CIT 591 – Homework 2

Due – Oct 4, 2016 at 12.00pm

Part 1 – Theory (30 points)

Please do the following REVIEW EXERCISES from the class textbook:

1. R5.4, R6.2, R6.4, R6.15 (5 points each)
2. Read Paul Graham's "Hackers and Painters"
   http://www.paulgraham.com/hp.html
   Write a 1-page (normal margins, normal font sizes, single line spacing) response to the above. (10 points)
   Some things to think about while writing the response:
   a. Do you agree with the analogy? If yes, why? If no, why not?
   b. Are there other similarities or differences between hackers and painters?
   c. Do you agree with the article? If yes, why? If not, why not? Etc.

Part 2 – Programming (70 points)

Please do the following programming projects from the class textbook.

1. Write a program that prints all real solutions to the quadratic equation ax^2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b^2 – 4ac is negative, display a message stating that there are no real solutions.
   Implement a class `QuadraticEquation` whose constructor receives the coefficients a, b, c of the quadratic equation. Supply a method `evaluate` that evaluates the value of the quadratic equation given a specific x. Also provide accessor methods for a, b, c. Implement a class `QuadraticEquationSolver` whose constructor receives an object of type `QuadraticEquation`. Supply methods `getSolution1` and `getSolution2` that get the solutions, using the quadratic formula, or 0 if no solution exists. The `getSolution1` method should return the smaller of the two solutions. Supply a mutator method for `QuadraticEquation`. Finally, implement a QuadraticEquationTester class whose `main` method reads user input for a, b, c, constructs the appropriate objects, and prints the solutions out. (30 points)
2. Programming Project P6.4 (Prime Numbers). (20 points)
3. Programming Project P6.8 (The Buffon Needle Experiment). (20 points)

Part 2 – Extra Credit (10 points)

For the programming parts 2 and 3 above, you used `while` or `for` loops. For Extra Credit, replace all loops with the other kind, if possible. E.g., if you used a `while` loop, try to replace it with a `for` loop. If it's not possible to do so, explain why.

Explain, from a design/style point of view, which loop is better in which situation and why.

Please submit both versions of your code, if you attempt the EC.

For the EC part, you cannot have any help from the TAs/instructor.

Grading Criteria (for the Programming part)

10% for compilation – If your code compiles, you get full credit. If not, you get a 0.
80% for functionality – Does the code work as required? Does it crash while running? Are there bugs? …
10% for style – Do you have good variable names? Is your code well commented? …
0% for design – There are no points for design for this hw. So this is just a general guideline for the future. You will gain points in future assignments for good design.

Programming – General Comments

Here are some guidelines with respect to programming style.

Please use Javadoc-style comments.

For things like naming conventions, please see Appendix I (Page A-79) of the Horstmann book. You can also install the Checkstyle plugin (http://eclipse-cs.sourceforge.net/) in Eclipse, which will automatically warn you about style violations.

Submission Instructions

We recommend submitting the theory part electronically also. However, you can turn in a physical copy at the start of class, if you prefer. Please **do not** print out the Java source.

In addition to the theory writeup, you should also submit a text file titled readme.txt. That is, write in plain English, instructions for using your software, explanations for how and why you chose to design your code the way you did. The readme.txt file is also an opportunity for you to get partial credit when certain requirements of the assignment are not met. Think of the readme as a combination of instructions for the user and a chance for you to get partial credit.

Please create a folder called YOUR_PENNKEY. Places all your files inside this – theory writeup, the Java files, the readme.txt file. Zip up this folder. It will thus be called YOUR_PENNKEY.zip. So, e.g., my homework submission would be swapneel.zip. Please submit this zip file via canvas.