Software Development

Mini Assignment 1
Due: January 21, 2018

This mini assignment practices the software techniques covered in lecture #2 and some things you saw from COMP 250. Primarily this will be a Java programming assignment that explores the concept of writing well designed software. You may be aware of other software techniques. Please do not use them. Just focus on what you have seen from Lecture #2 and supporting ideas from COMP 250, like Big Oh and how to program in Java.

Specifically, writing well designed software includes the following: optimality in speed (Big Oh) and memory (bytes), simplicity of solution, correctness, robustness, easy to read code, and commenting as documentation. Provide a Java software solution to the given problem below that expresses the well-designed software categories outlined in this paragraph.

Write a well-designed software solution for the following problem:

An application stores a two-digit pin. Each digit of the pin is a positive integer number from zero to nine. The pin can have leading zeros. For example, valid pins would include: 00, 01, to 99. Your application initializes the pin to the digits 00 and then asks the user to update the pin to a new value. The pin is only updated if the new digits are valid and they are not the same digits as the current pin. The program repeatedly prompts the user for pins until a successful update occurs. The program terminates when the user successfully changes the pin to a new value. The program displays the new pin value when it terminates.

The output of the program is as follows:

```
PROMPT $ ./java PinMain.class

Welcome to pin update!

Please input your old pin: 01

That is not your old pin.

Please input your old pin: 02

That is not your old pin.

Please input your old pin: 00

Old pin confirmed.

Please input your new pin: 00

Invalid pin.

Please input your new pin: AB

Invalid pin.
```

```
Please input your new pin:

Invalid pin.

Please input your new pin: 01

New pin confirmed.

Your pin has been updated to 01.

PROMPT $
```

The program has now terminated.

Your program will use three Java source files. One source file will contain all the pin code (Pin.java). The second file will contain the user interface code (PinMain.java). The last file will contain the testing code (PinTest.java).

PinMain.java and PinTest.java use Pin.java. PinMain.java has a main method. PinTest.java also has a main method. They both share the Pin.java file. Hence, you are creating two programs. One program, PinMain.java and Pin.java is the application described above. The other program, PinTest.java and Pin.java, will be used to show the correctness of your Pin.java code, exhaustively. Do **not** use tools like jUnit and Java Doc. You are just writing simple Java code. You are building things from scratch.

I do not think I need to further describe the application PinMain.java. You see the output and problem description above.

Pin.java will store the current pin. It will have a constructor and the methods you need to satisfy the problem description. It has an optimal number of methods. It uses optimal algorithms and data structures. You will need to determine the optimality and correctness of your algorithm and data structure choices. Think carefully.

PinTest.java has a main method that will exhaustively test every method in Pin.java. To do this correctly, each method from Pin.java will have a corresponding testing loop within PinTest.java. Each testing loop will exhaustively test the entire data range of parameter values for the corresponding method providing all the valid and invalid and boundary value arguments. The loop outputs to the screen the arguments passed to the method and the value returned by the method. This output is used by the developer to verify the correctness of the method. Make your output look something like this:

```
Test case for method <<method name goes here>> :

Arguments: <<all arguments for call>> Result: <<result from call>>

Arguments: <<all arguments for call>> Result: <<result from call>>

Etc.

Test case for method <<next method name goes here>> :

Arguments: <<all arguments for call>> Result: <<result from call>>

Arguments: <<all arguments for call>> Result: <<result from call>>

Etc.
```

You are creating a well-designed program. I expect to see optimality, pretty code, and comments as documentation.

WHAT TO HAND IN

- PinMain.java
- PinTest.java
- Pin.java

HOW IT WILL BE GRADED

For your assignment to be graded your program (a) must run, (b) did not use tools, and (c) followed the assignment instructions. You are doing this assignment on your own.

- +5 - Optimality (Big Oh)
- +5 - Optimality (memory)
- +5 - Simplicity of solution
- +5 - Correctness (PinTest.java)
- +5 - Well written code that is easy to read (see lecture 2 for examples)
- +5 - Comments as documentation (see lecture 2 for examples)