
5.2. Types of software: Compilers, Interpreters, Assemblers, Linkers, Loaders
5.3. Operating System: Introduction
5.4. Main function of operating system
5.5. Files and directories
5.6. Types of OS

**UNIT IV : Operating system AND Internet**

**6. Operating system**

6.1. Fundamentals of DOS,Booting procedure of DOS
6.2. DOS commands (Internal & External)
6.3. Configuration of DOS (Confis.sys) ,Batch file concepts(Autoexe.bat)
6.4. Introduction to WINDOWS
6.5. Introduction to LINUX

**7. Internet :Basic services**

7.1. Email, File Transfer Protocol, Telnet
7.2. Internet search tools
7.3. www browsers
7.4. uses of the internet

## Reference Books:

1. Fundamentals of computer science-P.K.Sinha.

2 Fundamentals of computer science-V RAJARAMAN.

**Course Title: Introduction to C**
**Course Code: U-INC-180**

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

---

**Learning Objectives**

- Learn writing algorithms
- drawing flowchart to solve given problem
- C-syntax, function, Operators, Array, File Handling etc.

**Course Outcome**

- Student should write Algorithm to solve given problem
- drawing flowchart to solve given problem
- able to convert algorithm to flowchart
- write program to print output on console
- writing complex program, manage file with c program etc.

---

## Syllabus

**UNIT I: Algorithm , Flowchart & C Basic**

**1. Algorithm and flowcharts**

1.1. Definition and properties
1.2. Developing well known algorithms
1.3. Principles of flowcharting
1.4. Flow charting symbols
1.5. Converting algorithms to flowcharts

**2. Overview of C**

2.1. Introduction to C
2.2. Structure of C program
2.3. Character set C tokens, Keywords,Identifiers,Constants ,variables, Data types,Declaration of variables , Defining Symbolic Constants
2.4. Operators, Formatted input and output.

**UNIT II : Statements & Function**

**3. Decision making and Branching and Arrays**

3.1. Decision making with if statement - simple if statement- The if else Statement - Nesting of if ...else statements - The else if ladder
3.2. The switch statement ,Thegoto statement ,The break statement.The continue statement
3.3. Decision Making and Looping: The while statement - The do statement - The

for statement - Jumps in Loop:,
3.4. Arrays :One Dimensional - Two Dimensional - Multidimensional arrays –
3.5. String handling Functions.

**UNIT III : Function & Pointer**

**4. Introduction to function**

4.1. Return values and their types - Calling a function
4.2. Types of user defined functions
4.3. No Arguments and no return values
4.4. Arguments but no return values
4.5. Arguments with return values
4.6. Recursion.
4.7. Storage Classes-Automatic storage class, Register storage class, Static storage class, External storage class.

**5. Pointers**

5.1. Understanding pointers, Accessing the Address of' a variable
5.2. Declaring and initializing pointers
5.3. Accessing a variable through its pointers.

**UNIT IV : Structure, Union & File Operations**

**6. Introduction to Structure**

6.1. Structure initialization
6.2. Arrays of structures
6.3. Structures within structures
6.4. Introduction to Union

**7. File management in C**

7.1. Defining and opening a file - closing file
7.2. I/O operations on files
7.3. Error handling during I/0 operations
7.4. Random access to files
7.5. Command line arguments
7.6. Case study – Library details maintenance system.

**REFERENCE BOOKS:**

1. Let us C-YashwantKanetkar.
2. Programming in C- Balguruswamy
3. The C programming Lang., Pearson Ecl – Dennis Ritchie
4. Structured programming approach using C- Forouzah&Ceilberg Thomson learning publication.
5. Pointers in C – YashwantKanetkar
6. Byron Gottfried, - “Programming with C” (Schaum's Outline Series ) – Tata McGrawHill Publishing Company – 1998

**Course Code: U-WPD-181**
**Course Title: Web Page Designing**

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

---

**Learning Objective:** The course has been designed to provide the basic knowledge for design of the web page / site.

**Course Outcomes:**

- A website is Published using application Server i.e. Tomcat/ IIS
- Apply Java script Code to sites.
- Develop Motion Graphic with Flash.
- Apply proper layout and interactive website design

## Syllabus

**UNIT I:**

1. **Introduction to web designing &HTML Documents:**

1.1. Web page, web site, web browser, www.
1.2. Developing web documents , web design process.
1.3. Publishing documents, web publishing.
1.4. Maintaining documents, maintenance phase of web page
1.5. Overview of HTML
1.6. Rules of HTML Documents.
1.7. Structure of HTML documents.

