Inheritance

Inheritance:

In C++, inheritance is **a process in which one object acquires all the properties and behaviors of its parent object automatically**. In such way, you can reuse, extend or modify the attributes and behaviors which are defined in other class.

Types of Inheriance:

* Single Inheritance
* Multiple Inheritance
* Multilevel Inheritance
* Hierarchical Inheritance
* Hybrid Inheritance

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 subclass_name : access_mode
base_class
{
  //body of subclass
};
```

Example:

```
#include<bits/stdc++.h>
#include<string.h>
using namespace std;
class Student
{

    public:

    string name = "Rokesh";
    int roll_no = 20361101;
    string school_name = "ABC
matric. Hr. secondary school";

    public:

    void scl_name(){
        cout<<"School name :
"<<school_name<<endl;
    }

};

class student1 : public Student{
public:
    void print(){
```

```
        cout<<"Student name is :
"<<name<<endl;
        cout<<"Student roll no is :
"<<roll_no<<endl;
    }
};
int main(){
    /* derived class*/
    student1 s;
    s.print();

    /*base class*/
    Student S;
    S.scl_name();

}
```

Output:

```
Student name is : Rokesh
Student roll no is : 234
School name = "ABC matric.
Hr. secondary school"
```

Multiple Inheritance:

Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes.
i.e 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
};
```

Here, the number of base classes will be separated by a comma (‘, ‘) and access mode for every base class must be specified.

Example:

```
#include<bits/stdc++.h>
#include<string.h>
using namespace std;
string school_name = "ABC matric.
Hr. secondary school";
class Student1
{
    public:

    string name = "Santhosh";
    int roll_no = 20361101;


    public:

    Student1(){
        cout<<"Student1 name is :
"<<name<<endl;
        cout<<"Student1 roll no is :
"<<roll_no<<endl;
        cout<<"school name :
"<<school_name<<endl;
    }
```

```
};
class Student2
{

    public:

    string name = "Akash";
    int roll_no = 20361102;

    public:

    Student2(){
        cout<<endl<<"Student2 name
is : "<<name<<endl;
        cout<<"Student2 roll no is :
"<<roll_no<<endl;
        cout<<"school name :
"<<school_name<<endl;
    }

};

/*  Multiple inheritance */
class Students : public
Student1,public Student2{

};
int main(){
```

```
    Students S;
    return 0;


}
```

Output:

Student1 name is : Santhosh
Student1 roll no is : 123
school name : ABC matric. Hr. secondary school

Student1 name is : Akash
Student1 roll no is : 124
school name : ABC matric. Hr. secondary school

Multilevel Inheritance

In this type of inheritance, a derived class is created from another derived class.

When one class inherits another class which is further inherited by another class, it is known as multi level inheritance in C++.

Inheritance is transitive so the last derived class acquires all the members of all its base classes.

Example:

```
#include<bits/stdc++.h>
#include<string.h>
using namespace std;
string school_name = "ABC matric.
Hr. secondary school";
class Student
{
    public:

    string name = "Santhosh";
    int roll_no = 20361101;


    public:

    Student(){
         cout<<"Student name is :
"<<name<<endl;
        cout<<"Student roll no is :
"<<roll_no<<endl;

    }

};
class group : public Student
{
```

```
    public:

    string Group = "Bio-logy ";
    string Section = "A-2";

    public:

    group(){
        cout<<"Group :
"<<Group<<endl;
        cout<<"Section :
"<<Section<<endl;
    }

};

/*  Multi level  inheritance */
class full_detail : public group{

    public:
    full_detail(){
        cout<<"School name :
"<<school_name<<endl;
    }


};
int main(){
```

```
    full_detail f;
    return 0;


}
```

Output:

Student name is : Santhosh
Student roll no is :
Group : Bio-logy
Section : A-2
School name : ABC matric.Hr . Secondary school

Hierarchical Inheritance

In this type of inheritance, more than one sub class is inherited from a single base class.

i.e. more than one derived class is created from a single base class.

