

CSC 110: Fundamentals of Programming I

Assignment #4: More methods, while-loops, Random

Due date

Sunday, October 18th, 2015 at 11:55 pm via submission to connex.

How to hand in your work

Submit the requested file (`Pig.java`) through the Assignment #4 link on connex.

Learning outcomes

When you have completed this assignment, you should understand:

- How to pass *parameters* and *return* values using `static` methods.
- How to use `while` loops.
- How to use the `Random` class and its `nextInt` method.
- How to logically decompose a program into smaller parts.
- How to *indent* and *document* a Java program.

Write the code for a Java program named `Pig.java`. The instructions on how to play Pig, as well as the specification for the methods required in your program can be found in the `Pig.html` file linked [here](#). Read the specification document carefully, as it describes what each method should do. Each of the methods **must** be present in the `Pig` class, with the headers (parameters and return values) exactly as described. (Note that for the `Scanner` and `Random` parameters, you can omit the `java.util.` prefix by writing `import` statements at the top of the source code.)

For this assignment, *testing* is crucial. All tests for the methods are implemented in the main method. During and after the writing of each method, call the current method from main, setting up enough tests to satisfy yourself that the method works. See **Appendix A** for examples of good testing output. As you progress, comment out the previous tests (don't delete them; we'll be checking), and carry on with the next test.

Recommended steps to follow in order

1. The `diceRoll` method is a nice place to start. Focus on the simple responsibility this method has, without thinking about the rest of the game. It simply rolls a die and returns the value that of the roll.
2. The `playerTurn` and the `computerTurn` methods are similar and yet still different. One requires some user interaction and the other does not. Choose whichever one you would like to do first. Make sure you finish and test them both before moving to the next step.
3. The final method puts the whole game together and should be implemented only after the previous three are tested successfully. When you test this method in main and it produces output that is similar to the final output in **Appendix A**, you can play Pig!

Marking

Your mark will be based on the following criteria:

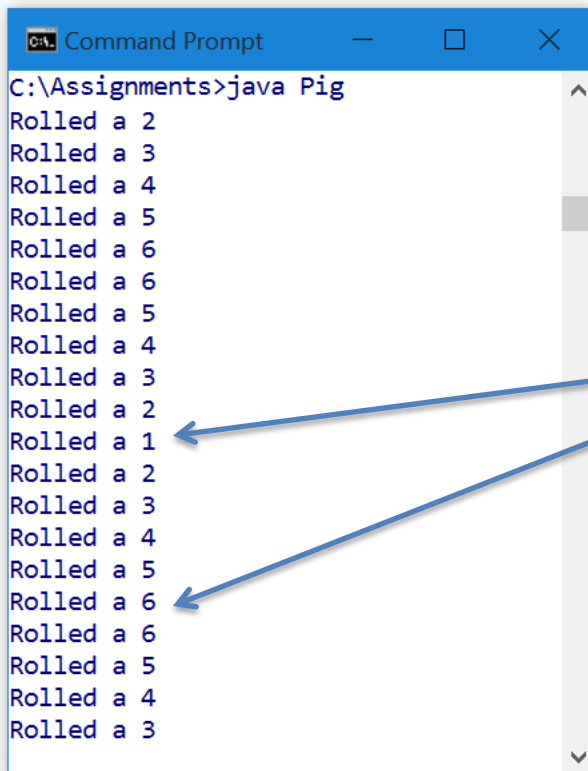
- Your code *must compile and run*. It must prompt the user, generate Random numbers, and produce the expected output as demonstrated in **Appendix A**.
- Your code must conform to all the requirements mentioned in the [specification document](#).
- The main method must show all of your testing code. It may be commented out once you have tested parts of it. The final call to `gameLoop()` should not be commented out.
- Your code must follow the guidelines outlined in `Style_Guidelines.pdf`, found through the Lectures & Stuff link in the Lab Resources folder on `conneX`. You may note that the specifications provide some very nice comments you are welcome to borrow.

Appendix A – Testing your code

As you work through a solution, it is recommended that you save, compile and test your code after every line or two of code that you write. This can be something as easy as printing out the value of a variable, or calling a method to print out the value returned. It is important to do this to confirm a component of your code works correctly, so you can be confident using that component throughout your code later.

