ESC 151 –C Programming (ANSI C)

Class:
Section 1: Remote, quizzes Monday at 9:30
Section 2: TTH 10:00 AM-11:15 AM

Suggested External Information (in place of a book): http://www.cprogramming.com/tutorial/c-tutorial.html

Instructor: Robert Fiske
Office: FH 315,
Office Hours/Availability:

Officially: TTH 6:00-8:00
in general I'm available at any time except TTH: 9:00-12:30. Send me an email and if need be we can meet via zoom, if I don't reply in a timely fashion send another email and/or post on the class discussion board.

Note on medical reasons for extensions/other requirements

Educational access is the provision of classroom accommodations, auxiliary aids and services
to ensure equal educational opportunities for all students regardless of their disability. Any student who feels he or she may need an accommodation based on the impact of a disability should contact the Office of Disability Services at (216)687-2015. The Office is located in MC 147. Accommodations need to be requested in advance and will not be granted retroactively.

Any medical/family issues you have need to be brought to my attention before you take any quiz/exam, once the exam is in your hands no makeup will be offered. Once you begin to take an exam/quiz you may not leave the room until you turn in the quiz/exam.

Course Objectives: This course is designed to:
1. Introduce the C programming language.
2. Introduce the concepts of computer programming.
3. Teach students to think about how to approach programming challenges.
4. Teach students to debug computer programs.

Expected Outcomes: Upon completion of this course, students should be able to:
1. Identify, formulate, and solve engineering problems using C language
2. Use techniques, skills, and tools that the C language offers for modern engineering practice

Fulfills the following Engineering Program Objectives and Outcomes:

1. Knowledge/Understanding of Application of Mathematics, Science, and Engineering Principles
2. Knowledge/Understanding of Identification, Formulation, and Solution of Engineering Problems
3. Knowledge/Understanding of Techniques, Skills, and Tools for Modern Engineering Practice

Grading

Quizzes 15%
Programming Projects 15%
Class Participation/Attendance 10%
Weekly Programming tasks 30%
Final Exam: 30%

Section 1 (Remote) will be graded on participation, details will be given in class, and will involve posting to the class discussion board.

Section 2 (face to face) will be graded on attendance. Attendance points are earned for showing up to class or notifying me (with an appropriate excuse*) of the need to miss class before the start of class. If you are kicked out of class for not following safety protocols you will not get attendance credit for that class period, in addition to any other potential penalties.

*Appropriate excuses include medical issues, family emergencies and so on. In addition things such as oversleep can be accepted if they happen rarely and will be allowed at my discretion. Basically if you usually show up to class an miss a day on occasion you won't be penalized, if you repeatedly miss class you will.

Grading Scale:

A	100-94
A-	93-90
B+	89-87
B	86-83
B-	82-80
C+	79-75
C	74-70
D	69-60
F	0%-59%

Programming assignments are to be submitted via blackboard before you leave class.

THIS COURSE IS HANDS ON PRACTICE INTENSIVE AND ADDITIONAL PROGRAMING PRACTICE BEYOND CLASS PERIOD IS HIGHLY RECOMMENDED TO SUCCEED IN THIS COURSE.

Important Notes:

• For a successful completion of the course, students are strongly advised to attend all lectures and computer lab sessions, and to take notes because they are a fundamental instructional part in addition to the textbook.
• If a student misses a class, he/she will be responsible for the missed material. To succeed you have to work consistently; please do not take the attitude that this week I will ignore and catch-up during the next week. For this course that is a sure recipe for failure.

Schedule of topics (subject to change):

Week 1: Environment, variables, math
Week 2: Input, Output
Week 3: Switch
Week 4: If, else if, else
Week 5: While/Do While loops
Week 6: For Loops
Week 7: Arrays
Week 8: Strings (may start files this week)
Week 9: Files
Week 10 Functions
Week 11: Points
Week 12: Pointers Continued
Week 13: Structures
Week 14: Libraries (may swap this out)
Week 15: Review, additional topics (any new will not be on the final)

Other recommended sites (comments about if I actually use this site, or just found it after a quick search):

https://stackoverflow.com/ (I use this site often, very helpful)

https://www.cheatography.com/ashlyn-black/cheat-sheets/c-reference/ (found this in a quick searc)

http://www.cprogramming.com/tutorial/c-tutorial.html (found this on google a few years ago, haven't used it extensively, but have checked it out somewhat in depth over time)

https://www.tutorialspoint.com/compile_c_online.php (have used this a few times, it provides an online Linux compilation environment, not as useful as having your own system to work with, but can do most of what you need in this class)

https://en.wikibooks.org/wiki/Algorithm_Implementation/ (I've used this for sample class code a few times, as well as a refresher for things I don't recall how they worked exactly off the top of my head

If you use Visual studio for homework follow these steps to set up your project.

Instructions for Microsoft Visual Studio 2015 C++ Compiler

To create a C Project in Visual Studio 2015:

Open Visual Studio
Select File->New->Project
Select Other Languages->Visual C++->Empty Project
Give the project a name and a place to save the project
Right click “Source Files”
Select Add->New Item...
Select “C++ File (.cpp)”
Give the file a name, but rename the extension to .c instead of .cpp
Select Project->_Project_Name_ Properties.
Select Configuration Properties->C/C++->All Options
Change the dropdown on “Compile As” to say “Compile as C Code (/TC)”
You now have a C project.
If you are using certain functions (such as the base scanf function) you may need to tell visual studio to allow that function.
To do this go back to the project properties.
Select Configuration Properties->C/C++->Preproccessor
Select the text field next to “Preprocessor Definitions”
Add “_CRT_SECURE_NO_WARNINGS;” (minus the quotation marks) to the beginning of that line.

For assignments you are allowed to talk to one another (in fact I encourage you to share ideas for how to do the assignment), however you must write your own code, if you copy and paste off of each other this is considered cheating.

If you use code you find online you must cite the source of where you found the code, if the code isn't cited it is considered cheating.

Per CSU policy cheating on a quiz/assignment is a minor infraction.

Two minor Infractions are considered a major infraction.

Minor infractions result in a 0 on the assignment.
Major infractions result in failure of the course.

Health and Safety syllabus statement

Approved by Faculty Senate on 07/15/2020

The COVID-19 pandemic is still present and serious. Before entering class, you should have completed your daily health assessment. While you are in class on campus, you are required to: sit in your designated seat, maintain physical distance**,** wear your facial covering (e.g., masks or face shields), always cough or sneeze into your elbow or tissue, use the materials provided to clean your desk and chair before and after use, and adhere to other public safety protocols and directives for your specific classroom/lab/studio. Students who do not follow these health and safety requirements will be instructed to leave class immediately. Students who violate this protocol will need to leave the classroom and MAY be marked absent. Repeated violations of these health-saving protocols may lead to sanctions under the Student Code of Conduct **(**3344-83-04 [E] and [Z]) up to and including suspension or expulsion. The CSU community thanks you for your cooperation!

The above notice was copied directly from CSU's announcement

Required skills:
If this is your first programming class the most important skill you need to have is problem solving. In class I will be going over what code looks like and how the various features work, it will be up to you to spend time practicing writing code. Get used to using the debugger, the debugger is the most useful tool you have at your disposal.

When I took ESC 151 as an undergrad the vast majority of what I learned came from experimenting at home with the debugger, the lectures introduced the concepts to me, but I did not truly understand the material until I spent time working with code at home.