

CS 135

Design Assignment 4 (DA4-10/04)

Programming Assignment 5 (PA5-10/08)

As specified in your syllabus, you must turn your assignments in by 6:00 pm on the due date specified. If it is turned in late, but prior to 12:00 midnight the day it is due, credit will be reduced by 50% of the earned score. Any laboratories turned in more than 6 hours late will not earn any credit.

Objectives:

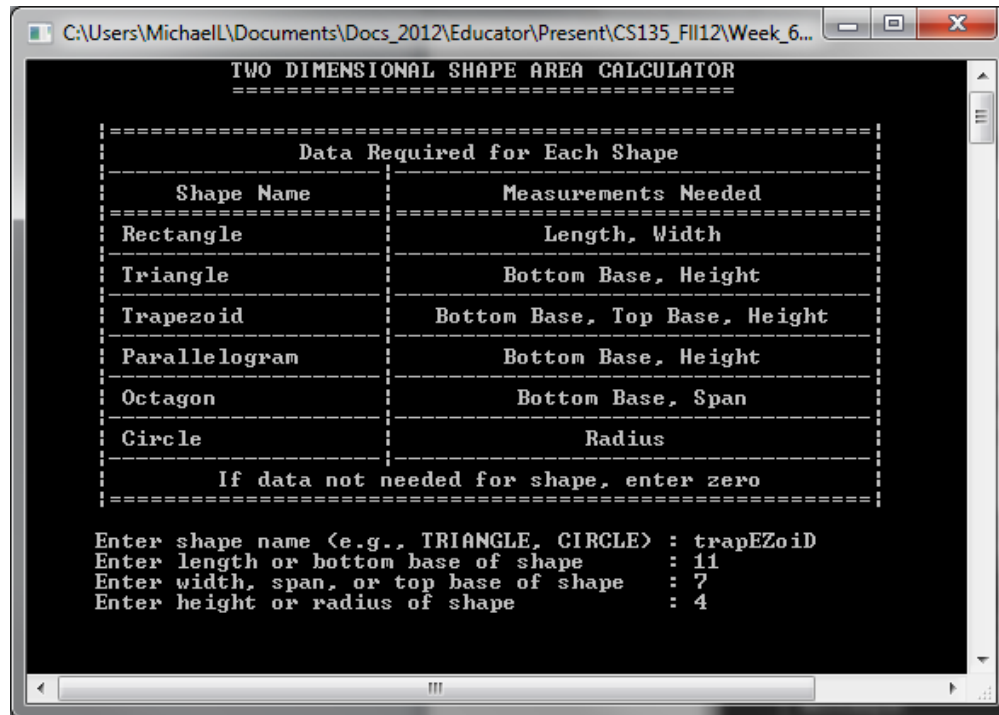
- 1) You will use a systematic development process to create a program
- 2) You will use a set of standardized functions to implement command-line I/O operations in a formatted command line system
- 3) You will use global constants to assist with program clarity
- 4) You will create simple functions as part of applying program modularity, and to allow for reuse of functions when code components are repeated
- 5) You will implement mathematical operations in order to solve problems
- 6) You will develop and implement decision-making actions in your program

Tasks:

Calculating Two-Dimensional Shape Areas (Part Deux)

