

UNIT I Introduction to C++: Difference between C and C++, Evolution of C++, The Object Oriented Technology, Disadvantage of Conventional Programming, Key Concepts of Object Oriented Programming, Advantage of OOP, Object Oriented Language.

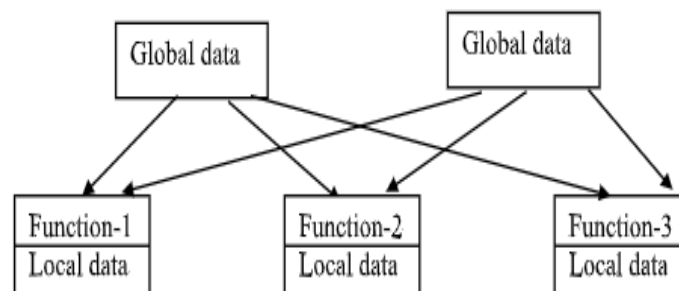
Introduction:

C++ is an object oriented programming language, C++ was developed by Jarney Stroustrup in 1983 at AT & T Bell laboratories, USA. C++ was developed from C and simula 67 language. C++ was early called 'C with classes'.

- C++ is derived from C Language.
- It is a Superset of C.
- Earlier C++ was known as C with classes.
- In C++, the major change was the addition of classes and a mechanism for inheriting class objects into other classes.
- Most C Programs can be compiled in C++ compiler.
- C++ expressions are the same as C expressions.
- All C operators are valid in C++.

Procedural oriented programming language

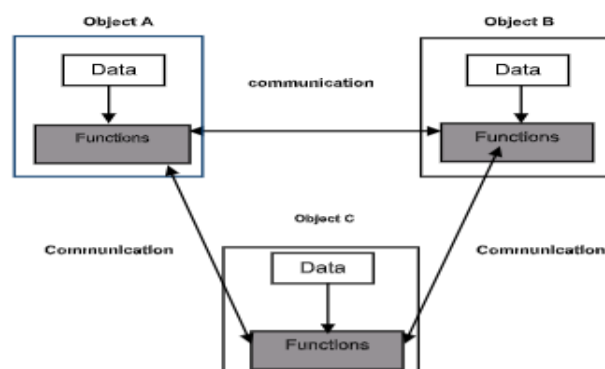
- Emphasis (importance) is given to creation of functions, not on data and large programs are divided into smaller programs known as functions.
- Most of the functions share global data.
- Data move openly around the system from function to function.
- Functions transfer data from one form to another.
- It employs 'top – down' approach in program design.
- Examples: C, BASIC, COBOL, FORTRAN



Object-oriented programming language

- In object oriented languages, the data has the principal role so it revolves around data.
- The main programming unit of OOP is the object.
- An object is a representation of a real-time entity and consists of data and methods or functions that operate on data.

- This way, data, and functions are closely bound and data security is ensured.
- Methods belong to the data, without the data, the method does not have any meaning.
- In OOP, everything is represented as an object and when programs are executed, the objects interact with each other by passing messages.
- An object need not know the implementation details of another object for communicating.
- It follows bottom-up approach.
- OOP tries to model the real world.
- It supports Polymorphism, Inheritance, Data Abstraction, Encapsulation.
- Examples –C++,JAVA,C#(C SHARP),.net



Advantages of OOP

1. **Simplicity:** Software objects model real world objects, so the complexity is reduced and the program structure is very clear.
2. **Modularity:** Each object forms a separate entity whose internal workings are decoupled from other parts of the system.
3. **Modifiability:** It is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods.
4. **Extensibility:** Adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.
5. **Maintainability:** Objects can be maintained separately, making locating and fixing problems easier.
6. **Re-usability:** Objects can be reused in different programs.

