Single Inheritance

```
/* C++ program to demonstrate an Example of Single Inheritance  */

#include<iostream>
using namespace std;
class B
{
   int a;
   public:
   int b;
   void get_ab();
   int get_a();
   void show_a();
};

class D: private B
{
   int c;
   public:
   void mul();
   void display();
};

void B::get_ab()
{
   cout<<"\nEnter Values for a and b :: ";
   cin>>a>>b;
}

int B::get_a()
{
   return a;
}

void B::show_a()
{
   cout<<"\na = "<<a<<"\n";
}

void D::mul()
{
   get_ab();
   c=b*get_a();
}
```

```
void D::display()
{
   show_a();
   cout<<"\nb = "<<b<<"\n";
   cout<<"\nc = "<<c<<"\n\n";
}


int main()
{

            D d;
            d.mul();
            d.display();
            d.mul();
            d.display();
            return 0;

}
```

Multiple Inheritance

```
/*  C++ Program to demonstrate an Example of Multiple Inheritance  */

#include<iostream>
using namespace std;

    class M
    {
        protected:
        int m;
        public :
        void get_M(int );
    };
    class N
    {
        protected:
        int n;
        public:
        void get_N(int);
    };
    class P: public M, public N
    {
        public:
        void display(void);
    };

    void M::get_M(int x)
    {
        m=x;
    }
    void N::get_N(int y)
    {
        n=y;
    }
    void P::display(void)
    {
    cout<<"\n\tm = "<<m<<endl;
    cout<<"\n\tn = "<<n<<endl;
    cout<<"\n\tm*n = "<<m*n<<endl;
    }
    int main()
    {
        P p;
        p.get_M(10);
        p.get_N(20);
        p.display();
        return 0;
    }
```

Hybrid Inheritance:

```
/*  C++ Program to demonstrate an Example of Hybrid Inheritance  */

#include<iostream>
using namespace std;
class stu
{
protected:
    int rno;
public:
    void get_no(int a)
{
rno=a;
    }
void put_no(void)
    {
        cout<<"Roll no :: "<<rno<<"\n";
    }
```

```
};
class test:public stu
{
        protected:
        float part1,part2;
public:
  void get_mark(float x,float y)
  {
        part1=x;
        part2=y;
  }
void put_marks()
{
 cout<<"Marks obtained :\n"<<"part1 = "<<part1<<"\n"<<"part2 = "<<part2<<"\n";
}
};
class sports
{
        protected:
        float score;
public:
 void getscore(float s)
 {
        score=s;
 }
void putscore(void)
{
        cout<<"Sports : "<<score<<"\n";

}
};

class result: public test, public sports
{
        float total;
public:
        void display(void);
};
void result::display(void)
{
        total=part1+part2+score;
        put_no();
        put_marks();
        putscore();
```

```
    cout<<"Total Score = "<<total<<"\n";
}
int main()
{

    result stu;
    stu.get_no(123);
    stu.get_mark(27.5,33.0);
    stu.getscore(6.0);
    stu.display();
    return 0;
}
```

