

Fundamentals of Programming-I

Semester No 2	Code CS-116	Credit Hours 2-1
-------------------------	-----------------------	----------------------------

COURSE DESCRIPTION:

This course gives an introduction to fundamental concepts in modern computer programming. Students are taught how to write programs to solve well-defined problems. Importance is given on mastering basic programming skills and related theoretical concepts and principals. Students are taught both procedural and object-oriented styles of programming.

TEXT AND MATERIAL

Text Book: 1. Engineering Problem Solving With C++ 3rd Edition, Delores M. Etter

- Reference Books:**
1. C++ How to Program, 5th Edition by Deitel and Deitel
 2. C++ in Action: Industrial Strength Programming by Bartosz Milewski
 3. (Online Edition freely available at <http://www.relisoft.com/book/>)
 4. Object-Oriented Programming in C++, by Robert Lafore
 5. Big C++, by Cay Horstmann and Timothy Budd

COURSE LEARNING OUTCOMES:

Upon successful completion of the course, the student should be able to:

S No	CLO Statement	PLO	Learning Domain and level
1	Determine programming construct and articulate how it is used to achieve desired output using C++	PLO 2	C3
2	Develop a computer program to a well-defined problem. This includes developing a general flow of logic, identifying the variables, conditional/iterative execution, fail conditions	PLO 5	C3
3	Employ correct programming methodology for a given problem based on optimal/efficient design.	PLO 2	C3

ASSESSMENT SYSTEM:

Quizzes	10%
Assignments	10%
OHTs	40%
ESE	40%

Week No	Topics	Learning Outcomes	Assessment
1	Introduction to Computers	Course Outline, objectives, and teaching plan, assessment method. A brief history. Recent engineering achievements. A discussion of Numbering Systems. A discussion of hardware and software. A five-step problem-solving methodology	OHT and ESE
2	Simple C++ Programs:	Understand C++ Development Environment, Preprocessor Directives, Compilers, Linkers, Variables, Comments, Common C++ Data Types, C++ Keywords, I/O Stream, cin and cout commands, Pseudo code and Flowchart.	OHT and ESE
3	Arithmetic and Logic Operations:	Learn arithmetic operations (like addition, subtraction, multiplication, division, modulus etc.), rules of operator precedence, relational operators (like equality, greater than, less than etc.), logical operators (like and, or, not etc.)	OHT and ESE
4&5	Control Statements:	Understanding includes top down algorithm development, transfer of control, basis of conditional control, if conditional statement, if else double selection statement, nested control statements, while repetition statement, increment and decrement operators.	OHT and ESE
7&8	Loops:	Develop understanding of basis of counter controlled repetition, for repetition statement, do while repetition statement, switch multiple selection statement, usage of break and continue commands.	OHT and ESE
9&10	Functions:	Develop ability to program components in C++, function prototypes, function scope, signature and arguments, storage classes, pass by value, pass by reference and recursive functions.	OHT and ESE
11	Arrays:	Understanding of declaration of array, one dimensional array, multi-dimensional array, passing array to functions, searching arrays with linear search, sorting arrays, sorting arrays with insertion sort, bubble sort and quick sort.	OHT and ESE
12	Classes-I:	Develop understanding of classes, objects, member functions, data members, constructors, destructors, defining a class with a member function, defining a member function with a parameter, initializing objects with constructors.	OHT and ESE
14	Classes-II	<ul style="list-style-type: none"> Understanding of class scope, accessing class members, access functions, utility 	OHT and ESE

		functions, default member wise assignment, constant objects and member functions, friend functions and classes, static class members.	
15	Strings & File Handling:	<ul style="list-style-type: none"> Understanding includes Programming with the String data type, Finding characters in a string, Files and Streams, creating and updating a Sequential File, Creating and Updating a Random-Access File. 	OHT and ESE
16&17	Course Revision	<ul style="list-style-type: none"> Explain the studied material in an overall context Revise the topics if necessary 	OHT and ESE
18	End Semester Exam		

LAB PRACTICALS:

Lab No	Description
1	Introduction to Programming, Hands on practice on basic tool (creating, updating and editing of the source code and compiling of object code).
2	Syntax of a program with usage of header files, main function and “return” reserved word
3	Knowing the syntax of data types (int, float, char and strings), usage of variables, constants and how to declare the variable as constant.
4	Arithmetic, Relational and Logical Operators, for and while loop. Break and continue statements.
5	Assignment operators, increment / decrement operators,
6	Difference between post-increment / post-decrement and pre-increment / pre-decrement logics.
7	Decision statements (if-else, ladder if-else, use of else-if and nested conditions)
8	Loops nested with decision statements and vice versa
9	Do-while loop and Switch statement. Syntax and examples Nested switch statement, break and continue keywords
10	User-defined functions and function files. Sub functions and Nested functions (routines, methods, calling of functions)
11	Arrays and matrices (Multi- dimension Arrays, Linear Array), Addition and multiplication of Arrays
12	Pointers (defining, declaration and calling)
13	Link lists (sorting, merging, traversing)
14	Inserting and deleting link lists
15	Structures (defining, calling, accessing and inserting the entities of structures)