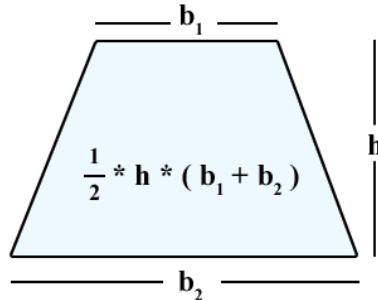


COSC 2347: Homework 2

Due: 11:59 PM on Wednesday, October 9, 2016

PROGRAM DESCRIPTION:

The purpose of this programming project is to write a C++ program that uses loops and programmer-defined functions to print out an isosceles trapezoid and calculate its area. For reference, refer to the following diagram of an isosceles trapezoid with its formula to compute its area:



REQUIREMENTS:

- As with all homework programs in this course, your program's output initially display the department and course number, your name, your SAMID, and your e-mail address.
- You will prompt the user to enter an odd integer between 1 and 17, inclusively, to use as the top base, b_1 , of the trapezoid. You will validate the user's input by creating a programmer-defined function to ensure that the integer is an odd integer in the range 1 to 17, inclusively. The integer entered by the user should be passed as a parameter to this function at a minimum (read: more parameters may be passed as needed). If the integer is not valid, you will continually re-prompt the user to enter the integer again until the user enters a valid integer. The return type of this function should be a Boolean data type and you are to use this Boolean result in determining whether or not the user input is valid.
- Once the user has entered a valid integer for the top base of the trapezoid, you will then prompt the user to enter an odd integer between 3 and 19, inclusively, to use as the bottom base, b_2 , of the trapezoid with the condition that the bottom base, b_2 , is greater than the top base, b_1 . To validate the bottom base, b_2 , you will use (and modify as needed) the programmer-defined function used to validate the top base, b_1 , being sure to enforce the new range and the inequality that b_2 is greater than b_1 . Similarly, if the integer is not valid, you will continually re-prompt the user to enter the integer again until the user enters a valid integer.
- You will prompt the user for and read in a printable character that will be used to draw the trapezoid. You may assume that the user enters a printable character.

COSC 2347: Homework 2

- You will draw the trapezoid using a programmer-defined function that accepts both integers (for each trapezoid base, b_1 and b_2) and the printable character entered by the user as parameters to the function. This function should be a `void`-function that does not return a value. It should print a trapezoid of the appropriate size using the printable character entered by the user. The trapezoid shall be drawn as follows:
 - The bottom base, b_2 , represents the number of user-entered printable characters used in the bottom row of the trapezoid.
 - The top base, b_1 , represents the number of user-entered printable characters used in the top row of the trapezoid.
 - Each row shall contain two less user-entered printable characters than the row beneath it. For example, if the bottom base, b_2 , is specified by 15 printable characters, then the row on top of the bottom base shall be drawn with 13 printable characters, and so forth.
 - The height of the trapezoid will be given by the number of rows it takes to print the bottom base, b_2 , to the top base, b_1 . For example, if $b_2 = 13$ and $b_1 = 5$, then this trapezoid will have a height of 5 rows ($b_2 = 13$, next row 11, then 9, then 7, and finally $b_1 = 5$). You may not prompt for the height of the trapezoid.
 - You may only use `cout` statements that print a single character (i.e., that printable character entered by the user), a single space, or a single new-line character (such as `'\n'` or `endl`). The goal is to maximize the use of repetition with nested `for` loops and minimize the number of `cout` statements.
- After you have drawn the trapezoid, you will compute the “area” of the trapezoid using the area formula: $area = \frac{1}{2} * h * (b_1 + b_2)$. You should find that this “area” also corresponds to the number of printable characters used to print the trapezoid itself! To compute the “area” of the trapezoid, you will create a programmer-defined function that accepts only the two bases, b_1 and b_2 , of the trapezoid; thus, you must calculate the height of the trapezoid and may not prompt the user for the height. The return type of this function should be an integer data type and you are to use this result in printing out the value of the area of the trapezoid.
- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code.
- Your program source code should be named “**homework2.cpp**”, without the quotes.

COSC 2347: Homework 2

- Your program will be graded based largely on whether it works correctly on the Linux machines
- This is an individual programming assignment that must be the sole work of the individual student.

You may assume that all input will be of the appropriate data type, although the value itself may not be valid.

DESIGN (ALGORITHM):

On a piece of paper (or word processor), write down the algorithm, or sequence of steps, that you will use to solve the problem. You may think of this as a “recipe” for someone else to follow. Continue to refine your “recipe” until it is clear and deterministically solves the problem. Be sure to include the steps for prompting for input, performing calculations, and displaying output.

