Assignment #3
CSE110 - Arizona State University

Topics

- Chapter 4 Loops - Conditional and repetition (loop) statements

Coding Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.
- Use comments after the ending brace of classes, methods, and blocks to identify to which block it belongs.

Assignment Documentation

At the beginning of each programming assignment you must have a comment block with the following information:

```
/*-------------------------------------------------------------------------
// AUTHOR: your name
// FILENAME: title of the source file
// SPECIFICATION: description of the program
// FOR: CSE 110- homework #- days and time of your class
// TIME SPENT: how long it took you to complete the assignment
//-------------------------------------------------------------------------*/
```

Part #1: Written Exercises (10 pts)

1. (4 pts) What do the following loops print?

   a)
   ```
   for (int i = 10; i > 1; i--)
         System.out.print(i + " ");
   ```

   b)
   ```
   int j = 1;
   while (j < 20) {
         if (j % 5 == 0)
               System.out.print(j + " ");
         j += 2;
   }
   ```

2. (6 pts) Given the following code segment that reads in a `String str`:

```
String str = in.nextLine();
```

Write a loop that will print out the value entered into `str` in reverse. That is if the user entered "Harry", the loop should print "yrraH". This does not have to be a complete program, but you should test your loop inside a complete program. Section 4.7.2 in the book and the Examples for Loops lecture video show some `String` loop examples with a `for` loop.

Part #2 - Programming (10 pts)

Write a program called `Assignment3` (saved in a file `Assignment3.java`) that computes the greatest common divisor of two given integers.

One of the oldest numerical algorithms was described by the Greek mathematician, Euclid, in 300 B.C. It is a simple but very effective algorithm that computes the greatest common divisor of two given integers.

For instance, given integers 24 and 18, the greatest common divisor is 6, because 6 is the largest integer that divides evenly into both 24 and 18. We will denote the greatest common divisor of x and y as $gcd(x, y)$.

The algorithm is based on the clever idea that the $gcd(x, y) = gcd(x - y, y)$ if $x >= y$ and $gcd(x, y) = gcd(x, y - x)$ if $x < y$. The algorithm consists of a series of steps (loop iterations) where the "larger" integer is replaced by the difference of the larger and smaller integer. This continues until the two values are equal. That is then the gcd.

You are required to do the following:

- Prompt the user to enter two integers and read them from the keyboard.
- Create a loop, and inside of it replace the larger integer with the difference between the larger integer and the smaller integer (if the integers are equal you may choose either one as the "larger") during each iteration. You will need to use an if/else statement to determine the larger value.
- Continue looping until the two integers are equal.
- Print out the remaining value as the gcd. See the sample output below.

Sample Outputs

Sample 1:
Enter the first integer: **72**
Enter the second integer: **54**
The gcd of 72 and 54 is 18

Sample 2:
Enter the first integer: **18**
Enter the second integer: **24**
The gcd of 18 and 24 is 6

Submission

- Go to the course web site (my.asu.edu), and then click on the on-line Submission tab.
- Submit your **Assignment3.java** file on-line. Make sure to choose Hw3 from drop-down box.

- `Assignment3.java` should have the following, in order:
  - In comments, the assignment header.
  - In comments, the answers to questions in Part #1.
  - The working Java code requested in Part #2.
  - The Assignment3.java file must compile and run as you submit it. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission.

NO LATE ASSIGNMENTS WILL BE ACCEPTED.