OOP LABORATORY 4

Name: **ANIRBAN HAZRA**
Section: **B-12**
Roll : **2005643**

1) WAP to swap private data member of two classes. [The classes have no relation with each other].

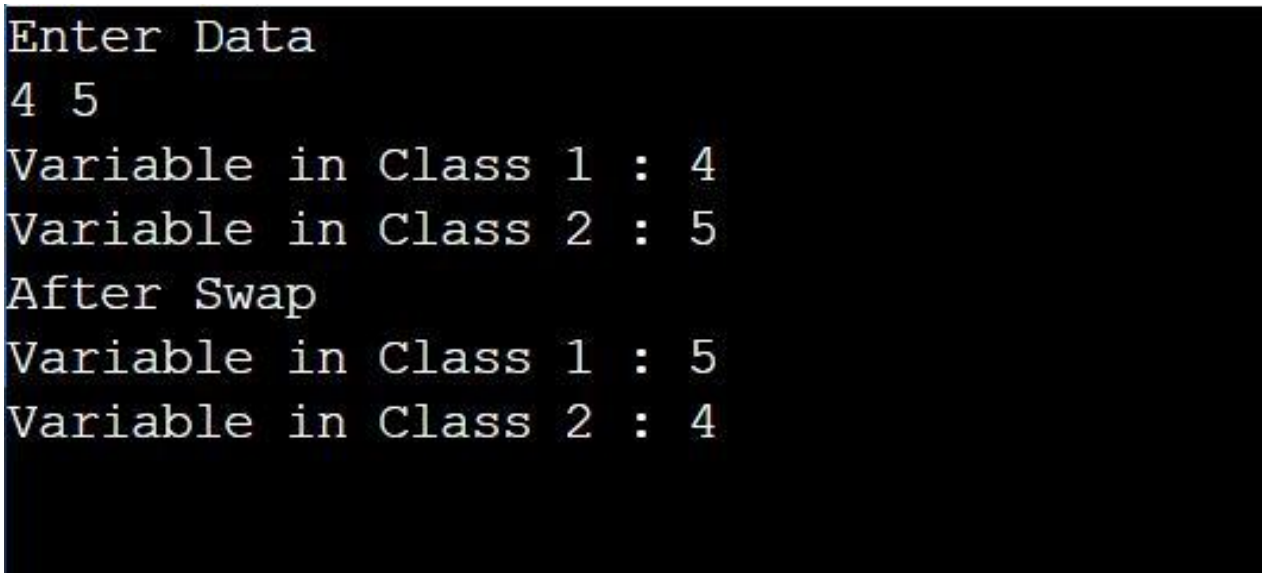
PROGRAM CODE:-

```
#include <iostream>

using namespace std;
class test2;
class test1
{
        private:
        int a;
        public:
        void display();
        friend void swap(test1 &, test2 &);
        friend void input(test1 &, test2 &);
};
class test2
{
        private:
        int a;
        public:
        void display();
        friend void swap(test1 &, test2 &);
        friend void input(test1 &, test2 &);
};
void  test1::display()
{
        cout<<"Variable in Class 1 : "<<a<<endl;
}
void  test2::display()
{
        cout<<"Variable in Class 2 : "<<a<<endl;
}
void input(test1 &t1, test2 &t2)
{
        cin>>t1.a>>t2.a;
}
```

```
void  swap(test1 &t1, test2 &t2)
{
        int temp= t1.a;
        t1.a=t2.a;
        t2.a=temp;
}
int main()
{
        test1 t1;
        test2 t2;
        cout<<"Enter Data"<<endl;
        input(t1,t2);
        t1.display();
        t2.display();
        cout<<"After Swap"<<endl;
        swap(t1,t2);
        t1.display();
        t2.display();
}
```

OUTPUT:-



2) Modify program 1) to make the swap function member of one class and friend to another class.

