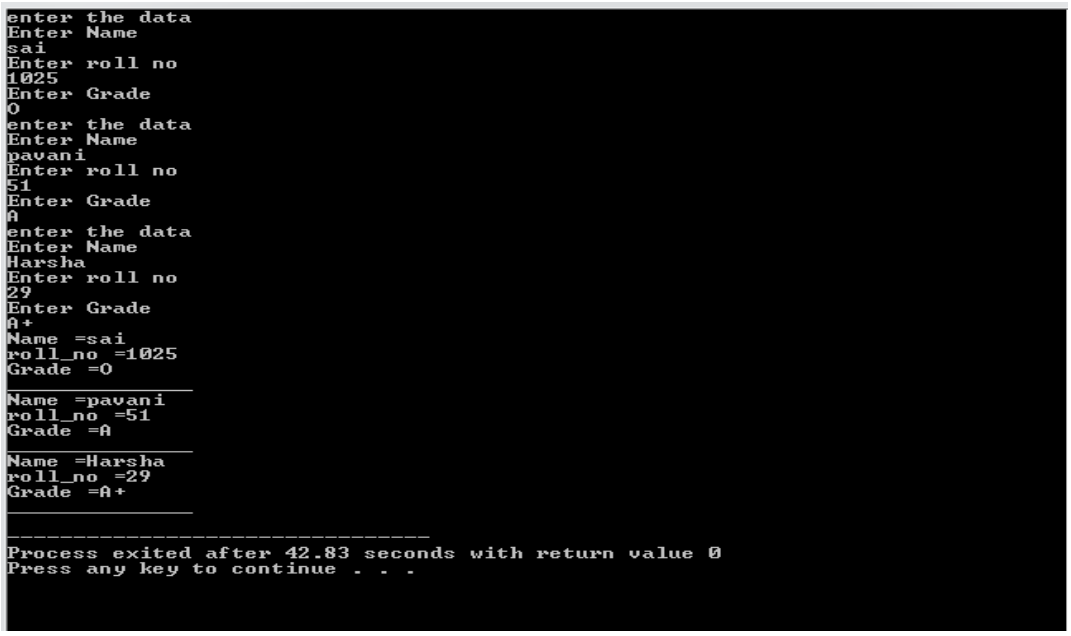
Aim: Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.

Program:

```
#include<iostream>

using namespace std;

class Student_data

{

        public:

        char name[20];

        int roll_no;

        char grade[5];

        void Read()

        {

                cout<<"enter the data"<<endl;

                cout<<"Enter Name \n";

                cin>>name;

                cout<<"Enter roll no \n";

                cin>>roll_no;

                cout<<"Enter Grade \n";

                cin>>grade;

        }

        display()

        {

                cout<<"Name ="<<name<<endl;

                cout<<"roll_no ="<<roll_no<<endl;

                cout<<"Grade ="<<grade<<endl;
```

```
                cout<<"_____\n";
        }


};
int main()
{
        Student_data s[3];
        int i;
        for(i=0;i<3;i++)
        {
                s[i].Read();
        }
        for(i=0;i<3;i++)
        {
                s[i].display();
        }
        return 0;
}
```

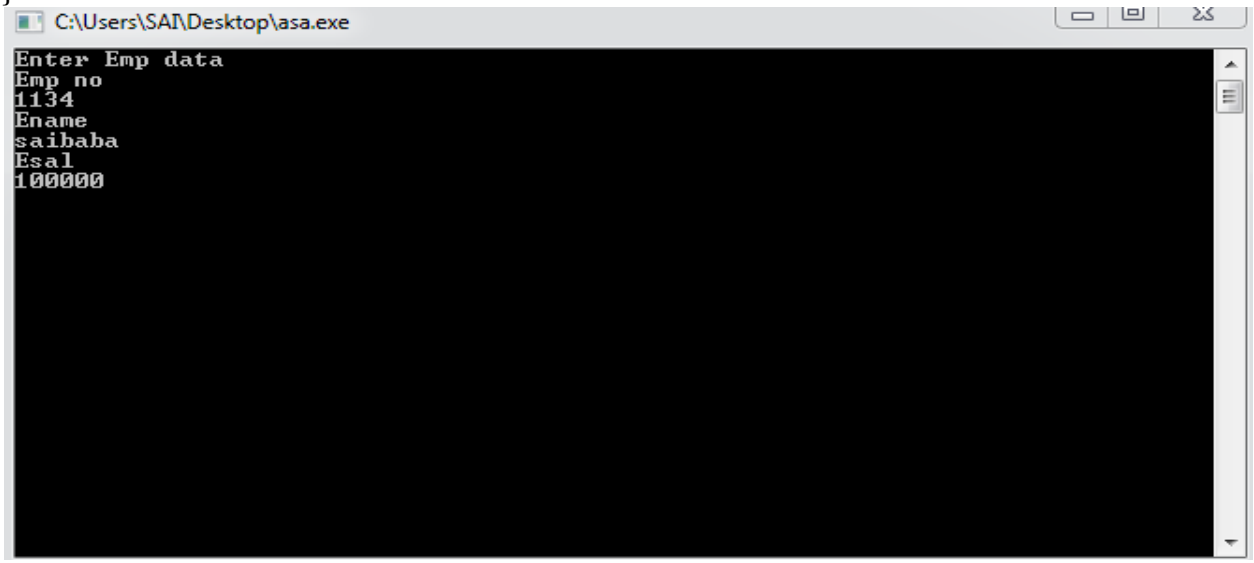


2.)Aim: Write a C++ program to declare Struct. Initialize and display contents of member variables.

