14. COMPUTER SCIENCE (Code No. 083)
(2017-18)

Learning Objectives:

- To understand basics of computers.
- To develop logic for Problem Solving.
- To develop problem solving skills and their implementation using C++.
- To understand and implement the concept of Object Oriented Methodology.
- To understand the concept of working with Relational Database.
- To understand the basic concept of Computing Logic.
- To understand the basic concepts of Communication and Networking technologies.
- To understand Open Source concepts.

Class XI (Theory) - C++

Duration: 3 hours **Total Marks: 70**

Unit No.	Unit Name	Marks
1.	COMPUTER FUNDAMENTALS	10
2.	PROGRAMMING METHODOLOGY	12
3.	INTRODUCTION TO C++	14
4.	PROGRAMMING IN C++	34
	Total	**70**

Unit 1: Computer Fundamentals (18 Theory + 6 Practical) Periods

Classification of computers: Basics of computer and its operation; Functional Components and their interconnections, concept of Booting.

Software concepts: Types of Software – System Software, Utility Software and Application Software

System Software: Operating System, Compiler, Interpreter and Assembler;

Operating System: Need for Operating System, Functions of Operating System (Processor Management, Memory Management, File Management and Device Management), Types of Operating System-Interactive (GUI based), Time Sharing, Real Time and Distributed, Commonly used Operating System: UNIX, LINUX, Windows, Solaris, BOSS (Bharat Operating System Solutions); Mobile OS – Android, Symbian, IOS.

Utility Software: Anti Virus, File Management tools, Compression tools and Disk Management tools (Disk Cleanup, Disk Defragmenter, Backup).

Open Source Concepts: Open Source Software, Freeware, Shareware, and Proprietary Software.

Application Software: Office Tools – Word Processor, Presentation Tool, Spreadsheet Package, Database Management System; Domain Specific tools – School Management System, Inventory

Management System, Payroll System, Financial Accounting, Hotel Management, Reservation System and Weather Forecasting System.

Number System: Binary, Octal, Decimal, Hexadecimal and conversion between different number systems.

Internal Storage encoding of Characters: ASCII, ISCII (Indian Scripts Standard Code for Information Interchange), and UNICODE (for multilingual computing)

Microprocessor: Basic concepts, Clock speed (MHz, GHz), 16 bit, 32 bit, 64 bit, 128 bit processors; Types – CISC Processors (Complex Instruction Set Computing), RISC Processors (Reduced Instruction Set Computing), and EPIC (Explicitly Parallel Instruction Computing).

Memory Concepts: Units: Byte, Kilo Byte, Mega Byte, Giga Byte, Tera Byte, Peta Byte, Exa Byte, Zetta Byte, Yotta Byte.

Primary Memory: Cache, RAM, ROM

Secondary Memory: Fixed and Removable storage – Hard Disk Drive, CD/DVD Drive, Pen Drive, Blue Ray Disk.

Input Output Ports/ Connections: Serial, Parallel and Universal Serial Bus, PS-2 port, Infrared port, Bluetooth, Firewire.

Unit 2: Programming Methodology (28 Theory + 10 Practical) Periods

General Concepts: Modular Approach, Clarity and Simplicity of Expressions, Use of proper names for Identifiers, Comments, Indentation; Documentation and Program Maintenance; Running and Debugging programs, Syntax Errors, Run-Time Errors, Logical Errors

Problem Solving Methodologies: Understanding of the problem, Solution for the problem, Identifying minimum number of inputs required for output, Writing code to optimizing execution time and memory storage, step by step solution for the problem, breaking down solution into simple steps (modular approach), Identification of arithmetic and logical operations required for solution; Control Structure- Conditional control and looping (finite and infinite).

Problem Solving: Introduction to Algorithms/Flowcharts.

Unit-3: Introduction to C++ (44 Theory + 36 Practical) Periods

Getting Started: C++ character set, C++ Tokens (Identifiers, Keywords, Constants, Operators,), Structure of a C++ Program (include files, main function), Header files – iostream.h, iomanip.h, **cout, cin;** use of I/O operators (<<and>>), Use of endl and setw ( ), Cascading of I/O operators, compilation , Error Messages; Use of editor, basic commands of editor, compilation, linking and execution.

