

## Project 0

### Question 1

Create a class called **Complex** for performing arithmetic with complex numbers. Use the given driver program to test your class. Complex numbers have the form

$$\text{realPart} + \text{imaginaryPart} * i$$

where  $i$  is  $(-1)^{1/2}$ .

Use integer or floating-point to represent the realPart and imaginaryPart components as the **private** members of the class. Provide a constructor function that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initial values are provided.

Provide **public** member functions for each of the following:

- Addition of two **Complex** numbers: The real parts are added together and the imaginary parts are added together.
- Subtraction of two **Complex** numbers: The real part of the right operand is subtracted from the real part of the left operand and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.
- Printing **Complex** numbers in the form **(a, b)** where **a** is the real part and **b** is the imaginary part.

### Hints:

#### Class declaration

```
// Complex.h
class Complex
{
public:
    Complex(int a=0, int b=0);
    Complex add(Complex &,Complex &);
    Complex sub(Complex &,Complex &);
    void print();
private:
    int real;
    int imaginary;
};
```

#### Test driver (client program)

```
// Test driver
#include "complex.h"
int main()
{
    Complex num1(2, 5), num2(7, 8), num3, num4 ;
    num1.print();
    num2.print();
    num3.add(num1,num2); // num3=num1+num2
    num3.print();
    num4.sub(num1,num2); //num4= num1-num2
    num4.print();
    return 0;
}
```

## Question 2

Write a grading program for a class with the following grading policies:

1. There are two quizzes, each graded on the basis of 10 points.
2. There is one midterm exam and one final exam, each graded on the basis of 100 points.
3. The final exam counts for 50% of the grade, the midterm counts for 25%, and two quizzes together count for a total of 25%. (Do **not** forget to normalize the quiz scores. They should be converted to a percent before they are averaged in.)

Any grades of 90 or more is an A, any grades of 80 or more (but less than 90) is a B, any grades of 70 or more (but less than 80) is a C, any grades of 60 or more (but less than 70) is a D, and any grades below 60 is an F.

The program will read in students' scores and output the students' records, which consist of two quizzes and two exam scores as well as the students' average numeric score for the entire course and final letter grade. Define and use a class for a student record. Also, input/output should be done with files.

### Hints:

#### Class declaration

```
// grade.h
const int maxsize=20;
struct Student
{
    char stuID[5];
    int midterm;
    int final;
    int quiz1;
    int quiz2;
    double average;
    char grade;
};

class Stuclass
{
public:
    Stuclass();//constructor
    void input();// read data from a file
    void printout();// output data to a file
    void calculation(); //calculate the average grade and final
                        // letter grade for each student

private:
    Student myclass[maxsize];
    int numberofstudents;
};
```

The organization for the input file is as follows:

The number at the top represents the number of students in a class. Following are students' records, which consist of 5 items:

1. student ID
2. quiz 1

3. quiz 2
4. midterm exam
5. final exam

**“input.dat”**

```

10
0001 8 9 100 90
0002 7 9 70 90
0003 7 8 75 80
0004 9 9 90 85
0005 6 7 70 75
0006 5 5 60 60
0007 8 7 100 90
0008 7 7 50 60
0009 9 9 85 92
0010 7 6 60 70

```

**Note:** Input and output **must be done** with files. An example is given in following.

```

void Stuclass::input()
{
    ifstream in_stream;
    in_stream.open("input.dat");

    if (in_stream.fail())
    {
        cout << "Intput file opening failed.\n";
        exit(1);
    }
    int x;
    // read number of students
    in_stream >> x;
    numberofstudents = x;

    for (int i = 0; i < numberofstudents; i++)
    {
        // read Student ID (4-digit)
        in_stream >> myclass[i].stuID;
        // read Quiz1
        in_stream >> myclass[i].quiz1;
        // read Quiz2
        in_stream >> myclass[i].quiz2;
        // read Midterm
        in_stream >> myclass[i].midterm;
        // read Final
        in_stream >> myclass[i].final;
    }
    in_stream.close();
}

void Stuclass::printout()
{
    ofstream out_stream("output.dat");

```

```

    if (out_stream.fail())
    {
        cout << "Output file opening failed.\n";
        exit(1);
    }
    out_stream << "StuID\tQuiz1\tQuiz2\tMidterm\tFinal\tAverage\tFinal Grade\n";

    for (int i = 0; i < numberofstudents; i++)
    {
        out_stream << myclass[i].stuID << "\t" <<
            myclass[i].quiz1 << "\t" <<
            myclass[i].quiz2 << "\t" <<
            myclass[i].midterm << "\t" <<
            myclass[i].final << "\t" <<
            myclass[i].average << "\t" <<
            myclass[i].grade << endl;
    }
    out_stream.close();
}

```

### Test driver (client program)

```

// Test driver
#include "grade.h"
int main()
{
    Stuclass myclassgrade;
    myclassgrade.input();
    myclassgrade.calculation();
    myclassgrade.printout();
    return 0;
}

```