Pig

Your names:

This programming project is to be done by a pair of students. (If your class has an odd number of students, you may have been assigned to a three-person "pair".) Work together, discussing your current problem and how to solve it. Regularly trade off who is driving (actually editing the code) and who is navigating (making suggestions and consulting documentation).

By the time you are done with this activity, you and your pair should be able to:

- use Java data types and their operators.
- use Java control structures.
- work more effectively as a pair.

After you complete this activity, please fill out the short survey at

http://goo.gl/forms/HXjyuUb2ou

to improve this project for future users.

Playing the Game

Included in the Pig project is a file Pig.jar. This is a compiled version of the working game. To play it, use a terminal to navigate to the directory containing the file and type this on the command line:

java -jar Pig.jar

This is a two-player game. Play once or twice within your pair so that you understand the rules.

Implementing Pig

Write a program Pig.java that behaves *exactly* like Pig.jar. (Following a specification is a useful skill for you. Having all students' programs all behave the same way makes life easier for your grader.)

Take notes as you work. (This is good work for the navigator.) How well are you working as a pair? What bugs and conceptual difficulties did you encounter? How did you overcome them? What did you learn?

There are two approaches that you might take to writing this program:

1. Write out the entire program, test it, and then debug it.

2. Implement one small feature, test it, and debug it. Add another feature, test it, and debug it. Repeat until the program is done.

Experienced software developers know that the second approach takes *vastly* less time. Here's a reasonable sequence of stages your program might go through as it grows:

1. Roll the die and print the result.

2. Roll the die five times, printing each roll and the running total.

3. Play a complete turn, allowing the user to keep going or stop. The turn ends when the player either rolls a 1 or chooses to stop.

4. Allow two players to alternate turns. Maintain both players' scores.

5. Play complete game, including detecting the end of the game.

You may end up writing some code that doesn't end up in the final program, such as the "five times" loop or printlns for debugging. That's fine! It's like construction workers putting up scaffolding that won't remain in place when the building is finished.

If you don't know how to do something (like breaking out of a loop or generating a random number to simulate rolling a die), you have many resources at your disposal:

- Ask your partner.
- Consult your notes or textbook.
- Look through previous programs you have examined or written.
- Search the internet. Some students are reluctant to do this, thinking that it might be "cheating". It certainly would be plagiarism to simply find a working Pig program and hand it in. On the other hand, the internet is perfectly valid as a way to remind yourself of syntax, find examples, or expand your understanding. Real programmers do this all the time.

  Note that some internet sources may use different terminology, refer to obscure Java features, or be out of date. The official website for your textbook or your course are therefore particularly good sources.
- Post to your course email list or forum.
- Ask your instructor or teaching assistant.

After you're done, please fill out the survey at http://goo.gl/forms/HXjyuUb2ou.