
600.107 Introductory Programming in Java, Summer 2015
Homework 4

Due: Thursday, July 9th at 11:59pm

Please review the **Homework Guidelines** handout for this course before you begin.

Restriction: For all tasks in this assignment, you are not permitted to use arrays or ArrayLists.

Task 1. Write a Java class named `Identification` that simulates the interaction a bank customer might have attempting to log into an automated teller machine (ATM). To log in, the customer must demonstrate that he knows his own personal identification number (PIN). However, to stop criminals from guessing someone else's PIN, a customer is given only three chances to enter his PIN correctly; then the customer is locked out until he makes personal contact with someone at the bank.

To simplify things in this simulation, we'll assume (each time the program is run) that the correct 4-digit PIN number is 1234. In fact, you must include in your class the following declaration of a named constant, and make use of it when checking the user's guesses:

```
//named constant representing desired PIN value
final String TARGET_PIN = "1234";
```

Your program should ask the user for his PIN number up to three times. If he enters it correctly (that is, if he enters the `TARGET_PIN` value 1234), indicate that he has successfully logged in, and end the program. If he enters it incorrectly one or two times, allow him to try to enter it again. If his third guess is also incorrect, output a message saying that the account is now locked and that he should contact the bank.

In addition to using the named constant specified above, you must also solve this problem using either a `while` or `do-while` loop. You may not use a `for` loop to solve this problem.

Here are three sample executions, with user input shown in bold:

```
Please enter your PIN number: 1234
You are successfully logged in.
Goodbye.
```

```
Please enter your PIN number: 1492
Please enter your PIN number: 1776
Please enter your PIN number: 4321
You made 3 unsuccessful login attempts.
Your account is locked. Please contact the bank.
Goodbye.
```

```
Please enter your PIN number: 1233
Please enter your PIN number: 12345
Please enter your PIN number: 1234
You are successfully logged in.
Goodbye.
```

Task 2. Write a Java class named `Mirror` which contains a main method that collects a line of input from the user and then determines and outputs the longest sequence of characters at the start of the line which is a mirror image of the sequence of characters at the very end of the line. The two mirror image sequences of characters may overlap. For example, the input line "abX Zba" has longest mirror end "ab", so your program would output "ab". The input "aba" has longest mirror end "aba", and the input line "abcdefg" has an empty `String` (a.k.a. "", the `String` with length 0) as its longest mirror end.

Task 3. Recall that a divisor of integer n is an integer that can be multiplied by some other integer to produce n . Which integer between 1 and 25 has the largest number of positive divisors, and how many divisors does it have? What about the collection of integers between 1 and 10000? Or between 4000 and 12500?

Write a Java class called `Divisors` that contains a main method which can find and print out the answer to any one of these questions, with any set of positive endpoints. Your program should collect the two positive endpoint values (inclusive) of the range as input. As long as either of the two entered values is not positive; your program should prompt the user for two more integer endpoints and collect them. Once two positive integers are successfully collected, your program will then report an integer from within the range which has the greatest number of positive divisors. It is possible that several integers in this range have the same maximum number of divisors. Your program only needs to print out one of them (along with its number of divisors). Be sure that your program works whether the user enters the smaller positive endpoint first or the larger positive endpoint first.

Below is one sample execution, but you'll need to test many more cases than this one! Also, notice that the program below reports the answer as 6, but either 8 or 10 would also be correct in this case.

Given a specific range of positive integers to search, this program will determine a value in the range which has the highest number of divisors. You get to select the range.

```
Please specify the range by entering two positive integers: 0 10
Please specify the range by entering two positive integers: 1 10
Okay, checking values between 1 and 10...
The winning number is 6 with 4 divisors.
```

Turn-in. Before the due date listed at the top of this file, please submit via the course Blackboard site one complete `.zip` file named `HW4-jhed.zip` (where you replace 'jhed' with your own personal JHED) containing all `.java` files required for the above tasks. Be sure that all submitted code compiles! If you were unable to complete a task, please include as part of your `.zip` submission a text file named `README` explaining anything you'd like the graders to know.

Need to fix something that you already submitted? At any point up until the due date for this assignment, you may re-submit a complete zip file. The submission most recently received is the one that will be graded. Any earlier submissions will be ignored, so please include *all* files in each submission.