Concordia University
Comp 248 – Winter 2016
Introduction to Programming
Combined Assignments 3 & 4 - Due by 11:59 PM Sunday March 20, 2016

Assignments 3 & 4

Purpose: The purpose of these assignments is to help you practice selection and repetition statements as well as user-defined classes.

TEAMS:
The assignment can be done individually or in teams of two from the **same** section. Submit one assignment per team.

General Guidelines When Writing Programs:
- Refer to information given in Assignment 1

Assignment 3:

Write a Java program that prints **one** of the following patterns based on the user choice of a *pattern number*, which must be between 1 and 4, or 5 to quit (See examples below), and an *input value*, which must be larger than 0 and smaller than 10, and according to the following:

A) If the user enters any invalid pattern number, then the program should display a message indicating that input was invalid and request the user to either enters a correct pattern number or 5 to quit the entire program. That is, an entry of 5 would simply terminate the program.
B) Once the pattern number is correctly supplied, the program requests the user to enter the *input value*. If the user enters any invalid value that is not within the expected range, then the program should reject this entry and asks the user to re-enter another value; this would repeat indefinitely until a good value is entered.
C) Upon the entry of a good input value, the program must check whether this value is odd or even. If the user enters an odd number, then the program would draw a pattern that is similar to the following (i.e. the shown pattern is drawn if the user enters 5. You should notice that this is only an example; your program must allow for the general case for different appropriate values as indicated above.) You should notice that the patterns are actually quite similar whether the input is odd or even, with the exception of pattern # 4.

```
54321          1     12345          1
5432          12      2345        123
543          123       345      12345
54          1234        45        123
5          12345         5          1
```

If the user enters an even number, the pattern would look as one of the following (i.e. the shown pattern is drawn if the user enters 4. Again this is just an example; your program must work for the general case.)
`

```
4321           1     1234           1
432           12      234         123
43           123       34         123
4           1234        4           1
```

Again, notice the behavior of pattern # 4 for an even and odd number of rows.

D) Once good values are entered and the pattern is drawn, the program will repeat; that is prompting the user to again enter a pattern number and an input value. This will repeat until the user enters 5 for a pattern number, which would stop the entire program.

<u>Restrictions</u>:

- You can use any combinations of the selection and repetition statements that we have seen in class. You must however write the needed code yourself (i.e. do not use any built-in methods that may perform what is needed.)

<u>General Algorithm:</u>

In general your program should:

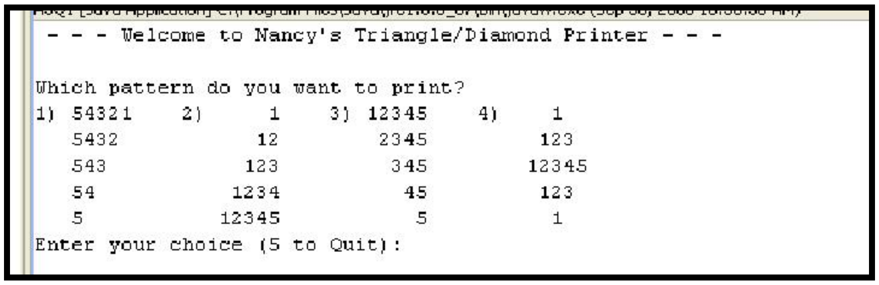
1. Display a welcome message with your name(s) on it (i.e. *"Welcome to Mike and Linda Triangle/Diamond Printer"*).
2. Prompt the user for the pattern they want to display. Make sure the pattern requested is a legal one.
3. Prompt the user for the input value. The input value should be greater than 0 and less than 10. Keep prompting the user until the input value is valid.
4. Display the requested pattern.
5. As long as the user wants to display another pattern your program repeats steps 2 to 4
6. Display a closing message

<u>A few hints:</u>

- For the displaying of the patterns (to allow the user to choose), you just need few System.out.println() statements to show how the patterns would look like. That is, you do not actually draw the patterns at that time.
- Work on one pattern at a time.
- For the fourth pattern you may want to divide the work into displaying the top half of the diamond, then display the bottom half of the diamond.