difference between c and c++

C	C++
C was developed by Dennis Ritchie between the year 1969 and 1973 at AT&T Bell Labs.	C++ was developed by Bjarne Stroustrup in 1979.
C is Procedural Language.	C++ is Object oriented Language.
Top down approach is used in Program Design.	Bottom up approach adopted in Program Design.
No namespace Feature is present in C Language.	Namespace Feature is present in C++ for avoiding Name collision.
Multiple Declaration of global variables are allowed	Multiple Declarations of global variables are not allowed.
In C programming language, a big program code is divided into small pieces which is called functions.	In C++ programming language, a big program code is divided into Objects and Classes.
Focuses on method or process instead of data.	Focuses on data instead of method or procedure.
In C , scanf() Function used for Input. printf() Function used for output.	In C++ , Cin>> Function used for Input. Cout<< Function used for output
C requires all the variables to be defined at the starting of a scope	C++ allows the declaration of variable anywhere in the scope i.e at time of its First use.
C contains 32 keywords.	C++ contains 63 keywords
In C, malloc() and calloc() Functions are used for Memory Allocation and free() function for memory Deallocating	In C++, new and delete operators are used for Memory Allocating and Deallocating.
In procedural oriented programming, there is no specific accessing mode to access attributes or functions in the program.	In Object oriented programming, there are three accessing modes “public”, “private”, “protected” that are used as an accessing share to access attributes or functions
In C language the data is not secured.	Data is secure, so it can't be accessed by external functions. (Using Encapsulation concept of OOPs)
C is not compatible with another language.	C++ is compatible with the other generic programming languages.
C does not support function overloading.	C++ supports function overloading.
Operator overloading is not possible in C.	Operator overloading is possible in C++.
C language does not allow you to declare String or Boolean data types. It supports built-in and primitive data types.	C++ supports String and Boolean data types.
In C, does not provide direct support to Exception handling.	In C++, Exception Handling is done with Try and Catch block.
Supports built-in data types.	Supports built-in & user-defined data types.
C does not support oops concepts i.e., Polymorphism, Inheritance, Encapsulation	C++ supports oops concepts i.e., Polymorphism, Inheritance, Encapsulation
C supports only Pointers.	C++ supports both pointers and references.
No concept of virtual classes	Concept of virtual function appears during inheritance.

C, VB, Fortran and Pascal are common examples of procedure oriented languages.	Examples of object oriented programming languages include: C++, Java, VB.NET, C#.NET and Python
--	---

Evolution of C++

C++ is an object oriented programming language and also considered as an extension of C. Bjarne Stroustrup at AT&T Bell Laboratories in Murray Hill, New Jersey (USA) developed this language in the early 1980s. Stroustrup, a master of Simula67 and C, wanted to combine the features of both the languages and he developed a powerful language that supports object-oriented programming with features of C. The outcome was C++ as per Fig. 1.1. Various features were derived from SIMULA67 and ALGOL68. Stroustrup called the new language 'C with classes'. However, in 1983, the name was changed to C++.

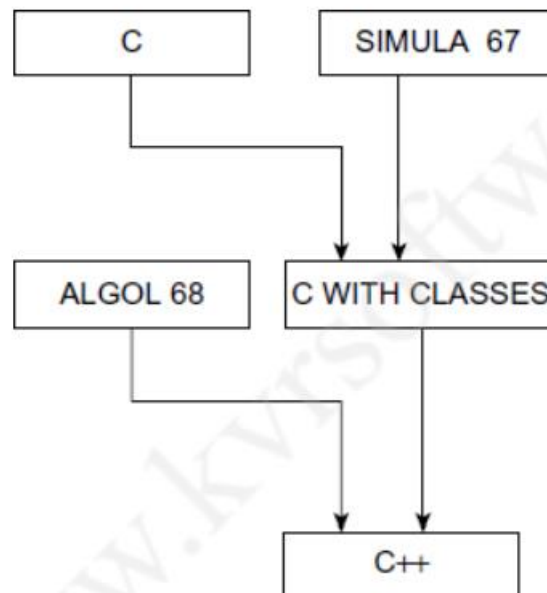


Fig. 1.1 Evolution of C++

For developing complicated applications, object oriented language such as C++ is the most convenient and easy. Hence, a programmer must be aware of its features.

Introduction to C++

- C++ is a general-purpose, high-level, compiler based, object oriented programming language.
- It is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features.
- It was developed by Bjarne Stroustrup starting in 1979 at Bell Labs as an enhancement to the C programming language and originally named "C with Classes".
- C++ is widely used in the software industry, and remains one of the most popular languages ever created.

