

Subject: Computer Science Major
DSC - I
FUNDAMENTALS OF INFORMATION TECHNOLOGY and PROGRAMMING IN 'C'
Course Code: UCS110T

Credits: 6 (4 Th, 2 Pr)

No. of Lectures: 60

No. of Practical: 60

Course Outline

Unit I

Evolution of Computers, Generations of Computer, Characteristics of a Computer, Computer Classification.

Basic Components of Digital Computers: Block Diagram. CPU: Functions of Each Unit: Primary Memory, ALU and CU, Instruction format. Bus: Data, Control and Address Bus. Number Systems: Binary, Octal, Decimal, Hexadecimal, Their Conversions, Binary Arithmetic. ASCII, BCD, EBCDIC.

Language Evolution: Generation of Languages: Machine, Assembly, High Level Languages. Characteristics of Good Language Translators: Compiler, Interpreter and Assembler. Source and object Program.

Unit II

Memory: Memory hierarchy, RAM, Static & dynamic RAM, Types of RAM (SDRAM, RDRAM, DDR), ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blu Ray and SSD.

Input Devices: Keyboard, Mouse, Light Pen, Touch Screen, Voice Input, MICR, OCR, OMR, Barcode Reader and Flatbed Scanner. Output Devices: VDU, Printers: Dot Matrix, Laser and Inkjet. Plotters: Drum, Flat-Bed and Inkjet.

Unit III

Computer Network: Goals and applications - Business Application, Home Application, Network terminology, Topologies: Linear, Circular, Tree and Mesh. Types of Networks: LAN, WAN, MAN. Repeaters, Bridge, Routers, Brouters and Gateway. Modem for Communication between PC's, Wi-Fi network, Introduction of Bluetooth and Infrared devices. Network protocols. Architecture: Peer-to-Peer, Client/Server. TCP/IP Reference Model with functionality of each layer.

Internet services (Introduction only): WWW – Web browser, URL, Internet search engines, WWW development languages, Electronic mail – E-mail address, e-mail message format, e-mail services (application based e-mail, webmail), how email works (client-server model), File Transfer Protocol – How FTP works (client-server model), Terminal network, Uses of Internet

Unit IV

Programming Languages and Tools, Machine language, Assembly language, High level languages. Problem Solving techniques: Development Tools: Algorithm,

Flowcharts and Pseudo code (Definition and its characteristics) Developing Algorithm and Drawing flowcharts.

C character set, Tokens, Identifiers, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions: Maths, character and string handling Functions.

Unit V

Control Structure: Compound statement, Selection statement: if, if-else, nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving).

Arrays: Need, Types: Single and Two Dimensional Array. Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions. Macro: single-line and multi-line. Storage Classes. (Special emphasis on problem solving)

Unit VI

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures. Union: Unions, Differences between Structure and Union

Pointer: Introduction, Address Operator (&), Pointer variables, void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

Graphics: Initializing graphics, Drawing basic shapes, Graphics Color system, Types of pens and brushes.

Minor - I
FUNDAMENTALS OF INFORMATION TECHNOLOGY and PROGRAMMING IN 'C'
Course Code: UCS111T

Credits: 6 (4 Th, 2 Pr)

No. of Lectures: 60

No. of Practical: 60

Course Outline

Unit I

Evolution of Computers, Generations of Computer, Characteristics of a Computer, Computer Classification.

Basic Components of Digital Computers: Block Diagram. CPU: Functions of Each Unit: Primary Memory, ALU and CU, Instruction format. Bus: Data, Control and Address Bus. Number Systems: Binary, Octal, Decimal, Hexadecimal, Their Conversions, Binary Arithmetic. ASCII, BCD, EBCDIC.

Language Evolution: Generation of Languages: Machine, Assembly, High Level Languages. Characteristics of Good Language Translators: Compiler, Interpreter and Assembler. Source and object Program.

Unit II

Memory: Memory hierarchy, RAM, Static & dynamic RAM, Types of RAM (SDRAM, RDRAM, DDR), ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blu Ray and SSD.

Input Devices: Keyboard, Mouse, Light Pen, Touch Screen, Voice Input, MICR, OCR, OMR, Barcode Reader and Flatbed Scanner. Output Devices: VDU, Printers: Dot Matrix, Laser and Inkjet. Plotters: Drum, Flat-Bed and Inkjet.

Unit III

Computer Network: Goals and applications - Business Application, Home Application, Network terminology, Topologies: Linear, Circular, Tree and Mesh. Types of Networks: LAN, WAN, MAN. Repeaters, Bridge, Routers, Brouters and Gateway. Modem for Communication between PC's, Wi-Fi network, Introduction of Bluetooth and Infrared devices. Network protocols. Architecture: Peer-to-Peer, Client/Server. TCP/IP Reference Model with functionality of each layer.