Program:

```
#include<iostream>

using namespace std;

struct Emp

{

        int Eno;

        string  Ename;

        int sal;


};

int main()

{

        Emp e;

        cout<<"Enter Emp data \n";

        cout<<"Emp no \n";

        cin>>e.Eno;
```

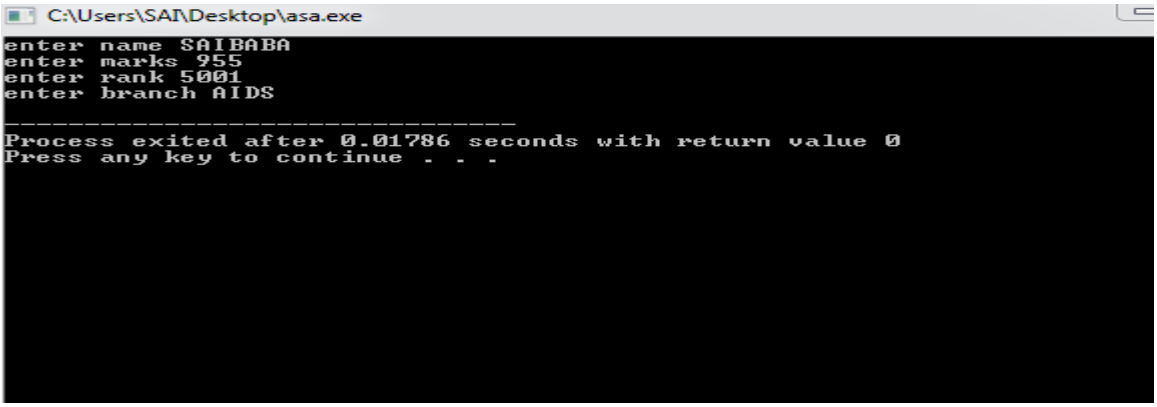
```
        cout<<"Ename \n";
        cin>>e.Ename;
        cout<<"Esal \n";
        cin>>e.sal;
        return 0;
}
```



3.)Aim: Write a C++ to illustrate the static methods and members.
Program:

```
#include<iostream>
using namespace std;
class Myclass
{
        public:
        static  string name;
        static int marks;
        static int Rank;
        static string branch;

        int static Data()
```

```
        {
                cout<<"enter name "<<name<<endl;
                cout<<"enter marks "<<marks<<endl;
                cout<<"enter rank "<<Rank<<endl;
                cout<<"enter branch "<<branch<<endl;

        }

};
string Myclass::name="SAIBABA";
int Myclass::marks=955;
int Myclass::Rank=5001;
string Myclass::branch="AIDS";
int main()
{
        Myclass::Data();
        return 0;}
```



4.)Aim: Write a C++ program to find the sum for the given variables using function with default arguments.

Program:

```
#include<iostream>

using namespace std;

class Myclass

{

        public:

        int a,b,c,sum;

        int Total(int x,int y,int z)

        {

                return sum=x+y+z;

        }

        void display()

        {

                Total(10,25,35);

                int final;

                final=sum;

                cout<<"total sum of values"<<final;




        }


};

int main()

{

        Myclass mc;
```

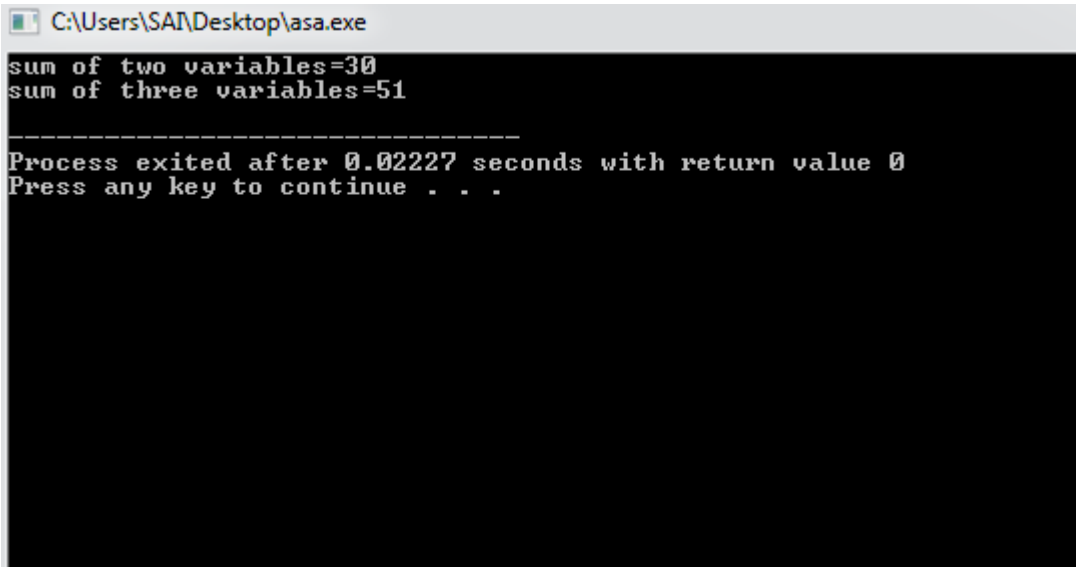
```
        //mc.Total(10,20,30);
        mc.display();

        //int final=mc.sum;
        return 0;
}
```

```
total sum of values70
--------------------------------
Process exited after 0.0208 seconds with return value 0
Press any key to continue . . .
```

5.)Aim: Write a C++ Program to illustrate Enumeration and Function Overloading
Program :

```
#include<iostream>
using namespace std;
class Student
{
        public:
        int Fun(int x,int y)
        {
                return x+y;
        }
        int Fun(int x,int y,int z)
        {
                return x+y+z;
        }
```

```
};

int main()
{
        Student s;
        enum Sum{a=10,b=20,c,d};//enumeration
        //cout<<"d value ="<<d;
        cout<<"sum of two variables="<<s.Fun(a,b)<<endl;
        cout<<"sum of three variables="<<s.Fun(a,b,c)<<endl;
        return 0;
}
```



