**Title of Subject : Programming Fundamentals (SW-112)**

**Discipline :** Software Engineering (1st Semester)

**Effective :** 19 Batch & onwards

**Pre-requisite :** None

**Assessment :** Theory**:** 20% Sessional, 80% Written Semester Examination

## (20% Mid, 60% Final)

Practical: 40% Sessional, 60% Final Examination

**Credit Hours :** 03 + 01 **Marks :** 100 + 50

**Minimum Contact Hours:** 45 + 45

# Specific Objectives of course:

* To understand the fundamentals of computer programing.
* To let students learn to create piece of code to perform tasks.

# Course Learning Outcomes (CLOs):

|  |  |  |  |
| --- | --- | --- | --- |
| CLO | Description | Taxonomy Level | PLO |
| 1 | Understanding and implementing concepts of functions, structures, pointers in C++ | C3 | 1 |
| 2 | Design algorithms to solve complex problems using arrays and pointers | C4 | 3 |
| 3 | To code, document, test and implement a well-structured, robust computer program using C++ programming language. | P2 | 3 |

**PROGRAM LEARNING OUTCOMES (PLOs):**

The course is designed so that students will achieve the following PLOs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Engineering Knowledge | ☑ | 7 | Environment and Sustainability | ☐ |
| 2 | Problem Analysis |  | 8 | Ethics | ☐ |
| 3 | Design/Development of Solutions | ☑ | 9 | Individual and Team Work | ☐ |
| 4 | Investigation | ☐ | 10 | Communication | ☐ |
| 5 | Modern Tool Usage | ☐ | 11 | Project Management | ☐ |
| 6 | The Engineer and Society | ☐ | 12 | Lifelong Learning | ☐ |

# Course outline:

* **BASIC OF C++ PROGRAMMING**

Constants and variables, -keywords, Identifiers, variable types: Integer, long, float, double and character, Types of variables: automatic/local, static, external/global, Standard and user defined function.

* **INPUT AND OUTPUT FUNCTIONS**

Introduction to Function, Difference between pre- defined/standard function and user defined function, Using more than one function, use of external variable, Prototype, function that return a value, Using arguments to pass Data to a function, Passing variables as arguments.

INPUT: single character, word and multiword, OUTPUT: single character and other data types, strings, Format specifiers, -field width specifiers, Escape sequence, printing strings, characters and graphic characters.

* **OPERATORS**

Address operator (&), Arithmetic Operators, -operator precedence, Arithmetic Assignment operator, Special assignment operators, Relational Operators, -Increment and Decrement Operator.

* **DECISION-MAKING STATEMENTS & LOOPS**

The if statement, The if-else statement, The else-if construct, Switch statement, goto statement, Conditional operator

For loop, Nested for loop, The while loop, The do while loop, Continue & break statement.

* **ARRAYS AND STRINGS**

Define an Array, Initializing an array, Multidimensional arrays, Arrays as function arguments, Strings, null character, string functions.

* **POINTERS**

Pointer variable, Returning multiple values from functions, Pointers and arrays, Pointers arithmetic, Pointers and strings, Double indirection: Pointers to pointers.

* **STRUCTURES UNION :**

Structures, Nested structures, Arrays of structures, Linked Lists, Routines, unions, Union of structures, Bit wise operator.

* **OBJECTS AND CLASSES**

Objects and Classes, Member Functions and Data, Private and Public, Constructors and Destructors, Objects and the Real World When to use Objects.

* **OPERATOR OVERLOADING**

The operator Keyword, Overloading Unary Operators, Overloading Binary Operators, Constructors as Conversion Routines, Converting between BASIC & user Defined Types, Thoughts on Overloading.

* **FILES**

Standard file I/O, Character, string and formatted I/O, Block I/O, Binary and Text file modes, System level I/O, Random access, & redirection.

# Practical Work to be carried out:

|  |
| --- |
| 1. Basics of C++ Programming and IDE Environment 2. Variables and Constants 3. Data Types and Expressions |
| 1. LOOPS |
| 1. Decision making statements |
| 1. Structures. |
| 1. Functions. |
| 1. Pointers. |
| 1. Arrays |
| 1. Objects and Classes. 2. Inheritance and Polymorphism |
| 1. Strings. |
| 1. Operator overloading. |
| 1. Streams and Files. |
| 1. Case Study/ Semester Project |

# Recommended Books:

* Peter Norton, Introduction to computers, Latest Edition.
* Robert Lafore, Object Oriented Programming in Turbo C++, Latest Edition.
* Deitel and Deitel, C++ How to Program, Prentice Hall Publications, Latest Edition

|  |  |  |
| --- | --- | --- |
| **Approval:** |  | |
| **Board of Studies:** | **Resolution No. 02** | **Dated: 29-08-2019** |
| **Board of Faculty:** | **Resolution No. 01** | **Dated: 07-10-2019** |
| **Academic Council:** | **Resolution No. 96.10** | **Dated: 07-10-2019** |