Sunny kumar
26 june class task
batch-Linux System Programming Track

Inheritance

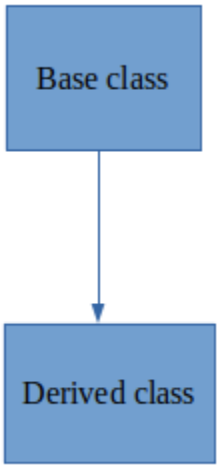
Inheritance in C++ is a feature that allows a class to inherit properties and behaviors (methods) from another class. It promotes code reusability and establishes a relationship between the base class and derived classes.

Types of inheritance:-

1.single inheritance
2.Multiple Inheritance
3.Hierarchical Inheritance
4.Multilevel Inheritance
5.Hybrid Inheritance

single inheritance

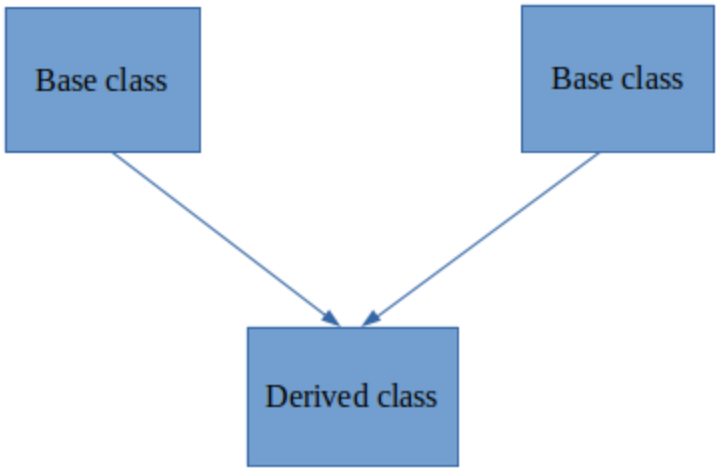
When a single derived class is created from a single base class is called single inheritance.



```cpp
#include <iostream>
using namespace std;

class Account {
public:
   float salary = 60000;
};

class Programmer: public Account {
public:
   float bonus = 5000;
};

int main(void) {
   Programmer p1;
   cout << "Salary: " << p1.salary << endl;
   cout << "Bonus: " << p1.bonus << endl;
   return 0;
}
```

Multiple Inheritance

When a derived class is created from more than one base class is called multiple inheritance.

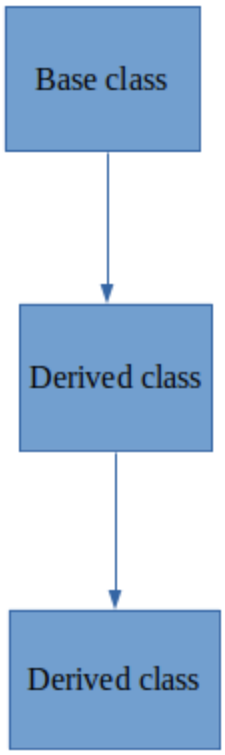



```
#include<iostream>
using namespace std;

class A{
   protected:
   int a;
   public:
   void get_a(int n)
   {
       a=n;
   }
};
class B{
   protected:
   int b;
   public:
   void get_b(int n)
   {
       b=n;
   }
};
class C:public:A,publicB{
   public:
   void Display()
   {
```

```
        cout<<"the value of a is: "<<a<<endl;
        cout<<"the value of b is: "<<<b<endl;
        cout<<"addition of a and b s: "<<a+b<<endl;
    }
};

int main()
{
    C c1;
    c1.get_a(10);
    c1.get_b(20);
    c1.Display();
    return 0;
}
```

Multilevel Inheritance

When a derived class is created from another derived class is called multilevel inheritance.

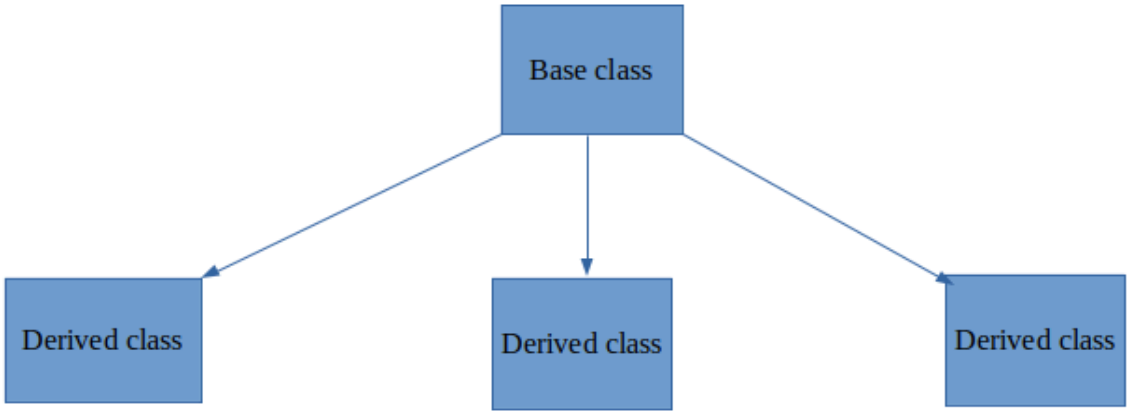



```
#include<iostream>
using namespace std;

class A{
   protected:
   int a;
   public:
   void get_a(int n)
   {
       a=n;
   }
};
class B{
   protected:
   int b;
   public:
   void get_b(int n)
   {
       b=n;
   }
```

```
};
class C:public:A,publicB{
   public:
   void Display()
   {
       cout<<"the value of a is: "<<a<<endl;
       cout<<"the value of b is: "<<<b<endl;
       cout<<"addition of a and b s: "<<a+b<<endl;
   }
};

int main()
{
   C c1;
   c1.get_a(10);
   c1.get_b(20);
   c1.Display();
   return 0;
}
```

Hierarchical Inheritance

When more than one derived class created from a single base class is called Hierarchical Inheritance.