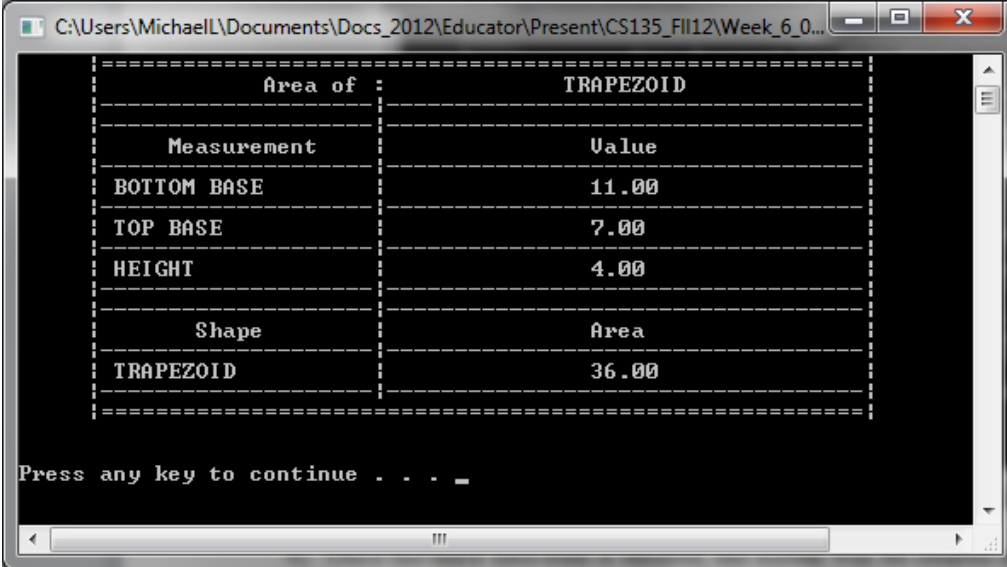
- 1) The program, called **shapes2.cpp** will continue to require the use of the **formatted_cmdline_io_v08.h** so that nice looking tables can be displayed. You may not use any of the other I/O tool sets (e.g., `iostream` or formatted console I/O), and you may not use any other header files or tools for this assignment.
- 2) The program will accept the kind of two-dimensional shape that is to be calculated and then the data necessary to do the calculations. The output table will then show the results related to the specific shape entered.

- 3) The initial part of the program will show a title and it will show the necessary parameters for calculating the areas of one of six shapes. The initial screen and the data entry is shown here. Note that the screen will have to be extended vertically to show all of this part of the program. Also note that the letter case does not matter for the shape entered as long as it is spelled correctly.



```
C:\Users\Michael\Documents\Docs_2012\Educator\Present\CS135_Fll12\Week_6...
TWO DIMENSIONAL SHAPE AREA CALCULATOR
=====
Data Required for Each Shape
=====
Shape Name      Measurements Needed
-----
Rectangle      Length, Width
Triangle       Bottom Base, Height
Trapezoid      Bottom Base, Top Base, Height
Parallelogram  Bottom Base, Height
Octagon        Bottom Base, Span
Circle         Radius
=====
If data not needed for shape, enter zero
=====
Enter shape name (e.g., TRIANGLE, CIRCLE) : trapEZoid
Enter length or bottom base of shape      : 11
Enter width, span, or top base of shape   : 7
Enter height or radius of shape           : 4
```

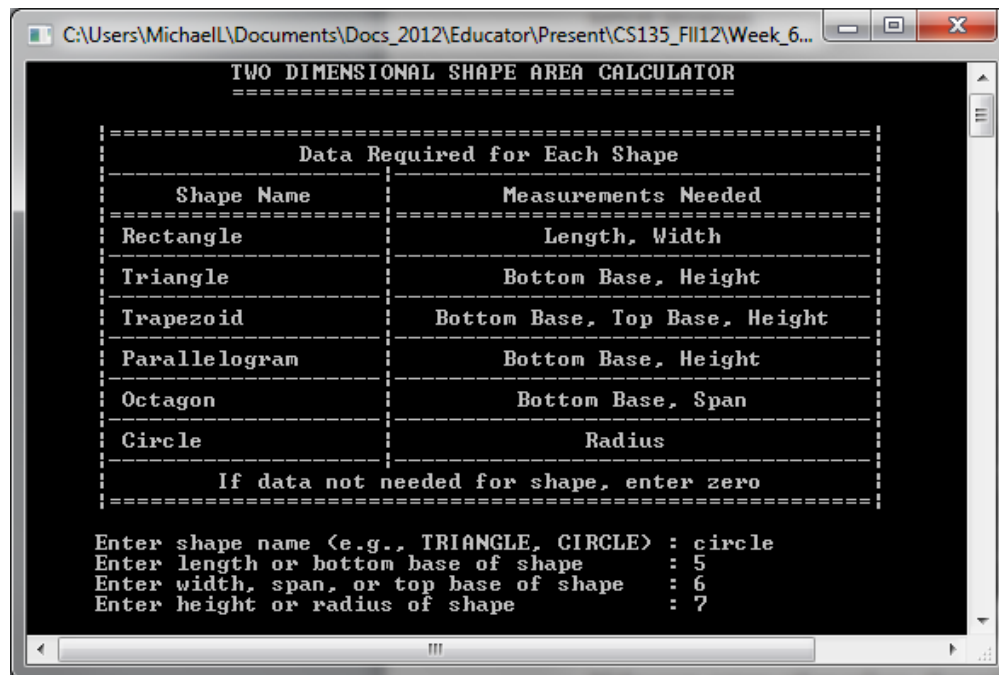
- 4) Once the third data item is entered, the screen will be cleared and the results will be displayed. For this program, the result table will only show the measurements and results for the specified shape. The area of this trapezoid is 36.00, as shown here. Also note that even though the shape name was entered with varying cases, the display shows an upper case word. This will be discussed later in this document.