Data Types, Variables and Constants: Concept of Data types; Built-in Data types: **char, int , float** and **double;** Constants: Integer Constants, Character constants (- \n, \t, \b), Floating Point Constants, String Constants; Access modifier: **const;** Variables of built-in-data types, Declaration/Initialization of variables, Assignment statement, Type modifier: **signed, unsigned, long**

Operator and Expressions: Operators: Arithmetic operators (-,+,*,/,%), Assignment operator (=), c++ shorthands (+=, -=,*=,/=,%=) Unary operators (-), Increment (++) and Decrement (--) Operators, Relational operator (>,>=,<=,=,!=), Logical operators (!,&&,||), Conditional operator: <condition>?<if—true>:<if false>; Precedence of Operators; Automatic type conversion in expressions, Type casting;

UNIT 4: PROGRAMMING IN C++ (50 Theory + 48 Practical) Periods

Flow of control

Conditional statements: if else, Nested **if, switch..case..default,** use of conditional operator, Nested **switch..case, break** statement (to be used in **switch..case only); Loops**: **while, do - while, for** and Nested loops

Inbuilt Functions

Header file Categorization	Header File	Function
Standard input/output functions	**stdio.h**	**gets ( ), puts ( )**
Character Functions	**ctype.h**	**isalnum ( ), isalpha ( ), isdigit ( ), islower ( ), isupper ( ), tolower ( ), toupper ( )**
String Function	**string.h**	**strcpy ( ), strcat ( ), strlen ( ), strcmp ( ), strcmpi ( ), strrev ( ), strupr ( ), strlwr ( )**
Mathematical Functions	**math.h**	**fabs ( ), pow ( ), sqrt ( ), sin ( ), cos ( ), abs ( )**

Introduction to user-defined function and its requirements.

Defining a function; function prototype, Invoking/calling a function, passing arguments to function, specifying argument data types, default argument, constant argument, call by value, call by reference, returning values from a function, calling functions with arrays, scope rules of variables: local and global variables.

Relating to Parameters and return type concepts in built-in functions.

Structured Data Type

Arrays: Introduction to Array and its advantages.

One Dimensional Array: Declaration/initialization of One-dimensional array, Accepting array elements, accessing array elements, manipulation of array elements (sum of elements, product of elements, average of elements, linear search, finding maximum/minimum value)

Declaration / Initialization of a String, string manipulations (counting vowels/ consonants/ digits/ special characters, case conversion, reversing a string, reversing each word of a string)

Two-dimensional Array: Declaration/initialization of a two-dimensional array, inputting array elements, accessing array elements, manipulation of Array elements (sum of row element, column elements, diagonal elements, finding maximum / minimum values)

User-defined Data Types: Introduction to user defined data types.

Structure: Defining a Structure (Keyword Structure), declaring structure variables, accessing structure elements, passing structure to functions as value and reference, argument/parameter, function returning structure, array of structure, passing an array of structure as an argument/ a parameter to a function.

Defining a symbol name using **typedef** keyword and defining a macro using **#define preprocessor** directive.

Class XI (Practical) – C++

Duration: 3 hours **Total Marks: 30**

1. **Programming in C++** **10**

   One programming problem in C++ to be developed and tested on Computer during the examination. Marks are allotted on the basis of following:

Logic	:	6 Marks
Documentation	:	2 Marks
Output presentation	:	2 Marks

2. **Project Work** **06 + 4***

   Problems using String, Number, array and structure manipulation

   General Guidelines: Initial Requirement, developing an interface for user (it is advised to use text based interface screen), developing logic for playing the game and developing logic for scoring points

   - Memory game: A number guessing game with application of 2 dimensional arrays containing randomly generated numbers in pairs hidden inside boxes.
   - Hollywood/Hangman: A word Guessing game
   - Cows 'N Bulls: A word/number Guessing game
   - Random Number Guessing Game (High\Low)
   - A game to check whether a word does not use any of the forbidden letters
   - Cross'N knots game: A regular tic-tac –toe game.

   or

   Similar projects may be undertaken in other domains. (As mentioned in general guidelines for project, given at the end of the curriculum in a group of 2-4 students)

* Collaboration and Presentation of the project

3. **Practical File** **5+1***

   (a) Record of the configuration of computer system used by the student in the computer lab (by exploring inside computer system in the first 2 lab classes).