6.)Aim: Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member

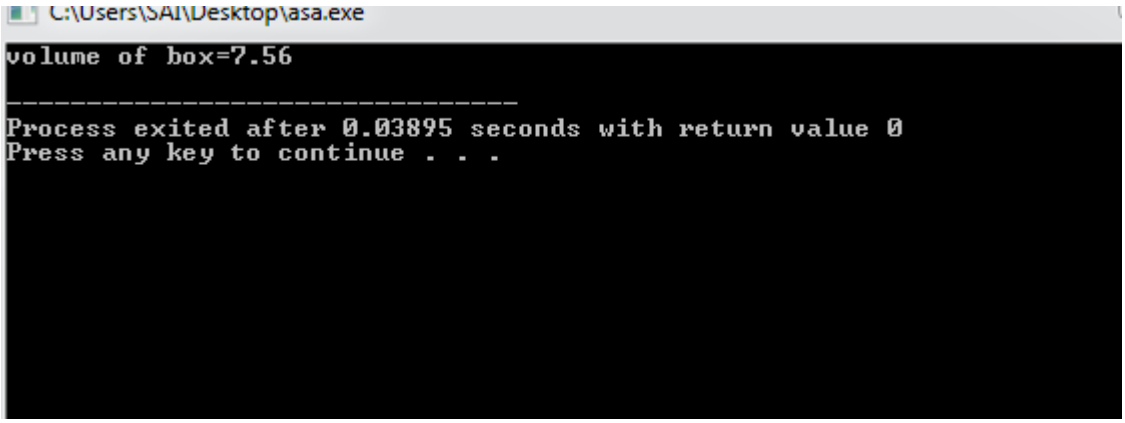
Program:

```
#include<iostream>

using namespace std;

class Box
{
        public:
```

```
        double length;
        double breadth;
        double height;
        public:
        int get_data(double l,double b,double h)
        {
                length=l;
                breadth=b;
                height=h;
        }
        double volume()
        {
                return length*breadth*height;
        }

};
int main()
{


        Box *ptr;
        Box d;
        ptr=&d;
        ptr->get_data(1.2,2.1,3.0);


        cout<<"volume of box="<<ptr->volume()<<endl;
```

return 0;

}



7)Aim: To write a C++ program to find the value of a number raised to its power that demonstrates a function using call by value.
Program:

```
#include<iostream>

#include<cmath>

using namespace std;

class Myclass

{

        public:

        int fun(int base,int powers)

        {


                return pow(base,powers);


        }


};

int main()

{
```

```
int x,y;
cout<<"enter a base values \n";
cin>>x;
cout<<"enter a power value \n";
cin>>y;
Myclass m;
cout<<"power of number value="<<m.fun(x,y)<<endl;
return 0;
}
```

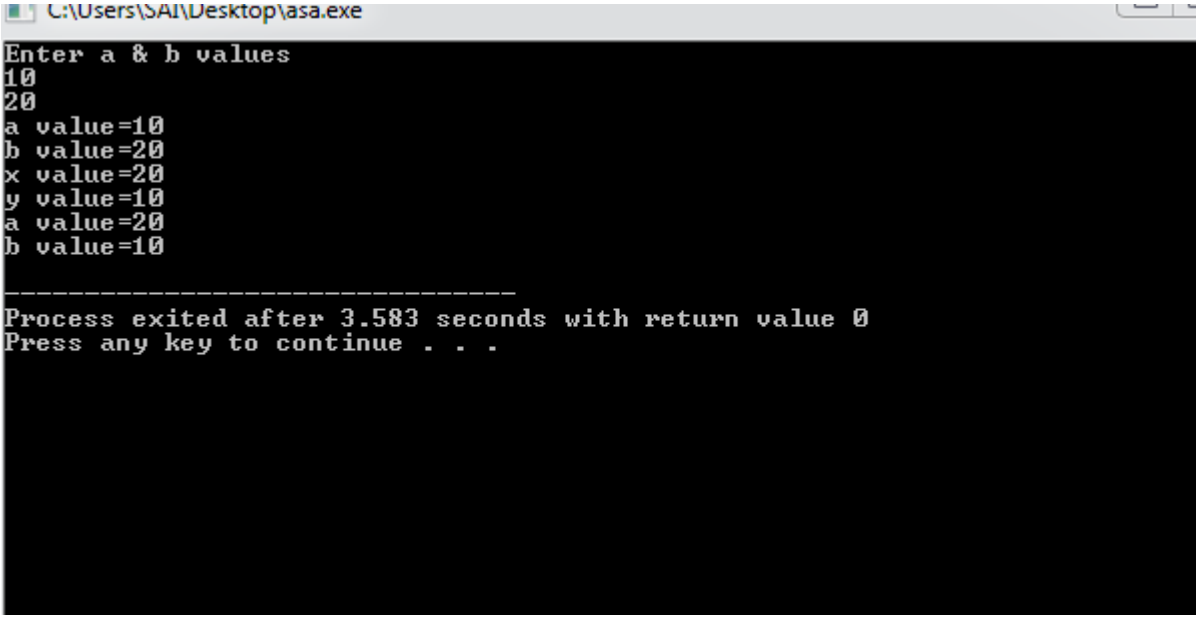
```
enter a base values
25
enter a power value
3
power of number value=15625

--------------------------------
Process exited after 9.338 seconds with return value 0
Press any key to continue . . .
```

8.)Aim: To write a C++ program and to implement the concept of Call by Address

Parogram:

```
#include<iostream>
using namespace std;
class Swap
{       public:
;
        int swap(int &x,int &y)
        {
                int z;
                z=x;
```

```
            x=y;
            y=z;
            cout<<"x value="<<x<<endl;
            cout<<"y value="<<y<<endl;

        }
};
int main()
{
        Swap s;
        int a,b;
        cout<<"Enter a & b values"<<endl;
        cin>>a>>b;
        cout<<"a value="<<a<<endl;
        cout<<"b value="<<b<<endl;
        s.swap(a,b);
        cout<<"a value="<<a<<endl;
        cout<<"b value="<<b<<endl;
        return 0;

}
```



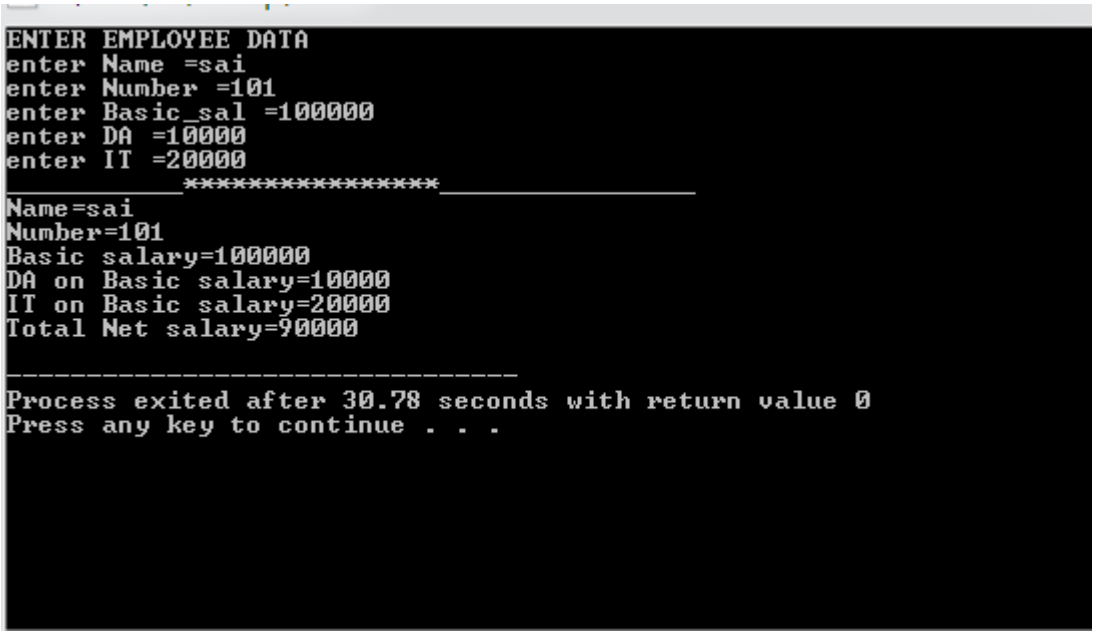
9.)Aim: Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members.

