Course Title	**Programming Fundamentals (Lab)**		
Course Code	**CC-112L**		
Credit Hours	1 (0,1)		
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.	Lab Content	Assignments/Readings
Week 1	Lab Orientation: Introduction to problem solving, algorithms, pseudo-code, flowcharts; writing first pseudo-code	Gaddis: Chapter 1 Handouts Lab Tasks
Week 2	Practice implementation of sequence, selection and iteration	Gaddis: Chapter 1 Handouts Lab Tasks
Week 3	overview of C++ IDE (Dev C++, Visual Studio etc.), First C++ Programs: Structure of C++ program, compiling & running	Gaddis: Chapter 2 Lab Tasks
Week 4	Printing text, simple input/output using cin/cout; Variables & Data Types: Declaring variables, basic data types, arithmetic expressions, assignment statements; Operators & Expressions	Gaddis: Chapter 2-3 Lab Tasks
Week 5	Decision Making I: Relational and logical operators; if and if-else programs (marks grading, max/min, etc.)	Gaddis: Chapter 4 Lab Tasks
Week 6	Decision Making II: Nested if, introduction to while loop; solving repetitive problems	Gaddis: Chapter 4-5 Lab Tasks
Week 7	Iterative Structures: for, do-while, switch, break, continue; menu-driven programs	Gaddis: Chapter 5 Lab Tasks
Week 8	Functions I: Function definition, prototypes, calling functions, passing arguments by value; basic modular programs	Gaddis: Chapter 6 Lab Tasks
Week 9	Functions II: Pass by reference, scope rules, recursion (factorial, Fibonacci), random number generation	Gaddis: Chapter 6 Lab Tasks
Week 10	Arrays I: One-dimensional arrays, input/output, summation, searching elements	Gaddis: Chapter 7 Lab Tasks

Week 11	Arrays II: Sorting (bubble/selection), multidimensional arrays, matrix operations	Gaddis: Chapter 8 Lab Tasks
Week 12	Strings & Structures: Character arrays vs string, string handling, defining structures, accessing members	Gaddis: Chapter 9-10 Lab Tasks
Week 13	Pointers I: Pointer basics, initialization, pointer operators, pointers with arrays	Gaddis: Chapter 9-10 Lab Tasks
Week 14	Pointers II & Dynamic Memory: Pointer arithmetic, dynamic memory (new/delete), passing by reference using pointers	Gaddis: Chapter 9-10 Lab Tasks
Week 15	File Handling & Preprocessor: File streams (ifstream, ofstream), reading/writing files, intro to macros (#define)	Gaddis: Chapter 12 Lab Tasks
Week 16	Advanced Topics & Final Practice: Bitwise operators, enums, debugging practice, mini problem-solving session / lab test	Gaddis: Chapter 11 Lab Tasks