**INTRODUCTION TO C++**

* C++ is an object-oriented programming language.
* It is an extension to [C programming](https://www.javatpoint.com/c-programming-language-tutorial).

**Object-Oriented Programming (OOPs)**

C++ supports the object-oriented programming, the four major pillar of object-oriented programming ([OOPs](https://www.javatpoint.com/cpp-oops-concepts)) used in C++ are:

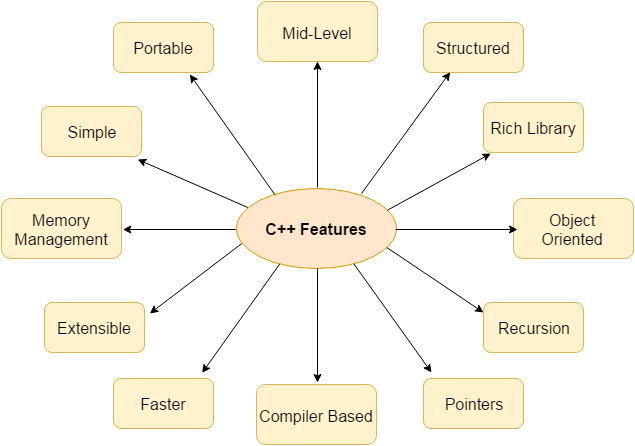
1. Inheritance
2. Polymorphism
3. Encapsulation
4. Abstraction

**APPLICATION OF C++**

* Window application
* Client-Server application
* Device drivers
* Embedded firmware etc

# **C++ Features**

C++ is object-oriented programming language. It provides a lot of **features** that are given below.



### **1) Simple**

C++ is a simple language in the sense that it provides structured approach (to break the problem into parts), rich set of library functions, data types etc.

### **2) Machine Independent or Portable**

Unlike assembly language, c programs can be executed in many machines with little bit or no change. But it is not platform-independent.

### **3) Mid-level programming language**

C++ support both high and low language. So it is known as mid-level language.

### **4) Structured programming language**

C++ is a structured programming language (i.e)we can break the program into parts using functions. So, it is easy to understand and modify.

### **5) Rich Library**

C++ provides a lot of inbuilt functions that makes the development fast.

### **6) Memory Management**

It supports the feature of dynamic memory allocation.

### **7) Speed**

The compilation and execution time of C++ language is fast.

### **8) Pointer**

C++ provides the feature of pointers. We can directly interact with the memory by using the pointers.

### **9) Recursion**

In C++, we can call the function within the function. It provides code reusability for every function.

### **10) Extensible**

C++ language is extensible because it can easily adopt new features.

### **11) Object Oriented**

C++ is object-oriented programming language. OOPs makes development and maintenance easier.

### **12) Compiler based**

C++ is a compiler based programming language, it means without compilation no C++ program can be executed.

# **C++ Program**

#include <iostream> -preprocessor directives

**using** **namespace** std;  -instead of .h

**int** main ( )

 {

**int** age;

   cout << "Enter your age: ";   - cout writing the output

   cin >> age; -cin read from input

   cout << "Your age is: " << age << endl;

}

## Standard output stream (cout)

* The **cout** is a predefined object of **ostream** class.
* It is connected with the standard output device, which is usually a display screen.
* Syntax: cout is represented by <<.

## Standard input stream (cin)

* The **cin** is a predefined object of **istream** class.
* It is connected with the standard input device, which is usually a keyboard.
* Syntax: cin is represented by >>.

# **C++ Variable**

* A variable is a name of memory location.
* It is used to store data.
* Its value can be changed during the execution of program.
* Syntax: **datatype variable\_list;**
* Examples : int a; , float y;

## Rules for defining variables

1. A variable can have alphabets, digits and underscore.
2. A variable name can start with alphabet and underscore only. It can't start with digit.
3. No white space is allowed within variable name.
4. A variable name must not be any reserved word or keyword e.g. char, float etc.

# **C++ Data Types:**

A data type specifies the type of data that a variable can store such as integer, floating, character etc.

|  |  |
| --- | --- |
| **Types** | **Data Types** |
| Basic Data Type | int, char, float, double, etc |
| Derived Data Type | array, pointer, etc |
| Enumeration Data Type | Enum |
| User Defined Data Type | Structure |

## Basic Data Types:

* The basic data types are integer-based and floating-point based. C++ language supports both signed and unsigned literals.
* The memory size of basic data types may change according to 32 or 64 bit operating system.

|  |  |  |
| --- | --- | --- |
| **Data Types** | **Memory Size** | **Range** |
| char | 1 byte | -128 to 127 |
| signed char | 1 byte | -128 to 127 |
| unsigned char | 1 byte | 0 to 127 |
| short | 2 byte | -32,768 to 32,767 |
| signed short | 2 byte | -32,768 to 32,767 |
| unsigned short | 2 byte | 0 to 32,767 |
| int | 2 byte | -32,768 to 32,767 |
| signed int | 2 byte | -32,768 to 32,767 |
| unsigned int | 2 byte | 0 to 32,767 |
| short int | 2 byte | -32,768 to 32,767 |
| signed short int | 2 byte | -32,768 to 32,767 |
| unsigned short int | 2 byte | 0 to 32,767 |
| long int | 4 byte |  |
| signed long int | 4 byte |  |
| unsigned long int | 4 byte |  |
| float | 4 byte |  |
| double | 8 byte |  |
| long double | 10 byte |  |

# **C++ Keywords:**

A keyword is a reserved word. You cannot use it as a variable name, constant name etc.

**A list of 32 Keywords in C++ Language which are also available in C language are given below.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| auto | break | case | char | const | continue | default | do |
| double | else | enum | extern | float | for | goto | if |
| int | long | register | return | short | signed | sizeof | static |
| struct | switch | typedef | union | unsigned | void | volatile | while |

# **C++ Operators:**

An operator is simply a symbol that is used to perform operations. There can be many types of operations like arithmetic, logical, bitwise etc.

There are following types of operators to perform different types of operations in C language.

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operator
6. Unary operator
7. Ternary or Conditional Operator
8. Misc Operator