Program:

```
#include<iostream>

using namespace std;

class EMPLOYEE

{

        private:

        string Name;

        int Number;

        int Basic_sal;

        int DA;

        int IT;

        int Net_sal;

        public:

        void get_data()

        {
```

```
        cout<<"ENTER EMPLOYEE DATA \n";
        cout<<"enter Name =";
        cin>>Name;
        cout<<"enter Number =";
        cin>>Number;
        cout<<"enter Basic_sal =";
        cin>>Basic_sal;
        cout<<"enter DA =";
        cin>>DA;
        cout<<"enter IT =";
        cin>>IT;
//      cout<<"enter Net_sal =";
//      cin>>Net_sal;
                }
void Print()
{
        cout<<"___________**********_____________\n";
        Net_sal =Basic_sal+DA;
        Net_sal -=IT;
        cout<<"Name="<<Name<<endl;
        cout<<"Number="<<Number<<endl;
        cout<<"Basic salary="<<Basic_sal<<endl;
        cout<<"DA on Basic salary="<<DA<<endl;
        cout<<"IT on Basic salary="<<IT<<endl;
        cout<<"Total Net salary="<<Net_sal<<endl;
```

```
        }


};
int main()
{
        EMPLOYEE e;
        e.get_data();
        e.Print();
        return 0;
}
```



10)Aim:

Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).

Program:

```
#include<iostream>

using namespace std;

class EMPLOYEE

{
```

```
private:
string Name;
int Number;
int Basic_sal;
int DA;
int IT;
int Net_sal;
public:
void get_data(int i)
{
        cout<<"ENTER EMPLOYEE DATA "<<i+1<<"="<<endl;
        cout<<"enter Name =";
        cin>>Name;
        cout<<"enter Number =";
        cin>>Number;
        cout<<"enter Basic_sal =";
        cin>>Basic_sal;
        //cout<<"enter DA =";
        //cin>>DA;
        //cout<<"enter IT =";
        //cin>>IT;
//      cout<<"enter Net_sal =";
//      cin>>Net_sal;
}
void Find_net()
{
        DA=(0.52)*Basic_sal;
```

```
            IT=(0.30)*(Basic_sal+DA);
            //Net_sal =Basic_sal+DA;
            Net_sal =Basic_sal+DA-IT;
        }
        void Print()
        {
            cout<<"_____**********_____\n";

            cout<<"Name="<<Name<<endl;
            cout<<"Number="<<Number<<endl;
            cout<<"Basic salary="<<Basic_sal<<endl;
            cout<<"DA on Basic salary="<<DA<<endl;
            cout<<"IT on Basic salary="<<IT<<endl;
            cout<<"Total Net salary="<<Net_sal<<endl;



        }

};
int main()
{
        EMPLOYEE e[3];
        int i;
        for(i=0;i<3;i++)
        {
        e[i].get_data(i);
        }
```

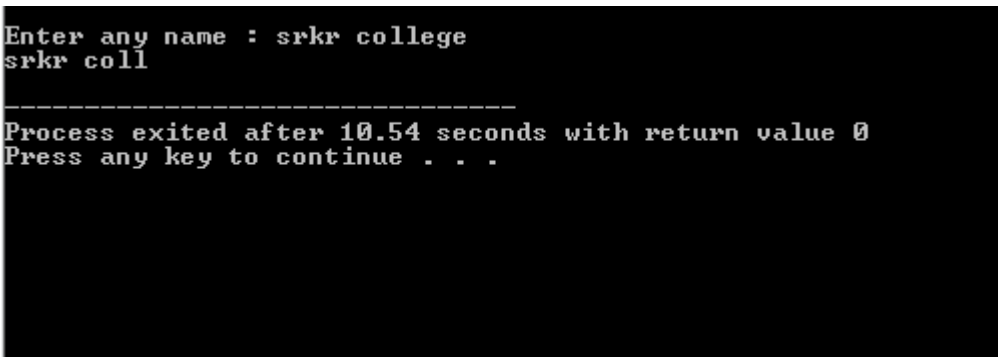
```
    for(i=0;i<3;i++)
    {
    e[i].Find_net();
    }
    for(i=0;i<3;i++)
    {
    e[i].Print();
    }

    return 0;
}
```

```
ENTER EMPLOYEE DATA 1=
enter Name =sai
enter Number =101
enter Basic_sal =100000
ENTER EMPLOYEE DATA 2=
enter Name =valli
enter Number =102
enter Basic_sal =200000
ENTER EMPLOYEE DATA 3=
enter Name =pavan
enter Number =103
enter Basic_sal =300000
__________**********_________________
Name=sai
Number=101
Basic salary=100000
DA on Basic salary=52000
IT on Basic salary=45600
Total Net salary=106400
__________**********_________________
Name=valli
Number=102
Basic salary=200000
DA on Basic salary=104000
IT on Basic salary=91200
Total Net salary=212800
__________**********_________________
Name=pavan
Number=103
Basic salary=300000
DA on Basic salary=156000
IT on Basic salary=136800
Total Net salary=319200

--------------------------------
```

11)Aim: Write a C++ to illustrate the concepts of console I/O operations.
Program:

```
#include <iostream>
```

```
#include<conio.h>

using namespace std;

int main() {

cout<<"\nEnter any name : ";

char c[10];

cin.getline(c,10); //It takes 10 charcters as input;

cout.write(c,9); //It reads only 9 character from buffer c;

cout<<"\n";

return 0;

}
```



12)Aim: Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels

Program:

```
#include <iostream>

#include<conio.h>

using namespace std;

int x;

int main() {

int x = 10; // Local x

cout << "\nValue of global x is " << ::x;

cout << "\nValue of local x is " << x <<endl;

return 0;
```

}

```
Value of global x is 0
Value of local x is 10

--------------------------------
Process exited after 0.01378 seconds with return value 0
Press any key to continue . . .
```

13)Aim: Write a C++ program to allocate memory using new operator.

