

CSC 1301 SPRING SESSION 2020

ASSIGNMENT # 06

“WORKING WITH NESTED FOR LOOPS”

THERE ARE SEVERAL PROGRAM + (BONUS) 2 CHALLENGE PROGRAM

DUE 04/14/2020 @ 11:59 PM

(Please create a new java file for each program)

Program 1

1) Write the code to display the content below using user input.

The program Name is **Half_XmasTree**. This program MUST use (**ONLY**) for loops to display the output below from 10 to 1 therefore ...1st row prints 10 star 2nd row prints 9... 3rd print 8 stars and so forth...

This program is controlled by the user to input the amount of row.

A good test condition is the value of ten rows. The user will demand the size of the tree.
Remember the purpose of print() and println()

Enter the desired number of rows :

8

```
*****
*****
*****
*****
****
***
**
*
```

What to submit : Java source file, **Half_XmasTree.java**

Program 2

2) Write a program named **Ascii_Tree** using a nested for loop that will generate the exact output. You must code for 10 rows total.

Expected output:



What to submit : Java source file, **Ascii_Tree.java**

Program 3

3) Write a program named **Full_XmasTree** using a nested for loop that will generate the exact output.

This program **MUST** use (ONLY) for loops to display the output below. For example the 1st row prints 1 star 2nd row prints 2, the 3rd row print 3 stars and so forth...

This program is controlled by the user to input for the amount of row. "Prompt the user to enter the dimensions of the tree"

A good test condition is the value of ten rows.

(hint***)This program should account for white spaces print(" "). Remember the purpose of print() and println()

Expected output:

Please enter the desired number of rows :

6



Part B (additional to the same code/program #3)add a second loop of any sort to control the footing of the tree.

Here are some sample footing you must use one of these samples:

Please keep in mind I am just displaying the flow of the footing which is intergrated to the full tree

A) *****

 *

b) *****

 *

c) *****

 *
 *

d) *****

What to submit : Java source file, **Full_XmasTree.java**

Program 4 - Guess Number program

Program description

Write a program named **Guess_A_Number** that plays a game in which the program picks a secret number and the user tries to guess it.

1. The program first asks the user to enter the maximum value for the secret number.
2. Next, it chooses a random number that is ≥ 1 and \leq the maximum number.
3. Then the user must try to guess the number.
4. When the user succeeds, the program asks the user whether or not to play another game.

The following Sample shows what the user will see on the screen (user input is in **bold**):

2. Input and output

Guess the secret number.

Enter maximum value for secret number: **10**

A new secret number has been chosen.

Enter guess: **3** Too

low; try again.

Enter guess: **8**

Too low; try again. Enter

guess: **9**

Too low; try again.

Enter guess: **10** You

won in 4 guesses!

Play again? (Y/N) **y**

A new secret number has been chosen.

Enter guess: **7**

Too high; try again.

Enter guess: **3** Too

low; try again.

Enter guess: **5**

You won in 3 guesses!

Play again? (Y/N) **n**

The user may enter any number of spaces before and after each input. The program should terminate if the user enters any input other than y or Y when asked whether to play again.

Hints

- 1) Use two while statement (nested) for the whole program. Maybe you can incorporate if statemement(s)
- 2) Use the following statement to pick the secret number:
`int secretNumber = (int) (Math.random() * maxNumber) + 1;`
- 3) Use trim() method to trim any number of spaces in an input.
- 4) Use the equalsIgnoreCase method to test whether the user entered y or Y.

What to submit : Java source file, **Guess_A_Number.java**

CHALLENGE QUESTION #01 (5 POINTS)

1) Write a program that takes argument for: N and prints a $(2N + 1)$ -by- $(2N + 1)$. The sample **Expected Output** is shown below.

Hint ** Use two (2) `for` loops and one (1) `if-else` statement to accomplish this task.

```
/*****
```

```
*****
```

```
*
```

```
* Execution:
```

```
*
```

```
* Prints out an XMarksDaSpot.
```

```
*
```

```
*
```

```
*
```

```
* * . . . . . * *
```

```
* . * . . . . . * .
```

```
* . . * . . . . . * .
```

```
* . . . * . . . * . .
```

```
* . . . . * . * . . .
```

```
* . . . . . * . . . .
```

```
* . . . . * . * . . .
```

```
* . . . * . . . * . .
```

```
* . . * . . . . . * .
```

```
* . * . . . . . . * .
```

```
* * . . . . . . . *
```

```
*
```

```
*****/
```

CHALLENGE QUESTION #02 (10 POINTS)

2) Write a program name `Diamonds_R_4_Ever` that takes in the argument `N` and then prints a $(2N + 1)$ -by- $(2N + 1)$. The diamond example Expected Output is shown below.

```

/*****
 *   Compilation:  Diamonds_R_4_Ever.java
 *
 *
 *
 *
 *   Diamond_R_4_Ever
 *
 *   . . . . * . . . .
 *   . . . * * * . . .
 *   . . * * * * * . .
 *   . * * * * * * * .
 *   * * * * * * * * *
 *   . * * * * * * * .
 *   . . * * * * * . .
 *   . . . * * * . . .
 *   . . . . * . . . .
 *
 *****/

```

```

*****/

```