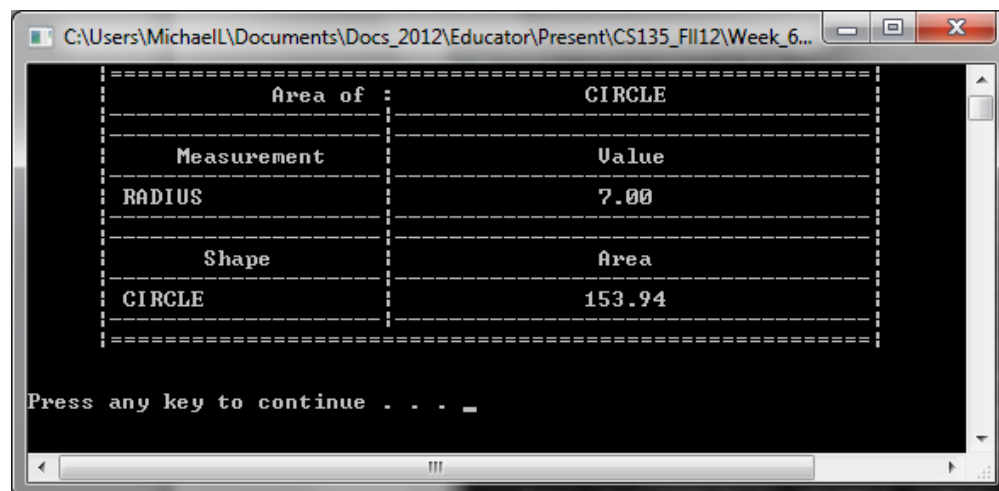
Area of		TRAPEZOID
Measurement		Value
BOTTOM BASE		11.00
TOP BASE		7.00
HEIGHT		4.00
Shape		Area
TRAPEZOID		36.00

Press any key to continue . . . _

- 5) Some items, such as a circle, need only one length (i.e., the radius). When this is the case, the user has the option to enter zero for all the other input items, but this is not required for this newer program.



- 6) If the user enters other values, they will be ignored if they are not needed, as shown



- 7) The challenges for this program are that you need to decide which shape you have before you can calculate the area and you must make decisions as to which measurement components to display when showing the output. For example, the rectangle has three measurements being the length, the width, and the height while the trapezoid also has three measurements which are bottom base, top base, and height. Your output display must appropriately show these measurement names along with the measurements themselves. Before starting this assignment, make sure you thoroughly analyze the example program so you can understand what your program must be able to do.
- 8) You may reuse any appropriate code you wish to use from the previous assignment but make sure you develop your design before trying to use the functions. This design will be different enough that trying to build your program around the old design will cost you much more time than it should.
- 9) You will also need another pair of functions to support your program. As mentioned previously, the user may enter the shape name with any case (i.e., lower or upper) she wishes to use although she still has to spell the name correctly. In order for your program to handle this and make good decisions, you will have a function called **toUpperCaseWord** which will depend on another function called **toUpperCaseLetter**. Both of these functions are provided for you in a separate text file. You can copy them as needed into your program but make sure you put the right parts into the right places. Remember that even though you have been given the implementation code, you must not show any code in your Design Assignment (i.e., steps 1 – 5). You will need to use the specifications, prototypes, and stubs in appropriate places but remember that you will lose credit if you show any code in supporting functions in step 5. Also note that since you will be given the specifications, it is expected that you will know how to appropriately use the **toUpperCaseWord** function (you will not use the **toUpperCaseLetter** function but **toUpperCaseWord** depends on it).
- 10) This is now your third program with functions, and you have the addition of some simple decision-making operations to the program. Your functions will not be specified this time and for most or all of the programs from now on. However, you must demonstrate good judgment in designing and implementing the appropriate functions for this assignment.
- 11) If you have any questions for following the six step process, review the pertinent sections of the online reference and/or the online code examples and videos, or check with the CS 135 Instruction Team.