Key features of C++

1. **Simple:** It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
2. **Machine Independent but Platform Dependent:** A C++ executable is not platform-independent (compiled programs on Linux won't run on Windows), however they are machine independent.
3. **Mid-level language:** It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
4. **Rich library support:** Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development.
5. **Speed of execution:** C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as garbage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast.
6. **Pointer and direct Memory-Access:** C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables).
7. **Object-Oriented:** One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
8. **Compiled Language:** C++ is a compiled language, contributing to its speed.
9. **Some of its application domains include**
 - ❖ Operating systems.
 - ❖ Editors
 - ❖ Application software
 - ❖ Device drivers
 - ❖ Databases
 - ❖ Translators
 - ❖ Entertainment software such as video games.
 - ❖ Commercial applications.

The language began as enhancements to C, first adding classes, then virtual functions, operator overloading, multiple inheritance, templates, and exception handling among other features.

Key concepts of Object oriented programming

1. Object
2. Class
3. Encapsulation
4. Abstraction
5. Polymorphism
6. Inheritance
7. Message passing

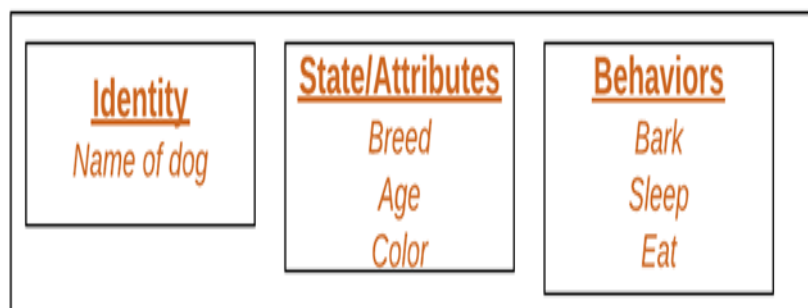
Object

- An object can be defined as an instance of a class, and there can be multiple instances of a class in a program.
- An Object contains both the data and the function, which operates on the data. For example - chair, bike, marker, pen, table, car, etc.
- An object consists of :

State : It is represented by attributes of an object

Behavior : It is represented by methods of an object.

Identity : It gives a unique name to an object.



2. Class

- A class is a data-type that has its own members i.e. data members and member functions.
- It is the blueprint for an object in objects oriented programming language. It is the basic building block of object oriented programming in c++.
- The members of a class are accessed in programming language by creating an instance of the class.
- Some important properties of class are –
 - ❖ A class can have multiple objects which have properties and behaviour that in common for all of them.
 - ❖ **Class** is a user-defined data-type.

- ❖ A class contains members like data members and member functions.
- ❖ **Data members** are variables of the class. Data members define the properties of the class whereas the member functions define the behaviour of the class.
- ❖ **Member functions** are the methods that are used to manipulate data members.

For example, if you had a class called “Expensive Cars” it could have objects like BMW, Toyota, etc. Its properties(data) can be price or speed of these cars. While the methods may be performed with these cars are driving, reverse, braking etc.

Syntax:

Class class_name

```
{
    Datatype datamembers;

    return_type method_name(parameters);
}
```

OBJECT	CLASS
Object is an instance of a class.	Class is a blue print from which objects are created
Object is a real world entity such as chair, pen, table, laptop etc.	Class is a group of similar objects.
Object is a physical entity.	Class is a logical entity.
Object is created many times as per requirement.	Class is declared once.
Object allocates memory when it is created.	Class doesn't allocated memory when it is created.

3. Inheritance

- When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability.
- It is used to achieve runtime polymorphism.

Sub class (child) - The class that inherits from another class

Super class (parent) - The class being inherited from.

4. Polymorphism

- Polymorphism refers to the ability of a variable, object or function to take on multiple forms.
- The name defines polymorphism is multiple forms which means polymorphism is the ability of object oriented programming to do some work using multiple forms.
- The behaviour of the method is dependent on the type or the situation in which the method is called.
- In c++ programming language, polymorphism is achieved using two ways. They are operator overloading and function overloading.
- **Function overloading** and **Operator overloading** are examples of polymorphism.
- **In function overloading** we can have more than one function with same name but different numbers, type or sequence of arguments.

Example - In English, the verb run has a different meaning if you use it with a laptop, a foot race, and business. Here, we understand the meaning of run based on the other words used along with it. The same also applied to Polymorphism.

5. Abstraction

- An abstraction is an act of representing essential features without including background details. It is a technique of creating a new data type that is suited for a specific application.
- Hiding of unwanted details and show the essential one is called Data Abstraction.
Example- While driving a car, you do not have to be concerned with its internal working. Here you just need to concern about parts like steering wheel, Gears, accelerator, etc.

