**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ENGE 2314 Engineering Problem Solving With C++**

**Module 6 — User-defined Functions Part I**

**Summary of Tasks**

1. Read Sections 5.1 – 5.2 in the text ……………………........................................pp. 183 – 195
2. Answer review questions 1, 2, 5, 6-10 ...……………….…….………………….pp. 242 – 243
3. Watch:

“HW6 Intro”

two .avi’s in “User Defined Functions”

…………………………………….in the Classroom AVI Files folder in Scholar’s Resources

1. Review Section 2.7 in the text………………………………………………………pp. 73 - 78
2. Review Appendix A <cmath> and <cstdlib>…………………………………….pp. 523 - 525
3. Read “Reading Assignment Amplifications and Clarifications”…...provided in this document
4. Contribute to the forum by answering some of the “Questions about Week 6 Material,” or starting a thread on anything you’re confused about.
5. **Complete and submit take-home test HW6………………………individual graded effort**

**WARNING:**

HW6 is considered a take-home test, worth 10% of the final grade. Be very careful about following the honor code on this test:

* You may work through the engineering problem – NOT THE PROGRAM – with each other. You may do this informally or via the forum.
* You may discuss, post and share the code for any example from the .avi’s, text, modules, practice problems, etc. – just not any HW code.
* You may discuss general difficulties you’re having with the HW6 program. See the forum for examples of this.

You may NOT:

* show any HW code to another person.
* look at anyone else’s HW code.
* copy any HW code from any source other than that provided by the instructors. You may re-use your own code, but make very sure that you developed that code entirely on your own.
* post any HW code to the forum or chat room – even a single line – even a line of code that you believe is incorrect or has an error.

As always, you have seven submission opportunities. So, start early!

As always, I will be available for general help on the forum, in office hours in Torgersen, on Dyknow, by email and by appointment (see the policy on how to get help in this course). Sam, our TA is available on Monday nights in the Chat Room. But for the take-home tests, we will not look at or trouble-shoot your code. You must develop your code on your own. So, start early!

**Reading assignment amplifications and clarifications**

This chapter may be the beginning of truly new concepts for many students so don’t be surprised if you spend more time attempting to grasp the reading material and spend more time doing the homework. However, once you grasp it you will probably wonder why it seemed so difficult because it is really very easy.

When you think of **modularity**, think of it in terms of standardized pieces for building complex systems. Any complex, man-made system can be de-composed into smaller and smaller pieces until you get to the smallest possible sub-division. You are simply breaking a large task into easily understood, and therefore easily programmable, subtasks. For example, constructing a building usually involves many subcontractors who are experts in their particular field, they do one thing extremely well and that’s all they are asked to do. Programmers should write large programs the same way. A particular example is shown below.

Problem – Find the cost of manufacturing a cylindrical tank with a conical top when given the total volume of the tank, the radius of the tank, the fact that the radius and height of the cone are the same, the cost-per-unit-area of the cone portion and the cost-per-unit-area of the cylindrical portion.

Initial modular break down:

1. Find surface area of cone
2. Find surface area of cylinder
3. Find cost of the cone
4. Find cost of the cylinder
5. Find total cost

Further investigation indicates that there is a need to know the volume of the cone and the volume of the cylinder so that the height of the cylinder can be computed, enabling the computation of the surface area of the cylinder. This leads to:

1. Find volume of the cone
2. Find volume of the cylinder
3. Find height of the cylinder
4. Find surface area of the cone
5. Find surface area of the cylinder
6. Find cost of the cylinder
7. Find cost of the cone
8. Find total cost.

In keeping with the concept of modularity, the programmer would write a function for each of these calculations.

Make sure that you understand the concept and use of the function prototype (page 194). The standard style rule in this course is that all functions are prototyped above the main() and defined below the main() with the exception of those very short functions that take only one line to completely define.

Remember, the more modular your code is, the easier it is to read, troubleshoot, and update.

**Complete and submit homework 6**

**PROBLEM STATEMENT:** A containing tank is to be constructed that will hold some given amount of oil when filled. The shape of the tank is to be a rectangular prism (including a base) surmounted by a rectangular pyramid. The pyramid's height is always equal to half the length of the prism. The cost to construct the prism portion of the tank is $300 per square meter, less a 10% discount for prism surface area in excess of 300 square meters and a 15% discount for prism surface area in excess of 500 square meters, while the cost for the pyramid portion is $400 per square meter, less a 20% discount for pyramid surface area in excess of 150 square meters and a 40% discount for pyramid surface area in excess of 300 square meters.