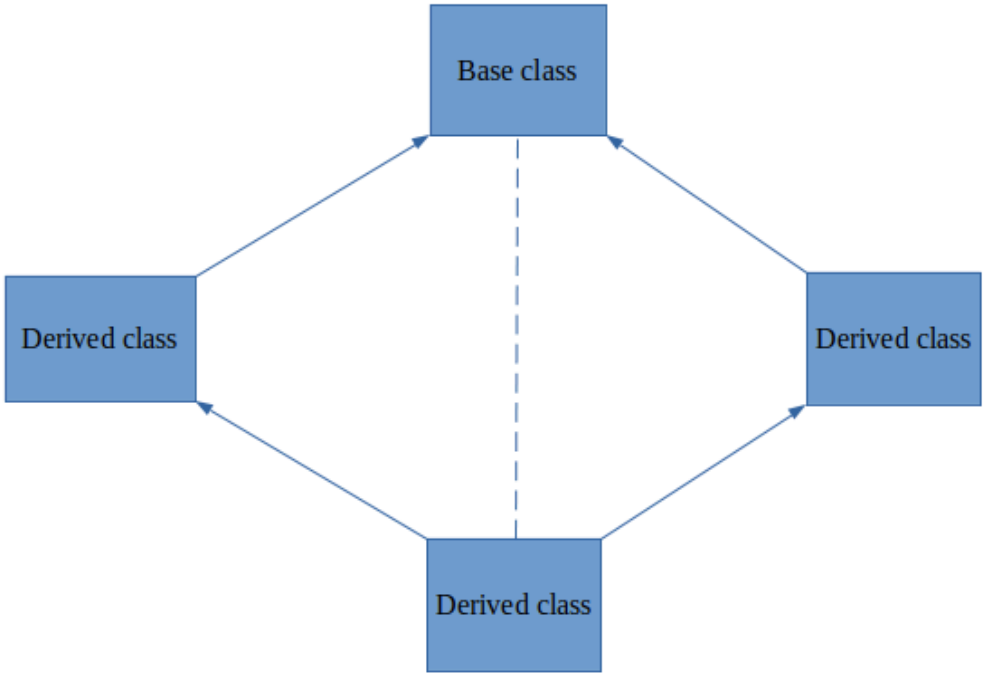



```cpp
#include <iostream>
using namespace std;
class Person {
protected:
   string name;
   int age;
public:
   void getDetails() {
       cout << "Enter name: ";
       cin >> name;
       cout << "Enter age: ";
       cin >> age;
   }
   void displayDetails() {
       cout << "Name: " << name << ", Age: " << age << endl;
   }
};
class Student : public Person {
private:
   int studentID;
public:
   void getStudentID() {
       cout << "Enter student ID: ";
       cin >> studentID;
   }
```

```
    void displayStudentDetails() {
        displayDetails();
        cout << "Student ID: " << studentID << endl;
    }
};
class Teacher : public Person {
private:
    int teacherID;
public:
    void getTeacherID() {
        cout << "Enter teacher ID: ";
        cin >> teacherID;
    }

    void displayTeacherDetails() {
        displayDetails();
        cout << "Teacher ID: " << teacherID << endl;
    }
};

int main() {
    Student student;
    Teacher teacher;

    cout << "Enter details for student:" << endl;
    student.getDetails();
    student.getStudentID();

    cout << "Enter details for teacher:" << endl;
    teacher.getDetails();
    teacher.getTeacherID();

    cout << "\nStudent Details:" << endl;
    student.displayStudentDetails();

    cout << "\nTeacher Details:" << endl;
    teacher.displayTeacherDetails();

    return 0;
}
```

Hybrid Inheritance

Combination of single , multiple and hierarchical inheritance is called Hybrid Inheritance.




```
#include <iostream>
using namespace std;

class A {
protected:
   int a;
public:
   void get_a() {
       cout << "Enter the value of 'a': " << endl;
       cin >> a;
   }
};

class B : public A {
protected:
   int b;
public:
```

```
    void get_b() {
        cout << "Enter the value of 'b': " << endl;
        cin >> b;
    }
};

class C {
protected:
    int c;
public:
    void get_c() {
        cout << "Enter the value of 'c': " << endl;
        cin >> c;
    }
};

class D : public B, public C {
public:
    void mul() {
        cout << "Product of a, b, and c is: " << a * b * c << endl;
    }
};

int main() {
    D d;
     d.get_a();
    d.get_b();
    d.get_c();
    d.mul();
    return 0;
}
```