2. **HTML Markup tags:**

2.1. Tags definition, Classification of tags.
2.2. Basic tags, HTML HEAD,TITLE,BODY
2.3. Paragraph tags, list tags, horizontal rule tag, PRE tags, Block quote tags, Address tags, Font tag div tags, Span tag &other different formatting tags.

**UNIT II**

3. **Linking in HTML**

3.1. U.R.L. concept
3.2. Hyperlink(anchor) tag & its all attributes,
3.3. Creating email hyperlinks (using mailto anchor)

4. **Image in HTML**

4.1. Image and image formats
4.2. <img> tag and its all attributes
4.3. Inline and floating images
4.4. Using images as links
4.5. Image map, client side and server side image maps.

**Unit III**

5. **Tables in HTML**

5.1. Introduction
5.2. <table> tag, <tr>, <th>, <td> and attributes
5.3. Rowspan, colspan, cellspecing, cellpading

5.4. example
6. **Frames in HTML**
6.1. Overview
6.2. <FRAMESET> and <FRAME> tag, and attributes
6.3. Simple Frame
6.4. Use of <NOFRAME> tag
6.5. Frame targeting
6.6. Floating frame

**UNIT IV:**
7. **Forms in HTML**
7.1. Introduction
7.2. <FORM> tag and attributes
7.3. Form controls like text filed, submit, reset, radio button, etc.
8. **DHTML**
8.1. Introduction
8.2. DOM
8.3. Introduction CSS
9. **VB and Java Script**
9.1. Introduction, adding script to document, data types, operators
9.2. Variables, global and local
9.3. I/O statements
9.4. Control Statements: if, if…else, select case.
9.5. Looping statements for---next, do-while, do until
9.6. Adding script to documents
9.7. If statements, looping statements
9.8. Events in java script.

**Reference Books:**


- HTML5 Programming for ASP.NET Developers Learning
- Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics

**Course Title: FUNDAMENTALS OF STATISTICS**
**Course Code: U-FST-182**

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

---

**Learning Objectives**

- To acquaint students with various statistical methods.
- To cultivate statistical thinking among students.
- To prepare students for future courses having quantitative components.

**Course Outcome**

- Understand and appreciate descriptive statistics.
- Understand the concepts of probability and random variables.

---

**Syllabus**

**UNIT I:**
**1. INDTRODUCTION AND GRAPHICAL REPRESENTION**

1.1. Definitions of Statistics.

1.2. Importance of statistics.

1.3. Advantages and Limitations.

1.4. Scope of Statistics

1.5. Collection of Data

1.6. Types of Data

1.7. Attributes and variables

1.8. Construction of Frequency, Cumulative and Relative

1.9. Frequency distributions.

1.10. Graphical representation of Frequency distribution:

(Histogram, Frequency Polygon, Frequency Curve and Cumulative Frequency curves (Ogive curves))

1.11. Diagrammatic representations: Simple bar, Subdivided bar, Pie diagrams.

**UNIT II:**
**2. MEASURES OF CENTRAL TENDENCY**

2.1. Concept of central tendency

2.2. Arithmetic Mean: Definition, Formulae and computation for ungrouped and grouped data, Merits and Demerits.

2.3. Median: Definition, Formulae and Computation for ungrouped and grouped data, Merits and Demerits.

2.4. Quartiles: Definition, Formulae and Computation for ungrouped and grouped data.

2.5. Mode: Definition, Formulae and Computation for ungrouped and grouped data, Merits and Demerits.

**UNIT III:**

**3. MEASURES OF DISPERSION**

3.1. Concept of Dispersion.

3.2. Range: Definition, Formulae and Computation for ungrouped and grouped data.

3.3. Standard Deviation: Definition, Formulae and Computation for ungrouped and grouped data.

3.4. Variance: Definition, Formulae and Computation for ungrouped and grouped data.

3.5. Coefficient of variance: Definition, Formulae and Computation for ungrouped and grouped data

**UNIT IV:**

**4. CORRELATIONS AND TIME SERIES**

4.1. Definition of Correlation

4.2. Types of Correlation

4.3. Karl Pearson's coefficient of correlations for ungrouped data and problems

4.4. Definition and components of time series

4.5. Measures of trends

4.6. Moving average method and least square method and problems.

REFERENCE BOOKS:

