

Course Outline

Programming Fundamental using C++

Books: we used following books during study

1. C++ How to program by DIETAL and DIETAL
2. Let us C by Yashwant Kantaker
3. Object-Oriented-Programming using C++ by IT Series

Week No.	Lecture No.	Contents
1	1	<ul style="list-style-type: none"> • Course introduction • Objectives and learning outcomes • Natural languages and Programming Languages • Comparison of natural language and computer language • Problem solving cycle/ programming process
	2	<ul style="list-style-type: none"> • Overview of different programming languages <ul style="list-style-type: none"> ○ Imperative programming paradigm <ul style="list-style-type: none"> ▪ Structural programming languages ▪ Procedural programming languages ○ Declarative programming paradigm <ul style="list-style-type: none"> ▪ Functional programming languages ○ Modular programming paradigm <ul style="list-style-type: none"> ▪ Object oriented programming languages
2	3	<ul style="list-style-type: none"> • Overview of Algorithms and Flow Charts
	4	<ul style="list-style-type: none"> • Introduction about Tool and its Installation
3	5	<ul style="list-style-type: none"> • The First C++ program <ul style="list-style-type: none"> ○ Different parts of a program
	6	<ul style="list-style-type: none"> • Output <ul style="list-style-type: none"> ○ Creating Output ○ Escape Sequence
4	7	<ul style="list-style-type: none"> • Data Types
	8	<ul style="list-style-type: none"> • Inputs from user
5	9	<ul style="list-style-type: none"> • Calculation of different formula <ul style="list-style-type: none"> ○ Practice with multiple examples
	10	<ul style="list-style-type: none"> • Conditional Structure <ul style="list-style-type: none"> ○ Types of control structures ○ Relational Operators ○ IF Statements <ul style="list-style-type: none"> ▪ Its structure and practice with multiple examples
6	11	<ul style="list-style-type: none"> ○ Multiple if-else-if structure ○ Nested IF structure ○ Compound Condition

	12	<ul style="list-style-type: none"> ○ Switch Structure ○ GOTO statement
7	13	<ul style="list-style-type: none"> ● Loop Structure <ul style="list-style-type: none"> ○ Counter controlled loops ○ Sentinel controlled loops <ul style="list-style-type: none"> ▪ WHILE loop ▪ DO-WHILE loop
	14	<ul style="list-style-type: none"> ▪ Practice multiple examples for above LOOPS
8	15	<ul style="list-style-type: none"> ○ FOR Loop ○ CONTINUE and BREAK statements ○ Nested Loops
	16	<ul style="list-style-type: none"> ● Arrays <ul style="list-style-type: none"> ○ Advantages of Arrays ○ Declaration of one-dimensional array ○ Array initialization ○ Accessing array elements without using loop
9	17	<ul style="list-style-type: none"> ○ Accessing array elements using loops ○ Input and output values of array with and without using loop
	18	<ul style="list-style-type: none"> ○ Two-Dimensional (2D) Arrays <ul style="list-style-type: none"> ▪ Advantages of 2D array ▪ Declaration of 2D array ▪ Array initialization ▪ Accessing array elements with and without using loop ▪ Input and output values of array with and without using loop
10	19	<ul style="list-style-type: none"> ○ Multidimensional Array
	20	<ul style="list-style-type: none"> ● Functions <ul style="list-style-type: none"> ○ Advantages of functions ○ Types of functions in C++ ○ User-Defined functions <ul style="list-style-type: none"> ▪ Function declaration or function prototype ▪ Function definition ▪ Function call ○ Scope of function
11	21	<ul style="list-style-type: none"> ○ Passing parameters to functions <ul style="list-style-type: none"> ▪ Pass by Value ▪ Pass by reference ○ Returning values from function
	22	<ul style="list-style-type: none"> ○ Types of variables with respect to functions <ul style="list-style-type: none"> ▪ Local variables ▪ Global variables ▪ Static variables ○ Recursion

12	23	<ul style="list-style-type: none"> ○ Function and Arrays <ul style="list-style-type: none"> ▪ Practice multiple examples with combination of above two concepts i.e., functions and arrays
	24	<ul style="list-style-type: none"> ● Pointers <ul style="list-style-type: none"> ○ Basic introduction about memory and references ○ Pointer declaration ○ The VOID pointer ○ Dereference operator ○ Pointer initialization ○ Operations on pointers <ul style="list-style-type: none"> ▪ Pointer Addition ▪ Pointer Subtraction
13	25	<ul style="list-style-type: none"> ○ Memory Management with Pointers <ul style="list-style-type: none"> ▪ Dynamics Variables ▪ The NEW operator ▪ The DELETE operator ○ Practice multiple examples related to memory management
	26	<ul style="list-style-type: none"> ● Structure <ul style="list-style-type: none"> ○ Declaration of structure ○ Definition of structure variables ○ Accessing members of structure variable ○ Initializing structure variables
14	27	<ul style="list-style-type: none"> ○ Nested Structure <ul style="list-style-type: none"> ▪ Accessing members of nested structure ▪ Initializing nested structure ● Union ● Enumerations
	28	<ul style="list-style-type: none"> ● File Handling <ul style="list-style-type: none"> ○ Advantages of file ○ Type of files ○ File accessing methods <ul style="list-style-type: none"> ▪ Sequential access method ▪ Random access method
15	29	<ul style="list-style-type: none"> ○ Stream <ul style="list-style-type: none"> ▪ Types of streams ○ Opening Files <ul style="list-style-type: none"> ▪ Default opening modes ▪ Verifying file open ○ Writing files ○ Closing files
	30	<ul style="list-style-type: none"> ● Discussion and Evaluation of Project/Presentation
16	31	<ul style="list-style-type: none"> ● Discussion and Evaluation of Project/Presentation
	32	<ul style="list-style-type: none"> ● Discussion and Evaluation of Project/Presentation