Write a program that, for a given tank volume and a series of prism heights, finds the optimal prism length and width (to the closest 0.1 meters) that minimizes cost for that prism height. The program should ask the user for the minimum prism height and the maximum prism height and the tank volume. Prism heights and lengths should be checked for every tenth of a meter. Display the results in table form (prism height, prism length, prism width, cost). Now that we’re optimizing two parameters, the final costs will be closer. Therefore, output the exact cost, NOT the rounded cost. Your program should work for prism lengths between 2.0 and 40.0 meters and for a tank volume between 100 and 5,000 cubic meters. You will need to format your output so that it prints with the given number of decimal places. Finally, determine the dimensions that produce the overall optimal cost.

You are required to (A) use a class data type to hold the values of the prism and pyramid construction costs, respectively; and (B) initialize these two data members within the definition of this class data type.

You are required to (C) define and use another class data type to hold the optimal length and width and their associated cost for a particular prism height and tank volume, and (D) initialize these data members within the definition of this class data type.

You are required to (E) define and use a user-defined function to find the optimal width and length and their associated cost for a given prism height and tank volume. You are not allowed to use input (cin) or output (cout) statements within this function. The function must return an object which holds the optimal values.

Do not use the *pow(x,y)* function to square a number. It is faster and more accurate to simply multiply this number by itself.

Once you have it compiling and generating the correct output as shown below, submit the source code (\*.cpp file) to the Grader.

**FORMATTING:** Declare all floating point variables as double precision floating point numbers. Define floating point constants with a period. For instance, use "**1.0**" instead of only "**1**". Make certain that there is one space on each side of the ">>>" prompts.

**SAMPLE HOMEWORK OUTPUT:** Your program will be tested with several input sets, including the one shown here in RED.

Enter the minimum prism height (decimal number, meters) >>> 11.0

Enter the maximum prism height (decimal number, meters) >>> 11.5

Enter the tank volume (integer, cubic meters) >>> 4000

Prism Height (m) Prism Length (m) Prism Width (m) Minimum Cost ($)

---------------------------------------------------------------------------

11.0 17.0 17.0 415156.35

11.1 17.0 16.9 415142.54

11.2 16.9 16.9 415135.85

11.3 16.8 16.9 415138.10

11.4 16.8 16.8 415146.58

11.5 16.7 16.8 415162.07

The minimum cost is $415135.85.

It is achieved with a prism height of 11.2, a length of 16.9 and a width of 16.9.

While you can certainly use some of the code written for HW3 and HW5 in this assignment, be aware that undiagnosed errors in that code may result in problems with this HW. If you get errors, be sure to trace through what is happening at each step of the program, including those steps that were written earlier.

Be especially careful to increment the loop properly, as was discussed in the .avi lesson.

50% of your grade will be based on whether you write the function required in E above. Although your initial grade from the Grader may return 100, when we check the code by hand the next week, it will be reduced by 50 points for not following these instructions.

While it is easier to write this program using simple variables, you are required to use class objects (compound variables). 50% of your grade will be based on whether you define the classes as specified in the instructions and instantiate, initialize and use class objects. Although your initial grade from the Grader may return 100, when we check the code by hand the next week, it will be reduced by 50 points for not following these instructions.

Note that you must also initialize the class objects by writing a constructor function in the class definition. If you use the class but not the constructor, you will lose 25 points on your homework grade. Although your initial grade from the Grader may return 100, when we check the code by hand the next week, it will be reduced by 25 points for not following these instructions.

**IT IS NOT SUFFICIENT TO GET WORKING CODE AND THE ANSWER**

Up to 15 points may be deducted for not using proper style. Again, be careful not to control the loops with a floating point counter. Besides the style issues described in previous modules, be sure to prototype your function before the main function and define it after.

Note that this is a graded assignment and should be completed on an individual basis. Again, refer to the course policy document for further details on how this restricts how you may work with others on this and other graded assignments.

Make certain that your program matches the sample output shown online with the assignment instructions. Perfect character-by-character matching, including the same case, is required for a passing grade. The course policy document explains why this is so important.

Make certain that you “sign” the program you submit online and include the honor statement on Line 2. Programs submitted without this statement will receive a 0 at the time the homework is reviewed by hand for style and completeness. Be sure to log in to the Grader and sign the program using all lower-case letters. Programs submitted without “signatures” will be automatically rejected by the grading system. The course policy document describes how you will need to go about “signing” your program file.

Use the Grader to submit your assignment. Make certain that you select the correct file to submit (it should be the source code, which is the file with the lowercase .cpp extension). Once it has been submitted, you should immediately see your grade along with any errors that were found.

You will need to resubmit your assignment if there are problems with it (see the course policy document for details).

**Think About It**

Maybe you’ll work for GM when you graduate. Your first job could be to design the oil tank for a new automobile. How would you EVER work through that many calculations if you didn’t know how to program?