PROGRAM CODE:-

```
#include <iostream>

using namespace std;
class test2;
```

```
class test1
{
	private:
	int a;
	public:
	void display();
	void swap(test2 &);
	friend void input(test1 &, test2 &);
};
class test2
{
	private:
	int a;
	public:
	void display();
	friend void test1::swap(test2 &);
	friend void input(test1 &, test2 &);
};
void  test1::display()
{
	cout<<"Variable in Class 1 : "<<a<<endl;
}
void  test2::display()
{
	cout<<"Variable in Class 2 : "<<a<<endl;
}
void input(test1 &t1, test2 &t2)
{
	cin>>t1.a>>t2.a;
}
void test1::swap(test2 &t1)
{
	int temp= t1.a;
	t1.a=this->a;
	this->a=temp;
}
int main()
{
	test1 t1;
	test2 t2;
	cout<<"Enter Data"<<endl;
	input(t1,t2);
	t1.display();
	t2.display();
	cout<<"After Swap"<<endl;
	t1.swap(t2);
	t1.display();
	t2.display();
}
```

OUTPUT:-

3) Create two classes which stores distance in feet, inches and meter, centimeter format respectively. Write a function which compares distance in object of these classes and displays the larger one.

PROGRAM CODE:-

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

class Met;
class inc
{
   float feet, inches;
   public:
   float total;
   void getdata()
   {
      cout<<"Enter the distance in feet and inches: "<<endl;
      cin>>feet;
      cin>>inches;
      total = (feet * 12) + inches;
      total = total * 2.54;
   }
   friend void calc(Met, inc);
};
class Met
  {
   float met, cent;
   public:
   float total;
   void getdata()
   {
```

```
        cout<<"Enter the distance in metres and centimetres: "<<endl;
        cin>>met;
        cin>>cent;
        total = met * 100 + cent;
    }
    friend void calc(Met, inc);
};

void calc(Met m, inc i)
{
    if(m.total > i.total)
    {
        cout<<"The larger of distances is "<<m.met<<" metres and "<<m.cent<<" centimetres."<<endl;
    }
    Else
    {
        cout<<"The larger of distances is "<<i.feet<<" feet and "<<i.inches<<" inches."<<endl;
    }
}

int main()
{
    Met m;
    inc i;
    m.getdata();
    i.getdata();
    calc(m, i);
    return 0;
}
```

OUTPUT:-

```
Enter the distance in metres and centimetres:
5 60
Enter the distance in feet and inches:
5 6
The larger of distances is 5 metres and 60 centimetres.
```

4) Create a class which stores name, author and price of a book. Store information for n number of books. Display information of all the books in a given price range using friend function.

PROGRAM CODE:-

```
#include <iostream>
using namespace std;
class Amount;
class Book
{
   char name[51];
   char author[51];
   public:
   void read(Book* b);
   friend void display(Book* b,Amount* a,int u,int l);
};
class Amount
{
   int price;
   public:
   void read(Amount* a);
   friend void display(Book* b,Amount* a,int u,int l);
};

void Book::read(Book* b)
{
   cout<<"Enter the name of the book: ";
   cin>>b->name;
   cout<<"Enter the Author's name: ";
   cin>>b->author;
}
void Amount::read(Amount* b)
{
   cout<<"Enter the price of the book: ";
   cin>>b->price;
}

void display(Book* b,Amount* a,int u, int l)
{
   if((a->price >= l) && (a->price <= u))
   {
      cout<<"Book Name: "<<b->name<<endl;
      cout<<"Author Name: "<<b->author<<endl;
      cout<<"Price Name: "<<a->price<<endl;
   }
}

int main()
{
   Book *o_book;
   Amount *a;
```

```
    int size;
    cout<<"Enter number of books: ";
    cin>>size;
    o_book = new Book[size];
    a = new Amount[size];
    cout<<"Enter the input for all books: "<<endl;
    for(int i = 0;i<size;i++)
    {
            cout<<i+1<<": "<<endl;
            o_book->read(&o_book[i]);
           a->read(&a[i]);
    }
    cout<<endl<<"Details of books: "<<endl;
    int lower,upper;
    cout<<"Enter the range: ";
    cin>>lower>>upper;
    for(int i = 0;i<size;i++)
    {
            display(&o_book[i],&a[i],upper,lower);
    }

    delete [] o_book;
    delete [] a;
    return 0;
}
```

OUTPUT:-

```
Enter number of books: 4
Enter the input for all books:
1:
Enter the name of the book: Name1
Enter the Author's name: Author1
Enter the price of the book: 120
2:
Enter the name of the book: Name2
Enter the Author's name: Author2
Enter the price of the book: 500
3:
Enter the name of the book: Name3
Enter the Author's name: Author3
Enter the price of the book: 1200
4:
Enter the name of the book: Name4
Enter the Author's name: Author4
Enter the price of the book: 700