1. Fundamentals of Statistics by A.M. Gun, M.K.Gupta and B. Dasgupta

2. Statistical Methods by S.P. Gupta.

3. Business Statistics by S. Shaha

4. Modern Elementary Statistics by J.E. Freund

5. Fundamentals of Statistics by S C Gupta.

6. Fundamentals of Applied Statistics by Gupta and Kapoor.

**U-LAC-183**
**Lab Course I(FIT)**

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

## Learning Objectives:

- Techniques for solving problems
- Basic computational concepts and elementary data structures
- The edit-compile-link-run cycle from a user point of view
- Testing strategies
- The main activities of software development and their interactions, and some of the major problems of software development

## Course Outcome:

The curriculum leading to a baccalaureate degree in Computer Science prepares students for positions as computer scientists in business, industry and government, or for graduate study in computer science. The curriculum's main objectives are to impart students with an understanding of the basics of computer science

---

**Proposed Practical List:**

1)Study of BOOTING Procedure of O.S.
2)Study of classifiction of computer.
3)Study of windows O.S. i)Desktop ii)Icon iii)Taskbar
4)Study of Input and Output Devices.
5)Study of creating File and Folder.
6)Study of moving,copingfile,and folder from one loction to another.
7)Study of MS-DOS Internal Commands
8) Study of MS-DOS Enternal Commands
9)Study of File related commands
10) Study of Directory related Commands.
11)Introduction of MS-Word
12)Perpare time-table in MS-Word
13)letter writing in MS-word.
14)Introduction of MS-Excel
15) Introduction of MS-Powerpoint
16)Creating E-MAIL

**U-LAC-184**
**Lab Course II(Prog. In C)**

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

**Learning Objectives**

- Learn writing algorithms
- drawing flowchart to solve given problem
- C-syntax, function, Operators, Array, File Handling etc.

**Course Outcome**

- Student should write Algorithm to solve given problem
- drawing flowchart to solve given problem
- able to convert algorithm to flowchart
- write program to print output on console
- writing complex program, manage file with c program etc.

---

**Proposed Practical List:**

1. Hello World Program
2. Program to perform Addition of two number
3. Program to take input from User
4. Program for if else statement
5. Program for nested if else
6. Program for else if ladder
7. Program for if else statement
8. Program for while loop
9. Program to demonstratefor loop
10. Program for Array
11. Program for Function
12. Program for Function Recursion
13. Program for Pointer
14. Program for Structure
15. Program for fopen() and fclose()

**U-LAC-185**
**Lab Course III(WPD)**

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

---

**Learning Objective:** The course has been designed to provide the basic knowledge for design of the web page / site.

**Course Outcomes:**

- A website is Published using application Server i.e. Tomcat/ IIS
- Apply Java script Code to sites.
- Develop Motion Graphic with Flash.
- Apply proper layout and interactive website design

---

**Proposed Practical List:**

1 Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.
2 Create your class timetable using table tag.
3 Create user Student feedback form (use textbox, text area , checkbox, radio button, select box etc.)
4 Create a web page using frame. Divide the page into two parts with Navigation links on left hand side of page (width=20%) and content page on right hand side
of page (width = 80%). On clicking the navigation Links corresponding content must be shown on the right hand side.
5 Write html code to develop a webpage having two frames that divide the webpage into two equal rows and then divide the row into equal columns fill
each frame with a different background color.
6 Create your resume using HTML tags also experiment with colors, text , link , size and also other tags you studied.
7 Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
8 Use Inline CSS to format your resume that you created.
9 Use External CSS to format your class timetable as you created.
10 Use External, Internal, and Inline CSS to format college web page that you created.
11 Develop a JavaScript to display today's date.
12 Develop simple calculator for addition, subtraction, multiplication and division operation using JavaScript

**U-LAC-186**
**Lab Course IV(Stat)**

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

**Learning Objectives**

- To acquaint students with various statistical methods.
- To cultivate statistical thinking among students.
- To prepare students for future courses having quantitative components.

**Course Outcome**

- Understand and appreciate descriptive statistics.
- Understand the concepts of probability and random variables.

