object-oriented programming

(OOP)

KIAN _ ACADEMY

Inheritance :-

- **Inheritance :** is one of the most important feature of Object Oriented Programming.

- **Sub Class** : The class that inherits properties from another class is called Sub class or Derived Class.

- **Super Class** : The class whose properties are inherited by sub class is called Super Class or Base class

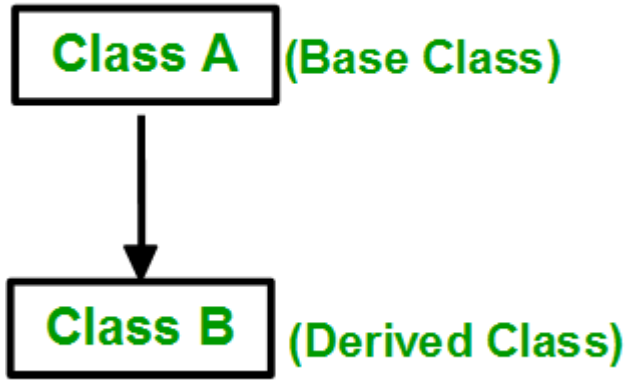
Inheritance :-

- **Types of Inheritance in C++:**

1) Single Inheritance : In single inheritance, a class is allowed to inherit from only one class. i.e. one sub class is inherited by one base class only.

Syntax :

```
class name : access specifer class name
{
  // body of subclass
};
```



Example

```
#include<iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
class Car : public Vehicle
{


};
int main()
{
    Car c;
    return 0;
}
```

Output:

This is a Vehicle

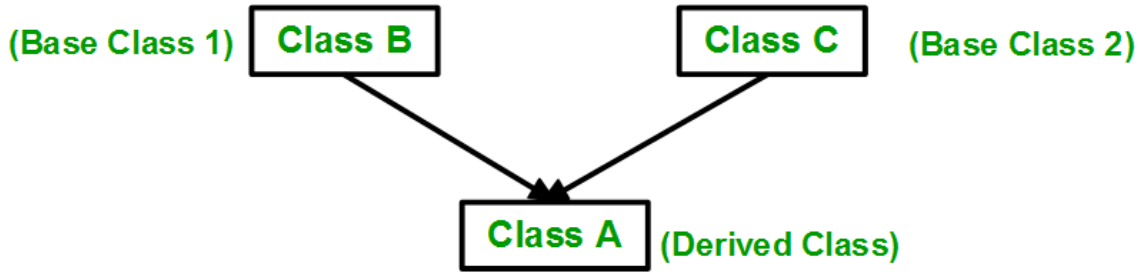
Inheritance :-

- **Types of Inheritance in C++:**

2) Multiple Inheritance : Multiple Inheritance where a class can inherit from more than one classes. one sub class is inherited from more than one base classes

Syntax :

```
class subclass_name : access_mode base_class1, access_mode base_class2, ....
{
  // body of subclass
};
```



Note : symbol used to create multiple is **comma**.

Example

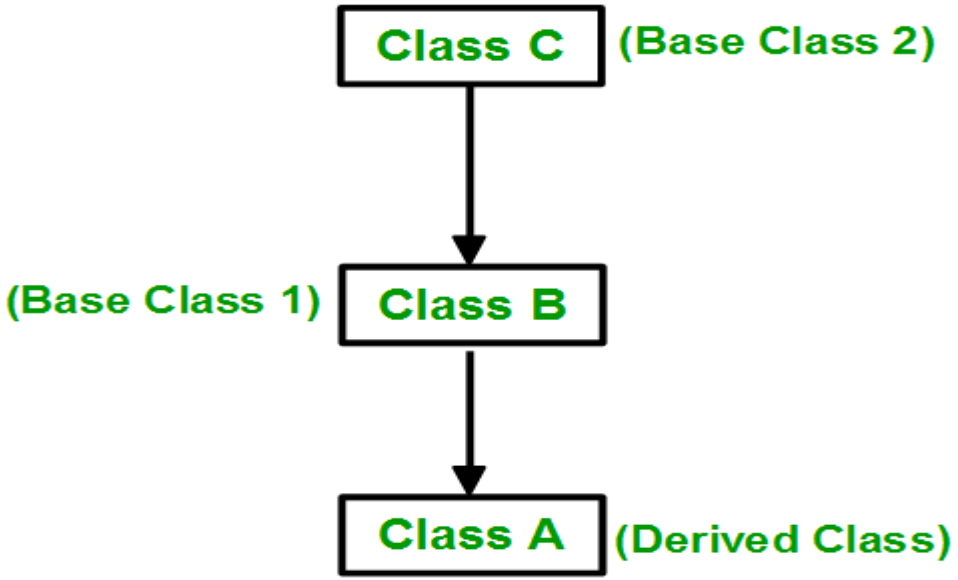
```
#include<iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
class FourWheeler
{
public:
    FourWheeler()
    {
        cout << "This is a 4 wheeler Vehicle\n";
    }
};
class Car : public Vehicle, public FourWheeler
{
};
int main()
{
    Car c;
    return 0;
}
```

