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

Common to both the options. Refer to Unit 1 mentioned in case of Python for further details

Unit-2: Programming Methodology

Common to both the options Refer to Unit 2 mentioned in case of Python for further details.

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, Relation operator (>,>=,<=,=,!=), Logical operators (!,&&,||), Conditional operator: <condition>?<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 ( ), strev ( ), strupur ( ), strlwr ( )
Mathematical Functions	math.h	fabs ( ), pow ( ), sqrt ( ), sin ( ), cos ( ), abs ( )
Other Functions	stdlib.h	randomize ( ), random ( )

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 functions and 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, Inputting 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. **One logical problem to be solved through flow charts.** **04**

3. **Project Work** **06**

   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)

4. Practical File **(5+1*)=6**

(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.

6. 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.