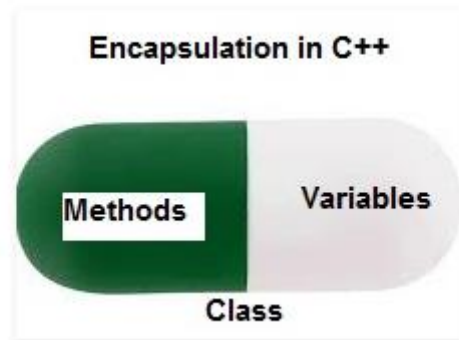
6. Encapsulation

- Encapsulation is an OOP technique of wrapping the data and code. In this OOPS concept, the variables of a class are always hidden from other classes. It can only be accessed using the methods of their current class.

(or)

- Binding (or wrapping) code and data together into a single unit are known as encapsulation.
- Due to the concept of encapsulation in object oriented programming another very important concept is possible, it is data abstraction or Data Hiding. I
- It is possible as encapsulating hides the data at show only the information that is required to be displayed.

Example –A capsule, it is wrapped with different medicines.



Disadvantages of conventional programming

- Emphasis (importance) is on doing things.
- Most of the functions share global data. So we cannot easily find out which function modifies global data if several hundreds of functions exists.
- Data is moving freely from function to function, therefore there is no security to data.

To overcome the above problems, many software developers turned into object oriented programming languages to solve these problems

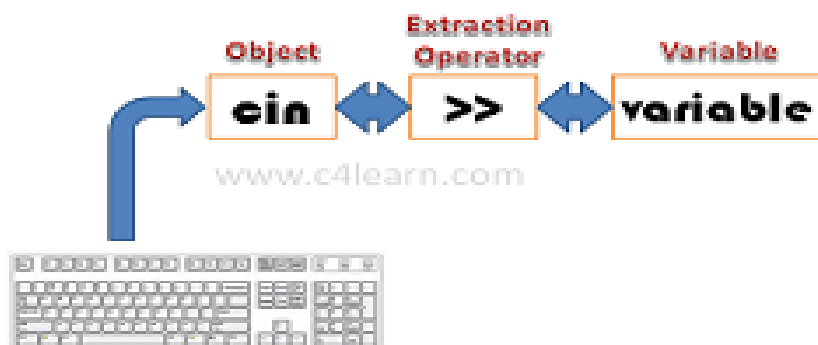
C++ I/O Operation Examples:

1. Cin-

- It takes input from keyboard:
- To input data from the keyboard, we use the word 'Cin' followed by 'extaction' operator, which is two greater than symbols >>

Ex: The following statement will wait for a value to be entered through the keyboard and when enter key is pressed, it will put value into the variable.

Cin>>a;



2. Cout

- It displays output on the Screen:
- To show output on the screen, we use the word 'Cout' followed by 'insertion' operator which is two less than symbols <<.

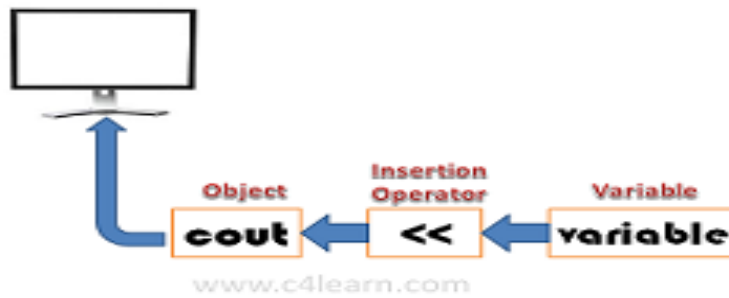
Ex: `Cout<<a;`

- The following are the different ways to print on the screen.