Program:

```
#include<iostream>
using namespace std;
int main()
{
    int x,i,n;//one dimensional array []
    cout<<"Enter array size \n";
    cin>>n;
    cout<<"Enter memory size \n";
    int *p;
    p=new int[n];
    for(i=0;i<n;i++)
    {
        //p[i]=i+1;
        cout<<"store values in heap add \n";
        cin>>x;
        p[i]=x;
    }
    for(i=0;i<n;i++)
    {
```

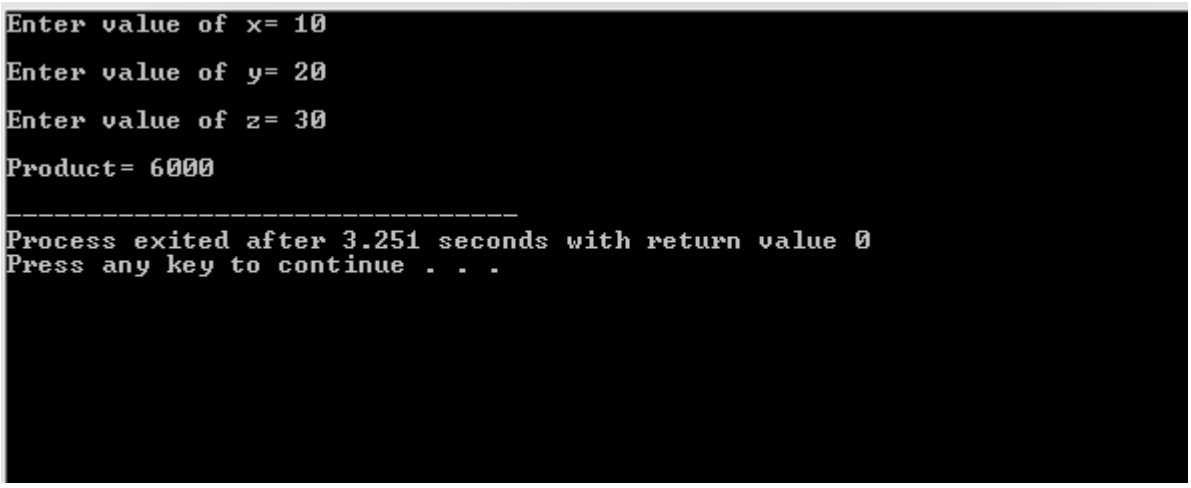
```
        cout<<p[i]<<" ";
    }delete p;
    return 0;
}
```

```
Enter array size
5
Enter memory size
store values in heap add
101
store values in heap add
102
store values in heap add
103
store values in heap add
104
store values in heap add
105
101 102 103 104 105
--------------------------------
Process exited after 10.95 seconds with return value 0
Press any key to continue . . .
```

14)Aim: Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3)
Program:

```
#include <iostream>
#include<conio.h>
using namespace std;
class base //single base class
{
public:
int x;
void getdata() {
cout << "Enter value of x= "; cin >> x;
}
};
class derive1 : public base // derived class from base class
{
public:
```

```
int y;
void readdata()
{
cout << "\nEnter value of y= "; cin >> y;
}
};
class derive2 : public derive1 // derived from class derive1
{
private:
int z;
public:
void indata() {
cout << "\nEnter value of z= "; cin >> z;
}
void product() {
cout << "\nProduct= " << x * y * z <<endl;
}
};
int main() {
derive2 a; //object of derived class
a.getdata();
a.readdata();
a.indata();
a.product();
return 0;
}
```



15)Aim: Write a C++ program to create an array of pointers. Invoke functions using array objects.

Program:

```
#include <iostream>
#include <string>
using namespace std;
class Student
{
string name;
int marks;
public:
void getName()
{
getline( cin, name );
}
void getMarks()
{
cin >> marks;
}
```

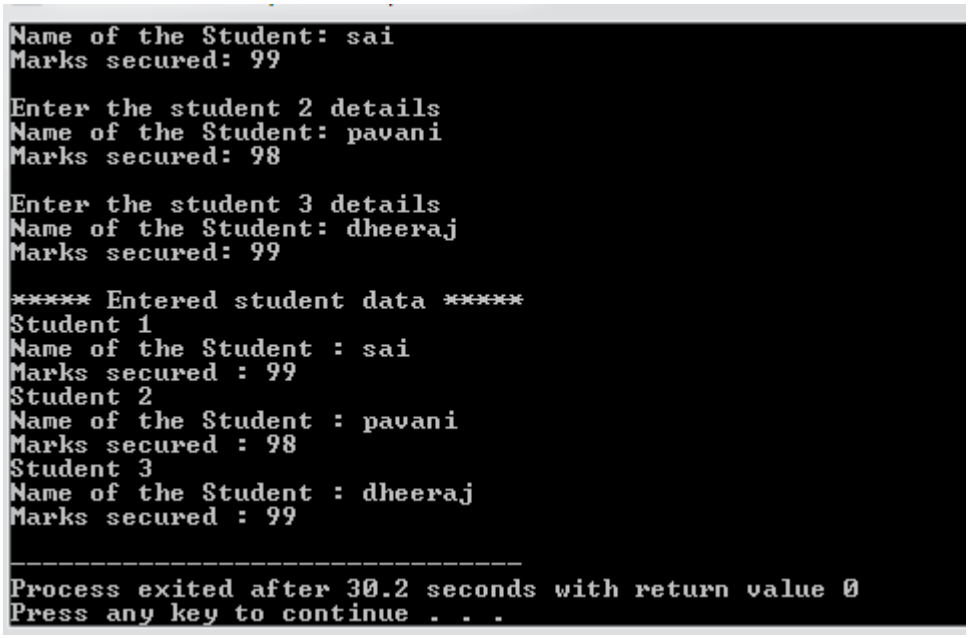
```
void displayInfo()
{
cout << "Name : " << name << endl;
cout << "Marks : " << marks << endl;
}
};
int main()
{
Student st[5].*ptr;
ptr=&st;
for( int i=0; i<5; i++ )
{
cout << "Student " << i + 1 << endl;
cout << "Enter name" << endl;
st[i]->getName();
cout << "Enter marks" << endl;
st[i]->getMarks();
}
for( int i=0; i<5; i++ )
{
cout << "Student " << i + 1 << endl;
st[i]->displayInfo();
}
return 0;
}
```

15)Aim: Write a C++ program to create an array of pointers. Invoke functions using array objects.