   (b) Must have minimum 20 programs from the topics covered in class XI course.

   - Programs on Control structures
   - Programs on String manipulations
   - Programs on array manipulations(1D & 2D)
   - Programs on structures.

     ***1 mark is for innovating while developing programmes.**

4. **Viva Voce** **04**

   Viva will be asked from the syllabus covered in class XI and the project developed by the student(s).
   *1 mark is for innovating while developing programme.

COMPUTER SCIENCE

Class XII (Theory) - Python

Duration: 3 Hours **Total Marks: 70**

Unit No.	Unit Name	Marks
1	Object Oriented Programming with Python	24
2	Advanced Programming with Python	20
3	Database Management System and SQL	8
4	Boolean Algebra	8
5	Communication Technologies	10
Total		**70**

Unit 1: Object Oriented Programming with Python (50 theory + 40 practical) periods

REVIEW: Python covered In Class-XI

Concept of Object Oriented Programming: Data Hiding, Data Encapsulation, Class and Object, Polymorphism, inheritance, advantages of Object Oriented Programming over earlier programming methodologies,

Classes: Defining classes (attributes, methods), creating instance objects, accessing attributes and methods, using built in class attributes (dict, doc, name, module, bases), Constructor(_ _init()_ _, _ _del()_ _ and _ _ str()_ _) methods in a class, private attributes (limited support), importance of **"self"** (acts as a pointer to current calling object) operator overloading with methods.

Inheritance:

Concept of base class and derived class: Single, multilevel and multiple inheritance- Overriding methods, using super( ) in derived class to invoke _ _init()_ _ or overridden methods of parent class.

Unit 2: Advance Programming with Python (42 Theory + 36 Practical) Periods

Linear List Manipulation: Sequential allocation, traversal,insertion of an element in a sorted list, deletion of an element from the list, searching (linear, binary), sorting (insertion, selection, bubble).

Stacks (List Implementation): Introduction to stack (LIFO Operations), operations on stack (PUSH and POP) and its implementation in python.

Converting expression from infix to postfix notation and evaluation of postfix expression.

Queues (List implementation) – Introduction to Queue(FIFO) , Operations on Queue (INSERT and DELETE) and its implementation in Python.

Data File

Need for data file, types of data file-text and binary,opening and closing files-open( ), close( ), access modes (output, input, default), file object, access_modes, reading and writing a file read(), readline(), readlines(), write(), writelines file positions (seek(), tell()), renaming and deleting a file.,flush();

Implementation of basic file operations on text and binary file in Python: Creating/writing data into file, reading and displaying data from file, searching for particular data from a file, insertion and deletion of data from an already existing file, modification of data in file.

Error and Exceptions – NameError, IndexError, TypeError, IO Error, ImportError, ValueError, EOFError.

Generator function using Yield.

Unit 3: Databases and SQL (20 Theory + 20 Practical) Periods

Data base Concepts: Introduction to data base concepts and its need.

Relational data model: Concept of domain, tuple, relation, key, primary key, alternate key, candidate key;

Relational algebra : Selection, Projection, Union and Cartesian product;

Structured Query Language:

General Concepts: Advantages of using SQL, Data Definition Language and Data Manipulation Language;

Data Types: NUMBER/DECIMAL, CHARACTER/VARCHAR/VARCHAR2, DATE;

SQL COMMANDS: CREATE TABLE, DROP TABLE, ALTER TABLE, UPDATE ….SET…., INSERT, DELETE; SELECT, DISTINCT, FROM, WHERE, IN, BETWEEN, GROUP BY, HAVING, ORDER BY;

SQL functions: SUM ( ), AVG ( ), COUNT ( ), MAX ( ) AND MIN ( ); Obtaining results (SELECT query) from 2 tables using equi-join, Cartesian product and Union

Note: Implementation of the above mentioned commands could be done on any SQL supported software on one or two tables.

