CSC 101 **Name: _____**

Lab Seven **Date: _____**

Objectives:

- Continue our understanding of loops
- Introduce nested loops
- File input and output
- Basic string functions
-

Procedures:

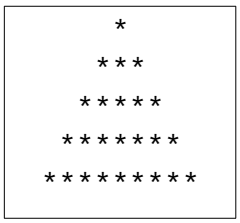
1. Examine the following program:
Use the space below to trace this program, giving the state of the variables, along with its output:

```
 1 #include<iostream>
 2 using namespace std;
 3
 4 int main()
 5 {
 6     for(int x=0; x<10; ++x)
 7     {
 8         for(int y=0; y<=x; ++y)
 9         {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

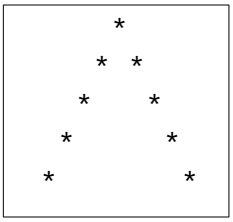
2. Alter the program above so that it prints the following pattern:
Print the code for your new program and attach it to this lab sheet.

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

3. Using what you have learned above, create a program that prints the following pattern:
Print the code along with the output of the program and attach it to this lab sheet.



4. Using what you have learned above, create a program that prints the following pattern:
Print the code along with the output of the program and attach it to this lab sheet.



5. Write a complete C++ program that prompts the user for a student name and 4 test grades and writes them to a file. Make sure to write a complete program and document your code. Compile and run your program, opening the created text file and making sure that it works properly. Attach the program along with the created text file to this lab sheet.

6. Alter the program you created in the previous step to read in 10 student names and their test grades. Write these names and grades to a file. Compile and run your program, opening the created text file and making sure that it works properly. Attach the program along with the created file to this lab sheet.

7. Write a complete C++ program that opens the text file you created in the previous step and reads each record from the file, reading the student name and his/her 4 grades. For each record read, print the student name and his/her average. Compile and run your program to make sure that it works properly. Attach the program, along with its output, to this lab sheet.

8. Write a gradebook application that minimally offers the user the following menu:
    1. Create a grade file for a class.
    2. Display the gradebook for a class.
    3. Calculate averages for class.
    4. Exit

The program should continue to execute until the user terminates by selecting choice 4. If choice 1 is selected, the user should be prompted for each student's name and four test grades, then create a file composed of this information with a filename selected by the user. If the user selects choice 2, the program should prompt for the name of the grade file, read each student and his/her grades, and print the information to the screen. If the user selects choice 3, the program should prompt for the name of the grade file, then read each student and his/her grades, calculate the student's average, and print the student's name and average.

Make sure you write a complete C++ program. Add judicious comments to document your code.

Include a header, like the one from previous labs, displaying your name, the date, and the course number.

When you have finished, print out the code and attach it to this lab sheet.