Details of books:
Enter the range: 100 999
Book Name: Name1
Author Name: Author1
Price Name: 120
Book Name: Name2
Author Name: Author2
Price Name: 500
Book Name: Name4
Author Name: Author4
Price Name: 700
```

5) Create a class complex which store real and imaginary part of a complex number. WAP to implement the following using friend function.
-add two objects of complex class.
-subtract one complex number from another
-find modulus of a complex number (√(r2 + i2)).

PROGRAM CODE:-

```
#include<iostream>
#include<math.h>

using namespace std;

class complex
{
        int real, imag;

public:

  void input ()
  {
   cout << "\nEnter real and imag part of the complex number: ";
   cin >> real >> imag;
  }


  friend complex subtract (complex, complex);

  void display ()
  {
  cout << "\nThe difference of complex numbers is: " << real << "+i" << imag;
  }


  friend void modl(complex a)
  {
       cout<<"\nModulus is : "<<sqrt((a.real*a.real)+(a.imag*a.imag))<<endl;
  }

};


complex subtract (complex a, complex b)
{
         complex t;
         t.real = a.real - b.real;
        t.imag = a.imag - b.imag;
         return t;
}
```

```
int main ()
{
  complex a, b, c;
  a.input ();
  b.input ();

  c = subtract (a, b);
  c.display ();

  cout<<"\nModulus of 1st complex";
  modl(a);

  cout<<"\nModulus of 2nd complex";
  modl(b);

  cout<<"\nModulus of 3rd complex";
  modl(c);

  return (0);
}
```

OUTPUT:-

```
Enter real and imag part of the complex number: 7 6

Enter real and imag part of the complex number: 5 4

The difference of complex numbers is: 2+i2
Modulus of 1st complex
Modulus is : 9.21954

Modulus of 2nd complex
Modulus is : 6.40312

Modulus of 3rd complex
Modulus is : 2.82843
```

6) WAP to create a class 'A' with an integer data member. Create another class ' B' with one integer data. Using friend class do the following.
-Take the input for the data members of both the classes using a member function of Class B.
-Display the data members of both the classes using a member function of Class B.
-Add the data members of both the classes and display the result.

PROGRAM CODE:-

```
#include<iostream>
using namespace std;
class A
{
        private:
                int x;
        public:
                friend class C;
};

class B
{
        private:
                int y;
        public:
                friend class C;
};

class C
{
private:
        int a;                  //Will store the value of x from class B
        int b;                  //Will store the value of y from class C
public:
  void inputAB ()
  {
    A s;
    B r;
      a = s.x;
      b = r.y;
      cout << "Enter the value of x\n";
      cin >> a;

      cout << "Enter the value of y\n";
      cin >> b;

  }

  void displayAB ()
  {
     cout << "The value of x is:  " << a << "\nThe value of y is:  " << b <<endl;
  }
  void addAB ()
  {

    cout << "The addition of " << a << " and " << b << " is " << a +b << endl;
  }
};
```

```
int main ()
{
  C c;
  c.inputAB ();
  c.displayAB ();
  c.addAB ();
  return 0;
}
```

OUTPUT:-

```
Enter the value of x
67
Enter the value of y
78
The value of x is:  67
The value of y is:  78
The addition of 67 and 78 is 145
```

7) Create a class with an integer data member. Include functions for input and output in class. Count the number of times each function is called and display it.

PROGRAM CODE:-

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

class Data
{
private:
  int x;
  static int Input;
  static int Output;

public:
  void input ();
  void output ();
  void display ();
};

int Data::Input = 0;
int Data::Output = 0;
```

```
void Data::input ()
{
  cout << "Enter the value of x\n";
  cin >> x;
  Input++;
}

void Data::output ()
{
  cout << "The value of x is " << x << endl;
  Output++;
}

void Data::display ()
{
  cout << "The input function has been called  " << Input << " times\n";
  cout << "The output function has been called  " << Output << " times\n";
}

int main ()
{
  Data d;

  d.input ();
  d.output ();

  d.input ();
  d.output ();

  d.display ();
}
```

OUTPUT:-

```
Enter the value of x
56
The value of x is 56
Enter the value of x
78
The value of x is 78
The input function has been called  2 times
The output function has been called  2 times
```

8) Create a class which stores name, roll number and total marks for a student. Input data for n students. Find the average marks scored by n students, store it as a data member of the class and display it using a function which may be called without object.

