Test 1 Practice

Test 1

- Wednesday, April 8th, in-class 3pm
- 5% of the total grade
- covers material from Chap. 1 – 6
- open book

Book chapters

1. Intro to Computers and Java
2. Intro to Java Applications
3. Intro to Classes, Objects, Methods and Strings
4. Control Statements: Part 1
5. Control Statements: Part 2
6. Methods: A Deeper Look

Test format

- **Prob. 1 – short answers**
- **Prob. 2 – declarations and writing short code**
- **Prob. 3 & 4 – programming methods**
- **Prob. 5 – implementing a class**

Problem 1a: Short answers

a) Each class declaration that begins with keyword _____ must be stored in a file that has exactly the same name as the class and ends with *.java*.

b) Return type _____ indicates that a method will perform a task, but will not return any information when it completes its task.

c) List four primitive types that store integer-type numbers.

Problem 1b: Short answers

State whether each of the following statements is *true* or *false*. If *false*, explain why.

d) An `import` declaration is not required when one class in a package uses another one in the same package.

e) Variables declared in the body of a particular method are known as *instance variables* and can be used in all methods of the class.

Problem 2a: Declarations and short code

a) Declare a private integer variable named `ONE` that is initialized to 1.

b) Declare a public character variable `lastNameInitial` that is set to your last name initial.

c) Declare a public float variable `PI` and set it to 3.14.

Problem 2a: Declarations and short code

(d) Write a `for` loop that increments integer variable `count` from its initial value of 0 to 10 in steps of 2.

(e) Write a `while` loop that decrements `count` from its last value of 10 back to 0 in steps of 5.

(f) Write an `if` statement that check if `count` variable is equal to zero, in which case, it prints out to the system console "Problem 2 is done".

Problem 3a: equalDigits method

Write a static method `equalDigits`, that returns true if a positive input integer has all identical digits, and false otherwise.

For example, integer 1111 has all identical digits, while 1024 does not.

Use the following method prototype:

```
static boolean equalDigits( int number )
```

Problem 3b: Rising number method

A *rising number* is a positive integer where each digit is greater than or equal to any digit appearing to its left. All of the single digit numbers are rising, as are numbers such as 12567, 144555899, and 888888. To be clear, the number 17689 is not rising, because the second digit 7 is followed by a 6.

Write a static method `risingDigits` that returns true if the input integer is a rising number, and false otherwise.

```
static boolean risingDigits( int number )
```

Problem 4a: sumDigits method

Write a static method `sumDigits`, that returns the sum of input number's digits.

For example, if the number is 123, then this methods returns 6 (1+2+3). We will only consider positive input integers and zero.

Use the following method prototype:

```
static int sumDigits( int number )
```

Problem 4b: Harmonic number method

The *harmonic number* of positive integer `n` is defined as the sum of the reciprocals of the first n positive integers. For example, the harmonic number of 5 is given by the sum:

```
harmonic(5) = 1 + 1/2 + 1/3 + 1/4 + 1/5
```

Write a static method `harmonic` that takes an integer number as the input and returns its harmonic number value.

```
static double harmonic( int number )
```

Problem 5a: Point2D class

We define a class named `Point2D` which models an `(x,y)` point in 2D space, where `x` is its horizontal and `y` is its vertical coordinate. These coordinates are implemented using private double variables `x` and `y`.

(a) Implement getter and setter for the private variables `x`.

Problem 5a (cont.): isOrigin method

(b) Define a public instance method `isOrigin`, which returns true if the point equals to the origin (0,0).

Problem 5a (cont.): negatePoint method

(c) Implement a public instance method `negatePoint`, which negates coordinates of the point. For example, negate of point (5,-1) will change its coordinates to (-5,1).

Problem 5b: Relational number class

We define a class named `Rational` that implements the basic methods and operations for rational numbers (numbers that can be represented as a fraction of two integers a/b, where a is the numerator and b is the denominator).

Note that the denominator cannot be set to zero, since division by zero is not well defined in ordinary arithmetic. Therefore, we need to print an error message to the console in case that the denominator is set to 0.

(a) Implement getter and setter for the private variables `x`.

Problem 5b (cont.): Class definition

```
// class definition
public class Rational {

     // instance variables
     private int num;
     private int den;

     // constructor (part a)
     public Rational(int num, int den) {}

     // getters and setters (part b and c)
     public int getNum() {}
     public int getDen() {}

     public void setNum(int num) {}
     public void setDen(int den) {}

     // public instance methods (part d and e)
     public boolean isPositive() {}
     public void invert() {}

}
```

Good luck!