Testing the `diceRoll` method:

One way to do this is to create a for-loop that loops 20 times. Inside the for-loop, call the `diceRoll` method and assign the value it returns to an integer variable `roll`. Then, print out the value of `roll`.



```
C:\Assignments>java Pig
Rolled a 2
Rolled a 3
Rolled a 4
Rolled a 5
Rolled a 6
Rolled a 6
Rolled a 5
Rolled a 4
Rolled a 3
Rolled a 2
Rolled a 1
Rolled a 2
Rolled a 3
Rolled a 4
Rolled a 5
Rolled a 6
Rolled a 6
Rolled a 5
Rolled a 4
Rolled a 3
```

What to look for:

- Is the minimum number a 1?
- Is the maximum number a 6?

Testing the `playerTurn` method: (user input underlined in red):

```
Command Prompt
C:\Users\Anthony Estey\Documents\csc110\Assignment
s>java Pig
You rolled a 6. Your score so far is 6
Do you want to roll again? (yes or no)
yes
You rolled a 3. Your score so far is 9
Do you want to roll again? (yes or no)
yes
You rolled a 4. Your score so far is 13
Do you want to roll again? (yes or no)
yes
You rolled a 5. Your score so far is 18
Do you want to roll again? (yes or no)
no
Ending your turn with a score of 18.

Score returned: 18

You rolled a 6. Your score so far is 6
Do you want to roll again? (yes or no)
yes
Uh oh, you rolled a 1!
Your turn is over and you get 0 points this round

Score returned: 0

You rolled a 5. Your score so far is 5
Do you want to roll again? (yes or no)
no
Ending your turn with a score of 5.

Score returned: 5
```

What to look for:

- Does it continually roll if the user chooses to roll again?
- Does the turn end if the user chooses to end the turn?
- Is the sum of all rolls returned at the end of a turn?
- If a 1 is rolled, does the turn end with a score of 0 returned

Testing the *computerTurn* method:

```
Command Prompt
C:\Users\Anthony Estey\Documents\csc110\Assignments>java Pig
The computer rolled a 2. Its total score this turn is 2.
The computer rolled a 6. Its total score this turn is 8.
The computer rolled a 3. Its total score this turn is 11.
The computer rolled a 4. Its total score this turn is 15.
Ending computer's turn with a score of 15.

Score returned: 15

The computer rolled a 6. Its total score this turn is 6.
The computer rolled a 5. Its total score this turn is 11.
The computer rolled a 1, ending its turn with a score of 0 this round

Score returned: 0

The computer rolled a 4. Its total score this turn is 4.
The computer rolled a 5. Its total score this turn is 9.
The computer rolled a 5. Its total score this turn is 14.
The computer rolled a 6. Its total score this turn is 20.
Ending computer's turn with a score of 20.

Score returned: 20
```

What to look for:

- Does the computer's turn always last 4 turns?
- Is the sum of all rolls returned at the end of a turn?
- If a 1 is rolled, does the turn end early with a score of 0 returned

Testing the *gameLoop* method: (with user input in red font)

```
C:\Users\Anthony Estey\Documents\csc110\Assignments>java Pig
```

```
Welcome to the game of Pig. Beginning the game...
```

```
You rolled a 5. Your score so far is 5  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 2. Your score so far is 7  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 6. Your score so far is 13  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 4. Your score so far is 17  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 5. Your score so far is 22  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 5. Your score so far is 27  
Do you want to roll again? (yes or no)
```

yes

```
You rolled a 5. Your score so far is 32  
Do you want to roll again? (yes or no)
```

no

```
Ending your turn with a score of 32.
```

playerTurn
method

computerTurn
method

The same process
is repeated until
someone reaches
100 points:

- a) *playerTurn*
- b) *computerTurn*
- c) Print scores

```
The computer rolled a 4. Its total score this turn is 4.  
The computer rolled a 5. Its total score this turn is 9.  
The computer rolled a 3. Its total score this turn is 12.  
The computer rolled a 1, ending its turn with a score of 0 this round
```

```
The scores at the end of the current round are:
```

```
Player: 32      Computer: 0
```

```
You rolled a 5. Your score so far is 5  
Do you want to roll again? (yes or no)
```

yes

```
Uh oh, you rolled a 1!
```

```
Your turn is over and you get 0 points this round
```

```
The computer rolled a 2. Its total score this turn is 2.  
The computer rolled a 5. Its total score this turn is 7.  
The computer rolled a 5. Its total score this turn is 12.  
The computer rolled a 2. Its total score this turn is 14.  
Ending computer's turn with a score of 14.
```

The scores at the end of the current round are:

Player: 32 Computer: 14

You rolled a 2. Your score so far is 2

Do you want to roll again? (yes or no)

yes

You rolled a 5. Your score so far is 7

Do you want to roll again? (yes or no)

yes

Uh oh, you rolled a 1!

Your turn is over and you get 0 points this round

The computer rolled a 1, ending its turn with a score of 0 this round

The scores at the end of the current round are:

Player: 32 Computer: 14

You rolled a 5. Your score so far is 5

Do you want to roll again? (yes or no)

yes

You rolled a 6. Your score so far is 11

Do you want to roll again? (yes or no)

yes

You rolled a 2. Your score so far is 13

Do you want to roll again? (yes or no)

no

Ending your turn with a score of 13.

The computer rolled a 4. Its total score this turn is 4.

The computer rolled a 3. Its total score this turn is 7.

The computer rolled a 5. Its total score this turn is 12.

The computer rolled a 2. Its total score this turn is 14.

Ending computer's turn with a score of 14.

The scores at the end of the current round are:

Player: 45 Computer: 28

You rolled a 5. Your score so far is 5

Do you want to roll again? (yes or no)

yes

You rolled a 3. Your score so far is 8

Do you want to roll again? (yes or no)

yes

Uh oh, you rolled a 1!

Your turn is over and you get 0 points this round

The computer rolled a 2. Its total score this turn is 2.

The computer rolled a 3. Its total score this turn is 5.

The computer rolled a 1, ending its turn with a score of 0 this round

The scores at the end of the current round are:

Player: 45 Computer: 28

You rolled a 4. Your score so far is 4

Do you want to roll again? (yes or no)

yes

You rolled a 6. Your score so far is 10

Do you want to roll again? (yes or no)

yes

You rolled a 3. Your score so far is 13

Do you want to roll again? (yes or no)

yes

You rolled a 5. Your score so far is 18

Do you want to roll again? (yes or no)

no

Ending your turn with a score of 18.

The computer rolled a 4. Its total score this turn is 4.

The computer rolled a 5. Its total score this turn is 9.

The computer rolled a 2. Its total score this turn is 11.

The computer rolled a 5. Its total score this turn is 16.

Ending computer's turn with a score of 16.

The scores at the end of the current round are:

Player: 63 Computer: 44

You rolled a 5. Your score so far is 5

Do you want to roll again? (yes or no)

yes

You rolled a 2. Your score so far is 7

Do you want to roll again? (yes or no)

yes

You rolled a 2. Your score so far is 9

Do you want to roll again? (yes or no)

yes

You rolled a 3. Your score so far is 12

Do you want to roll again? (yes or no)

yes

You rolled a 4. Your score so far is 16

Do you want to roll again? (yes or no)

yes

You rolled a 4. Your score so far is 20

Do you want to roll again? (yes or no)

no

Ending your turn with a score of 20.

The computer rolled a 1, ending its turn with a score of 0 this round

The scores at the end of the current round are:
Player: 83 Computer: 44

You rolled a 2. Your score so far is 2
Do you want to roll again? (yes or no)

yes

You rolled a 2. Your score so far is 4
Do you want to roll again? (yes or no)

yes

You rolled a 4. Your score so far is 8
Do you want to roll again? (yes or no)

yes

You rolled a 4. Your score so far is 12
Do you want to roll again? (yes or no)

yes

You rolled a 6. Your score so far is 18
Do you want to roll again? (yes or no)

no

Ending your turn with a score of 18.

The computer rolled a 5. Its total score this turn is 5.
The computer rolled a 6. Its total score this turn is 11.
The computer rolled a 2. Its total score this turn is 13.
The computer rolled a 1, ending its turn with a score of 0 this round

The scores at the end of the current round are:
Player: 101 Computer: 44

Congratulations, you win!!

Game ends
when 100
points are
reached

Print out
who wins at
the end!