Syntax of Hierarchical inheritance:

```
class A
{
    // body of the class A.
}
class B : public A
{
    // body of class B.
```

```
}
class C : public A
{
    // body of class C.
}
class D : public A
{
    // body of class D.
}
```

Example:

```
#include<bits/stdc++.h>
#include<string.h>
using namespace std;

string name = "Santhosh";
int roll_no = 20361101;
string school_name = "ABC matric. Hr.
secondary school";

class Student
{

};
class group : public Student
{

    public:

    string Group = "Bio-logy ";
    string Section = "A-2";
```

```
};
class group_name : public group
{

    public:

    group_name(){
        cout<<"Group : "<<Group<<endl;
    }

};
class sec : public group
{

    public:

    sec(){
        cout<<"Section : "<<Section<<endl;
    }

};
```

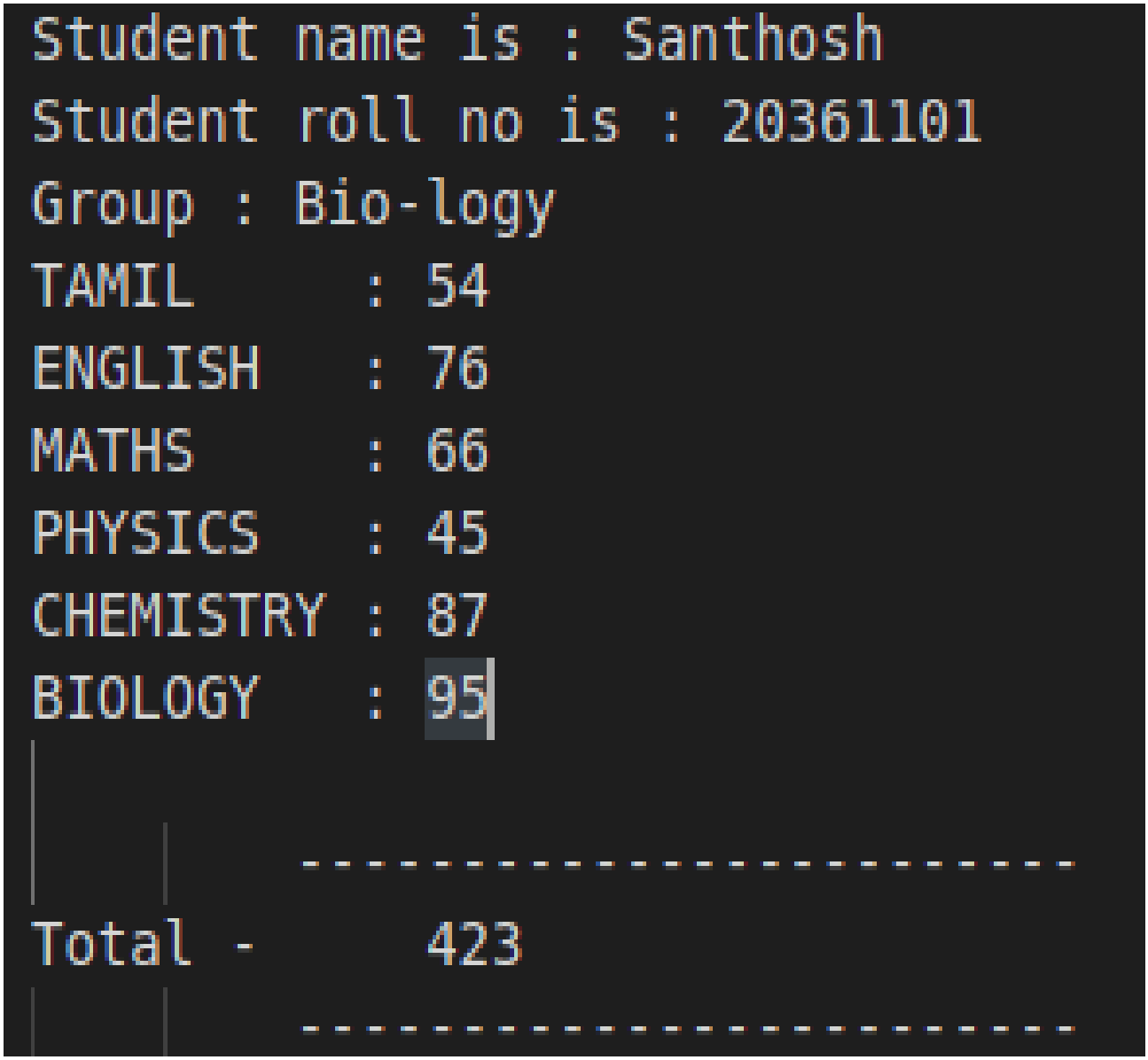
```
class marks : public Student
{

};
class marks1 : public marks
{
    public:

    int tamil = 54;
    int english = 76;
    int maths = 66;

    public:

    marks1(){
        cout<<"TAMIL     : "<<tamil<<endl;
        cout<<"ENGLISH   :
"<<english<<endl;
        cout<<"MATHS     :
"<<maths<<endl;

    }
```

```
};
class marks2 : public marks
{
    public:

    int phy = 45;
    int che = 87;
    int bio = 95;

    public:

    marks2(){
        cout<<"PHYSICS   : "<<phy<<endl;
        cout<<"CHEMISTRY :
"<<che<<endl;
        cout<<"BIOLOGY   : "<<bio<<endl;
    }

};

class total_ : public marks1 ,public marks2
```

```
{
    public:

    int total;

    public:
    total_(){
        total = tamil + english + maths + phy
+ che + bio;
        cout<<endl<<"
------------------------"<<endl;
        cout<<"Total -     "<<total<<endl;
        cout<<"
------------------------"<<endl;
    }
};

int main()
{
    cout<<"Student name is :
"<<name<<endl;
```

```
    cout<<"Student roll no is :
"<<roll_no<<endl;
    group_name g;
    total_ t;

}
```

