

**Rajarshi Shahu Mahavidyalaya, Latur  
(Autonomous)**

**Department of Computer Science and Information Technology  
Syllabus for UG Course (B.Sc.C.S. F.Y.)  
(Under CBCS)**

**(With Effect from Academic Year:2017-18)**

**Name of the Programme: Bachelor of Computer Science (First Year Semester I + II)**

**Semester : I**

Code No	Course Name	Maximum Marks		Total Marks	Credits
		Theory / Practical	Internal		
U-COE-101	Communicative English – I	30	20	50	3
U-FCS-171	Fundamentals of Computer Science	30	20	50	3
U-DIE-172	Digital Electronics	30	20	50	3
U-PWC-173	Programming with C	30	20	50	3
U-INL-174	Introduction to Linux OS	30	20	50	3
U-LAC-175	Lab Course – I	50	-	50	2
U-LAC-176	Lab Course – II	50	-	50	2
U-LAC-177	Lab Course – III	50	-	50	2
U-LAC-178	Lab Course – IV	50	-	50	2

**Semester : II**

Code No	Course Name	Maximum Marks		Total Marks	Credits
		Theory / Practical	Internal		
U-COE-201	Communicative English – II	30	20	50	3
U-DIM-271	Discrete Mathematics	30	20	50	3
U-DBM-272	Database Management System	30	20	50	3
U-ADC-273	Advance 'C'	30	20	50	3
U-WET-274	Web Technology	30	20	50	3
U-LAC-275	Lab Course – V	50	-	50	2
U-LAC-276	Lab Course – VI	50	-	50	2
U-LAC-277	Lab Course – VII	50	-	50	2
U-LAC-278	Lab Course – VIII	50	-	50	2
U-MOE-235	Morel Education (NCBC)				

**Class:- B.Sc C.S F.Y. (Sem – I)**  
**(Under CBCS)**

**Course Title: - Fundamentals Of Computer Science**  
**Teaching Hours: 60**

**Course Code:- U-FCS-171**  
**Credits:03**

---

**Learning Objectives:**

To produce programmers equipped with an understanding of

- fundamental computational concepts underlying most programming languages
- a range of problem solving techniques using computers
- the role of programming within the overall software development process
- attitudes and working practices appropriate for a professional programmer and skills supporting
- the solution of small problems using a programming language
- the clear expression of solutions at different levels of abstraction
- Independent and self-motivated study in Computing Science.

**Course Outcomes:**

1. Identify the parts of the computer system.
2. Adequately explain functioning of computer components.
3. Explain the process of problem solving using computer

## **SYLLABUS**

### **UNIT I: Computer System & 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 & Memory**

#### **3. Input Output Devices**

- 3.1. Input Devices: Keyboard, Point & Draw Devices, Data Scanning devices, Digitizer, Electronic Card Reader, Voice Recognition devices
- 3.2. 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 & Introduction to Operating System**

#### **5. Computer Software**

- 5.1. Definition of software
- 5.2. Types of software
- 5.3. Compilers, Interpreters, Assemblers, Linkers, Loaders

#### **6. Introduction to Operating System**

- 6.1. Introduction
- 6.2. Main function of operating system
- 6.3. Files and directories
- 6.4. Types of OS

## **UNIT IV: Study of Operating systems**

### **7. Study of Operating systems**

- 7.1. Introduction to DOS
- 7.2. File and directory structure
- 7.3. Introduction to WINDOWS
- 7.4. Introduction to LINUX
- 7.5. Comparative study of operating systems

### **Reference Books:**

1. Fundamentals of computer science-P.K.SINHA.
- 2 Fundamentals of computer science-V RAJARAMAN.

**Class:- B.Sc C.S F.Y. (Sem – I)  
(Under CBCS)**

**Course Title: - Digital Electronics**  
**Teaching Hours: 60**

**Course Code:- U-DIE-172**  
**Credits:03**

---

**Course Objectives:**

- 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 Outcomes:**

- 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 pulse waveforms
- 1.2 Different number systems and their conversions(decimal, binary, octal, hexadecimal)
- 1.3 Binary arithmetic addition, subtraction, multiplication
- 1.4 Binary subtraction using 1's and 2's complement method, decimal subtraction 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
- 2.2 Symbol, 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 Demorgan's 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 subtractor and FULL subtractor
- 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 flop 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
-

**Class:- B.Sc C.S F.Y. (Sem – I)  
(Under CBCS)**

**Course Code:- U-PWC-173  
Course Title:- Programming with ‘C’**

**Total Teaching Hours: 60**

**Total Marks: 50  
Credit: 03**

---

**Learning Objectives:**

- To gain experience about structured programming
- To help students to understand the implementation of C language
- To understand various features in C
- To teach basic principles of programming
- To develop skills for writing programs using ‘C’

**Course Outcomes:**

- 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
- Solve the given problem using the syntactical structures of C language
- Develop , execute and document computerized solution for various problems using the features of C language
- To read and write C program that uses pointers, structures and files

**Detail Syllabus:**

**UNIT- I: Introduction and Overview.**

**CHAPTER 1. Introduction to 'C' Language**

- 1.1. Algorithm and flowchart
- 1.2. Introduction to C Programming language
- 1.3. Introduction to C
- 1.4. Structure of C program

**CHAPTER 2. Overview of 'C'**

- 2.1. Character set, C Tokens
- 2.2. Keywords

- 2.3. Identifiers
- 2.4. Variables
- 2.5. Constants
- 2.6. Data Types
- 2.7. Operators
- 2.8. Console based I/O and related built-in I/O functions: printf( ),scanf( ), getch()
- 2.9 Command line argument

## **UNIT- II: Control Structures , Arrays, Pointers**

### **CHAPTER 3. Control Structures**

- 3.1. Decision making structures: If, If-else, Nested If –else, Switch.
- 3.2. Loop Control structures: While, Do-while,for, Nested for loop
- 3.3. Other statements: break, continue, goto, exit

### **CHAPTER 4. Arrays and Pointers**

- 4.1 Introduction to array
- 4.2. Arrays: One Dimensional , Two Dimensional, Multidimensional arrays
- 4.3 Introduction to pointers
- 4.4 Accessing the Address of a variable
- 4.5 Declaring and initializing pointers
- 4.6 Accessing a variable through its pointers
- 4.7 pointer to pointer

## **UNIT- III: Functions**

### **.CHAPTER 5. Introduction to Function**

- 5.1. Introduction
- 5.2. Function Types
- 5.3. Category of functions
  - 5.3.1 No Arguments and no return values
  - 5.3.2 Arguments but no return values
  - 5.3.3 Arguments with return values
- 5.4 Call by value, call by reference
- 5.7. Recursion.

### **CHAPTER 6. String and Math Functions**

- 6.1 String manipulation function
- 6.2 strlen(), strcpy(),strcat(),strcmp(),strlwr(),strupr().
- 6.3 Math Function.

## **UNIT- IV: Structure, Union and File Management**

### **7. structure and union**

- 7.1. Introduction,Structure definition
- 7.2. Structure initialization
- 7.3. Arrays of structures
- 7.4. Structures within structures



7.5. Unions

**8. File management in C**

8.1. Defining and opening a file

8.2. closing file

8.3. I/O operations on files

8.4. Random access to files

8.5. The Preprocessor.

**REFERENCE BOOKS:**

1. Let us C-Yashwant Kanetkar.

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 – Yashwant Kanetkar

6. Byron Gottfried, - “Programming with C” (Schaum's Outline Series ) – Tata McGrawHill Publishing Company - 1998

**Class:- B.Sc C.S F.Y. (Sem – I)**  
**(Under CBCS)**

**Course Title: -Introduction to Linux O.S.**  
**Teaching Hours: 60**

**Course Code:- U-INL-174**  
**Credits:03**

---

**Learning Objectives:**

- Teach Basics of Linux Operating System
- Teach ownership and permissions of the files and directories.
- Explain why these issues exist.
- How to set permissions files/directories
- How to manipulate files/directories – list files, create, delete, and move just about anything on the file system
- Mention vi - a standard Unix text editor

**Course Outcomes:**

- Master functions, structures and history of operating systems
  - Master understanding of design issues associated with operating systems
  - Master various process management concepts including scheduling, synchronization, deadlocks
  - Master concepts of memory management including virtual memory
  - Master system resources sharing among the users
  - Master issues related to file system interface and implementation, disk management
  - Be familiar with different Linux commands used for different purposes.
  - Be familiar with various types of operating systems including Unix.
- 

## **SYLLABUS**

### **UNIT I: Introduction to Linux and File System**

#### **1. Introduction to Linux O.S.**

- 1.1. Operating System.
- 1.2. Types of Operating System
- 1.3. Functions of Operating System
- 1.4. History of Linux.
- 1.5. Advantages of Linux.
- 1.6. Flavors of Linux, Linux Loader, Linux kernel,

#### **2. Linux File System**

- 2.1. Linux file system

- 2.2. Directory structure of Linux
- 2.3. File System concept ext3, ext2.