Internet services (Introduction only): WWW – Web browser, URL, Internet search engines, WWW development languages, Electronic mail – E-mail address, e-mail message format, e-mail services (application based e-mail, webmail), how email works (client-server model), File Transfer Protocol – How FTP works (client-server model), Terminal network, Uses of Internet

Unit IV

Programming Languages and Tools, Machine language, Assembly language, High level languages. Problem Solving techniques: Development Tools: Algorithm, Flowcharts and Pseudo code (Definition and its characteristics) Developing Algorithm and Drawing flowcharts.

C character set, Tokens, Identifiers, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions: Maths, character and string handling Functions.

Unit V

Control Structure: Compound statement, Selection statement: if, if-else, nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving).

Arrays: Need, Types: Single and Two Dimensional Array. Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions. Macro: single-line and multi-line. Storage Classes. (Special emphasis on problem solving)

Unit VI

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures. Union: Unions, Differences between Structure and Union

Pointer: Introduction, Address Operator (&), Pointer variables, void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

Graphics: Initializing graphics, Drawing basic shapes, Graphics Color system, Types of pens and brushes.

Syllabus for B.Sc. SEMESTER II
(To be implemented from Academic Session 2021-22)

Course Objectives	Course Outcomes
Students will try to learn:	After successful completion of the course student will be able to
1. To introduce the basics of Object Oriented Programming like classes, objects, abstraction, encapsulation and polymorphism, etc.	1. Learn to write programs using Object Oriented Programming concepts like class and object.
2. To understand the concepts of constructor and destructor; also familiar with operator overloading and dynamic objects.	2. Implementation of creation, deletion of objects and to make dynamic objects.
3. To get an idea of inheritance, virtual functions and exception handling.	3. Understand the implementation of inheritance, virtual functions and exception handling using programming.
4. To know about components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques used for data collection and data analysis.	4. Understand components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques.
5. To understand concepts associated with input and output design, form design, activities related to implementation and change strategies related to a computer system.	5. Know concepts associated with input and output design, form design, implementation activities and change strategies.
6. To understand the principles of testing strategies, conversion methods, planning and scheduling techniques, SCM, ISO, software re-engineering and software maintenance costs.	6. Get acquainted with the principles of testing strategies, conversion methods, planning and scheduling techniques, and advanced concepts.

**OBJECT ORIENTED PROGRAMMING USING ‘C++’ and
SYSTEM ANALYSIS AND DESIGN**

Course Code: UG-CS (05)-S2-T

Course Outline

Unit I

Object Oriented Methodology: Elements of Object Oriented programming, Objects, Classes, OOPs features.

Classes & Objects: Specifying a Class, Creating Objects, Accessing Class members, Defining member function, Outside Member Functions as inline, Accessing Member Functions within the class, Static data member, Access Specifiers: Private, Protected and Public Members.

Unit II

Constructors & Destructors: Introduction, Parameterized Constructors, Constructor Overloading, Constructors with Default Arguments, Copy Constructor, Dynamic Constructor, Destructor, Order of Construction and Destruction, Static data members with Constructor and Destructors, Virtual Destructor.

Operator Overloading: Definition, Overloadable Operators, Unary Operator Overloading, Binary overloading, Rules for Operator Overloading.

Dynamic Objects: Pointers to Objects, Creating and Deleting Dynamic Objects: New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, this Pointer.

Unit III

Inheritance: Definition, Abstract classes, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Constructor and Destructor in Derived Classes.

Virtual Functions: Need for Virtual Functions, definition, Pure Virtual Functions, Abstract Classes, Rules for Virtual Function. Runtime polymorphism.

Exception Handling: Exception Handling Model, List of Exceptions, catch with ellipses catch() Handling Uncaught Exceptions, Fault Tolerant Design Techniques, Memory Allocation Failure Exception, Rules for Handling Exception Successfully.

Unit IV

Introduction: System, Subsystems, Components of Computerized Information System, Systems Analysts, SDLC, Prototyping model and Spiral model.

Feasibility Study and Analysis: Identifying Problems, Organizing Feasibility Analysis: Economic, Financial, Organizational and Technological. Feasibility Decision, Choice of a solution.

Data Collection: Interviews, Brain Storming, Questionnaires, Document Search, Observation.

Structured tools and techniques of Data analysis: Structured English, Process Charts, SOP, Decision Tables and Decision Trees, Data Flow Diagram, Data Dictionary (Special emphasis on problem solving).

Unit V

System Design: Input design: Input Validation, Human factor Consideration, Messages, System Tolerance.

Output design: Categories of output, Design Principles, Control of Output. Forms: Principles of Form Design, Ways to ensure Quality Forms.