Unit 4: Boolean Algebra (16 Theory + 0 Practical) Periods

Role of Logical Operations in Computing.

Binary-valued Quantities, Boolean Variable, Boolean Constant and Boolean Operators: AND, OR, NOT; Truth Tables; Closure Property, Commutative Law, Associative Law, Identity law, Inverse Law, Principle of Duality, Idempotent Law, Distributive Law, Absorption Law, Involution Law, DeMorgan's Law and their applications;

Obtaining Sum of Product (SOP) and Product of Sum (POS) form the Truth Table, Reducing Boolean Expression (SOP and POS) to its minimal form, Use of Karnaugh Map for minimization of Boolean expressions (up to 4 variables);

Application of Boolean Logic: Digital electronic circuit design using basic Logic Gates (NOT, AND, OR, NAND, NOR)

Use of Boolean operators (NOT, AND, OR) in SQL SELECT statements

Use of Boolean operators (AND, OR) in search engine queries.

Unit 5: Communication Technologies (16 Theory + 3 Practical) Periods

Evolution of Networking: ARPANET, Internet, Interspace Different ways of sending data across the network with reference to switching techniques (Circuit and Packet switching).

Data Communication terminologies: Concept of Channel, Bandwidth (Hz, KHz, MHz) and Data transfer rate (bps, Kbps, Mbps, Gbps, Tbps).

Transmission media: Twisted pair cable, coaxial cable, optical fiber, infrared, radio link, microwave link and satellite link.

Network devices: Modem, RJ45 connector, Ethernet Card, Router, Switch, Gateway, wifi card.

Network Topologies and types: Bus, Star, Tree, PAN, LAN, WAN, MAN.

Network Protocol: TCP/IP, File Transfer Protocol (FTP), PPP, SMTP, POP3 Remote Login (Talent), and Internet Wireless/Mobile Communication protocol such as GSM, CDMA, GPRS, and WLL.

Mobile Telecommunication Technologies : 1G, 2G, 3G and 4G; Mobile processors;

Electronic mail protocols such as SMTP, POP3

Protocols for Chat and Video Conferencing VOIP

Wireless technologies such as Wi-Fi and WiMax

Network Security Concepts:

Threats and prevention from Viruses, Worms, Trojan horse, Spams

Use of Cookies, Protection using Firewall, https;

India IT Act, Cyber Law, Cyber Crimes, IPR issues, hacking.

Introduction To Web services: WWW, Hyper Text Markup Language (HTML), Extensible Markup Language (XML); Hyper Text Transfer Protocol (HTTP); Domain Names; URL; Website, Web browser, Web Servers; Web Hosting, Web Scripting – Client side (VB Script, Java Script, PHP) and Server side (ASP, JSP, PHP), Web 2.0 (for social networking)

E-commerce payment transactions using online banking, mobile banking and payment apps and services.

Class XII (Practical)
Python

Duration: 3 hours **Total Marks: 30**

1. Programming in Python **10**

One programming problem in Python to be developed and tested in Computer Lab during the Examination. Marks are allotted on the basis of following:

Logic	:	6 Marks
Documentation/Indentation	:	2 Marks
Output presentation	:	2 Marks

Note: The types of problems to be given will be of application type developed using classes from the following topics

- String
- Lists/Tuples
- Dictionary
- File Operations

2. **SQL Commands** **05**

   Five Query questions based on a particular Table/Relation to be tested practically on Computer during the examination. The command along with the result must be written in the answer sheet.

3. **Project Work** **05**

   The project has to be developed in Python language with Object Oriented Technology.

   (The project is required to be developed in a group of 2-4 students)

   - Presentation on the computer
   - Project report (Listing, Sample, Outputs, Documentation)
   - Viva

4. **Practical File** **06**

   Must have minimum 20 programs from the following topics

   - Lists (general, stack, queue)
   - File Handling
   - Programs based on Object Oriented Concepts
   - 15 SQL commands along with the output based on any table/relation

     ***1 mark is for innovating while developing programme.**

5. **Viva Voce** **04**

   Viva will be asked from syllabus covered in class XII and the project developed by student.

