Bangladesh University of Business and Technology

(BUBT)

Lab Report

Lab Report No: 05

Course: Object Oriented Programming Lab

Course Code:CSE 112

Date of Submission:

Submitted By:
Name: Argho Ghosh
ID: 20254103276
Intake: 56
Section: 05
Program: B.Sc. in CSE

Bangladesh University of Business and Technology (BUBT)

Submitted To:
Iffat Ara Sanzida
Lecturer
Department of Computer Science & Engineering

Bangladesh University of Business and Technology (BUBT)

Teacher's Signature

Task 01*: Create a class Complex with private data members real and imag to represent a complex number. The class should provide public member functions to input and display complex numbers. Write a friend function named addComplex() that takes two Complex objects as parameters, accesses their private data members, adds the corresponding real and imaginary parts, and returns a new Complex object containing the result. In the main program, input two complex numbers from the user, call the friend function to compute their sum, and display the resulting complex number in the form a + bi.*

Code:

```
#include<bits/stdc++.h>
using namespace std;
class Complex
{
private:
int real;
int img;
public:
Complex(){}
Complex(int r, int i)
{
real = r;
img = i;
}
void display()
{
cout<<"Output : "<< real <<" + "<<img<<"i"<<endl;
```

```
}
friend Complex addComplex(Complex c1, Complex c2);
};
Complex addComplex(Complex c1, Complex c2)
{
int sumreal = c1.real + c2.real;
int sumimg = c1.img + c2.img;
Complex c3(sumreal,sumimg);
return c3;
}
int main()
{
Complex s1(2,6);
Complex s2(7,8);
Complex s3;
s3 = addComplex(s1,s2);
s3.display();
}
```

Output:

Task 02: *Write a C++ program to demonstrate the concept of function overloading by defining an overloaded function named Max(). The function should perform different operations based on the number and type of parameters passed to it. The Max() function must be overloaded to: (i) find the maximum of two integers, (ii) find the maximum of three floating-point numbers,*

and (iii) find the maximum element in an array of integers, given the array and its size. In the
main() function, take the necessary inputs from the user, call the appropriate overloaded
versions of the Max() function, and display the results to clearly illustrate function overloading in
C++.

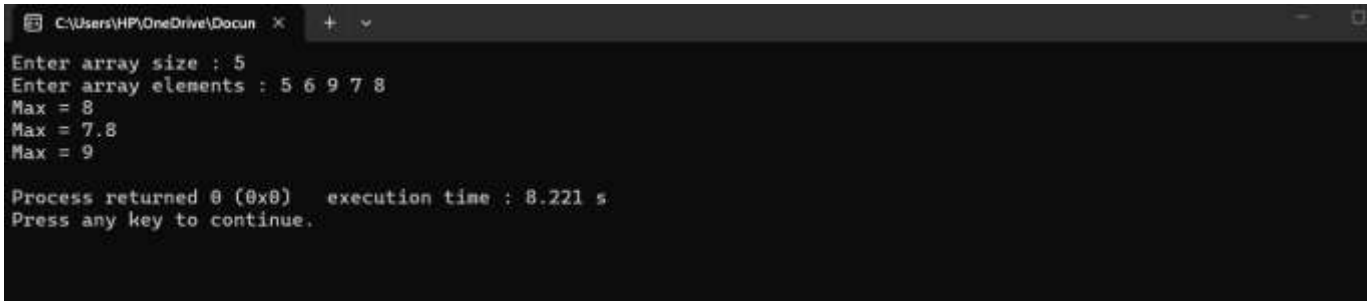
Code:

```
#include<bits/stdc++.h>
using namespace std;
class Overload
{
public:
int Maximum(int a, int b)
{
return max(a,b);
}
float Maximum(float a, float b, float c)
{
return max({a,b,c});
}
int Maximum(int a[], int n)
{
int mx = INT_MIN;
for(int i=0; i<n; i++)
{
mx = max(mx, a[i]);
}
return mx;
}
};
int main()
```

```
{
cout<<"Enter array size : ";
int n;
cin>>n;
int a[n];
cout<<"Enter array elements : ";
for(int i=0; i<n; i++)
{
cin>>a[i];
}
Overload s1;
cout<<"Max = "<<s1.Maximum(2,8)<<endl;
cout<<"Max = "<<s1.Maximum(2.6,6.8,7.8)<<endl;

cout<<"Max = "<<s1.Maximum(a,n)<<endl;
}
```

Output:



Task 03*: Create a class Student with attributes: name, id, and marks. The program should create an array (or list) of 5 Student objects, take input for their details, and perform the following operations:*

1. Calculate and display the average marks of all students.

2. Ask the user to input a threshold mark.

3. Display the names and marks of students who scored above the given

threshold.
To achieve this, include a member function inside the Student class that displays details of
students whose marks are greater than the user-given value.

Code:

```
#include<bits/stdc++.h>
using namespace std;
class student
{
public:
   string name;
   int id;
   int mark;
   void set_mark(int m)
   {
      mark = m;
   }
   int get_mark()
   {
      return mark;
   }
   student() {};
   student(string s, int i, int m)
   {
      name = s;
      id = i;
      mark = m;
   }
   void input()
   {
```

```
        cout<<"Enter name: ";
        cin>>name;
        cout<<"Enter id: ";
        cin>>id;
    }
    void display()
    {
        cout<<"Name : "<<name<<endl;
        cout<<"ID : "<<id<<endl;
        cout<<"Mark : "<<mark<<endl;

    }

};
int main()
{
    cout<<"Enter the value of n: ";
    int n;
    cin>>n;
    int num;
    cout<<"Enter a mark to checked out: ";
    cin>>num;
    student s[n];
    int sum = 0;
    for(int i=0; i<n; i++)
    {
        s[i].input();
        int m;
        cout<<"Enter mark : ";
        cin>>m;
        s[i].set_mark(m);
        sum += m;
```

```
    }
    for(int i=0; i<n; i++)
    {
        if(s[i].get_mark()>num)
        {
            s[i].display();
        }
    }
    cout<<"Average = "<<(double)sum/n;
}
```

Output: