CoSci 440 SYLLABUS
Programming in C++

INSTRUCTOR **Mari Rettke** rettkem@lamission.edu 818.470.8419 - cell
SECTION : **3102**

Note: ***Should you become unable to complete the class you must withdraw from the class prior to the cutoff date. Students that are enrolled in a class after the last date to drop must receive a grade.***

Student Learning Outcomes:
Students will at the completion of the course be able to:

- Demonstrate an understanding of how C++ works by coding, debugging, and executing programs written in C++ using Object Oriented Programming techniques
- Demonstrate the use of Pointers, Dynamic Arrays, Overloaded Operators in executable programs.
- Apply programming logic by using selection, decision and repetition structures.
- Demonstrate an understanding of scope by referencing elements passed as parameters between modules in a project.

Assessment of student learning outcomes:
2 – EXEMPLARY **–** demonstrate sound understanding of program development by coding programs using C++ which are completely correct according to project specifications and will meet all criteria specified; adds specified optional program enhancements
1 **–** ACCEPTABLE – demonstrate sound understanding of program development by coding programs using C++ which are completely correct according to project specifications and will meet all criteria specified
0 – NOT ACCEPTABLE – does not meet the standards of acceptable above

DETERMINATION OF GRADE

1. **Attendance:**
   The class meets from 5:45 – 10:00 Monday. Attendance will affect your grade in a negative manner only. You may miss two class meetings without penalty. If you miss more than two meetings, 1 point will be subtracted from your final grade for each additional class missed. *Three cases of tardiness equal one absence.* Please call in 818.364.7742 or 818.470.8419 if you are going to miss class.

2. **Assignments, Exercises, and Quizzes:** (25%)
   Short programming exercises will be given each week. You are expected to complete these exercises during the lab time.

   There will be a chapter quiz on the material from the book and from the lecture every week. Tests will be written and will consist of questions, debugging exercises, and code fragments.

   Assignments are due at the beginning of the next lab period. If you cannot complete an assignment – please see me.

All assignments and tests MUST have a comment section at the top of the program. Include your name as programmer, the name of the assignment (Assignment 1 or Assignment 1 – page 56), and a brief description of what the program is supposed to do.
If you do not have a comment section – you will lose points.

All programs should include a sufficient number of comments to explain what the code does. If you run into trouble and need to make corrections – use a double slash (**//**) in front of the line and include a comment as to why this particular line of code did not work.

3. **Tests:** (35%)

Test #1	5%
Test #2	10%
Test #3	10%

4. **Final Exam**: (20%)

5. **Final Project**: (program) (30%)

6. **Grading scale** – A = 90-100, B = 80-89, C = 70-79, D= 60-69, F – below 60

MATERIALS:

Text – C++ Primer Plus – 6th edition – Steven Prata – SAMS – ISBN 0-321-776-402
(If you have difficulty obtaining a book – please see me after the first or second class.)
Two flash drives

Office Hours:

Tue/Thur -1:00 to 2:30pm Mon/Tue/Thur – 4:00 – 5:30 Or call for appointment 470.8419

Additional information:

Students in this class are expected to have some prior experience in programming. Programming classes generally require 5 to 10 hours of outside work per week. This number varies according to the amount of experience that you have with programming.
If you do not have any experience in programming you will need to allow a significant of extra time to complete your assignments.

During the first five weeks of the class we will cover

All assignments and tests will be posted on the LAMC web site for this class. I will provide a paper copy when the assignment /test is given. If you lose your copy – please refer to the web site.

SPECIAL INSTRUCTIONS

Do not fall behind.

In this class we will spend the first few weeks reviewing the basics of C++ programming. We will cover one chapter per week, sometimes more than one chapter.

Problems and frustrations:

In the first five weeks of the semester 95 percent of your errors will be syntax errors. Syntax errors fall into the following categories:

- something is not spelled correctly (C++ is case sensitive)
- the punctuation is wrong
- something is missing
- the case is wrong – (upper case instead of lower)

The error is not in the book. It is on the screen. However, staring at the screen for hours **will not** help find the error.
Develop a process. Methodically check each of the possibilities. If you still cannot find the error, ASK me or ask someone else in the lab.
Our eyes play tricks on us and we cannot see what is really there until we become accustomed to the form of the language.

During the first five weeks, do not spend more than 15 minutes trying to find the error before you ask for help.

Outside help and Group projects

If you work in a group, you must have some differences in code. For example, variable names and comments. Projects turned in that are exactly the same as someone else's will share a grade unless each member of the group can verbally explain how the program works.

I am familiar with ALL the code used to solve the course assignments that is available on the internet. **Do not turn in code that is straight from the internet**. Use the internet as a resource, but do not copy code.

****

Tests (aka Projects/Programs) will be evaluated according to a variety of criteria:

40% - Does it work? (each test will have a standards sheet)
20%- Adherence to standards (see attached standards sheet)
20% - Structure (see attached standards sheet)
20% - Quality of code (see attached standards sheet)

ALL projects/programs must begin with a comment section which will include the following:

- Your name as programmer
- The name of the class and the due date of the assignment
- A description of the program and what it is supposed to do (problem analysis)

You **must use comments** throughout your program. Comments may be single line or a block.
Single line comments begin with a double forward slash (**//** **)**
Block comments may either begin with a double forward slash for each line or a forward slash followed by an asterisk at the beginning of the block **(/*)** and an asterisk followed by a forward slash at the end **(*/).**

If you do not comment your code, then you will lose points.