Here are a few sample runs (that the circled data has been entered by the user):

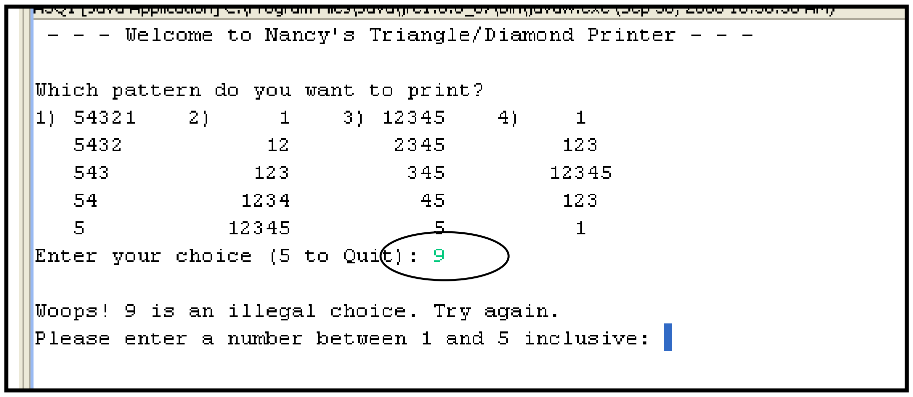
Sample Output # 1: Welcome message & Menu prompting user for pattern choice



```
 - - - Welcome to Nancy's Triangle/Diamond Printer - - -

Which pattern do you want to print?
1) 54321    2)      1    3) 12345    4)     1
   5432           12        2345          123
   543           123         345        12345
   54           1234          45          123
   5           12345           5            1
Enter your choice (5 to Quit):
```

Sample Output # 2: User entering illegal pattern choice (Not between 1 and 5 inclusive).

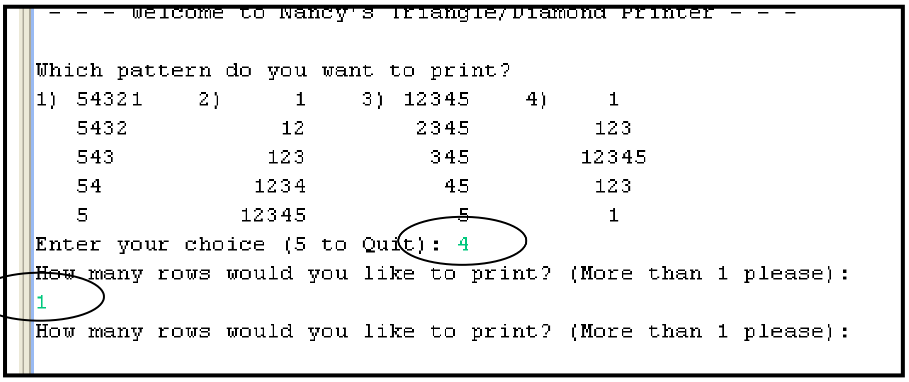


```
 - - - Welcome to Nancy's Triangle/Diamond Printer - - -

Which pattern do you want to print?
1) 54321    2)      1    3) 12345    4)     1
   5432            12        2345         123
   543            123         345        12345
   54            1234          45         123
   5            12345           5          1
Enter your choice (5 to Quit): 9

Woops! 9 is an illegal choice. Try again.
Please enter a number between 1 and 5 inclusive:
```

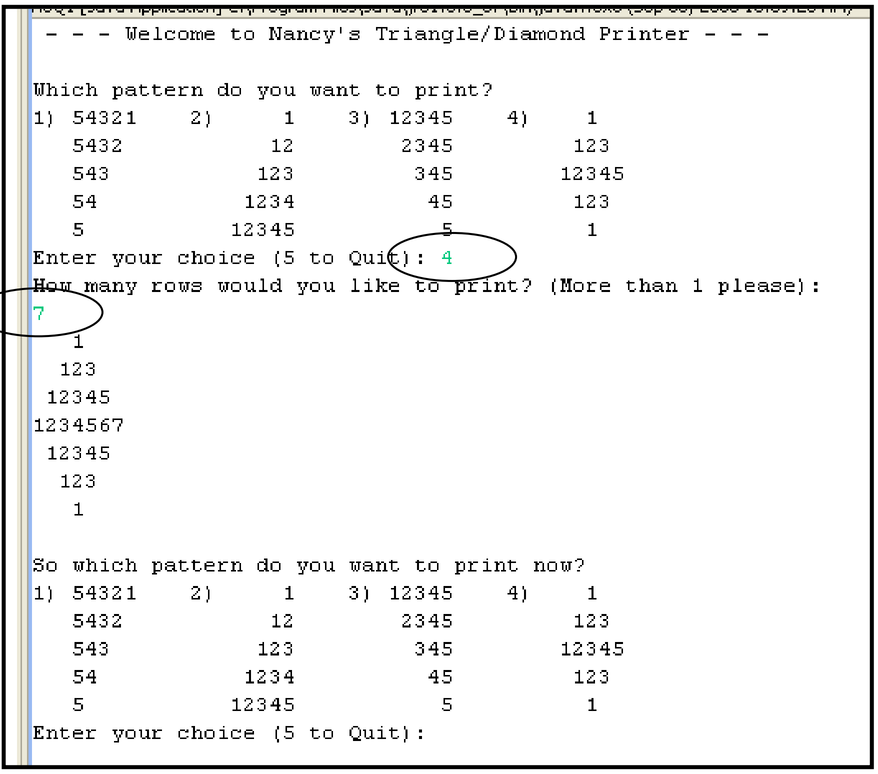
Sample Output # 3: User requesting number of rows less than 2.



```
 - - - Welcome to Nancy's Triangle/Diamond Printer - - -

Which pattern do you want to print?
1) 54321    2)      1    3) 12345    4)     1
   5432            12        2345         123
   543            123         345        12345
   54            1234          45         123
   5            12345           5          1
Enter your choice (5 to Quit): 4
How many rows would you like to print? (More than 1 please):
1
How many rows would you like to print? (More than 1 please):
```