## **UNIT II: Linux commands**

### **3. Linux commands**

- 3.1. General Purpose Utilities – - Man, who, login, logout, shutdown, passwd.
- 3.2. File management commands – cat, cp, rm, mv, wc,
- 3.3. Filter related commands – grep ,wc, head, tail, sort
- 3.4. Directory management commands – pwd, cd, mkdir, rmdir, ls
- 3.5. Checking directories and permissions – pwd, chmod, cd, ls
- 3.6. Compression commands zip ,unzip, gzip, gunzip,
- 3.7. printer related commands lpc ,lpr,lpq.lprm

## **UNIT III: VI Editor**

### **4. Working with permissions**

- 4.1. Assigning file permissions ,directory Permission
- 4.2. Using text editors
- 4.3. Editor ,use of vi
- 4.4. Features of vi
- 4.5. Vi basics,different modes and working with vi
- 4.6. Command mode -cursor movements(k,j,h,l),delete(character,line,word)

## **UNIT IV: RPM**

### **5. Working With RPM**

- 5.1. Performing system maintenance
- 5.2. Communication commands :- write, wall, talk, mesg, motd, Pre-login Message
- 5.3. Managing software with RPM :- Installing, Uninstalling, Upgrading  
Managing users Groups and passwords.

## **Reference Books:**

- 1 Bill Ball, David Pitts, “Red Hat Linux 7 Unleashed”, Techmedia SAMS Publication
- 2 EviNemeth, Garth Snyder, Scott Seebass, Trent R. Hein, “UNIX System Administration  
3 Handbook” Person Education Asia (LPE) (IIIrd Edition)
- 4 Red hat Linux & fedora unlashed Authors Bill Ball & Hoyt Dust.

**U-LAC-175**  
**Lab Course I(FIT)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives:**

To produce programmers equipped with an understanding of

- 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 Outcomes:**

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 classification 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, copying file, and folder from one location to another.
- 7) Study of MS-DOS Internal Commands
- 8) Study of MS-DOS External Commands
- 9) Study of File related commands
- 10) Study of Directory related Commands.
- 11) Introduction of MS-Word
- 12) Prepare time-table in MS-Word
- 13) Letter writing in MS-word.
- 14) Introduction of MS-Excel
- 15) Introduction of MS-Powerpoint

**U-LAC-176**  
**Lab Course II(DE)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives:**

1. Tell the history and development of digital electronics.
2. Identify and describe the six basic logic gates and combinational circuits in digital electronics.
3. Recognize the number systems use in digital logic design and its conversion.
4. Identify and describe flip-flop circuits, shift register circuits with types, counter circuit.

**Course Outcomes:**

1. Recognize the various codes such as ASCII,BCD,EBCDIC,GRAY to secure the information in communication
  2. Tell the basics of microprocessor with various block.
- 

**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 Subtractor.
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-177**  
**Lab Course III(Prog. With C)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives:**

- To gain experience about structured programming
- To help students to understand the implementation of C language
- To understand various features in C
- To teach basic principles of programming
- To develop skills for writing programs using 'C'

**Course Outcomes:**

- 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
  - Solve the given problem using the syntactical structures of C language
  - Develop , execute and document computerized solution for various problems using the features of C language
  - To read and write C program that uses pointers, structures and files
- 

**Proposed Practical List:**

1. Write a Structure of C Program.
2. Write a program to define console based input output functions.
3. Program to define arithmetic operators.
4. Write a program to define control and branching statements.
5. Write a program on Switch Case statement.
6. Write a program to define looping statements.
7. Write a program to define go-to , break and continue statement.
8. Write a program to accept n numbers and arrange them in ascending order using array.
9. Write a program to use of pointers.
10. Write a program to use of functions.
11. Write a program on String functions.
12. Write a program using math functions.
13. Write a program using Structure and Union.
14. Write a program on fprintf() and fscanf() function.
15. Write a program on fopen() and fclose().