Output :

```
This is a Vehicle
This is a 4 wheeler Vehicle
```

Inheritance :-

- **Types of Inheritance in C++:**
- **3) Multilevel Inheritance** : In this type of inheritance, a derived class is created from another derived class.



Example

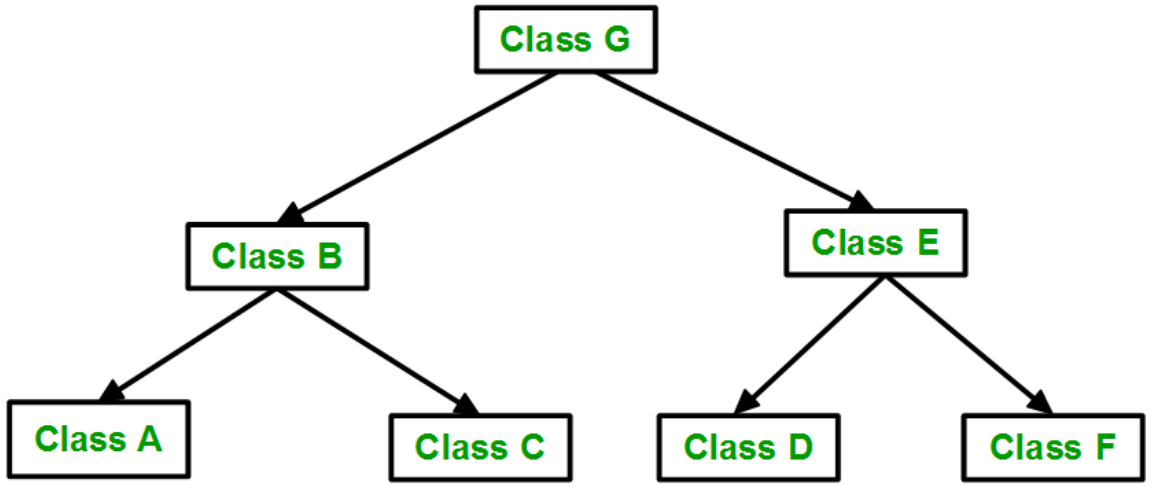
```
#include<iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
class fourWheeler : public Vehicle
{
public:
    fourWheeler()
    {
        cout << "Objects with 4 wheels are vehicles\n";
    }
};
class Car : public fourWheeler
{
public:
    Car()
    {
        cout << "Car has 4 Wheels\n";
    }
};
int main()
{
    Car c;
    return 0;
}
```

Output:

- This is a Vehicle
- Objects with 4 wheels are vehicles
- Car has 4 Wheels

Inheritance :-

- **Types of Inheritance in C++:**
- **4) Hierarchical Inheritance**: In this type of inheritance, more than one sub class is inherited from a single base class . more than one derived class is created from a single base class.



Example

```
#include<iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
class Car: public Vehicle
{
};
class Bus : public Vehicle
{
};
int main()
{
    Car c;
    Bus b;
    return 0;
}
```

Output :