Program:

```
#include <iostream>

using namespace std;


class Student

{

   string stud_name;

   int marks;

   public:

      void getStudentInfo(int i)

      {

         cout<< endl << "Enter the student " << i << " details" << endl;

         cout<< "Name of the Student: ";

         cin>> stud_name;

         cout<< "Marks secured: ";

         cin>> marks;

      }

      void displayStudentInfo()

      {

         cout << "Name of the Student : " << stud_name << endl;

         cout << "Marks secured : " << marks << endl;

      }

};

int main()

{

   Student stud[5],*ptr;
```

```
    int class_size;
    ptr=stud;

    cout<< "Enter the number of students in the class : ";
    cin>> class_size;

    for( int i=1; i<=class_size; i++ )
    {
       (ptr+i)->getStudentInfo(i);
    }

    cout<< endl << "***** Entered student data *****" << endl;

    for( int i=1; i<=class_size; i++ )
    {
       cout << "Student " << i << endl;
       (ptr+i)->displayStudentInfo();
    }
    return 0;
}
```



16)Aim: Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword

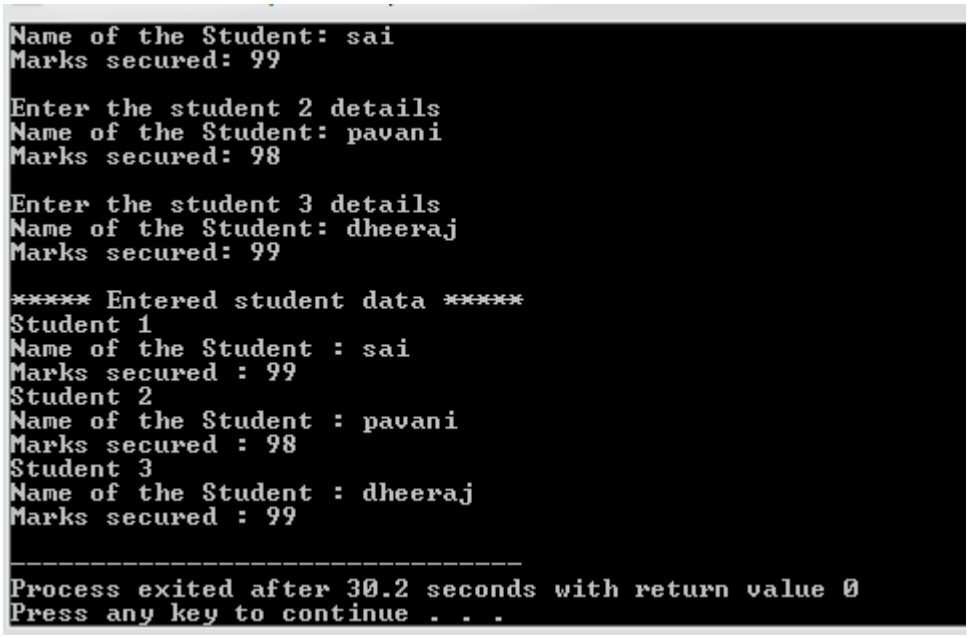
Program:

15)Aim: Write a C++ program to create an array of pointers. Invoke functions using array objects.

Program:

```
#include <iostream>

using namespace std;


class Student

{

   string stud_name;

   int marks;

   public:

      void getStudentInfo(int i)

      {

         cout<< endl << "Enter the student " << i << " details" << endl;

         cout<< "Name of the Student: ";

         cin>> stud_name;

         cout<< "Marks secured: ";

         cin>> marks;

      }

      void displayStudentInfo()

      {

         cout << "Name of the Student : " << stud_name << endl;

         cout << "Marks secured : " << marks << endl;

      }

};

int main()

{
```

```
    Student stud[5],*ptr;
    int class_size;
    ptr=stud;

    cout<< "Enter the number of students in the class : ";
    cin>> class_size;

    for( int i=1; i<=class_size; i++ )
    {
       (ptr+i)->getStudentInfo(i);
    }

    cout<< endl << "***** Entered student data *****" << endl;

    for( int i=1; i<=class_size; i++ )
    {
       cout << "Student " << i << endl;
       (ptr+i)->displayStudentInfo();
    }
    return 0;
}
```



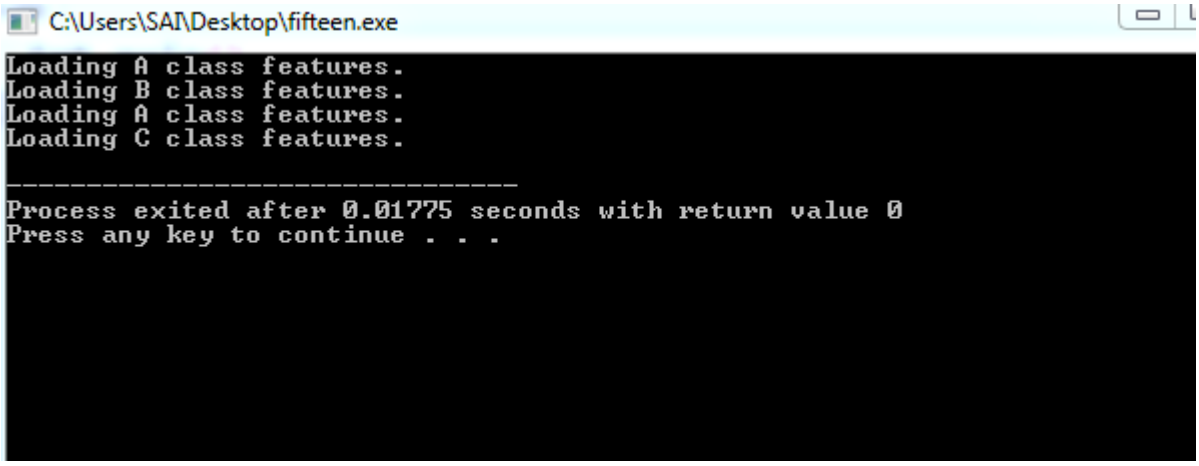
16)Aim: Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword

Program:

```
#include <iostream>

using namespace std;

class A {

public:

virtual void features()

{

 cout << "Loading A class features.\n";

 }

};


class B : public A

 {

public:

void features()

{
```

```
this->A::features();
cout << "Loading B class features.\n";
}
};

class C : public A
{
public:
void features()
{
this->A::features();
cout << "Loading C class features.\n";
}
};

class Loader
{
public:
void loadFeatures(A *a)
{
a->features();
}
};

int main()
{
Loader *l = new Loader;
```

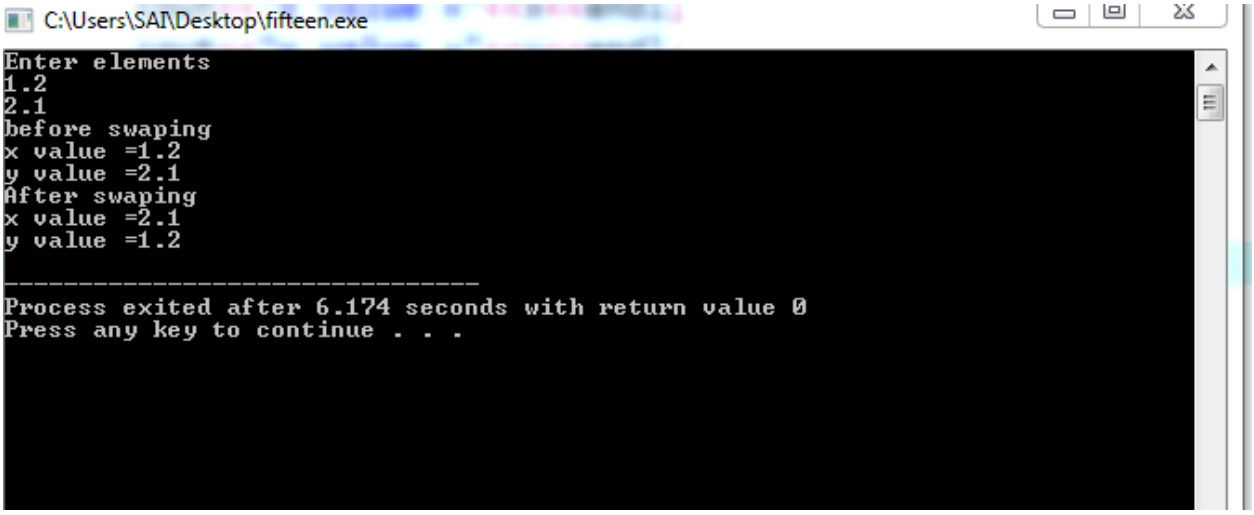
```
A *a;


B b;

C c;

a = &b;


l->loadFeatures(a);


a = &c;


l->loadFeatures(a);

return 0;

}
```



17)Aim: Write a C++ program for swapping two values using function templates
Program:

```
#include<iostream>

using namespace std;

template<class T>

T Swap(T x,T y)
```

```
{
        T temp;
        temp=x;
        x=y;
        y=temp;
        cout<<"After swaping \n";
        cout<<"x value ="<<x<<endl;
        cout<<"x value ="<<x<<endl;

}
int main()
{
        int x,y;
        cout<<"Enter elements \n";
        cin>>x>>y;

        cout<<"before swaping \n";
        cout<<"x value ="<<x<<endl;
        cout<<"x value ="<<x<<endl;
        return 0;

}
```


C:\Users\SAI\Desktop\fifteen.exe
Enter elements
1.2
2.1
before swaping
x value =1.2
y value =2.1
After swaping
x value =2.1
y value =1.2

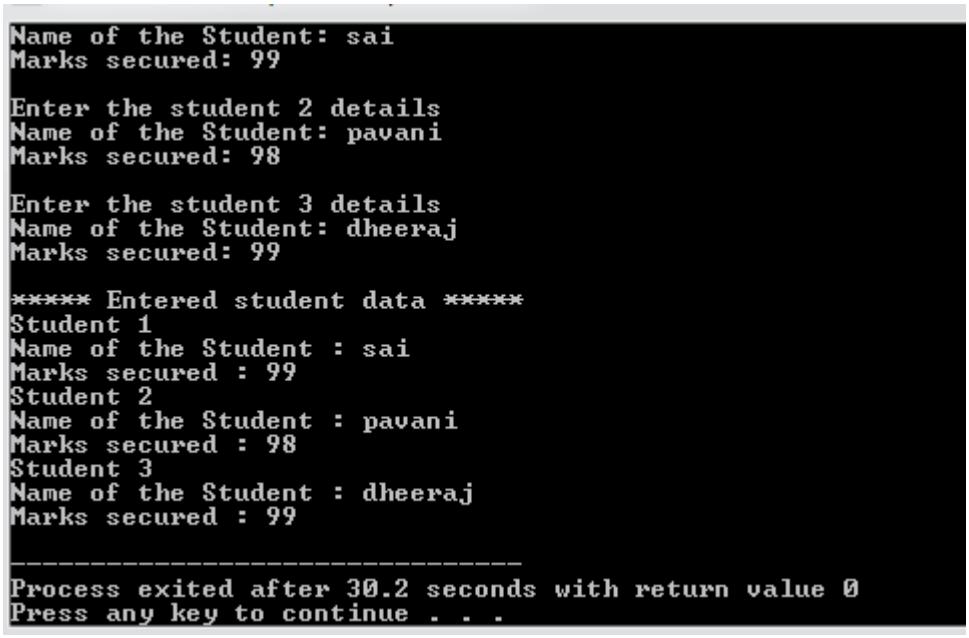
Process exited after 6.174 seconds with return value 0
Press any key to continue . . .

15)Aim: Write a C++ program to create an array of pointers. Invoke functions using array objects.