Sample Output # 4: Requesting an odd number (7) of rows for pattern 4. Notice program displays the requested pattern, displays the menu and prompts the user for their next choice.

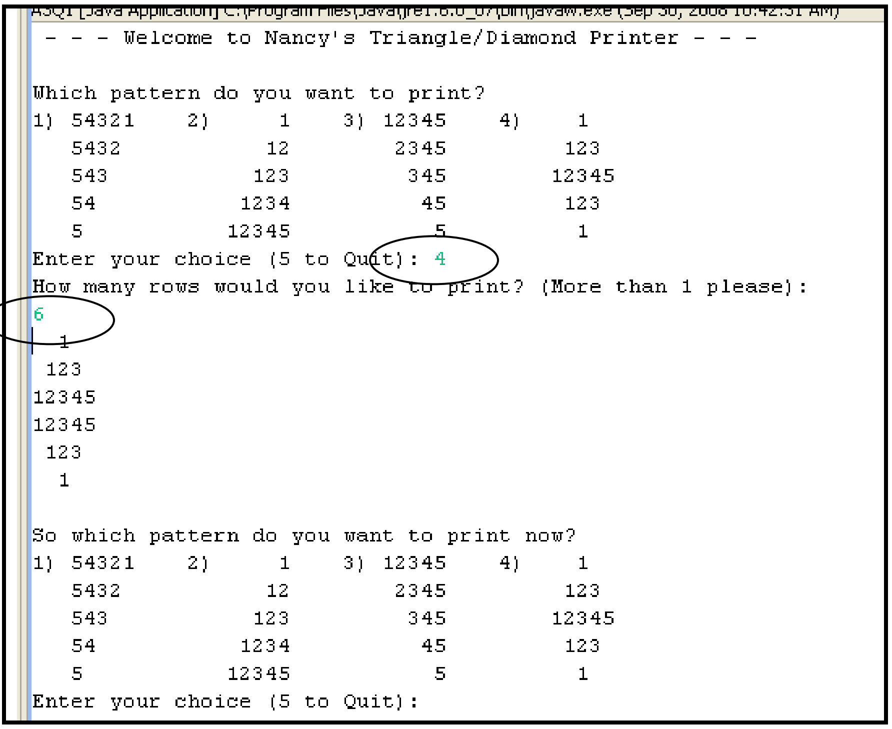


```
 - - - Welcome to Nancy's Triangle/Diamond Printer - - -

Which pattern do you want to print?
1) 54321    2)      1    3) 12345    4)     1
   5432            12        2345         123
   543            123         345        12345
   54            1234          45         123
   5            12345           5          1
Enter your choice (5 to Quit): 4
How many rows would you like to print? (More than 1 please):
7
   1
  123
 12345
1234567
 12345
  123
   1

So which pattern do you want to print now?
1) 54321    2)      1    3) 12345    4)     1
   5432            12        2345         123
   543            123         345        12345
   54            1234          45         123
   5            12345           5          1
Enter your choice (5 to Quit):
```

Sample Output # 5: Requesting an even number (6) of rows for pattern 4. Notice that the middle rows go from 1 to 6 (one less than the number of rows selected and that it is repeated twice to display 6 rows).