**U-LAC-178**  
**Lab Course IV(Linux OS)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives:**

- Teach Basics of Linux Operating System
- Teach ownership and permissions of the files and directories.
- Explain why these issues exist.
- How to set permissions files/directories
- How to manipulate files/directories – list files, create, delete, and move just about anything on the file system
- Mention vi - a standard Unix text editor

**Course Outcomes:**

- Be familiar with different Linux commands used for different purposes.
  - Be familiar with various types of operating systems including Unix.
  - Student should be able to general porpose related commands.
  - He also should be able to execute the file, directory and he also should be also to able to assign the permissions to file or directory.
  - Student must be able to create ,modify and delete the user.
  - Student should be able to compress and decompress the file.
  - Student should execute the programs in vi - editor.
- 

**Proposed Practical List:**

- 1) Introduction to Linux O.S.
- 2) Study of General Puropse related commands.
- 3) Study of File related commands.
- 4) Study of Directory related commands.
- 5) Study of Directory checking related commands.
- 6) Study of chmod and ls command
- 7)Study of Filter related commands.
- 8)Study of Compression related commands.
- 9) Study of Printer related commands.
- 10) Study of Communication commands.
- 11) Study of Vi editor.
- 12) Program to display your own information.

**Course Code: U-DIM-271**  
**DISCRETE MATHEMATICS**  
**(Under CBCS)**

**Total Teaching Hours: 50**

**Total Marks: 50**  
**Credit: 3**

---

**Learning Objectives**

To extend student's mathematical maturity and ability to deal with abstraction and to introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems. To provide fundamental ideas on graph theory required for the study of Computer Science.

**Course Outcomes**

- At the end of the course, students would
- Have knowledge of the concepts needed to test the logic of a program.
- Understand basic notions of Graph Theory
- Knowing Fundamental Theorems in Graph Theory
- Study of algorithmic Graph Theory
- Have an understanding in identifying patterns on many levels.
- Be aware of a class of functions which transform a finite set into another finite set which relates to input output functions in computer science.

**SYLLABUS**

**UNIT I:**

**1. SETS, RELATIONS AND FUNCTIONS**

- 1.1. Definition and types of sets
- 1.2. Equal sets, subsets, universal sets, Venn diagram.
- 1.3. Set operations
- 1.4. Properties of set union and intersections
- 1.5. Cartesian product
- 1.6. Relation , types of relation
- 1.7. Function, domain, range, Types of function

**UNIT II:**

**2. MATHEMATICAL LOGIC**

- 2.1. Propositions
- 2.2. Truth values and truth table
- 2.3. Logical connectives and compound statements
- 2.4. Statement pattern and logical equivalence
- 2.5. Tautology, contradiction, contingency



**UNIT III:****3. MATRICES AND DETERMINANTS**

- 3.1. Definition of Determinant
- 3.2. Definition and types of matrices
- 3.3. Equality of Matrices and transpose of matrices
- 3.4. Algebra of matrices : addition, subtraction of matrices, scalar
- 3.5. Multiplication of matrix ·
- 3.6. Adjoint of matrices
- 3.7. Inverse of matrices

**UNIT IV:****4. GRAPH THEORY**

- 4.1. Definition and types of graphs
- 4.2. Incidences and degree of vertices
- 4.3. Isomorphism of graphs
- 4.4. Connected and disconnected graphs
- 4.5. Walks, paths and circuits
- 4.6. Directed graph
- 4.7. Tree
- 4.8. Centre of Tree
- 4.9. Binary Tree
- 4.10. Spanning tree
- 4.11. Cut sets and Cut vertices – Fundamental circuits and cut sets
- 4.12. Edge Connectivity - Vertex connectivity
- 4.13. Hamiltonian Paths & Graphs
- 4.14. operations on graphs

**REFERENCE BOOKS:**

1. Elements of Discreet Mathematics by C.L. Liu
2. Discreet Mathematics by Olympia nicodemi
3. Mathematical Structures for Computer Science by Alon Doerr and k. Levasieur
4. A first step in graph theory by raghunathan, Nimkar & Solapurkar
5. Graphs theory with applications to computer science by Narsing Deo
6. Basic Mathematics by Mittal and Agarwal
7. Tremblay and Manohar : “Discrete Mathematical Structures with Application to Computer Science” , McGraw Hill Book Company..

