

ENGR170/SIE175

Computer Programming for Engineering Applications

University of Arizona

Course Objective:

1. Introduce undergraduate students to development and implementation of engineering programs using C programming language,
2. Introduce the students to the Unix operating System,
3. Introduce students to introductory concepts and techniques in engineering of software.
4. Prepare students for understanding hardware aspects including memory and its usage, and more advanced software concepts such as dynamic data structures.

Currently Used Text Book:

- Problem Solving and Program Design in C, by Hanley et. al, Addison Wesley, (latest edition).
- Just Enough UNIX, by Paul K. Andersen, Mc Graw Hill, (latest edition).

Laboratory and class computer account:

The course has scheduled Lab sessions in addition to lectures. Lab sessions will cover materials that supplement class lectures. You may address your questions to TA in lab or office hours and additionally you can send your inquiries to enr170@ece.engr.arizona.edu.

Tentative Exam Date:

Exam: The week prior the last week of classes

There is a class account on ECE network through which we will grade and return your assignments. This account is also used to answer your questions about the class and assignments. The account is enr@ece.arizona.edu. If you use other development

platforms, you need to make sure your programs can run correctly on the ece network and compiler used for this class.

Electronic submission of assignments:

In ENG170/SIE170 you will be submitting your programming assignments electronically to the class account. We will grade and return your graded assignments and scores electronically. For each of the assignments 1-5 and for your final project you will execute the "turn in" command to submit your program files. It is the responsibility of the student to turn in the correct file(s) and to submit the work on time by executing the turn in command correctly. The clock for timing purposes is the clock on the ECE computer. You can execute the turnin command any number of times before the final deadline for each assignment. Each new turnin will overwrite your previously turned in materials. **NO LATE ASSIGNMENTS WILL BE ACCEPTED.**

Course Outline:

To achieve the course's objectives three interrelated aspects will be studied:

- A programming environment: UNIX operating system.
- A programming language: the C programming language.
- Basic software concepts, techniques, notations, and coding guidelines.

The course will include a number of programming assignments with larger assignments towards the end of the semester. Each assignment focuses on a subset of the studied topics given below:

Introduction to UNIX Operating System

- a) Overview of computer organization
- b) Role of the operating system
- c) Basics of UNIX
- d) Commands and command interpreter
- e) The file system
 - i. Directories and filenames
 - ii. Permissions
 - iii. The directory hierarchy
 - iv. Devices
- f) C shell
- g) Processes, Foreground and background jobs, and history

- h) Input-Output redirection, Pipes

C Programming Language

- a) Data types, Elements of a C program. Statements and expressions;
- b) Examples of commonly used functions, assignment, output formatting;
- c) Simple data types; constants and variables; precedence rules; conversion between different data types.
- d) Conditions; relational operators; logical operators; if, if-else, and switch statements.
- e) Repetition and loops; while statement; for statement; increment and decrement operators; loop termination; nested loops; do-while statement; testing and debugging loops.
- f) Top-down design approach; modular programming; user functions; library functions; function declaration and definition; interaction between functions; scope rules; programs with multiple functions.
- g) Arrays; declaration and initialization of arrays; multi-dimensional arrays; searching and sorting arrays, passing arrays to functions.
- h) Pointers and memory addresses; pointers and arrays; pointers and function arguments; pointer arithmetic.
- i) Strings; string library functions; substrings; concatenation; strings vs. characters.
- j) File input and file output.
- k) Recursion; recursive functions with array parameters.
- l) Structures; arrays of structures; assigning values to structure variables; pointers to structures; Dynamic data structures; Linked lists; Trees; application to databases.
- m) Engineering applications, matrix manipulations, numerical integration and differentiation, numerical solution of equations, representing and computing with polynomials.

Software Design

- a) Design and coding guidelines
- b) Fundamental design concepts
- c) Documentation guidelines
- d) Overall system design

ABET Credits:

Engineering Science: 1.0

Engineering Design: 2.0

Office: [ECE 320G](#)

Grading:

There will be 5-6 assignments, one project and one exam. The break down will be as follows:

Assignments and Labs:	45 %
Project:	15 %
Exam:	40%
<hr/>	
TOTAL	100%

ACADEMIC INTEGRITY:

All the work that you submit in this course must be your own. Group efforts will be considered academic dishonesty (see [the *University of Arizona Code of Academic Integrity*](#)). Any marked similarity in form, notation, or content between two submissions will be regarded as evidence of academic dishonesty. **ACADEMIC DISHONESTY IS A SERIOUS OFFENSE THAT MAY RESULT IN SUSPENSION OR EXPULSION FROM THE UNIVERSITY.** All assignments will be checked.

You may discuss any approaches how to do the assignments with other students, but you may not consult or share written work.