

CoSci 439 SYLLABUS

INSTRUCTOR **Mari Rettke** rettkem@lamission.edu 818.470.8419 - cell
SECTION : **0187**

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.
- Demonstrate understanding and use functions and procedures.
- Demonstrate understanding and use of variables.
- 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 10:40 to 12:30 Tuesday and Thursday. 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 if you are going to miss class.

2. Assignments, Exercises, and Quizzes: (15%)

Short programming assignments will be given each week. These assignments will count as 15% of the final grade.

Quizzes will be written and will consist of True/False questions, debugging exercises, and code fragments

3. Tests: (25%)

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

4. Final Exam: (25%)

5. Final Project: (program) (35%)

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

MATERIALS:

Text – C how to program – 7th edition – Deitel/Deitel; Pearson/Prentice Hall 2010

Two flash drives

Office Hours: Tue/Thur -1 to 2pm and Mon/Tue/Thur – 4:30-5:30

Or call for appointment 470.8419

SPECIAL INSTRUCTIONS

Do not fall behind.

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
- 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 someone in the lab to help you.

Our eyes play tricks on us and we cannot see what is really there until we become accustomed to the form of the language.

Do not spend more than 45 minutes to an hour trying to find an error. During the first five weeks, do not spend more than 15 minutes trying to find the error before you ask for help.

In class or in the lab, if you do not understand something, ASK me or one of the tutors. We are building a very detailed and specific basis for this language and many others. (C++, C#, Java, scripting languages, etc.). You are NOT expected to know these things. This class is designed to teach you this material in a way that you understand it.

If you are shy about asking questions in class, I would like to say “get over it”, no one is going to make fun of you or think less of you. Your classmates will be grateful that you could articulate the question for them. If you truly are one of those people who become paralyzed at the thought of speaking out loud in public, then WRITE the question down and give it to me at the break or after class. It can be one word.

Example: scanf ??????
Function ??????
Define ??????

Or, of course, the question could be longer:

Example: Why do we have to use printf, why can't we use putc?

If I have already spent three days of class time on that very same question, then I will make an appointment with you to help you with the problem.

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.

Quizzes – these are all written – vocabulary, and debugging exercises **from the book**

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

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

Grading Tests

This class is taught as a beginning programming class. The projects emphasize various structures and methods of problem solving.

ALWAYS turn in the projects on time. A project that does not work or is incomplete can still earn a significant number of points.

Tests (projects) will be graded as follows:

- Adherence to coding standard presented in class: 20%
- Structure of the program: 20%
- Comments: 20%
- Naming of variables and constants: 20%
- Correct output: 10%

As you see, a program that is completely correct can earn 90%. To earn additional points you must do something additional. Each test will include a list of items that can be added to earn extra points.