Program	BS Data Science	
Course Code	CC-112	
Course Title	Programming Fundamentals	
Credit Hours	**Theory**	**Lab**
	3	1
Lecture Duration	90 minutes (1.5 Hours), 2 lectures per week, 1 LAB per week	
Semester	1	
Pre-requisites course / skills	**Courses**	**Knowledge**
	Nil	Nil
Follow Up Courses	Object Oriented Programming	

Course Learning Outcomes (CLOs)

CLO No	**Course Learning Outcome**	**Bloom Taxonomy**
CLO-1	**Understand** basic problem-solving steps and logic constructs	C2 (Understand)
CLO-2	**Apply** basic programming concepts	C3 (Apply)
CLO-3	**Design and implement** algorithms to solve real-world problems	C3 (Solve)

Objectives	1. Students should be able to translate their basic pseudo-code/flowcharts into some programming language that computer can understand so that they can get real feel of their efforts. 2. Student can translate of their logic into some programming language. 3. Students can learn basic principles of attacking a problem, a bit of performance factor and some basic structured design principles. 4. Students should be ready to take Object Oriented Programming course.

Learning Outcomes	• Students can write a program. • Students should be able to translate a computation problem into program. • Student can familiar with C++. • Student can design and implement algorithms to solve real world problems.
Syllabus	Topics: Flowcharts/Pseudo Codes, Basic C++ Language Constructs: Data types, Variable and Constants, Operator and Expressions, Input and Output (I/O), Formatted I/O, Escape Sequences. Structured Programming in C Language: Decision making using if control structure, Repetition using for and do while, multiple selection using switch and logical operators. Procedural Programming in C Language: functions, prototype, parameter and arguments, call by value and call by reference, library and header files, scope and life time of variables (storage classes), recursion. Composite data types arrays: definition, processing, and passing of array to a function, multidimensional arrays, searching and sorting. Pointers: pointer definition, pointer arithmetic, constant pointers, pointer and arrays. Strings: string and characters, string conversion functions, Dynamic Memory Allocation. User Defined Data Types: structures, definition, initialization, accessing members of structures, typedef, union and bitwise operators, enumerations. C File Processing: files and streams, Sequential Access File, Random Access File, Secondary Storage I/O. Miscellaneous Topics: Command Line Arguments.
Contents	1. Flow Charts/Pseudo Code 1.1. Sequence, Conditions, Repetition

	2. C++ Programming Language Introduction 3. Hello world in C++, COUT 3.1. Difference between Variables and Literals, Identifiers 4. Data Types 5. Cin, extraction operator 6. Formatted Output 7. Selection: 7.1. Relational operators and expression 7.2. If, if-else, switch 8. Repetition: 8.1. Loop, While, For, Do while 8.2. Sentinel-controlled loops, Nested loops 8.3. Increment and decrement operator 9. Function: 9.1. Defining, Calling, function prototype, passing arguments by value 9.2. Local and global variables, Static variables, 9.3. Default arguments 9.4. Overloading functions 10. Arrays: 10.1. Parallel Arrays, 2D Arrays 11. Pointers 12. CString 13. Structs, Union 14. Text and Binary File I/O
Teaching-learning Strategies	• Interactive class session • Hands on practices in class • Brainstorming and Group discussion sessions • Coding in LABS

Assignments	Coding Assignments 5			
Assessment and	Sr. #	Elements	Weightage	Details

Examinations	1	Formative Assessment	25%	It is continuous assessment. It includes: classroom participation, attendance, assignments and presentations, homework, attitude and behavior, hands-on-activities, short tests, quizzes etc.
	2	Midterm Assessment	35%	It takes place at the mid-point of the semester.
	3	Final Assessment	40%	It takes place at the end of the semester. It is mostly in the form of a test, but owing to the nature of the course the teacher may assess their students based on term paper, research proposal development, field work and report writing etc.
Textbooks	• Gaddis, T., & Sengupta, P. (2012). Starting Out with C++: From Control Structures Through Objects. Pearson.			
Reference Material/Suggested Readings	• **R1.** Reference from different books enlisted in reference material will be given as required or lecture notes for reading will be provided. • **R2.** Malik, D. S. (2011). JavaTM Programming: From Problem Analysis to Program Design. Cengage Learning. • **R3**. Ritchie, D. M., Kernighan, B. W., & Lesk, M. E. (1988). The C programming language. Englewood Cliffs: Prentice Hall.			
	• Handout provided by the teacher.			