Output:

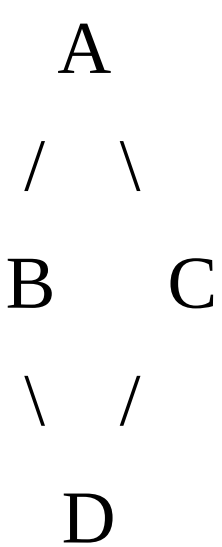


Hybrid Inheritance

Hybrid Inheritance is implemented by combining more than one type of inheritance.

For example: Combining Hierarchical inheritance and Multiple Inheritance.

Example:



Example:

```
#include<bits/stdc++.h>
#include<string.h>
using namespace std;

string name = "Santhosh";
int roll_no = 20361101;
string school_name = "ABC matric. Hr.
secondary school";

class Student
{

};
class group : public Student
{

    public:

    string Group = "Bio-logy ";
    string Section = "A-2";

    public:
```

```
    group(){
        cout<<"Group : "<<Group<<endl;
        cout<<"Section : "<<Section<<endl;
    }

};
class marks : private Student
{

    public:

    int tamil = 54;
    int english = 76;
    int phy = 45;
    int che = 87;
    int maths = 66;
    int bio = 95;

    public:

    marks(){
        cout<<"TAMIL     : "<<tamil<<endl;
        cout<<"ENGLISH   : "<<english<<endl;
        cout<<"MATHS     : "<<maths<<endl;
        cout<<"PHYSICS   : "<<phy<<endl;
        cout<<"CHEMISTRY : "<<che<<endl;
```

```
        cout<<"BIOLOGY   : "<<bio<<endl;
    }

};

/*  hybrid  inheritance */
class mid_term : public group,public marks{
    public:

    int total;

    public:
    mid_term(){
        total = tamil + english + maths + phy + che
+ bio;
        cout<<endl<<"
-----------------------"<<endl;
        cout<<"Total -     "<<total<<endl;
        cout<<"        -----------------------"<<endl;
    }



};
int main(){
    cout<<"Student name is : "<<name<<endl;
    cout<<"Student roll no is : "<<roll_no<<endl;
```

```
    mid_term m;
    return 0;


}
```

Output:

Previous and this program has same output , but code is different.