Course Code	Course Name	L-T-P-Credits	Year of Introduction
BT206	**C++ Programming**	**3-0-0-3**	**2016**

Prerequisite : Nil

Course Objectives

- To provide the fundamentals of object-oriented programming through a study of the concepts of program specification, algorithm development, and coding.
- To learn to write programs in an object-oriented high-level programming language.
- To develop knowledge in the fundamentals of algorithms, flowcharts, problem solving, programming concepts, classes and methods, control structures, arrays, and strings.

Syllabus

Introduction to computer, basic concepts of computer programming, control statements, arrays, function, structure, class object oriented programming, searching methods, searching, sorting, data structure and simple programs to apply all the programming concepts.

Expected outcome

Upon successful completion of this course, the students will

- become familiar with the basic problem solving skills using the high level language.
- become conversant with the approaches to software problems in C++
- write small-scale C++ programs using the above skills.
- acquire the knowledge to isolate and fix common errors in C++ programs
- apply the principles of object-oriented concepts to solve engineering problems that require computation.

Reference Books

1. Hubbard J. R., *Schaum's Outline of Programming with C++*, Tata McGraw Hill, 2004.
2. Lafore R., *Object-Oriented Programming in C++*, SAMS Publishing, 2001.
3. Kamthane A. M., *Object Oriented programming with ANSI and TURBO C++*, Pearson Education, 2006.
4. Balaguruswamy E., *Object Oriented programming with C++*, Tata McGraw Hill, 2013.
5. D'Orazio, T. B, *Programming in C++: Lessons and Applications*, McGraw-Hill, 2003.

Course Plan

Module	Contents	Hours	Sem. Exam Marks
I	Basic information on computers, Algorithm, flowchart, pseudo code, program development steps, programming languages, Character set, tokens, data types, variables, operators, expressions, Input and Output, directory, header files, return, computer hardware and software.	5	15%
II	Simple algorithms, flowcharts and programs, Control statements – if, switch statements with sample programs, Looping statements - for, while, do-while statements, Jump statements – break, continue, goto and exit (), simple programs	5	15%
FIRST INTERNAL EXAM			

III	Arrays - single and multi-dimensional arrays, initializing array elements, Character arrays, string functions. Functions – Arguments, returning function results, call by value and call by reference, functions calling functions, functions and arrays - Global variables, automatic, static and register variables, pointer, pointers and arrays, recursive functions.	8	15%
IV	Structures - functions and structures - Arrays of structures - structures within structures, Structures containing arrays. Files - Input and Output, Pointer and structure	8	15%
SECOND INTERNAL EXAM			
V	Basic concepts of object oriented programming, advantages of object oriented programming, Definition of a class, members of a class, data members and member functions, Declaration of objects, array of objects, Constructors and Destructors, Inheritance – Simple programs.	8	20%
VI	New data type creation, new header file creation, Searching – Linear search methods, sorting – Bubble sort, data structures - linked lists, stacks and queues in C++	8	20%
END SEMESTER EXAMINATION			

QUESTION PAPER PATTERN:

Maximum Marks: 100 Exam Duration: 3 hours

The question paper consists of Part A, Part B and Part C.

Part A consists of three questions of 15 marks each uniformly covering Modules I and II. The student has to answer two questions (15×2=30 marks).

Part B consists of three questions of 15 marks each uniformly covering Modules III and IV. The student has to answer two questions (15×2=30 marks).

Part C consists of three questions of 20 marks each uniformly covering Modules V and VI. The student has to answer two questions (20×2=40 marks).

Note : Each question can have a maximum of 4 subparts, if needed.