You should attempt to solve the problem by hand first (using a calculator as needed) to work out what the answer should be for a few set of inputs. You should also use these hand-calculations to help you figure out how to print the trapezoid (e.g., look for relationships between the number of spaces needed versus the number of printable characters on a row) and compute the area of the trapezoid.

Type these steps and calculations into a document (i.e., Word, text, PDF) that will be submitted along with your source code. Note that if you do any work by hand, images (such as pictures) may be used, but they must be clear and easily readable. This document shall contain both the algorithm and any supporting hand-calculations you used in verifying your results.

SAMPLE OUTPUT (input shown in **bold green**):

```
islam0299@faculty:~/cosc2347$ ./a.out
+-----+
|               Computer Science               |
|          COSCE 2347 - Special topic          |
| Student Name      SamID                      |
+-----+

Enter odd integer for base1 of trapezoid (1 - 17): 4
Enter odd integer for base1 of trapezoid (1 - 17): 19
Enter odd integer for base1 of trapezoid (1 - 17): 5
Enter odd integer for base2 > base1 of trapezoid (3 - 19): 3
Enter odd integer for base2 > base1 of trapezoid (3 - 19): 20
Enter odd integer for base2 > base1 of trapezoid (3 - 19): 18
```

COSC 2347: Homework 2

Enter odd integer for base2 > base1 of trapezoid (3 - 19): 13

Please enter a printable character to draw trapezoid: @

```
    @@@@
  @@@@@@
@@@@@@@@
@@@@@@@@@@@@
@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
```

The area of the trapezoid is 45 units squared.

islam0299@faculty:~/cosc2347\$./a.out

```
+-----+
|               Computer Science               |
|           COSCE 2347 - Special topic          |
| Student Name      SamID                      |
+-----+
```

Enter odd integer for base1 of trapezoid (1 - 17): 11

Enter odd integer for base2 > base1 of trapezoid (3 - 19): 19

Please enter a printable character to draw trapezoid: \$

```
    $$$$$$$$$$
  $$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$
```

The area of the trapezoid is 75 units squared.

islam0299@faculty:~/cosc2347\$./a.out

```
+-----+
| Computer Science and Engineering              |
|           COSCE 2347 - Special topic          |
| Student Name      SamID                      |
+-----+
```

Enter odd integer for base1 of trapezoid (1 - 17): 3

Enter odd integer for base2 > base1 of trapezoid (3 - 19): 5

Please enter a printable character to draw trapezoid: &

```
    &&&
&&&&&
```

The area of the trapezoid is 8 units squared.

TESTING:

Test your program to check that it operates as desired with a variety of inputs, especially boundary values or error conditions. Then, compare the answers your code gives with the ones you get from hand calculations.

SUBMISSION:

Your program will be graded based largely upon whether it works correctly on the CSE machines, so you should make sure your program compiles and runs on the CSE machines.

CSCE 1030: Homework 3

Your program will also be graded based upon your program style. This means that you should use comments (as directed), meaningful variable names, and a consistent indentation style as recommended in the textbook and in class.

- Program Header Example:

```
/*
=====
Name       : homework2.cpp
Author      : Mark A. Thompson
Version     :
Copyright   : 2015
Description : The program performs simple arithmetic operations based on in-
                put from the user.
=====
*/
```

- Function Header Example:

```
/*
=====
Function    : deposit
Parameters  : a double representing account balance and a double represent-
                ing the deposit amount
Return      : a double representing account balance after the deposit
Description : This function computes the account balance after a deposit.
=====
*/
```

We will be using an electronic homework submission on Blackboard to make sure that all students hand their programming projects on time. You will submit both (1) the program source code file and (2) the algorithm design document to the **Homework 3** dropbox on Blackboard by the due date and time.

Note that this project must be done individually. Program submissions will be checked using a code plagiarism tool against other solutions, so please ensure that all work submitted is your own.

Note that the dates on your electronic submission will be used to verify that you met the due date and time above. All homework up to 24 hours late will receive a 50% grade penalty. Later submissions will receive zero credit, so hand in your best effort on the due date.

As a safety precaution, do not edit your program (using `vi` or `pico`) after you have submitted your program where you might accidentally re-save the program, causing the timestamp on your file to be later than the due date. If you want to look (or work on it) after submitting, make a copy of your submission and work off of that copy. Should there be any issues with your submission, this timestamp on your code on the CSE machines will be used to validate when the program was completed.