- 12) You will develop a Design Assignment which includes the step 1 through 5 source code files by Thursday at 6:00 pm, and then you must develop the program code for next Monday at 6:00 pm. Further explanation of the required components and uploading process are provided near the end of this document.
- 13) You must also acquire six screen shots of the input and resulting output of your program operation (two screen shots each for a total of twelve). The first two pairs of screen shots should show the input used and the output displayed in the examples shown previously in this document. The other four should show examples of operations on the remaining shapes (e.g., rectangle, triangle, parallelogram, octagon). Again remember to annotate your screen shot document.

Turning in your Design Assignment:

Information:

Week: 6

Laboratory: 6

Design Assignment: 4

Due Date: 10/04, 6:00 pm

To turn in:

The first five steps of the Six Step Programming Process, including:

1. shapes2_s1.cpp
2. shapes2_s2.cpp
3. shapes2_s3.cpp
4. shapes2_s4.cpp
5. shapes2_s5.cpp

Upload these as separate files. Note that following the instructions for uploading your work is critical; you will lose points if you do not follow these instructions. Do not upload any files or in any format other than that specified here.

For information on how to turn in Design Assignments, refer to the "How to Turn in Design Assignments" in the "General Course Information" folder

Important grading note: Due to the large number of students and the small number of people grading assignments, a fraction of the Design Assignments will be graded each week. If your DA is not graded that week, you will receive full credit; if it is graded, you will receive feedback and the appropriate grade. The random process will ensure that all students will be graded an equal number of times across the semester, and it also provides the possibility of getting graded more than one week in a row. You must do your best on each DA and assume you will be graded each week.

Important DA feedback note: If your DA does not get graded in one particular week, you should do the following:

1. At least look at the provided sample DA when it is uncovered on the Tuesday after the PA is due; make sure your structure, organization, and problem-solving strategies are comparable
2. Stop by one of the Instruction Team offices (i.e., Instructors, TAs, Tutors, etc.) and ask them to help you review your DA so you can be sure to do well when you are graded

Turning in your Programming Assignment:

Information:

Week: 6

Laboratory: 6

Programming Assignment: 5

Due Date: 10/08, 6:00 pm

To turn in:

1. The Word file containing the following:
 - a. There should be at least six pairs screen shots (i.e., a total of twelve screen shots) for the programs as specified in item #12 previously in this document
 - b. Remember to clearly annotate every displayed result
2. The executable file:
 - a. shapes2EX.exe (shapes2_s6.exe is also acceptable)
3. The source code file:
 - a. shapes2_6.cpp

These files must be compressed and uploaded as one zip file. To do this, select all of the required files, right click on them, and select “Send To”, then select “Compressed (zipped) Folder”.

Once the folder is created, it will be placed in the same folder in which you are working. Change the name of the zipped folder to “LastnameFirstname_PAX” (where ‘X’ is the number of the Programming Assignment) as shown in the following example: “LeveringtonMichael_PA3” (no quotes). After you have renamed the zipped folder, double click on it to verify that it has all the files it is supposed to have.

Note that following the instructions for uploading your work is critical; you will lose points if you do not follow these instructions. Do not upload any files or in any format other than that specified here.

For information on how to turn in Programming Assignments, refer to the "How to Turn in Programming Assignments" in the "General Course Information" folder