CSc 110 Assignment 4
Focusing on If / Else, While Loops, and Methods

GameOfPig.java

Due:

Friday Oct. 24, 8pm.

Learning Outcomes:

When you have completed this assignment, you should understand:

- How to implement conditional statements (if-statements and if/else-statements)
- How to compare Strings.
- How to analyze the flow of choices to ensure correct program logic.
- How to design a suite of tests that will check the various cases in a branching program (i.e. one that contains conditional statements).
- Logical decomposition using methods.
- How to design and use indefinite (while) loops.
- How to generate random numbers.

Programming Problem Description:

This assignment requires you to implement the game of "Pig". Rules for Pig are available here: http://en.wikipedia.org/wiki/Pig_(dice). Try playing the game with a friend to see how it works.

In your version of Pig, the user will play against the computer using 1 die. Your "die" is a random number generator that simulates a 6-sided die. The player will go first. When the player plays, they choose when to end their turn. The computer should end its turn when either of the following happens: (a) the computer rolls a 1 or (b) the computer has done 4 rolls.

The program (GameOfPig.java) will…

- Use a **single** Random object in your program to simulate a roll of the die (i.e. a random number between 1 and 6). See the text (2nd ed: p. 307, 3rd ed: p. 317) to learn how to use Random. Note that this object will be passed to some of your methods as an actual parameter.
- Use a **single** Scanner object in your program. Note that this may need to be passed to some of your methods as a parameter.
- Use at least one if statement and at least one while loop. It may use more than 1 of each.

- Provide informative prompts when user input is required. The user will input "y" or "n" for whether or not to roll again.
- Give informative output: the number rolled each time the player rolls, the total earned at the end of each turn, the cumulative score at the end of each turn, and whether the player won or lost the game (once a score of at least 100 is reached).
- Be logically decomposed into smaller parts, using methods. It is recommended that you create separate methods for actions such as a computer turn and a player turn. You will need to decide what parameters to pass and what type of values to return.
- Be well documented, following the style guidelines.

HAND IN: Submit GameOfPig.java using the 'Assignments' link of the course connex site.

Bonus

For up to 10% bonus points, create a second program, PigTwo.java, that does **one of the following**:

(a) Implement the Pig game with two dice. The player always rolls 2 dice. The player loses all their points (total score returns to zero) if they roll two 1's simultaneously.
(b) Implement better rules for the computer to use on its turn, to make the computer "smarter". For example, the computer could decide whether to keep rolling based on a combination of the number of rolls it has done, plus whether or not it has reached a threshold number of points on that turn, like 15.

Hand in PigTwo.java. In the comment block at the top of the file, explain to the marker what you have changed and how this improves the game.

(Be sure to also hand in the basic program GameOfPig.java from the main assignment above.) You must receive at least 9/10 on the main assignment to be considered for the bonus.

Grading – The marker will look for:

- Your code correctly uses Strings, loops, Random, and if statements.
- Your methods must define a logical decomposition of the program into smaller parts.
- Compiles and produces correct output: The code constitutes a valid Java program (i.e. compiles *and* runs without error on the ECS 250 lab machines). It must accept input and print output as described above.
- Appropriate prompts are given to the user when input is needed.
- Documentation and style: Documentation must include the author's identification and the purpose of the program. Each group of instructions must be preceded with a comment describing their common purpose. Each method must be preceded with a comment describing its purpose as well as an indication of the input to and the output from the method. Whitespace and indentation is expected. Refer to the Style Guidelines in the course notes for details.