Codes: Types, Physical Representation of Codes, Principle of Code Design. Implementation: Training, Operational Training and Related Activities, Planning to Implement Change, Change Strategies.

Unit VI

Testing: Preparation for Testing, Test Execution: Levels of Testing, Component, Function, Subsystem, System, Test Evaluation, Acceptance.

System Evaluation Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Scheduling: Work Breakdown Structure, Activity Networks and CPM, Gantt Charts, PERT Charts, Project Monitoring and Control. Risk Management.

Software Configuration Management: Necessity, Configuring Management Activities Software Reliability and Quality Management: Software Reliability, Software Quality, ISO 9000. Software Maintenance: Characteristics of Software Maintenance, Maintenance Process Models, Estimation of Maintenance Cost. Software Reuse: What can be reused, Why no reuse so far, Basic Issues.

B. Sc. Semester II
COMPUTER SCIENCE PRACTICALS
Section A

1. Demonstrate the usage of Constructor and Destructor. Define a class data with data member acct_no, balance containing constructor data to initialize data member and a member function display () for output.
2. Program to demonstrate usage of a constructor and Destructor function. Declare a class with public data member count. The class containing one constructor and destructor to maintain updated information about active objects i.e. i) No of objects created. ii) No of objects Destroyed.
3. Program to accept the distance between city 1st & 2nd, city 2nd & 3rd. Calculate the distance between city 1st & 3rd. Define a class road with private data member km, m, d1, d2, d3 containing member function getdata () to accept values of d1, d2 and calculate for calculating distance.
4. Demonstrate the use of operators overloading (string manipulation: + for concatenation and relational operators for alphabetical comparison).
5. In a bank N depositor deposit the amount. Write a program to find total amount deposited in the bank. Declare a class deposit with private data member Rupee and Paise containing member function getdata(), putdata(). Use array of objects, use operator '+' for overloading.
6. Declare class event and accept time of first event and second event and find the difference between 1st and 2nd event containing public member function getdata() and display with private data member hour, minute, second and total. Use operator '-' for overloading.
7. Program to demonstrate Single Inheritance. Declare a class B and derive publically class D from B.
 - a. The class B contains private data member a, public data member b with member function get_ab(), get_a(), show_a().
 - b. The derived class D contains data member c with member function mul() and display().
8. Program to demonstrate Multiple Inheritances. Declare class M and N and derive publically class P from M and N.
 - a. Declare a class M with protected data member m and public member function get_m().
 - b. Declare a class N with protected data member n containing member function get_n().
 - c. Declare class P containing member function display ()

9. Program to demonstrate Multilevel Inheritance. Declare a class student and derive publically a class test and derive publically class result from class test.
 - a. The class student contains protected data member roll_number with public member functions get_number() and put_number().
 - b. The class test containing protected data member sub1, sub2 with public member function get_marks() and put_marks().
 - c. The class result contains data member total and public member function display ().
10. Program to demonstrate Hierarchical Inheritance. Declare a class Side and derive publically class Square from base class Side and also derive publically class Cube from base class Side.
 - a. Class Side contains protected data member L with a member function set_values().
 - b. Class Square contains member function sq().
 - c. Class Cube contains member function cub().
11. Program to demonstrate usage of normal virtual function and pure virtual function with abstract class.
12. Program to determine whether the input is +ve or –ve through exception.
13. Program to raise exception if an attempt is made to perform divide by zero.
14. Program to accept customer account information like Customer Name, Account Number, Account Type. Implement the following operations using inheritance:
 - a. Deposit
 - b. Withdraw
 - c. Display Balance
 - d. Display Account details.

Section B

Minimum five study experiments based on unit IV through Unit VI

REFERENCE BOOKS:

1. Mastering C++ by K R Venugopal Tata McGraw-Hill, New Delhi.
2. The C++ Programming Language –Bjarne Stroustrup
3. Programming with C++ - Ravichandran
4. Programming with C++ - Robert Lafore

5. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill
6. Information Systems Analysis, Design and Implementation By K. M. Hussain, Donna Hussain [Tata McGraw-Hill Publishing Company Ltd, New Delhi]
7. Fundamentals of Software Engineering by Rajib Mall [PHI Publication]
8. Workbook on Systems Analysis & Design by V. Garg [PHI Publication]
9. System Analysis and Design- Don Yeates, Shiebls, Helmy (M).
10. System Analysis & Design - Edward –TMH
11. System Analysis and Design – Satzinger, Robert Jackson and Stephen Burd, Thomson Learning

Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. www.tutorialspoint.com/cplusplus
2. www.cplusplus.com/doc/tutorial
3. www.cprogramming.com/tutorial
4. www.subjectcoach.com/tutorials
5. www.tutorialspoint.com/sdlc