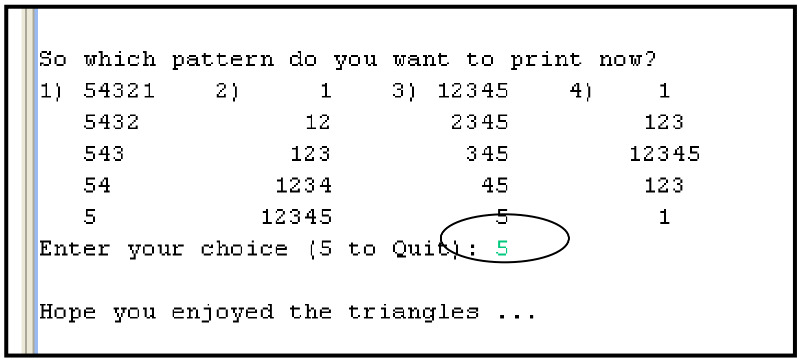


```
A3Q1 [Java Application] C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Sep 30, 2008 10:42:31 AM)
 - - - Welcome to Nancy's Triangle/Diamond Printer - - -

Which pattern do you want to print?
1) 54321    2)     1    3) 12345    4)     1
   5432           12        2345         123
   543           123         345        12345
   54           1234          45         123
   5           12345           5          1
Enter your choice (5 to Quit): 4
How many rows would you like to print? (More than 1 please):
6
  1
 123
12345
12345
 123
  1

So which pattern do you want to print now?
1) 54321    2)     1    3) 12345    4)     1
   5432           12        2345         123
   543           123         345        12345
   54           1234          45         123
   5           12345           5          1
Enter your choice (5 to Quit):
```

Sample Output # 6: Closing message



```
So which pattern do you want to print now?
1) 54321    2)     1    3) 12345    4)     1
   5432           12        2345         123
   543           123         345        12345
   54           1234          45         123
   5           12345           5          1
Enter your choice (5 to Quit): 5

Hope you enjoyed the triangles ...
```

Submitting Assignments 3

- <u>IMPORTANT:</u> You are allowed to work on a team of 2 students at most (including yourself!). Any teams of 3 or more students will result in 0 marks for all team members. If your work on a team, ONLY one copy of the assignment is to be submitted for both members.
- Zip together the source codes. (Please use WINZIP).
- Naming convention for zip file: Create one zip file, containing all source files for your assignment using the following naming convention:

  The zip file should be called *a#_studentID*, where # is the number of the assignment *studentID* is your student ID(s) number. For example, for the first assignment, student 1234567 would submit a zip file named *a1_1234567.zip*. If you work on a team and your

IDs are 12345678 and 34567890, you would submit a zip file named *a1_12345678_34567890.zip*.

- Submit your zip file at: https://fis.encs.concordia.ca/eas/ as **Programming Assignment** and submission **#3**. Assignments submitted to the wrong directory would be discarded and no replacement submission will be allowed.
- Submit only ONE version of an assignment. If more than one version is submitted the first one will be graded and all others will be disregarded.

Evaluation Criteria for Assignments 3 (10 points)

Comments - description of variables/ description of the steps in code/ purpose of program/ Indentation and readability of program/ Choice of variable names	2 pts
Clarity of output	0.5 pt
Menu operation / error checking	1 pts
Number or rows / error checking	1 pts
Pattern 1	1 pts
Pattern 2	1.5 pts
Pattern 3	1.5 pts
Pattern 4	1.5 pts

Assignment 4:

In this assignment you are required to write and use some user-defined classes. Here is what is given. A cell phone object has three attributes, a brand (String), a serial number (long), and a price (double).

Part 1:

You are required to design and implement the **CellPhone** class according to the following specifications:

- Upon the creation of a cell phone object, the object must immediately be initialized with valid values; that is brand, serial number and price.

- The design should allow enough flexibility so that the value of any of these attributes can be modified later on. For example, it should be possible to create a cell phone object with a given price then change its price later on.

- The design should allow the user of the class to obtain the value of any of the attributes individually. The design should also allow the user to view all cell phone information at once by passing cell phone objects to the print/println() methods;

- The design should allow for one cell phone to be compared to another cell phone for equality. Two cell phone objects are considered equal if they have the same brand and the same price.

Part 2:

In that part, you are required to write a public class called **UtilizeCellPhones**, which is going to utilize the CellPhone class that you have created in part1. In the main method of this class, you should perform the following operations:

1) Create three cell phone objects; notice that you must assign a brand, serial number (you can initialize it as any 9-digit number) and a price to each of these objects upon creation.
2) Show all information of each of the three objects.
3) Change the price of the first object; the price and the brand of the second object, and the serial number value of the third object.
4) Display only the modified attributes of the three objects. All outputs showing only the price of a cell phone (i.e. not the rest of the attributes) must be formatted.
5) Compare some of these cell phone objects for equality and display whether or not they are equal.

Submitting Assignment 4

- Same as indicated above for submitting Assignment 3; with the exception that you must submit your zip file at: https://fis.encs.concordia.ca/eas/ as Programming **Assignment #4.**

Evaluation Criteria for Assignment 4 (10 points)

Comments - description of variables/ description of the steps in code/ purpose of program/ Indentation and readability of program/ Choice of variable names	0.5 pts
Clarity of output	0.5 pt
Design and implementation of the CellPhone class and its methods	4 pts
Design of the **UtilizeCellPhones** class and satisfaction of the required operations	5 pts