**Proposed Practical List:**

1 Construction of histogram for given statistical data.

2 Construction of frequency polygon for given statistical data.

3 Construction of frequency curve for given statistical data.

4 Construction of ogive for given statistical data.

5 Construction of simple bar diagram for given statistical data.

6 Construction of subdivided bar diagram for given statistical data

7 Construction of pie diagram for given statistical data

8 To compute various measure of central tendency ( mean, median, mode)

9. To compute various measure of dispersion ( range, standard deviation, CV)

10 To compute coefficient of correlation

**Course Code: U-OPS-279**

**Course Title: Operating System**

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

**Credit:03**

---

**Learning objectives:**

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
- To know the components and management aspects of concurrency management
- To learn programmatically to implement simple OS mechanisms

**Course Outcomes**: Students will be able to:

- Analyze the structure of OS and basic architectural components involved in OS design
- Analyze and design the applications to run in parallel either using process or thread models of different OS
- Analyze the various device and resource management techniques for timesharing and distributed systems
- Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Interpret the mechanisms adopted for file sharing in distributed Applications
- Conceptualize the components involved in designing a contemporary OS

**SYLLABUS**

**UNIT I:**

**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

**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

**3. Features of MS-Windows**

3.1. GUI, Multitasking, Multi-user, network etc.

3.2. Important files of windows

**Unit II:**

**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:**

**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:**

**6. Device management**

6.1. Techniques for Device management

6.2. Devise management characteristics

6.3. Channels and control units

6.4. Device allocation consideration

**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. D.M Dhamdhere: Operating systems - A concept based Approach, 3rd Edition, Tata McGraw-Hill, 2012.
2. Avi Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, 9th Edition, John wiley & Sons, Inc. ISBN 978-1-118-06333-0, 2012

**Course Code: U-FDE-280**
**Course Title: Fundamentals of Digital Electronics**

**Total Teaching Hours:-60** **Marks-50**
**Credits:03**

---

**Learning Objective**

- To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
- To prepare students to perform the analysis and design of various digital electronic circuits.
- To impart to you a formalism of logic and enabling you to analyse logical processes.
- To enable you to implement simple logical operations using combinational logic circuits.
- To enable you to understand common forms of number representation in digital electronic circuits and to be able to convert between different representations.
- To enable you to understand the logical operation of simple arithmetic and other MSI circuits (Medium Scale Integrated Circuits).
- To impart to you the concepts of sequential circuits enabling you to analyse sequential systems in terms of state machines.
- To enable you to implement synchronous state machines using flip-flops.

**Course Outcome**

- To have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
- To understand and examine the structure of various number systems and its application in digital design.
- The ability to understand, analyze and design various combinational and sequential circuits.
- The Ability to identify basic requirements for a design application and propose a cost effective solution.
- The ability to identify and prevent various hazards and timing problems in a digital design.
- To develop skill to build, and troubleshoot digital circuits.

**Syllabus**

**UNIT I: Introduction to Number system and codes**

1.1 Logic levels and pukse waveforms
1.2 Different number systems and their conversions(decimal,binay,octal,hexadecimal)
1.3 Binary arithmetic addition,substraction,multiplication
1.4 Binary sustraction using 1's and 2's complement method, decimal substraction using 9's and 10's complement method
1.5 BCD numbers ,ASCII codes ,GRAY code

**UNIT II: Boolean algebra and gate networks**

2.1 Fundamental concepts of Boolean algebra
2.2Ssymbol, boolean equation, truth tables of different types of gate such as inverter or NOT gate, AND gate, OR gate, NAND gate, NOR gate, X-OR gate ,X-NOR gate
2.3 Basic laws of Boolean algebra and simplification of Boolean
2.4 Demorgans theorem 1st and 2nd
2.5 Universal property of NAND and NOR gate
2.6 Boolean expressions for gate networks for SOP and POS format
2.7 Karnaugh map for SOP and POS format with examples to find simplified Boolean equation

**UNIT III: Combinational logic circuit**

3.1 Half adder and FULL adder, parallel binary adders
3.2 HALF substractor and FULL substractor
3.3 Multiplexer and demultiplexer with types & examples
3.4 Encoder and decoder with types

**UNIT IV:Sequential logic circuit with Microprocessor**

4.1 Flip-flop-latches,edge triggered and level triggered flip flop with types SR flip flop ,D flip flop,JK flip flip and T flip-flop
4.2 Buffer registers , modes of operation of registers (SISO,SIPO,PISO,PIPO)
4.3 Asynchronous counters and types
4.4 Synchronous counters and types
4.5 Introduction to microprocessor
4.6 Basic components of a microprocessor

**Reference Books**

1. Floyd, Thomas L: " Digital Computer Fundamentals" , 3 rd Edition 1997.
2. Malvino, Pual Albert and Leach, Donald P: " Digital Computer Fundamentals", 3rd Edition, 1995.TMH.
3. Modern digital electronics by R.P.Jain
4. Bartee, Thomas C: " Digital Computr Fundamentals" 6 th Edition,1995.TMH