GUIDELINES FOR PROJECTS (Class XI and XII)

1. **Preamble**

   1.1 The academic course in Computer Science includes one Project in each year. The Purpose behind this is to consolidate the concepts and practices imparted during the course and to serve as a record of competence.

   1.2 A group of 2-4 students as team may be allowed to work on one project.

2. **Project content**

   2.1 Project for class XI can be selected from the topics mentioned in the syllabus or domains on similar lines.

   2.2 Project for class XII should ensure the coverage of following areas of curriculum:

   a) Problem Solving

   b) Object Oriented Programming in Python

   c) File Handling

Theme of the project can be

- Any Scientific or a fairly complex algorithmic situation
- Quizzes/Games
- Tutor/Computer Aided Learning Systems

2.3 The aim of the project is to highlight the abilities of algorithmic formulation, modular programming, systematic documentation and other associated aspects of Software Development.

2.4 The assessment would be through the project demonstration and the Project Report, which should portray Programming Style, Structured Design, Good documentation of the code to ensure readability and ease of maintenance.

Class XII (Theory) – C++

Duration: 3 hours **Total Marks: 70**

Unit No.	Unit Name	Marks
1.	OBJECT ORIENTED PROGRAMMING IN C++	30
2.	DATA STRUCTURE	14
3.	DATABASE MANAGEMENT SYSTEM AND SQL	8
4.	BOOLEAN ALGEBRA	8
5.	COMMUNICATION TECHNOLOGIES	10
Total		70

Unit 1: Object Oriented Programming in C++ (50 Theory + 40 Practical) Periods

REVIEW: C++ covered In Class – XI,

Object Oriented Programming: Concept of Object Oriented Programming – Data hiding, Data encapsulation, Class and Object, Abstract class and Concrete class, Polymorphism (Implementation of polymorphism using Function overloading as an example in C++); Inheritance, Advantages of Object Oriented Programming over earlier programming methodologies,

Implementation of Object Oriented Programming concepts in C++: Definition of a class, Member of a class – Data Members and Member Functions (methods), Using Private and Public visibility modes, default visibility mode (private); Member function definition: inside class definition and outside class definition using scope resolution operator (::); accessing members from object (s), Objects as function arguments–pass by value and pass by reference;

Constructor and Destructor: Constructor: special characteristics, declaration and definition of a constructor, default constructor, overloaded constructors, copy constructor, constructor with default arguments;

Destructor: Special Characteristics, declaration and definition of destructor;

Inheritance (Extending Classes): Concept of Inheritances, Base Class, Derived classes, protected visibility mode; Single level inheritance, Multilevel inheritance and Multiple inheritance, Privately derived, publicly derived and Protectedly derived class, accessibility of members from objects and within derived class (es);

Data File Handling: Need for a data file, Types of data files – Text file and Binary file;

Text File: Basic file operations on text file**:** Creating/Writing text into file, Reading and Manipulation of text from an already existing text File (accessing sequentially).

Binary File: Creation of file, Writing data into file, Searching for required data from file, Appending data to a file, Insertion of data in sorted file, Deletion of data from file, Modification of data in a file;

Implementation of above mentioned data file handling in C++;

Components of C++ to be used with file handling:

Header file: fstream.h; ifstream, ofstream, classes;

Opening a text file in—in, out, and app modes;

Using cascading operators (>>,<<) for writing text to the file and reading text from the file; open (), get (), read (), put (), write(), getline() and close() functions; Detecting end-of-file (with or without using eof() function), tellg(), tellp(), seekg(), seekp();

Pointers:

Introduction to Pointer, Declaration and Initialization of Pointer; Dynamic memory allocation/de-allocation operators: **new, delete;** Pointers and Arrays: Array of Pointers, Pointer to an array (1 dimensional array), Function returning a pointer, Reference variables and use of alias; Function call by reference. Pointer to structure: De-reference/Deference operator: *, ->; self referential structure;

Unit 2: Data Structures (42 Theory + 36 Practical) Periods

Introduction to data structure- array, stack queues primitive and non-primitive data structure, linear and non-linear structure, static and dynamic data structure.

