CSCI 201 – Computer Science 1

Closed lab assignment 5: Creating regular shapes with for loops.
Due date: Friday February 9

Objectives: *Design an algorithm for generating a two-dimennsional pattern.*
Employ the concept of passing parameters in algorithm implementation.
Learn about the need for reference parameters in C++.

IMPORTANT
The sample code for this assignment is available in `CourseInfo/TriangleANDSqrt/trianglefunc.cpp` *in* `CourseFiles`.

As we have seen, the for loop can be employed to create regular shapes (i.e., two-dimensional patterns) using ASCII characters as "pixels". In this assignment we create Vee-shaped figures with a specified height and line thickness.

Input: The program prompts for the height(h) of the Vee, the line thickness(t), and a character(`ch`). (These quantities are referred to as the input parameters for the shape we need to construct.)

Output: A Vee-shaped figure with height h, (the figure occupies h lines of input), line thickness t (the two arms of the Vee have a width of t characters on each line) shaded with the specified character.

Example: If the input parameters are 7(height), 6(line thickness) and *(character), the output would be as shown below.

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

Note that all lines of the image have the same structure and can be formed by writing the following:
(i) a sequence of blank space characters (call this "gap1")
(ii) a sequence of visible characters (call this "dash1")
(iii) another sequence of blank space characters ("gap2")
(iv) another sequence of visible characters (call this "dash2")
As in the examples in class, we use the following function:
`void printChars(int num, char ch, ostream& output)` (The function can be found in `courseinfo/TriangleANDSqrt` in CourseFiles) Thus each line of a Vee can be printed by calling `printchars` four times, with appropriate parameters for `num` and `ch`. For instance, the second line of the figure can be generated on the screen by the following sequence of operations: *Write one space, write 6 stars, write 12 spaces, write 6 stars.* These operations can be carried out using the

the following calls to `printChars()`

```
printchars(1,' ',cout); printchars(6,'*',cout);
printchars(12,' ',cout); printchars(6,'*',cout);
cout << endl;
```

Question 1: Count the number of spaces and stars on each line for the above example and come up with a sequence of operations that will construct the figure. Repeat this process for a figure with a height of 9 and line thickness of 7.

Question 2: Looking at the above examples and the numbers in the function calls for drawing these, identify the pattern(s) in the sequence(s) of dashes and gaps. What would be the sequence for a height **'h'** and thickness **'t'**?

Question 3: Construct the loop for this process using **psuedocode** (not on Raptor). Pseudocode is an artificial and informal language that helps programmers develop algorithms. This video is a helpful introduction.

Question 4: Write a C++ program that implements the flowchart using a `for` loop. Use the `printchars` function described above. In a script session, display ("cat") the code, compile, and test it. Upload the answers to questions 1, 2 and 3, the `.cpp` file and the script file into a folder named `Lab5` inside your CourseFiles folder.