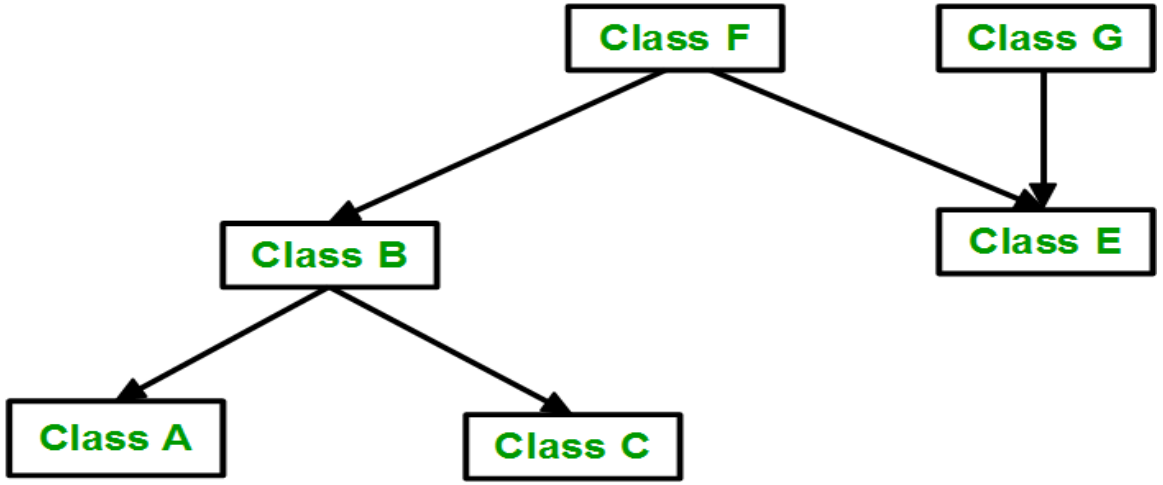
This is a Vehicle

This is a Vehicle

Inheritance :-

- **Types of Inheritance in C++:**
- **5) Hybrid (Virtual) Inheritance** : Hybrid Inheritance is implemented by combining more than one type of inheritance.

  → For example : Combining Hierarchical inheritance and Multiple Inheritance.



Example

```
#include<iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
```

```
class Fare
{
public:
    Fare()
    {
        cout << "Fare of Vehicle\n";
    }
};
class Car : public Vehicle
{
};
```

```
class Bus : public Vehicle, public F
{
};
int main()
{
    Bus b;
    return 0;
}
```

Output :

This is a Vehicle

Fare of Vehicle

Inheritance :-

- **Types of Inheritance in C++:**
- A special case of hybrid inheritance: **Multipath inheritance**:

A derived class with two base classes and these have one common base class is called multipath inheritance

Example

```
#include<iostream>
using namespace std;
class ClassA
{
public:
   int a;
};
class ClassB : public ClassA
{
public:
   int b;
};
class ClassC : public ClassA
{
public:
   int c;
};
class ClassD : public ClassB, public ClassC
{
public:
   int d;
};
int main()
{
   ClassD obj;
   obj.ClassB::b = 10;
   obj.ClassC::a = 100;
   obj.b = 20;
   obj.c = 30;
   obj.d = 40;
cout << " a from ClassB : " << obj.ClassB::a<<endl;
   cout << " a from ClassC : " << obj.ClassC::a<<endl;
   cout << " b : " << obj.b<<endl;
   cout << " c : " << obj.c<<endl;
   cout << " d : " << obj.d <<endl;
}
```

Output:

```
 a from ClassB : 8
 a from ClassC : 100
 b : 20
 c : 30
 d : 40
```

Exam questions:

- **(1)** What does inheriatance allows you to do?

- A) create a class
- B) create a hierarchy of classes
- C) access methods
- D) None of the mentioned

- **(2)** What is the syntax of inheritance of class?

- A) class name
- B) class name : access specifer
- C) class name : access specifer class name
- D) None of the mentioned

- **(3)** How many types of inheritance are there in c++?

- A) 2
- B) 3
- C) 4
- D) 5

- **(4)** What is meant by containership?

- A) class contains objects of other class types as its members
- B) class contains objects of other class types as its objects
- C) both a & b
- D) none of the mentioned

- **(5)** How many types of constructor are there in C++?

- A) 1
- B) 2
- C) 3
- D) 4

- **(6)** How many constructors can present in a class?

- A) 1
- B) 2
- C) 3
- D) multiple

- **(7)** What should be the name of constructor?

- A) same as object
- B) same as member
- C) same as class
- D) none of the mentioned

- **(8)** What does derived class does not inherit from the base class?

- A) constructor and destructor
- B) friends
- C) operator = () members
- D) all of the mentioned

- **(9)** What is meant by multiple inheritance?

- A) Deriving a base class from derived class
- B) Deriving a derived class from base class
- C) Deriving a derived class from more than one base class
- D) None of the mentioned

- **(10)** Which symbol is used to create multiple inheritance?

- A) Dot
- B) Comma
- C) Dollar
- D) None of the mentioned

Answer the questions :-

- 1 - b
- 2 - c
- 3 - d
- 4 - a
- 5 - c
- 6 - d
- 7 - c
- 8 - d
- 9 - c
- 10 - b