**Course Title: Advance 'C'**
**Course Code: U-ADC-281**

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

---

**Learning Objectives:**
**On comple**tion of the course, the student should **Knowledge** - know about:

- Techniques for solving problems
- Basic computational concepts and elementary data structures
- The edit-compile-link-run cycle from a user point of view
- Testing strategies
- The main activities of software development and their interactions, and some of the major problems of software development.
- Hand-execute simple programs, showing how input data is processed, output data is produced, and how the values of internal variables change
- Explain at various levels the behavior of fragments of programming language code

**Course Outcomes:**

- Identify the parts of the computer system.
- Adequately explain functioning of computer components.
- Explain the process of problem solving using computer
- Design an algorithmic solution for a given problem
- Write a maintainable C program for a given algorithm.
- Trace the given C program manually.
- Write C program for simple applications of real life using structures and files.
- Explain role of Operating system in computer system and applications of computer networks.

## SYLLABUS

**UNIT I: Pointer , Structure & Union**

**CHAPTER : 1. Pointer & Dynamic Memory Allocation**

1.1. Pointer to Pointer
1.2. Pointer & Functions
1.3. Pointer & Array
1.4. Array of Pointer
1.5. Dynamic Memory Allocation,
  1.5.1. Introduction to
    Calloc( ), Malloc( ), Free( )

**CHAPTER : 2. Structure & Union**

2.1 Introduction of structure,

2.2 Declaration & initialization.
2.3 Arrays of Structures, Structure within structure.
2.4 Introduction to Union.
2.5 Declaration & initialization.

**UNIT II: File Handling and Bit Operations**

**CHAPTER : 3. File Handling**

3.1 Why we need a file,
3.2 File operations(create, open, read, move , write, close),
3.3 File opening Mode,
3.4 Closing a file,
3.5 Input/output operations,
3.6 Creating and reading a file

**CHAPTER : 4. Bit Operations**

4.1 One's Compliment operator,
4.2 Right shift Operator,
4.3 Left Shift Operator,
4.4 Bit wise operator- AND, OR, XOR.

**UNIT III : Interaction with Hardware through 'C' and VDU Basic**

**CHAPTER : 5. Interaction with Hardware through 'C'**

5.1 Interrupt & Interrupt Vector table,
5.2 ROMBIOS Philosophy,
5.3 Invoking ROMBIOS function
5.4 int 86 ( ) function,
5.5 Interrupts to access ROMBIOS Services .

**CHAPTER : 6. VDU Basic**

6.1 Components of VDU, Monitors & Display Adaptors,
6.2 Video Display Modes:

- Text or Graphics,
- colors in text mode,
- colors in graphics mode,
- colors in SVGA, Video page,

6.3 Writing to video memory in text mode,
6.4 Video interrupt services:

- set video mode (0h),
- set cursor size(1h),
- set cursor position(2h)

**UNIT IV: Keyboard , Mouse programming and Introduction to C++**

7. **Keyboard And Mouse Programming**

7.1 Operation of keyboard ,Shift and toggle keys
7.2 keyboard interrupt(16 h)- get keyboard next character(0 h)
7.3 Mouse interrupt (33h),reset mouse (0h)
7.4 Show mouse pointer(1h), Hide mouse pointer(2h),
7.5 set mouse position and button status(3h).

**8. Introduction to C++**

8.1 Object oriented concepts, Features,
8.2 Advantages and Applications of OOPS
8.3 Data types, new operators and keywords, type conversion in C++
8.4 Classes & Objects

**REFRENCE BOOKS.**

1. Microprocessor System, the 8086/8088 family Architecture, programming & design By Chengliu A
2. Gibson.
3. Let us C by YeshwantKanetkar 3rd Edition
4. Pointers In C By YeshwantKanetkar 3rd Edition

**Course Code:- U-DSC-282**

**Course Title: DATA STRUCTURE**

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

**Credits:- 03**

---

**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

**Syllabus**

**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. Types of Data Structure

1.4. Array- definition.

1.5. Types-one, multi dimensional, character string array

**2. Algorithm analysis**

2.1. Algorithm – definition, characteristics

2.2. Space complexity, time complexity

**UNIT II:**

**3. Stack**

3.1. Definition of stack.

3.2. Operation on stack.

3.3. Declaration of stack.

**4. Queue**

4.1. Definition of queue.

4.2. Operations on queue.

4.3. Types of queue-Linear, Circular.

4.4. Applications of queues

**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 circular linked list, doubly linked list

**6. INTRODUCTION TO TREES**

6.1. Binary Trees

6.2. Expression Trees (Infix, Prefix, Postfix Traversals)

6.3. General Trees

6.4. Search Trees

6.5. Binary Search Trees

**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

**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

**U-LAC-283**
**Lab Course V(O.S)**

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

**Learning objectives:**

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
- To know the components and management aspects of concurrency management
- To learn programmatically to implement simple OS mechanisms

