Assignment #2 – Area/Volume of Shapes
Due: Thursday, 4/19/12, 11:59pm

(75 pts) Problem Statement: Write a program that calculates the area and/or the volume of shapes. The user chooses a shape, which should include the following: Circle, Rectangle, Trapezoid, and Triangle

Since we have not learned about strings, you can choose to read a number or character to determine the shape, but whichever you choose, you MUST be explicit in your prompt to the user. For example, if I choose to use numbers to correspond to the shapes, then I might ask the user to enter a shape with the following prompt:

To select a shape, enter a number from the following menu below:
[ 1 ] Circle
[ 2 ] Rectangle
[ 3 ] Trapezoid
Select shape:

After the user selects a shape, you need to determine if the user wants to calculate the area of the shape, the volume of the solid, or both. Once again, you need to choose between reading a number or character from the user. For instance if you choose to use numbers, then you might have a prompt like the following:

Would you like to calculate the area and/or volume for the specified shape? Select [ 1 ] Area, [ 2 ] Volume, or [ 0 ] both:

At this point, you must prompt the user for the specific dimensions based on the shape and the calculation to perform. For example, if the user chooses to find the area and volume of a circle, then you only need to read the radius for both calculations. Whereas, if the user chooses to find the area and volume of a rectangle, then you need to read the height and width for the area, in addition to the length for the volume. However, if the user chooses to calculate only the area of a rectangle, then you need to only read the height and width.

Here are the Area and Volume calculations for each shape:

Circle/Sphere: Area = pi * r^2  Volume: 4/3 * pi * r^3
Rectangle/Prism: Area = h * w  Volume: area * length
Trapezoid/Prism: Area = ½ * (b1 + b2) * h  Volume: area * length
Triangle/Prism: Area = ½ * b * h  Volume: area * length

(5 pts) Extra Credit: Ask the user if he/she wants to continue to select another shape and calculation to perform. Read a character corresponding to yes or no. The user should be able to enter an upper case or lower case letter for their response, i.e. y or Y for yes and n or N for no.
(10 pts) In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Read the class style guideline for more information: http://classes.engr.oregonstate.edu/eecs/spring2012/cs151-001/151_style_guideline.pdf

You are graded on having a header, proper comments, and readable code with indentation and vertical spacing that is CONSISTENT throughout your program. DO NOT align your entire program on the left side. This will cause you to automatically lose the full 10 points. In addition, do not forget your program header!!!

(15 pts) You are required to turn in a written document (as a pdf) addressing Polya’s steps to solving a problem with step 3 being the C code you write to carry out/implement your plan. With this said, your written document must include these three sections:

**Understanding the Problem**
In your own words, explain what YOU think the problem is asking you to do. In this section, document your uncertainties about the problem and anything else that you feel was unclear or vague. This is to ensure that YOUR understanding matches MY understanding of the problem☺

**Devising a Plan/Design**
At a minimum, provide an algorithm/pseudo code you designed to help solve the problem. In addition, include pictures/flow charts you used to help you devise your plan, as well as any other design decisions you made such as how to manage your time, how to decompose the problem, where to start first, etc. You can scan any handwritten work and attach it to the document as needed.

**Looking Back/Self-Reflection**
Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. Also, include a statement about what you learned from the assignment.

Electronically submit your C program file by the assignment due date, using TEACH.