**DEVIKTECH**

Curriculum for the course of

C++ programming language

Course Instructor –

**Learning Objectives**

C++ is a powerful general-purpose programming language. It can be used to develop operating systems, browsers, games, and so on. C++ supports different ways of programming like procedural, object oriented, functional and so on. This makes C++ powerful as well as flexible.

**Learning Outcomes**

At the end of the course participants should be able to -

* To describe the advantages of a high level language like C/C++, the programming process and the compilation process.
* To design, debug, implement and test programs using the fundamental elements of C/C++.

**Prerequisites**

There are no prerequisites required for the course. But a little knowledge of programming and high school mathematics is a benefit for candidate.

**Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Topic** |  | **Total hours** | |
| 1 | Introduction, control statement, loop functions | 4 | |  |
| 2 | Array, pointer, string | 6 | |  |
| 3 | Object-oriented programming(oops), inheritance, polymorphism, data encapsulation | 4 | |  |
| 4 | C++ exception, C++ file and stream | 2 | |  |
| 5 | Standard Template Library (STL) | 7 | |  |

**CURRICULUM**

1. **Introduction**
2. C++ programming
3. Data types
4. Variables
5. Keywords
6. Operators
7. Expressions
8. **Control statements**
9. if-else
10. switch
11. for loop
12. while loop
13. do-while loop
14. break statement
15. continue statement
16. goto statement
17. **Functions**
18. C++ functions
19. Call by value
20. Call by reference
21. Recursion
22. **Array**
23. C++ Array
24. Array of function
25. 2D Array
26. **Pointer**
27. C++ pointer
28. Sizeof() operator
29. Array of pointer
30. Function pointer
31. Malloc vs new operator
32. **Strings**
33. C++ Strings
34. **OOPs**
35. oops concept
36. classes
37. objects
38. encapsulation
39. polymorphism
40. Inheritance
41. types of inheritance
42. constructor
43. destructor
44. **C++ Exceptions**
45. Exception handling
46. Try, catch, throw
47. **C++ file and stream**
48. file and stream
49. getline()
50. **Standard Template Library (STL)**
51. Introduction to STL
52. Sorting
53. Searching
54. Vectors
55. List