Homework 6

**Name: Peter Dobbs**

**MU ID: 005803169**

**Date: 26 February, 2015**

**Homework Problem 1**

**Requirements**

Read in three integers from the command line and sort them from least to greatest value.

**Design**

*main*

Prompt the user to enter three integers. Create the scanner and variables and assign those variables to the input integers. Call the method “sort” and after the method has been used, close the scanner.

*sort*

Create a series of if statements that order the input integers.

**Iterative Development Steps**

1. Import the scanner utility into the program and create a new class.
2. Create two methods, one main and one named “sort.” Use the main method for assigning the scanner to a variable and creating and assigning variables to the input integers.
3. In the method “sort” create a series of if statements that orders the input integers from least to greatest, regardless of order of input.

**Tests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description of test | Input | Expected result | Actual result | Cause |
| method1 normal input | 5  9  8 | 5  8  9 | 5  8  9 | Good code |

**Homework Problem 2**

**Requirements**

Approximate pi using a random number generator.

**Design**

*main*

Import the Random class and create a new class. In the main method enable the random number generator for numbers between 0.0 and 1.0. Create a loop that finds the distance from the origin to a point that uses the random number generator for its coordinates. If that distance is less than one then add it to all the other distances that the random number generators compiles which are also less than one. Adjust this final value to approximate pi and display the approximation.

**Iterative Development Steps**

1. Import the Random class and create a new class. In the main method enable the random number generator for numbers between 0.0 and 1.0.
2. Create a loop that finds the distance from the origin to a point that uses the random number generator for its coordinates. If that distance is less than one then add it to all the other distances that the random number generators compiles which are also less than one.
3. Adjust this final value to approximate pi and display the approximation.

**Tests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description of test | Input | Expected result | Actual result | Cause |
| method1 normal input | N/A | Approximately 3.1415926535 | 3.11254742 | Good Code |

**Homework Problem 3**

**Requirements**

Create a program that checks three points for collinearity

**Design**

*main*

Initialize the scanner and assign the input values to variables. Call the method Copoints and print the result. Close the scanner

*Copoints*

Find the slopes of the lines from one point to another to another. Compare these slopes to one another and if they are all equal, then return that the points are collinear. If they are not equal then return that they are not collinear.

**Iterative Development Steps**

1. Import the Scanner class and create a new class. Create a method “Copoints” in addition to the main method. In main, initialize the scanner and assign the input values to variables.
2. In Copoints, calculate the slope of the line from one point to another and compare. If the slopes are all equal, then return that the points are collinear. If they are not equal then return that they are not collinear.

**Tests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description of test | Input | Expected result | Actual result | Cause |
| method1 normal input | (1, 1)  (2, 2)  (3, 3) | Points are collinear | Points are collinear | Good Code |

**References**

An Introduction to Programming Using Java, Anthony J Dos Reis.