Arrays:

One and two Dimensional arrays: Sequential allocation and address calculation;

One dimensional array: Traversal, Searching (Linear, Binary Search), Insertion of an element in an array, deletion of an element from an array, Sorting (Insertion, Selection, Bubble)

Two-dimensional arrays: Traversal Finding sum/difference of two NxM arrays containing numeric values, Interchanging Row and Column elements in a two dimensional array;

Stack (Array and Linked implementation of Stack):

Introduction to stack (LIFO: Last in First out Operations)

Operations on stack (PUSH and POP) and its Implementation in C++, Converting expressions from INFIX to POSTFIX notation and evaluation of Postfix expression;

Queue: (Array and Linked Implementation)

Introduction to Queue (FIFO: First in First out operations)

Operations on Queue (Insert and Delete and its Implementation in C++, circular queue using array.

Unit 3: Database Management Systema and SQL

Common to both the options. Refer to unit 3 DATABASE AND SQL mentioned in case of Python for further details.

Unit 4: Boolean Algebra

Common to both the options. Refer to unit 4 mentioned in case of Python for further details.

Unit 5: Networking and Open Source Software

Common to both the options. Refer to unit 5 COMMUNICATION_TECHNOLOGIES mentioned in case of Python for further details.

Class XII (Practical) - C++

Duration: 3 Hours **Total Marks : 30**

1. Programming in C++ 10

One programming problem in C++ to be developed and tested in Computer during the examination. Marks are allotted on the basis of following:

Logic	:	6 Marks
Documentation/Indentation	:	2 Marks
Output presentation	:	2 Marks

Notes: The types of problem to be given will be of application type from the following topics

- Arrays (One dimensional and two dimensional)
- Class(es) and objects
- Stack using arrays and or linked implementation
- Queue using arrays (circular) and or linked implementation
- Binary File operations (Creation, Displaying, Searching and modification)
- Text File operations (Creation, Displaying and modification)

2. SQL Commands 05

Five Query questions based on a particular Table / Relation to be tested practically on Computer during the examination. The command along with the result must be written in the answer sheet.

3. Project Work 05

The project has to be developed in C++ language with Object Oriented Technology and also should have use of Data files. (The project is required to be developed in a group of 2-4 students)

- Presentation on the computer
- Project report (Listing, Sample Outputs, Documentations)
- Viva
  * 1 mark is for innovation while writing programme.

4. Practical File 06

Must have minimum 20 programs from the following topics

- Arrays (One dimensional and two dimensional, sorting, searching, merging, deletion & insertion of elements)
- Class(es) and objects
- Stacks using arrays and linked implementation
- Queue using arrays & linked implementation (circular also).
- File (Binary and Text) operations (Creation, Updation, Query)
- Any computational Based problems
- 15 SQL commands along with the output based on any table/relation:

5. **Viva Voce** **04**

Viva will be asked from syllabus covered in class XII and the project developed by student.

Guidelines for Projects (Class XI and XII)

1. **Preamble**

1.1 The academic course in Computer Science includes on Project in each year. The Purpose behind this is to consolidate the concepts and practices imparted during the course and to serve as a record of competence.

1.2 A group of 2-3 students as team may be allowed to work on one project.

2. **Project content**

2.1 Project for class XI can be selected from the topics mentioned in the syllabus or domains on the similar lines

2.2 Project for class XII should ensure the coverage of following areas of curriculum:

- Flow of control
- Data Structure
- Object Oriented Programming C++
- Data File Handling

Theme of the project can be

- Any subsystem of a System Software or Tool
- Any Scientific or a fairly complex algorithmic situation
- School Management, Banking, Library Information System, Hotel or Hospital Management System, Transport query system
- Quizzes / Games;
- Tutor, Computer Aided Learning Systems

2.3 It is suggested to prepare a bilingual (English and other Indian language) user manual part of project file.

2.4 The aim of the project is to highlight the abilities of algorithmic formulation, modular programming, optimized code preparation, systematic documentation and other associated aspects of Software Development.