`cout<< "Output sentence";` *// prints Output sentence on screen*

`cout<< 120;` *// prints number 120 on screen*

`cout<< x;` *// prints the content of x on screen*



Example programs

1. C++ program to perform all arithmetic operations

```
#include <iostream.h>
```

```
Using namespace std;
```

```
void main ()
```

```
{
```

```
    int a, b, p, q, r, s;
```

```
    a = 10;
```

```
    b=4;
```

```
    p= a/b;
```

```
    q= a*b;
```

```
    r= a%b;
```

```
    s= b%a;
```

```
    cout<<p<<q<<r<<s;
```

```
}
```

2. C++ program to check whether the given number is positive or negative.

```
#include <iostream>

using namespace std;

main()
{
    int number;

    cout<< "Enter an integer: ";

    cin>> number;

    if (number >= 0)
    {
        cout<< "You entered a positive integer: " << number <<endl;
    }
    else
    {
        cout<< "You entered a negative integer: " << number <<endl;
    }
}
```

3. C++ program to create a calculator using switch.

```
#include <iostream>

using namespace std;

int main()
{
    Char oper;

    float num1, num2;

    cout<< "Enter an operator (+, -, *, /): ";

    cin>>oper;

    cout<< "Enter two numbers: " <<endl;
```

```

cin>> num1 >> num2;

switch (oper)
{
    case '+': cout<< num1 << " + " << num2 << " = " << num1 + num2;
        break;

    case '-': cout<< num1 << " - " << num2 << " = " << num1 - num2;
        break;

    case '*': cout<< num1 << " * " << num2 << " = " << num1 * num2;
        break;

    case '/': cout<< num1 << " / " << num2 << " = " << num1 / num2;
        break;

    default: cout<< "Error! The operator is not correct";
        break;

}

return 0;
}

```

4. C++ program for finding factorial of a given number

```

#include <iostream>

using namespace std;

int main()
{
    int i,fact=1,number;

    cout<<"Enter any Number: ";

    cin>>number;

    for(i=1;i<=number;i++)
    {
        fact=fact*i;
    }
}

```

```

    }

    cout<<"Factorial of " <<number<<" is: "<<fact<<endl;

    return 0;

}

```

5. C++ program to demonstrate strcat() function .

```

#include <cstring>

#include <iostream>

using namespace std;

int main()

{

    Char dest[50] = "object oriented";

    Char src[50] = " programming";

    strcat(dest, src);

    cout<<dest;

    return 0;

}

```

6. C++ program to perform addition of two matrices.

```

#include <iostream>

using namespace std;

int main()

{

    int row, col, a[10][10], b[10][10], sum[10][10], i, j;

    cout<< "Enter number of rows ";

    cin>> rows;

    cout<< "Enter number of columns ";

    cin>> col;

```

```

cout<<endl<< "Enter elements of 1st matrix: " <<endl;

// Storing elements of first matrix entered by user.
for(i = 0; i < row; ++i)
{
    for(j = 0; j < col; ++j)
    {
        cout<< "Enter element a" << i + 1 << j + 1 << " : ";
        cin>> a[i][j];
    }
}

// Storing elements of second matrix entered by user.
cout<<endl<< "Enter elements of 2nd matrix: " <<endl;
for(i = 0; i < row; ++i)
{
    for(j = 0; j < col; ++j)
    {
        cout<< "Enter element b" << i + 1 << j + 1 << " : ";
        cin>> b[i][j];
    }
}

// Adding Two matrices
for(i = 0; i < row; ++i)
{
    for(j = 0; j < col; ++j)
    {
        sum[i][j] = a[i][j] + b[i][j];
    }
    cout<<endl<< "Sum of two matrix is: " <<endl;
    for(i = 0; i < row; ++i)

```

```

        {
            for(j = 0; j < col; ++j)
            {
                cout<< sum[i][j] ;
                if(j == c - 1)
                    cout<<endl;
            }
        }
        return 0;
    }
}

```

7. C++ program to display array elements

```

#include <iostream>

using namespace std;

int main()
{
    int numbers[5] = {7, 5, 6, 12, 35};

    cout<< "The numbers are: ";

    for (int i = 0; i < 5; ++i)
    {
        cout<< numbers[i] ;
    }
}

```

8. C++ program for arithmetic operations using functions.

```
#include<iostream.h>
#include<conio.h>
int res;
void main()
{
    int sum(int,int);
    int sub(int,int);
    int mul(int,int);
    int a,b,mul,sub,add;
    cout<<"Enter two numbers:";
    cin>>a>>b;
    add=sum(a,b);
    sub=subt(a,b);
    mul=multi(a,b);
    cout<<"Sum:"<<s<<"Subtraction:"<<u<<"Multiplication:"<<m;
    getch();
}
Void sum(int a,int b)
{
    res=a+b;
    return res;
}

Void subt(int a,int b)
{
    res=a-b;
    return(res);
}

Void multi(int a,int b)
{
    res=a*b;
    return res;
}
```