Program:

```
#include <iostream>

using namespace std;


class Student

{

   string stud_name;

   int marks;

   public:

      void getStudentInfo(int i)

      {

         cout<< endl << "Enter the student " << i << " details" << endl;

         cout<< "Name of the Student: ";

         cin>> stud_name;

         cout<< "Marks secured: ";

         cin>> marks;

      }

      void displayStudentInfo()

      {

         cout << "Name of the Student : " << stud_name << endl;

         cout << "Marks secured : " << marks << endl;

      }

};

int main()

{
```

```
    Student stud[5],*ptr;
    int class_size;
    ptr=stud;

    cout<< "Enter the number of students in the class : ";
    cin>> class_size;

    for( int i=1; i<=class_size; i++ )
    {
       (ptr+i)->getStudentInfo(i);
    }

    cout<< endl << "***** Entered student data *****" << endl;

    for( int i=1; i<=class_size; i++ )
    {
       cout << "Student " << i << endl;
       (ptr+i)->displayStudentInfo();
    }
    return 0;
}
```



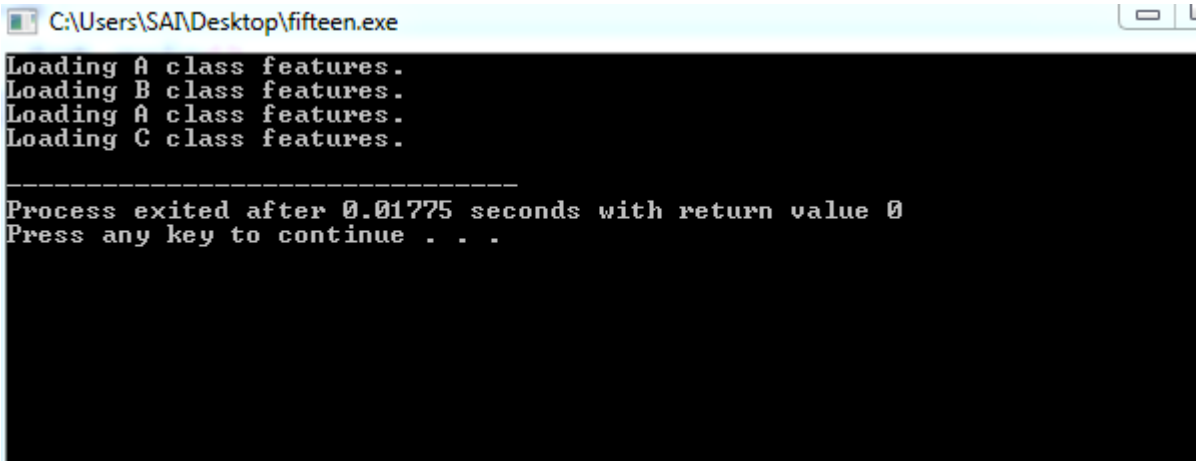
16)Aim: Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword

Program:

```
#include <iostream>

using namespace std;

class A {

public:

virtual void features()

{

 cout << "Loading A class features.\n";

 }

};


class B : public A

 {

public:

void features()

{
```

```
this->A::features();
cout << "Loading B class features.\n";
}
};

class C : public A
{
public:
void features()
{
this->A::features();
cout << "Loading C class features.\n";
}
};

class Loader
{
public:
void loadFeatures(A *a)
{
a->features();
}
};

int main()
{
Loader *l = new Loader;
```

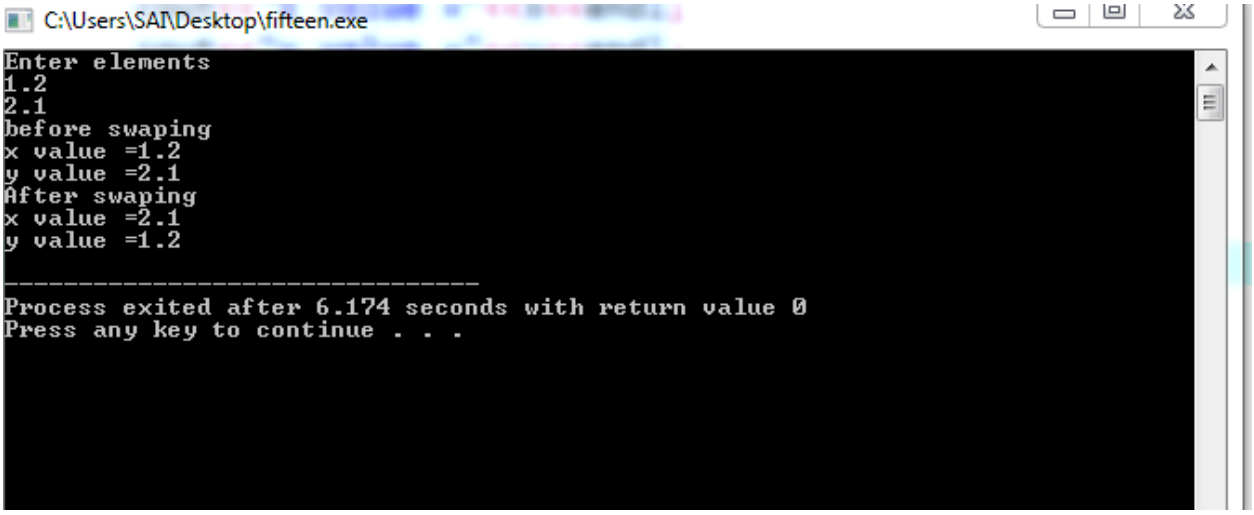
```
A *a;


B b;

C c;

a = &b;


l->loadFeatures(a);


a = &c;


l->loadFeatures(a);

return 0;

}
```



17)Aim: Write a C++ program for swapping two values using function templates
Program:

```
#include<iostream>

using namespace std;

template<class T>

T Swap(T x,T y)
```

```
{
        T temp;
        temp=x;
        x=y;
        y=temp;
        cout<<"After swaping \n";
        cout<<"x value ="<<x<<endl;
        cout<<"x value ="<<x<<endl;

}
int main()
{
        int x,y;
        cout<<"Enter elements \n";
        cin>>x>>y;

        cout<<"before swaping \n";
        cout<<"x value ="<<x<<endl;
        cout<<"x value ="<<x<<endl;
        return 0;

}
```


C:\Users\SAI\Desktop\fifteen.exe
Enter elements
1.2
2.1
before swaping
x value =1.2
y value =2.1
After swaping
x value =2.1
y value =1.2

Process exited after 6.174 seconds with return value 0
Press any key to continue . . .