Course Title	**Programming Fundamentals**		
Course Code	**CC-112**		
Credit Hours	3 (3,0)		
Category	Computer Science Core		
Prerequisite	None		
Co-Requisite	None		
Follow Up	CSC-111 Object Oriented Programming		
Course Introduction	This course provides fundamental concepts of programming to freshmen. The course is a prerequisite to many other courses; therefore, students are strongly advised to cover all the content and try to achieve CLOs to the maximum possible level.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	**BT**	**PLO**
	CLO1: Demonstrate proficiency in writing, debugging, and executing basic programs using programming languages such as C++ or Python.	C3 (Apply)	1,2,4,5
	CLO2: Explain core programming concepts including variables, control structures, functions, and data types.	C2 (Understand)	1,2
	CLO3: Develop simple algorithms to solve computational problems.	C6 (Create)	1,2,3,4
	CLO4: Apply best practices in coding to produce efficient and readable programs.	C3 (Apply)	1,2,4,5,8,9
	CLO5: Analyze program outputs and troubleshoot common errors effectively.	C4 (Analyze)	1,2,3,5,10
Text Book(s)	1. Tony Gaddis, Starting out with C++, 10th Edition, Pearson, 2024.		
Reference Material	1. Object Oriented Programming in C++ by Robert Lafore 2. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman		

Weekly Lesson Plan

Week No.	Course Content	Assignments/Readings
Week 1	Introduction to Problem Solving: Problem solving steps, algorithms, pseudo-code, flowcharts, overview of Von-Neumann Architecture	Gaddis: Chapter 1 Handouts
Week 2	Overview of the structured program theorem (sequence, selection and iteration)	Gaddis: Chapter 1 Handouts Quiz 1
Week 3	Introduction to C++: Structure of a C++ program, compilation process (compiler & linker), development environment, simple program (printing text), Variables and Data Types: Concept of variables, data types (int, float, char, bool)	Gaddis: Chapter 2 Assignment 1
Week 4	Input/output using cin/cout, arithmetic expressions, assignment statements, Operators and Expressions: Arithmetic operators, operator precedence, integer division, type casting, introduction to memory concepts	Gaddis: Chapter 3 Quiz 2
Week 5	Decision Making I: Relational and equality operators, if and if-else statements, nested if	Gaddis: Chapter 4
Week 6	Decision Making II & Loops: Logical operators, while loop, structured program development	Gaddis: Chapter 4-5 Quiz 3

Week 7	Iteration Control: for, do-while loops, break and continue, introduction to switch statement	Gaddis: Chapter 5 Assignment 2
Week 8	Functions I: Modular programming, function definition and prototypes, passing arguments by value	Gaddis: Chapter 6 Quiz 4
Week 9	Functions II: Passing by reference, scope rules, recursion, recursion vs iteration, random number generation	Gaddis: Chapter 6
Week 10	Arrays I: One-dimensional arrays, initialization, processing arrays, passing arrays to functions	Gaddis: Chapter 7 Quiz 5
Week 11	Arrays II: Searching and sorting arrays, multidimensional arrays, character arrays (strings basics)	Gaddis: Chapter 8 Assignment 3
Week 12	Strings and Structures: C++ strings, string handling functions, structures, accessing members, typedef, unions	Gaddis: Chapter 9-10 Quiz 6
Week 13	Pointers I: Pointer concepts, declaration and initialization, pointer operators, pointers and arrays	Gaddis: Chapter 9-10
Week 14	Pointers II & Dynamic Memory: Pointer arithmetic, array of pointers, function pointers (intro), dynamic memory allocation (new/delete)	Gaddis: Chapter 9-10 Quiz 7
Week 15	File Handling & Preprocessor: File streams (read/write), sequential files, basics of random access, preprocessor directives (#include, #define, macros, assertions)	Gaddis: Chapter 12 Assignment 4
Week 16	Advanced Topics & Review: Bitwise operators, enumerations, command-line/variable arguments, program termination (exit), course review	Gaddis: Chapter 11