Object-Oriented Programming

Lab 1
ENSIA 2021/2022

Exercise 1 (20 minutes)

Question 1.1

What is the output of the following code using the definition of the class `Time` given below it:

```
Time t1;
t1.setTime(18, 22, 9);
cout << "The Time is : ";
t1.printStandard();
```

You will assume the proper inclusion of `iostream`, `cout`, `cin`, and `endl`:

```
class Time {
    public:
        Time();
        void setTime( int, int, int );
        void printMilitary();
        void printStandard();
    private:
        int hour;
        int minute;
        int second;
};

Time::Time() { hour = minute = second = 0; }

void Time::setTime( int h, int m, int s ) {
    hour = ( h >= 0 && h < 24 ) ? h : 0;
    minute = ( m >= 0 && m < 60 ) ? m : 0;
    second = ( s >= 0 && s < 60 ) ? s : 0;
}

void Time::printMilitary() {
    cout << ( hour < 10 ? "0" : "" ) << hour << ":"
         << ( minute < 10 ? "0" : "" ) << minute;
}

void Time::printStandard() {
    cout << ( ( hour == 0 || hour == 12 ) ? 12 : hour % 12 )
         << ":" << ( minute < 10 ? "0" : "" ) << minute
```

```
        << ":" << ( second < 10 ? "0" : "" ) << second
        << ( hour < 12 ? " AM" : " PM" );
}
```

Question 1.2

Now with the same definition of `Time` given above, what is the output of the following code?

```
Time t(3, 4, 5);
t.printStandard();
cout <<endl;
t.printMilitary();
cout <<endl;
t.setTime(99, 3, 4);
t.printMilitary();
cout <<endl;
```

Exercise 2 (10 minutes)

In the following code, say if there are any errors. If so, correct them. Otherwise, say: "No errors."

```
#include <iostream>

struct Time {
    int hour;
    int minute;
    int second;
}

int main {
    Time clock;
    Time *clockPtr = &clock;
    clock_hour = 8;
    clock_minute = 12;
    *clockPtr.second = 0;
}
```

Exercise 3 (30 minutes)

Follow the template given after the problem statement to create a class called `Complex` for performing arithmetic with complex numbers. Write a driver program to test your class. Complex numbers have the form:

$$realPart + imaginaryPart \times i$$

where i is $\sqrt{-1}$.

Use floating-point variables to represent the private data for the class. Provide a constructor function that enables an object of this class to be initialised when it is declared. The constructor should contain default values in case no initialisers are provided. Provide public member functions for each of the following:

- Addition of two complex numbers;
- Subtraction of two complex numbers;
- Multiplication of two complex numbers;
- Printing complex numbers in the form `(a, b)` where `a` is the real part and `b` is the imaginary part. Make the print out as complete as possible about the operation.

```
// File: complex.h
// Class defintion
class Complex {
    public:
        Complex();                                // default constructor
        void addition(const Complex &);           // addition of a complex number to another
        void subtraction(const Complex &);        // subtract a complex number from another
        void multiplication(const Complex &m); // multiply a complex number by another
        void printComplex(void);                  // print a complex number
        void setComplexNumber(double, double); // set the real and imaginary parts
    private:
        ...
};

// File: complex.cpp
// Member functions definition
void Complex::Complex(double real, double imaginary) { ... }
void Complex::addition(const Complex &a) { ... }
void Complex::subtraction(const Complex &s) { ... }
void Complex::multiplication(const Complex &m) { ... }
void Complex::printComplex(void) { ... }
void Complex::setComplexNumber(double rp, double ip) { ... }

// File: main.cpp
// Driver program
#include <iostream>
using std::cout;
using std::endl;
int main() { ... }
```

Complete any missing details.

Exercise 4 (30 minutes)

Create a class called `Rational` for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private data of the class, the numerator and the denominator. Provide a constructor that enables an object of this class to be initialised when it is declared. The constructor should contain default values in case no initialisers are provided and should store the fraction in reduced form. For example the fraction $\frac{2}{4}$ would be stored in the object as 1 in the numerator and 2 in the denominator.

Hint You may wish to define a function `reduce` (properly defined) to do this.

Provide the public member functions that perform each of the following tasks:

- Adding two `Rational` numbers. The result should be stored in reduced form.
- Subtracting two `Rational` numbers. The result should be stored in reduced form.

- Multiplying two `Rational` numbers. The result should be stored in reduced form.
- Dividing two `Rational` numbers. The result should be stored in reduced form.
- Printing `Rational` numbers in the form `a / b`, where `a` is the numerator and `b` is the denominator.
- Printing `Rational` numbers in floating-point format.