CSCI 1301 – Programming Principles I
Georgia Southern University
Department of Computer Science
Fall 2024

Lab 6
Point Value: 20 points
Due: Friday September 27, 2024, end of lab

Objectives
1. Use different types of loops to write working programs.
2. Use nested loops to complete certain types of tasks.
3. Use the debugger to effectively diagnose and fix errors in code.
4. Archive Java source code files and other documents into a zip file.

Description
Complete the following steps as described and upload requested materials to Folio in the appropriate dropbox in a single zip file. Submit Java source code files to Gradescope.

- Students should work in groups of two for this lab. When submitting your zip file, fill out the comment header and indicate both group member names.

- Your submitted zip file should be named Lab#LastName1LastName2.zip. For instance, Lab6JonesSmith. Any requested screenshots and/or answers to questions should be included in a separate Word document.

- All programs should have a comment header. That comment header should include information similar to:
  ```
  /**
   * File: PAssign0.java
   * Class: CSCI 1301
   * Author: Christopher Williams
   * Created on: Jun 6, 2016
   * Last Modified: Aug 16, 2018
   * Description:  Display three messages to the console
   */
  ```

- Before writing any code, for each problem, discuss with your partner how you plan to approach solving the given problem. Once you both agree on a way to attempt the problem, only then should code be written.

- Pay attention to names and any other requested material. Failure to include material will result in the loss of points as described below.

Lab Problems
For all problems, use inline comments to describe major actions. Ask for assistance if needed.

1. do-while and Sentinel-Controlled Loops – Write a Java program named Lab06Prob01.java
    using a do-while loop. During execution of the loop, ask the user for floating-point values
    from -100.0 - 100.0 and determine the largest and smallest of those numbers. Any numbers
    outside of the range should be ignored. The loop should exit once the user enters a value of
    -19.5.[1] After the loop exits, provide descriptive text to indicate both the smallest and largest
    values entered up to one decimal place. The sentinel value should not be included in any
    calculations and you should account for the situation where NO valid values are entered
    and not print a max/min.

    For example, for the values 100.7, 50.2, -201.1, 701.7, 11.3, -15.7, 14.4, 37.4, -19.5 the max
    value is 50.2 and the min value is -15.7.

    Expected Output (valid values):
    ```
    The max value was 50.2
    The min value was -15.7
    ```

    Expected Output (no valid values):
    ```
    There were no valid values
    ```

    Note that Gradescope will provide its own input for this program and your program should
    correctly calculate the values for the given input, not just your tested values.

    After passing the Gradescope tests, show a working version of this program to the
    instructor or TA before moving on.

    **After instructor check-off, BOTH students should submit a final, identical version to
    Gradescope. This version is used for your final grade.**

2. Recall that for loops can be used to calculate approximations that are represented using
    running sums from one number to another number. For instance, a running sum of values
    from 0-50 (inclusive on both ends) can be represented mathematically as:

    $$\sum_{i=0}^{50} i$$

    and in Java code as:
    ```java
    int sum = 0;
    for (int i = 0; i <= 50; i++) {
        sum += i; // calculate each term
    }
    ```

---

[1] Typically, we would never use a double as a sentinel, but we will make an exception this time.

Write a Java program named Lab06Prob02.java that calculates the series represented by:

$$\sum_{i=5}^{250}(2i+1)^2$$

The program should calculate the series value based on the given summation using a for loop and display the final result with descriptive text, and auto-placed commas.

HINT: This should result in a sum of 21,084,086.

Expected Output:
`The sum is 21,084,086`

After passing the Gradescope tests, show a working version of this program to the instructor or TA before moving on.

After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.

3. while Loops (Sentinel-Controlled) – Write a Java program named Lab06Prob03.java that contains a while loop that can be controlled using a sentinel value. Write your loop so that only positive, even integer values are summed up and the resulting average of those numbers is output to the console once the sentinel value of 672 is entered.

   After the loop exits, provide output that includes descriptive text, the calculated sum, and the calculated average which should be limited to four decimal places. The sentinel value should not be included in any calculations. For instance, input of 4, 2, -5, -7, 8, 3, 7, 672 should provide a sum of (4+2+8) = 14 and an average of 14/3 ~= 4.6667.

   Expected Output:
   `For the positive, even numbers, the sum was XX and the average was X.XXXX`

   Note that Gradescope will provide its own input for this program and your program should correctly calculate the values for the given input, not just your tested values.

   After passing the Gradescope tests, show a working version of this program to the instructor or TA before moving on.

   After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.

4. Demonstrate all working programs to the instructor to receive credit for each problem. Turn in each problem inside the zip file detailed below.

5. Name the zip file Lab6LastName1LastName2.zip (e.g. Lab6JonesSmith) and submit that zip file to Folio. Both students must submit the exact same zip file for either to receive a grade.

Grade Breakdown

Demonstrate running application to instructor:

Problem 1	6 points
Problem 2	6 points
Problem 3	6 points
Zipped source files (in Folio):	2 points

Total Possible: 20 points

Last modified: September 23, 2024