**150018 - C++ Workshop**

**Homework Assignment #1**

**Object Oriented Programming**

1. Take care to make the program readable (including proper indentation)
2. Make sure the program does exactly what is required in each question.
3. In each of the questions, helper functions should be defined to improve program readability.
4. The exercise must be submitted according to the general instructions for submitting exercises (which appears on the course website), including:
   * Use meaningful names for the variables.
   * The program must document all functions defined, as well as conditional and loop statements and complex code snippets.
   * Include results of program execution as a comment at the end of the file!
5. You **may** submit in teams of two.

Important Note: You have one week to submit each homework assignment unless you instructed differently by your lecturer. An open submission box does not imply that you can submit after the due date.

**Question 1**

Define a class for rational numbers.

(Note: a denominator should not have neither 0 nor a negative number.)

The class should contain the following fields:

* + numerator
  + denominator

and the following methods:

* for each field, a setter method
  + if the parameter passed to the setDenominator() is 0 then it will set the denominator to 1
  + The denominator should not have a negative value. In the event that a negative value is passed, then the fields to be changed accordingly.
* for each field, a getter method
* print method in the format numerator/denominator (e.g., 1/2, 3/4, 54/56) according to the values that have been set.
* private method **Rational** **reduction()** which will reduce the object that called the function.

for example, after the following code segment:

Rational rat(2,4);

rat.reduction();

the numerator of rat is 1 and the denominator of rat is 2.

* a method **Rational** **add(Rational)** that returns a Rational that is the sum of the two rational numbers after the sum has been reduced.

for example, after the following code segment:

Rational r1, r2;

r1.setNumerator(1);

r1.setDenominator(6);

r2.setNumerator(1);

r1.setDenominator(3);

Rational r3;

r3 = r1.add(r2);

r1 remains 1/6

r2 remains 1/3

r3 is 1/2

* a boolean method **equal** that compares 2 rational numbers and checks if their values are equal (for the purposes of this exercise, equals means that the numerators are the same and the denominators are the same

aka 1/2 does not equal 3/6.

Use the main program that is found below to check the class that you created.

part 1. set/get

enter a rational number:

2/12

numerator: 2

denominator: 12

enter a rational number:

4/12

4/12

part 2. equals

The two numbers are different

part 3. add

2/12 + 4/12 = 1/2

You should use the following main program:

#include "Rational.h"

#include <iostream>

#include <string>

using namespace std;

int main()

{

int num1, num2, num3;

char junk;

cout << "part 1. set/get" << endl;

cout << "enter a rational number:" << endl;

cin >> num1 >> junk >> num2;

Rational rat1;

rat1.setNumerator(num1);

rat1.setDenominator(num2);

cout << "numerator: " << rat1.getNumerator()<<endl;

cout << "denominator: " << rat1.getDenominator()<<endl;

cout << "enter a rational number:" << endl;

cin >> num1 >> junk >> num2;

Rational rat2;

rat2.setNumerator(num1);

rat2.setDenominator(num2);

rat2.print();

cout << endl << endl;

cout << "part 2. equals" << endl;

if (rat1.equal(rat2))

cout << "The two numbers are equal" << endl;

else

cout << "The two numbers are different" << endl;

cout << endl << endl;

cout << "part 3. add" << endl;

Rational ans = rat1.add(rat2);

rat1.print();

cout << "+ ";

rat2.print();

cout << "= ";

ans.print();

cout << endl;

return 0;

}

**Question 2**

Define a class that represents employee data in the "Give from the Heart" organization for the purpose of calculating salaries.

The class should include the following fields:

* Social security number (int).
* Name - up to 20 characters (static array) (char name[21 [)
* Hourly wage (float)
* Number of hours worked (int)
* The amount of money collected for the organization (float).

And the following methods

For each field:

* Setter method.
* A getter method that returns its value
* A method to calculate the total salary according to the formula: number of hours worked \* wages per hour + percentage of amount collected

The percentages are calculated according to the following table:

|  |  |
| --- | --- |
| Salary | Percentage |
| up to 1000 nis (inclusive) | 10% |
| from 1000 nis to 2000 nis (inclusive) | 15% |
| from 2000 nis to 4000 nis (inclusive) | 20% |
| from 4000 nis to 5000 nis (inclusive) | 30% |
| above 5000 nis | 40% |

For example, a worker that collected 4500 NIS for the organization, the percentage of their salary would be 1000\*0.1+1000\*0.15+2000\*0.2+500\*0.3=800, that is in addition to the employee’s base salary the employee receives an additional 800 NIS.

Write a main program that will continue to read information about each of the employees until an SSN of 0 is entered. For each employee, the following data will be inputted, in order: SSN number, name, hourly wage, number of hours worked, the NIS amount of contributions collected. (It can be assumed that there is at least one employee).

The program should print out:

* the SSN and name of the employee who collected the least amount of money as well as how much money he/she collected
* the SSN and the name of the employee who received the highest salary as well as his/her salary

Note:

* The program should read all the data for an employee before checking if it is legal. On illegal input, the program should print ERROR. Each field should contain legal data relevant for that field.
* The main program should not allocate an array of employees.

enter details, to end enter 0:

123456789 moshe 50 40 2000

135792468 rivka 120 55 3450

97531246 sara 35 100 5632

0

minimum collected: 123456789 moshe 2000

highest salary: 1357924687 rivka 7140

enter details, to end enter 0:

111111111 doron 35 120 6000

222222222 tal 50 55 1400

444444444 levi 45 -4 100

ERROR

333333333 naomi 30 120 800

0

minimum collected: 333333333 naomi 800

highest salary: 111111111 doron 5550

**Question 3**

Write a class to represent circles (define a class Point to represent a point). The class contains the following fields:

* + center point
  + radius

and the following methods:

* getter/setter methods for all fields
* a method to compute area (define a constant PI=3.14)
* a method to compute circumference
* a method that receives a point as input and return 0 if the point lies on the circle, -1 if it is inside the circle, and 1 if it is outside the circle.

Write a main program the inputs data for 3 circles (A, B, and C) and for each circle, prints its area and circumference. The program then inputs data for points until it receives the point (0,0). The program outputs the number of points (not including the origin) that lie on or inside each circle.

enter the center point and radius of 3 circles:

(0,0) 3

(1,1) 2

(5,5) 2

perimeter: A: 18.84 B: 12.56 C: 12.56

area: A: 28.26 B: 12.56 C: 12.56

enter points until (0,0):

(0,1)

(1,0)

(0,4)

(0,0)

num of points in circle: A:2 B:2 C:0