**Course Code: U-DBM-272**  
**Course Title:- Database Management System**  
**(Under CBCS)**

**Total Teaching Hours: 60**

**Total Marks: 50**  
**Credits: 3**

---

**Learning Objectives**

- To implement the design of the tables in DBMS
- To write queries to get optimized outputs
- To store, retrieve and view the contents
- To generate report based on customized need

**Course Outcomes**

- Upon successful completion of the course, students would be able to :
- Identify the information that is needed to design a database management system for a business information problem.
- Create conceptual and logical database designs for a business information problem.
- Construct a database management system that satisfies relational theory and provides users with business queries, business forms, and business reports.
- Analyze the core terms, concepts, and tools of relational database management systems.
- Demonstrate skills to work in teams in teams and utilize effective group techniques to manage a complex project.

**SYLLABUS**

**UNIT I:**

**1. Introduction to Database**

- 1.1. Definition of DBMS, File processing Vs DBMS
- 1.2. Advantages and disadvantages of DBMS
- 1.3. Users of DBMS
- 1.4. DBMS Structure

**2. Elements of DBMS**

- 2.1. DBMS Languages: DDL, DML, DCL
- 2.2. Terms: Entity, Entity set, attributes
- 2.3. Keys: Primary, secondary, foreign, composite

**UNIT II:**

**3. Data Models**

- 3.1. Introduction, Object based logical model,
- 3.2. record based logical model (RDB, NDB, HDB)
- 3.3. E-R model, E-R diagram

#### **4. Relational Algebra and Calculus**

- 4.1. Introduction
- 4.2. Relation, Schemes, Domain, Tuples
- 4.3. Cardinality degree
- 4.4. Algebraic operation
- 4.5. Fundamental operation: Select, product, union
- 4.6. Set difference : Natural join, Cartesian product, rename
- 4.7. Relational calculus: Tuple and domain relational calculus

#### **UNIT III:**

#### **5. Relational Database Design**

- 5.1. Normalization: 1NF, 2NF, 3NF, BCNF
- 5.2. Class diagrams and E-R tables
- 5.3. Functional dependency
- 6. **SQL**
- 6.1. Data types
- 6.2. Table Creation, Modify
- 6.3. Selecting, Deleting records
- 6.4. Simple queries
- 6.5. Oracle constraints

#### **UNIT IV:**

#### **7. Use of Operators**

- 7.1. Comparison operators
- 7.1.1. Between, In, Not In, Like, Null
- 7.2. Logical operators
- 7.2.1. AND, OR, NOT

#### **8. Advance in SQL**

- 8.1. SQL function
- 8.2. Joins (Self and equi)
- 8.3. Sub-queries
- 8.4. Views

#### **Reference Books:**

1. “Database System Concepts” by Abraham Silberschatz, Henry Korth, and S. Sudarshan
2. “Database Management Systems” by Raghu Ramakrishnan

**Course Title: Advance 'C'**  
**Course Code: U-ADC-273**  
**(Under CBCS)**

**Total Teaching Hours: 50**

**Total Marks: 50**  
**Credits: 3**

---

**Aim:**

Programming is about writing the instructions which a computer follows to enable it to store knowledge, process knowledge, and communicate knowledge with the outside world. Stemming from storing knowledge we can move into data structures and databases. Stemming from processing knowledge we can move into algorithms and computations. Stemming from communicating knowledge we can move into human-computer interaction and network issues. We can look in more detail at what is actually happening when a computer runs programs, considering how the instructions we write are translated to real changes in the electronic mechanisms of computer machinery.

**Learning Objectives:**

**On completion** 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

### **REFERENCE 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-WET-274**  
**Course Title: - Web Technology**  
**(Under CBCS)**

**Total Teaching Hours: 60**

**Total Marks: 50**  
**Credits: 3**

---

**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. Web Publishing:**

- 1.1. Phases of website development
- 1.2. Web browser
- 1.3. Cross browser testing
- 1.4. How to publish a website
- 1.5. Web server
- 1.6. WWW
- 1.7. URL

**2. HTML Document:**

- 2.1. Overview
- 2.2. Rules & Guidelines of HTML
- 2.3. Structure of HTML document

**3. The Markup Tags**

- 3.1. Basic HTML Tags
- 3.2. Physical style tags
- 3.3. Paragraphs
- 3.4. Lists
- 3.5. Font
- 3.6. HR
- 3.7. Heading levels
- 3.8. Center
- 3.9. Div, Span, Address

**UNIT-II**

**4. Linking**

- 4.1. Hyperlinks
- 4.2. Mailto anchor

## **5. Inserting images**

- 5.1. Image file formats
- 5.2. <img> tag with its attributes
- 5.3. Images as background
- 5.4. Internal and External images
- 5.5. Image map: server side and client side image maps
- 5.6. Image as hyperlink

## **6. Adding multimedia Elements**

- 6.1. Audio file formats
- 6.2. Adding audio in html document
- 6.3. Video file formats
- 6.4. Adding video in html document

### **UNIT III:**

## **7. Tables**

- 7.1. <table> tag with its attributes
- 7.2. Rowspan, colspan
- 7.3. Table element
- 7.4. Frames, Overview of frames
- 7.5. <frameset> tag and all attributes
- 7.6. Frame targeting
- 7.7. Floating frames
- 7.8. Forms, <form> tag with its attributes, Form controls
- 7.9. <input> tag with its attributes

### **UNIT IV:**

## **8. DHTML**

- 8.1. DOM
- 8.2. Moving objects in DHTML
- 8.3. CSS(All types)

## **9. Vbscript and Javascript**

- 9.1. Adding script to document
- 9.2. Working with local and global variable
- 9.3. Vbscript
  - Datatypes
  - Operators
  - Some inbuilt function
  - Control statements
  - Looping statements
- 9.4. Javascript
  - Datatypes
  - Operators
  - Control statements
  - Looping statements

## **10. Introduction to ASP**

- 10.1. Introduction to active server pages
- 10.2. Installation of IIS server



### 10.3. Basic ASP script

#### **Reference Books:**

1. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Black Book: HTML, Javascript, PHP, Java, Jsp, XML and Ajax, Black Book

**U-LAC-275**  
**Lab Course V(DM)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives**

To extend student's mathematical maturity and ability to deal with abstraction and to introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems. To provide fundamental ideas on graph theory required for the study of Computer Science.

**Course Outcome**

- At the end of the course, students would
  - Have knowledge of the concepts needed to test the logic of a program.
  - Understand basic notions of Graph Theory
  - Knowing Fundamental Theorems in Graph Theory
  - Study of algorithmic Graph Theory
  - Have an understanding in identifying patterns on many levels.
  - Be aware of a class of functions which transform a finite set into another finite set which relates to input output functions in computer science.
- 

**Proposed Practical List:**

1. Set and types of sets
2. Relation and types of relations
3. Function and types of functions.
4. Logical connectives and truth tables
5. Creating matrix
6. Types of matrices and algebra of matrices
7. Inverse and adjoint matrix
8. Graph theory
9. Types of graph theory
10. Tree

**U-LAC-276**  
**Lab Course VI(DBMS)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives**

- To implement the design of the tables in DBMS
- To write queries to get optimized outputs
- To store, retrieve and view the contents
- To generate report based on customized need

**Course Outcome**

- Upon successful completion of the course, students would be able to :
  - Identify the information that is needed to design a database management system for a business information problem.
  - Create conceptual and logical database designs for a business information problem.
  - Construct a database management system that satisfies relational theory and provides users with business queries, business forms, and business reports.
  - Analyze the core terms, concepts, and tools of relational database management systems.
  - Demonstrate skills to work in teams in teams and utilize effective group techniques to manage a complex project.
- 

**Proposed Practical List:**

1. Data Definition Language (DDL) commands in RDBMS
2. Data Manipulation Language (DML) and Data Control Language (DCL) commands in RDBMS.
3. High-level language extension with Cursors.
4. High level language extension with Triggers
5. Procedures and Functions.
6. Embedded SQL.
7. Database design using E-R model and Normalization.
8. Design and implementation of Payroll Processing System.
9. Design and implementation of Banking System.
10. Design and implementation of Library Information System.

**U-LAC-277**  
**Lab Course VII(Adv C)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

---

**Learning Objectives:**

On completion 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.
- 

**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-278**  
**Lab Course VIII(WT)**

**Practical Total Marks: 50**

**CREDIT: [ PR-2]**

**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