**IT 1090C Computer Programming I**

**IT 6090C Java Programming**

**Prof. Tom Wulf**

# Lab 7 – Get Loopy

**20 Points (2 gr or extra credit)**

For this individual assignment, implement the Looping problems listed here. For each problem insert a screen shot of the output immediately after the loop description (I.E. don’t put them all at the end) Best to do these as you go rather than go back and do them after coding everything). Finally, give me your complete IntelliJ project. Name your Project **GettingLoopy**.  
  
If directed by your instructor, use GitHub source control for your project.

# Directions:

**Part A: Counting (for) Loops (2 points each)  
(Don’t be clever and use a while loop! Make these for loops)  
Paste a Screen capture or output window copy to show each one.**

1. A loop that counts (i.e. displays the count) up by 1 from 0 to 30

Graphical user interface

Description automatically generated

1. A loop that counts down by 1 from 30 to 0

Graphical user interface, text

Description automatically generated

1. A loop that counts up by 3 from 0 to 18  
     
   Graphical user interface, text

   Description automatically generated
2. A loop that counts down by 2 from 10 to 0

Graphical user interface, text

Description automatically generated

**Part B: Output for loops (2 points each)  
Use nested for loops to do these. Again, paste the output copy or screen shot after each one.**

1. A nested loop that creates this figure (there are no blank lines)  
     
   \*  
   \*\*

\*\*\*  
\*\*\*\*  
\*\*\*\*\*

Graphical user interface, text

Description automatically generated

1. A nested loop that creates this figure (there are no blank lines)  
     
   \*\*\*\*\*  
   \*\*\*\*  
   \*\*\*  
   \*\*  
   \*

Graphical user interface, text

Description automatically generated

1. A nested loop that creates this figure (there are no blank lines)  
     
   \*\*\*\*\*  
   \*\*\*\*\*  
   \*\*\*\*\*  
   \*\*\*\*\*  
   \*\*\*\*\*

Graphical user interface

Description automatically generated

1. (6 Points)  
   Redo either the F to C or C to F converter program.  
   Create a new Java Main File called **CtoF.java** or **FtoC.java**. (Don’t create a new project, just put it in the current project with the previous java main class.)

Use a do while loop to bulletproof the user input. Now, when the user fails to enter a valid number, the program will display an error msg and then loop and prompt them again to try again to input the value correctly. So, the program will block and repeat until they get it right. Be sure to thoroughly test your program. Include screen shots output copies here that show the tests and output.

Text

Description automatically generated

# Extra Credit or Graduate Option: (2 points)

Just add this code to the end of your main method.  
  
Write a program that outputs an elaborate **capital** version of your first initial using \* characters. (Note if your first initial is I use your last initial.) Use loop­**s** to print out duplicate lines in the character. Again no blank lines. Here the first loop would do the first 3 lines and a second would do the rest.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \*\*\*  
 \*\*\*  
 \*\*\*  
 \*\*\*  
 \*\*\*  
 \*\*\*

**Provide the screen shot here:**  
  
Submitting your work:

Submit this word.docx: **FirstName\_LastName\_Lab\_07.docx.**   
  
Include a zip archive of your entire IntelliJ project folder or a link to the GitHub repo for your project. Submit this along with this word.docx.