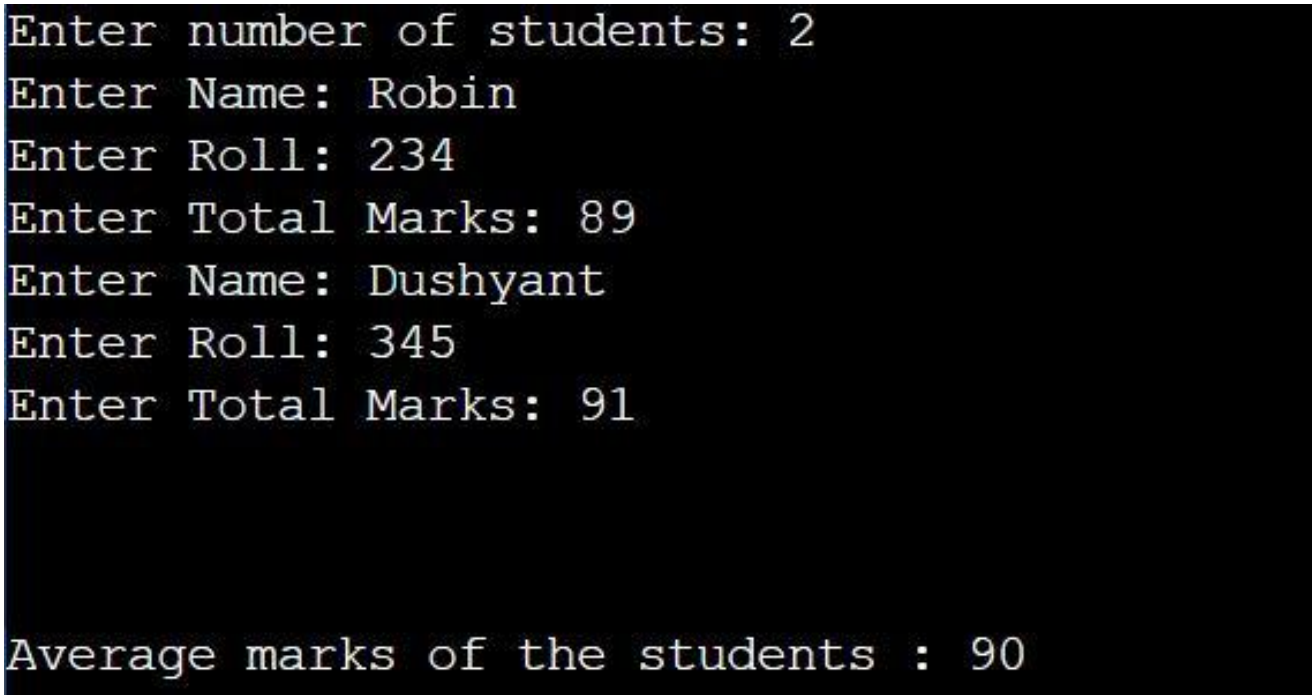
PROGRAM CODE:-

```
#include <iostream>
using namespace std;
class students
{
        string name;
        int roll;
        int tmarks;
        static int avgMarks;

        public:
                void getdata()
                {
                        cout<<"Enter Name: ";
                        cin>>name;
                        cout<<"Enter Roll: ";
                        cin>>roll;
                        cout<<"Enter Total Marks: ";
                        cin>>tmarks;
                        }
         static void display(students obj[],int n)
         {
            for(int i=0; i<n; i++)
            {
               avgMarks=avgMarks+obj[i].tmarks;
            }
            cout<<"\nAverage marks of the students : "<<(avgMarks/n);

         }
};
int students::avgMarks=0;
int main()
{
   int n;
   cout<<"Enter number of students: ";
   cin>>n;
   students ob[n];
   for(int i=0; i<n; i++)
   {
      ob[i].getdata();
   }
   cout << "\n\n";
   students::display(ob,n);
        return 0;
}
```

OUTPUT:-


Enter number of students: 2
Enter Name: Robin
Enter Roll: 234
Enter Total Marks: 89
Enter Name: Dushyant
Enter Roll: 345
Enter Total Marks: 91
Average marks of the students : 90