**Course Outcomes**: Students will be able to:

- Analyze the structure of OS and basic architectural components involved in OS design
- Analyze and design the applications to run in parallel either using process or thread models of different OS
- Analyze the various device and resource management techniques for timesharing and distributed systems
- Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Interpret the mechanisms adopted for file sharing in distributed Applications
- Conceptualize the components involved in designing a contemporary OS

**Proposed Practical List:**

1. Booting procedure of an Operating System.
2. Study of types of Operating System.
3. Text formatting in Word processor Package.
4. Creation of time table in open office.
5. Creation of resume.
6. Creation of marks memo in Excel.
7. Study of different formulas in Excel.
8. Creation of various charts in Excel.
9. Creation of Power Point Presentation.
10. Study of animation effects.
11. Introduction to Tally.
12. Creation of company in Tally.
13. Debit and Credit transaction for created company.

**U-LAC-284**
**Lab Course VI(D.E)**

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

---

**Learning Objective**

- To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
- To prepare students to perform the analysis and design of various digital electronic circuits.
- To impart to you a formalism of logic and enabling you to analyse logical processes.
- To enable you to implement simple logical operations using combinational logic circuits.
- To enable you to understand common forms of number representation in digital electronic circuits and to be able to convert between different representations.
- To enable you to understand the logical operation of simple arithmetic and other MSI circuits (Medium Scale Integrated Circuits).

**Course Outcome**

- To have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
- To understand and examine the structure of various number systems and its application in digital design.
- The ability to understand, analyze and design various combinational and sequential circuits.
- The Ability to identify basic requirements for a design application and propose a cost effective solution.
- The ability to identify and prevent various hazards and timing problems in a digital design.
- To develop skill to build, and troubleshoot digital circuits.

---

**Proposed Practical List:**

1. To perform and verify the truth tables of basic gates and derived gates.
2. To perform and verify the truth tables of EX-OR and EX-NOR gate.
3. To perform and verify LHS and RHS of Demorgan,s Theorem I & II.
4. To perform universal property of NAND gate.
5. To perform universal property of NOR gate.
6. To Implement and verify the truth table of any two Boolean equation.
7. To perform and verify the truth table of half adder.
8. To perform and verify the truth table of half Substractor.
9. To perform and verify the truth table of multiplexer.
10. To perform and verify the truth table of De-multiplexer.
11. To perform and verify the truth table of encoder
12. To perform and verify the truth table of Decoder

**U-LAC-285**
**Lab Course VII(Adv C)**

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

**Learning Objectives:**
**On comple**tion of the course, the student should **Knowledge** - know about:

- Techniques for solving problems
- Basic computational concepts and elementary data structures
- The edit-compile-link-run cycle from a user point of view
- Testing strategies
- Hand-execute simple programs, showing how input data is processed, output data is produced, and how the values of internal variables change
- Explain at various levels the behavior of fragments of programming language code

**Course Outcomes:**

- Identify the parts of the computer system.
- Adequately explain functioning of computer components.
- Explain the process of problem solving using computer
- Design an algorithmic solution for a given problem
- Write a maintainable C program for a given algorithm.
- Trace the given C program manually.
- Write C program for simple applications of real life using structures and files.
- Explain role of Operating system in computer system and applications of computer networks.

**Proposed Practical List:**

1. Program to demonstrate the concept of Pointer.
2. Program to demonstrate the concept of Pointer to Pointer.
3. Program to demonstrate the concept of command line argument.
4. Program to demonstrate the concept of structure.
5. Program to demonstrate the concept of Union..
6. Program to demonstrate the concept of file operations.
7. Program to demonstrate the concept of bitwise operations.
8. Program to demonstrate the concept of int86().
9. Program to set video goal.
10. Program to demonstrate the cursor size(1h).
11. Program to read the cursor position.
12. Program to demonstrate the concept of keyboard interrupt.
13. Program to demonstrate the concept of mouse interrupt.
14. Program for introduction of c++.
15. Program in c++ to demonstrate the concept of class and objects

**U-LAC-286**
**Lab CourseVIII(D.S.)**

**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.