Hierarchical Inheritance:

```
/*  C++ Program to enter Student details using Hierarchical Inheritance */

#include<iostream>
using namespace std;

const int len = 20 ;

class student  //  Base class
{
    public:
      char F_name[len];
    char L_name[len];
      int age;
    int roll_no ;
        void Enter_data(void)
        {
            cout << "\n\t Enter the first name: " ; cin >> F_name ;
            cout << "\t Enter the second name: "; cin >> L_name ;
            cout << "\t Enter the age: " ; cin >> age ;
            cout << "\t Enter the roll_no: " ; cin >> roll_no ;
        }
        void Display_data(void)
        {
            cout << "\n\t First Name = " << F_name ;
            cout << "\n\t Last Name = " << L_name ;
            cout << "\n\t Age = " << age ;
            cout << "\n\t Roll Number = " << roll_no ;
        }
};
```

```
class arts : public student
{
    private:
        char asub1[len] ;
        char asub2[len] ;
        char asub3[len] ;
    public:
        void Enter_data(void)
        {
            student :: Enter_data( );
            cout << "\t  Enter the subject1 of the arts student: "; cin >> asub1 ;
            cout << "\t  Enter the subject2 of the arts student: "; cin >> asub2 ;
            cout << "\t  Enter the subject3 of the arts student: "; cin >> asub3 ;
        }
        void Display_data(void)
        {
            student :: Display_data( );
            cout << "\n\t Subject1 of the arts student = " << asub1 ;
            cout << "\n\t Subject2 of the arts student = " << asub2 ;
            cout << "\n\t Subject3 of the arts student = " << asub3 ;
        }
};

class science : public student
{
    private:
        char ssub1[len] ;
        char ssub2[len] ;
        char ssub3[len] ;
    public:
        void Enter_data(void)
        {
            student :: Enter_data( );
            cout << "\t Enter the subject1 of the science student: "; cin >> ssub1;
            cout << "\t Enter the subject2 of the science student: "; cin >> ssub2;
            cout << "\t Enter the subject3 of the science student: "; cin >> ssub3;
        }
        void Display_data(void)
        {
            student :: Display_data( );
            cout << "\n\t Subject1 of the science student = " << ssub1 ;
            cout << "\n\t Subject2 of the science student = " << ssub2 ;
            cout << "\n\t Subject3 of the science student = " << ssub3 ;
        }
};

class commerce : public student
{
    private:   char csub1[len], csub2[len], csub3[len] ;
    public:
        void Enter_data(void)
        {
            student :: Enter_data( );
            cout << "\t Enter the subject1 of the commerce student: ";
            cin >> csub1;
            cout << "\t Enter the subject2 of the commerce student: ";
            cin >> csub2 ;
            cout << "\t Enter the subject3 of the commerce student: ";
            cin >> csub3 ;
        }
        void Display_data(void)
        {
            student :: Display_data( );
            cout << "\n\t Subject1 of the commerce student = " << csub1 ;
            cout << "\n\t Subject2 of the commerce student = " << csub2 ;
            cout << "\n\t Subject3 of the commerce student = " << csub3 ;
        }
};

int main()
{
        arts a ;
        cout << "\n Entering details of the arts student\n" ;
        a.Enter_data( );
        cout << "\n Displaying the details of the arts student\n" ;
        a.Display_data( );
        science s ;
        cout << "\n\n Entering details of the science student\n" ;
        s.Enter_data( );
        cout << "\n Displaying the details of the science student\n" ;
        s.Display_data( );
        commerce c ;
        cout << "\n\n Entering details of the commerce student\n" ;
        c.Enter_data( );
        cout << "\n Displaying the details of the commerce student\n";
        c.Display_data( );
return 0;
}
```

Over riding member functions in inheritance:

```
/*  C++ Program to Overriding member functions using Inheritance  */

#include<iostream>
using namespace std;
const int len = 20 ;
class Employee
{
    private:
        char F_name[len];
        int I_number ;
        int age ;
        float salary ;
    public:
        void Enter_data(void)
        {
            cout << "\n Enter the first name = " ; cin >> F_name ;
            cout << "\n Enter the identity number = " ; cin >> I_number ;
            cout << "\n Enter the age = " ; cin >> age ;
            cout << "\n Enter the salary = " ; cin >> salary ;
        }
        void Display_data(void)
        {
            cout << "\n Name = " << F_name ;
            cout << "\n Identity Number = " << I_number ;
            cout << "\n Age = " << age ;
            cout << "\n Salary = " << salary ;
        }
};  // End of the base class

class Engineer : public Employee
{
    private:
        char design[len] ; // S_Engineer, J_Engineer, Ex_Engineer etc

    public:
        void Enter_data( )
        {
            Employee :: Enter_data( ) ;  // Overriding of the member function
            cout << "\n Enter the designation of the Engineer: " ; cin >> design ;
        }
```

```
            void Display_data(void)
            {
                  cout << "\n *******Displaying the particulars of the Engineer**** \n" ;
                  Employee :: Display_data( ) ; // Overriding of the member function
                  cout << "\n Designition = " << design ;
            }
}; // End of the derived class


int main(void)
{
            Engineer er ;
            er.Enter_data( ) ;
            er.Display_data( );
return 0;
}
```