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

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

**Module 3 — Algorithm Development, Selection Control Structures**

­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­**Summary of Tasks**

1. Read Sections 3.1 – 3.3 in the text ……………………..........................................pp. 93 – 111
2. Read “Introduction to Algorithmic Thinking”………………………....in Scholar’s Resources
3. Read “Reading Assignment Amplifications and Clarifications”…...provided in this document
4. Answer review questions 1 – 5, 9, 10, 11, 12, 13, 17, 18…….………………….pp. 133 - 135
5. Read “Initializing Compound Data Types (Classes)”……………...provided in this document
6. Watch:

“Default Constructors”

“HW3 Intro”

“const keyword and selection structure”

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

1. Contribute to the forum by answering some of the “Questions about Week 3 material” or starting a thread on anything you’re confused about.
2. **Complete and submit homework 3………………………………individual graded effort**

**Reading Assignment Amplifications and Clarifications**

This first section of your reading assignment is one of the major reasons that programming languages are taught to engineering students, even when some of them will not explicitly write programming code after graduation – it develops and refines algorithmic problem-solving skills. Actually, it is an example of an old proverb:

To learn – teach!

You are learning problem-solving by teaching the computer how to solve problems, and if you can teach the computer, which has no understanding except on and off, you can teach anyone! By the same token, those students who say, “I just cannot learn how to program a computer,” are really saying that they cannot explain a multi-step solution process to anyone, including themselves. That’s not something you want to put on your résumé! The additional document provided in Scholar’s Resources, *Introduction to Algorithmic Thinking*, by Dr. Wayne Brown of the Air Force Academy, makes it very clear that this is not some esoteric ability that is genetically passed from generation to generation. If you can follow a recipe, you can follow an algorithm. By extension, if you can explain a multi-step process to a child, you can develop your algorithmic processing skills, regardless of your intended major.

Note the two common tools for refining a problem solution, flowcharting and pseudocode. In my experience, flowcharting has typically been used to document working code in the past simply because flowcharts had to be done by hand in the past and that time was much better spent actually writing code. On the other hand, pseudocode was and is typically used to sketch out a process before committing it to code and, even though Figure 3.1 on page 95 of the text might lead you to believe otherwise, pseudocode has no set syntax – as long as it’s understandable, it’s fine. That is not to say that the pseudocode in the text is a poor model – it is excellent. However, now that we have a software application like Raptor, it makes sense to develop your algorithms with it before you commit to C++ code because you are literally accomplishing two things at once – flowcharting and writing code! [On a personal note, why is it that these kinds of things weren’t available when I was learning programming? Whine, Whine.]

The material in section 3.2 on conditional expressions and logical operators is very straight forward. You probably have some experience with Boolean algebra, and that helps. One thing you should know is De Morgan’s Theorem and how to apply it. You can find multiple presentations of this important theorem on the web.

One little note, the character for the “or” symbol on your keyboard is entered using the shifted backslash (\) key twice. The symbol shown on that key shows two vertical dashes in line with a space between them but note when I use it here (|) the space does not show.

Read section 3.3 very carefully, making sure that you understand all that is being presented, textually and in the figures. Note the example at the bottom of page 103. Make sure that you understand the syntactical shortcuts being used. If you don’t, review the material on pages 55 through 57. One thing in particular needs to be clarified. The C++ statements

i++; and ++i;

both are interpreted the same way,

i = i + 1;

because they are stand alone statements, not part of an expression. However, the C++ statements

y = i++; and y = ++i;

do not meant the same thing. The first can be replaced by the following two statements in the order shown:

y = i; i = i + 1;

while the second has that order reversed to:

i = i + 1; y = i;

We will not use the conditional operator shown on page 107.

You are expected to be able to use and understand the switch statement so make sure that you read that section closely.

**Initializing Compound Data Types (Classes)**

Recall the class definition given in module 2:

class Student

{

public:

string Firstname;

string Lastname;

int studentID;

string Major;

int level;

char gender;

double GPA;

};

It would be beneficial to be able to initialize these user-defined data types. However, it doesn’t work if we try to do that in a manner that might seem straightforward as shown below:

class Student

{

public:

string Firstname = “John”;

string Lastname = “Doe”;

int studentID = 123456789;

string Major = “??”;

int level = 99;

char gender = ‘?’;

double GPA = -1.0;

};

Or

class Student

{

public:

string Firstname(“John”);

string Lastname(“Doe”);

int studentID(123456789);

string Major(“??”);

int level(99);

char gender(‘?’);

double GPA(-1.0);

};

The reason it doesn’t work is that you cannot initialize something that does not exist. Remember, a class definition is nothing but a blueprint. Using that analogy – you cannot paint the walls of a house on the blueprint, you have to wait until the walls are in place. However, you can put construction notes on a blueprint to paint the walls white when the walls are up. You can do the same thing in C++, that is, provide construction notes telling what to do every time one of these objects is built. This special functionality is provided by writing a *constructor* function as shown on the next page.

class Student

{

public:

string Firstname;

string Lastname;

int studentID;

string Major;

int level;

char gender;

double GPA;

Student()

{

Firstname = “John”;

Lastname = “Doe”;

studentID = 123456789;

Major = “??”;

level = 99;

gender = ‘?’;

GPA = -1.0;

}

};

Now, every time you build or *instantiate* one of these Student objects, it will be initialized to the values shown above without the user having to do anything else.

**Complete and submit homework 3**

**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 prism’s height and length are each 10 meters. 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 calculates and displays the width of the tank for a given volume that is input, and also calculates and displays the total cost of constructing the tank. Your program should produce valid results for volumes entered in the range of 100 to 5000 cubic meters. Your program will not be tested with volumes outside that range.

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

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 ">>>" prompt.

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

Enter tank volume >>> 200

The total cost of the tank is $100569.

The width of the tank is 1.71429 meters.

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 a class and instantiate, initialize and use class objects. The class should store the $300 per square meter and $400 per square meter. Although your initial grade from the Grader may return 100, when we check the code by hand the next week, it can 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. Another style issue, besides those listed last week, is to be sure to put your curly braces on lines all by themselves; this makes it easier for you to check that you haven’t forgotten one.

To make sure your code works, be sure to test it with all kinds of inputs, not just those provided in the assignment. As an engineer, one of the most important things you're going to need to do is to anticipate any possible input to the system you design, and make sure your system is built to handle it.  It's the same with programming.  It's hard to think through every possibility, but it's essential that you do it and do it well.  I would encourage you all to not only share sample inputs and the output they produce, but to brainstorm what KINDS of inputs you need to consider and test.

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 both 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**

Trying to start these HW’s without learning the material first? You won’t know it, or you won’t know it well. Shortcuts aren’t worth it. In college, HW should no longer be the focus of your work; if it is, you’re missing the point (and the points). The points are earned on the tests, and the way you earn them certainly isn’t by studying the stuff for a few days beforehand. It’s by studying it each day and each week. Then you can use the HW’s as a check for whether you’re learning it well enough or not. Put your time and effort where it counts – into preparing for the tests the right way by learning as you go along. If you do that, the HW’s won’t be any problem -- might even be fun.