Ira A. Fulton Schools of Engineering

school of computing, informatics, & decision systems engineering

CSE 110 – Assignment #4

Maximum points: 20 pts

Topics

- Loops (Chapter 4)

Use the following 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.

Important Note:

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. **(If not, -1 Pt)**

```
//***********************************************************
// Name: your name
// Title: title of the source file
// Author: (if not you, put the name of author here)
// Description: Write the description in your words.
// Time spent:  how long it took you to complete the assignment
// Date: the date you programmed
//**********************************************************
```

Your programming assignments require individual work and effort to be of any benefit. Every student must work independently on his or her assignments. This means that every student must ensure that neither a soft copy nor a hard copy of their work gets into the hands of another student. Sharing your assignments with others in any way is **NOT** permitted. Violations of the University Academic Integrity policy will not be ignored. The university academic integrity policy is found at http://www.asu.edu/studentlife/judicial/integrity.html

Part 1: Writing Exercise: (5 pts)

The following are the exercises about the loops in Java. Write the answers in a comment block before the code of Part2.

a. Consider the following code snippet. The programmer expected that it printed
`2014, ASU`
, but it did not. Explain the problem and solution in your word. **(1 pt)**

```
Scanner input = new Scanner ("2014\nASU\n");
int a = input.nextInt();
String b = input.nextLine();
System.out.println(a+ ","+b);
```

b. The following code generates the pattern (a) below. Explain the way to generate the pattern (b) in your word. It can be done by changing only one expression of for-loop. **(2 pts)**

```
for (int i = 0 ; i < 5; i++){
  for (int j = 0 ; j < 5; j++) System.out.print("*");
  System.out.println();
}
```

```
(a)             (b)
*****           *****
*****           ****
*****           ***
*****           **
*****           *
```

c. Consider the following code. Write the 2014th output and the values of **`i`** and **`j`** at the time. The first output is 0 when i is 0 and j is 0. *) Add a variable to keep the count, and add a condition to print out when the count is equal to 2014. **(2 pts)**

```
for (int i =0; i<100; i++)
  for (int j =0; j< 100; j++){
    System.out.println(i*j);
    j += 3;
}
```

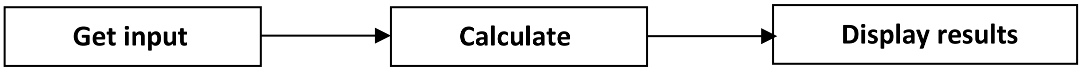
Note: The answers to the questions (a through c) above should be typed in the block of comments in the java file such as;

```
/*
a) The problem is …
b) To generate …
c) XXX when i is xxx, and  j is xxx
*/
```

Part 2: Programming (15 pts)

Write a Java program called **Assignment4.java**. The program is to display questions and read user inputs, then calculate and print out the requested data with a proper format.

This program will follow a very simple process.



The goal is to develop a program to continuously generate a list of numbers with a table format. Each time the user input has two integers. The first integer defines the number of column, and the second the number of row for the output table size. For example, when the input numbers are 4 and 3, it displays the numbers from 1 to 12 in a 4 x 3 table as shown below. Once a table is displayed, the program waits for the next input.

```
Input:     4 3

Output: *    |    1    2    3    4
        ------------------------
        1    |    1    2    3    4
        2    |    5    6    7    8
        3    |    9   10   11   12
```

This assignment has only one task, but the development process is decomposed into two steps. Follow the instruction one by one. Don't go to the next step if your program does not return the proper output in a step.

Step1 :
The first step is to develop a program to read the user input continuously as long as it is valid. The input should be a single line with two positive numbers such as "4 3". Once one of the input numbers is invalid (not integer or not positive), the program displays the message below and terminates.
`*** End of Program ***` **(5 Pts)**

Step2:
This is the main step in this assignment. Create nested loops to print out the numbers in a table format. The first integer `ROW` defines the number of rows, and the second integer `COL` defines the number of columns. Inside the loops, write the code to display the positive numbers from 1 to `Row*COL` **(5 Pts)**.
Use the `System.out.printf()` function with "%4d" and "%4s" formats for the integers and Strings to align them right. (See the section 2.3.2) **(5 Pts)**

Test your program in your development environment. If it works, submit the Assignment4.java to the online submission site. Check whether your code returns the correct answer like below.

```
Set the size of table: 4 3

   *   |   1   2   3   4
------------------------
   1   |   1   2   3   4
   2   |   5   6   7   8
   3   |   9  10  11  12

Set the size of table: 9 9

   *   |   1   2   3   4   5   6   7   8   9
--------------------------------------------
   1   |   1   2   3   4   5   6   7   8   9
   2   |  10  11  12  13  14  15  16  17  18
   3   |  19  20  21  22  23  24  25  26  27
   4   |  28  29  30  31  32  33  34  35  36
   5   |  37  38  39  40  41  42  43  44  45
   6   |  46  47  48  49  50  51  52  53  54
   7   |  55  56  57  58  59  60  61  62  63
   8   |  64  65  66  67  68  69  70  71  72
   9   |  73  74  75  76  77  78  79  80  81

Set the size of table: Q

*** End of Program ***
```

Use only the Java statements that have been covered in class to date. **DO NOT** use any other items out of the Chapter 1- 4 (Array, Array List, exit() etc.). If in doubt, ask. If you use them, then you lose the points of task. Complete each step one by one. Don't copy any code developed by others. Don't give your code to others. Don't use any algorithm, which you cannot understand. Your assignment file is checked by the MOSS (by Stanford Univ.), which is a program to detect cheatings.

Example Execution:

The following are example inputs and outputs. The user inputs are shown in red. Make your own questions rather than the examples.

Program Input (1)

```
4 3
9 9
-3 6
```

Expected Program Output (1)

```
Set the size of table: 4 3

   *   |   1   2   3   4
------------------------
   1   |   1   2   3   4
   2   |   5   6   7   8
   3   |   9  10  11  12

Set the size of table: 9 9

   *   |   1   2   3   4   5   6   7   8   9
--------------------------------------------
   1   |   1   2   3   4   5   6   7   8   9
   2   |  10  11  12  13  14  15  16  17  18
   3   |  19  20  21  22  23  24  25  26  27
   4   |  28  29  30  31  32  33  34  35  36
   5   |  37  38  39  40  41  42  43  44  45
   6   |  46  47  48  49  50  51  52  53  54
   7   |  55  56  57  58  59  60  61  62  63
   8   |  64  65  66  67  68  69  70  71  72
   9   |  73  74  75  76  77  78  79  80  81

Set the size of table: -3 6

*** End of Program ***
```

/***

Submit your homework by following the instructions below:

***/

• Go to the course web site (my.asu.edu), and then click on the on-line Submission tab. Log in the site using the account, which was registered at the first Lab session. Make sure you use the correct email address for registration. This will allow you to submit assignments. Please use your ASU e-mail address.

• Submit your **Assignment4.java** file on-line. Make sure to choose **HW4** from drop-down box.

• The **Assignment4.java** should have the following, in order:

- In comments, the assignment Header described in "Important Note".
- In comments, the answers to questions presented in Part#1.
- The working Java code requested in Part #2.
- The **Assignment4.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.**