TCSS 333 C for System Programming Master Syllabus
Version: August 2013
(Approved: Oct 2013)

Catalog Description

Introduces C as a language for exploring low-level machine characteristics and interacting with operating system services. Includes bit models for numeric data, pointers, arrays and structures, memory allocation, development of multiple file programs, libraries, system calls, tools for compiling and linking. Prerequisite: a minimum grade of 2.0 in TCSS 143

Preconditions

- Use loops and selection structures to implement a simple algorithm.
- Correctly choose suitable data types (int, double, String) to solve a problem.
- Write and make use of a set of functions, including parameters and return values.
- Develop, compile, run, and debug solutions to simple problems.

Student Learning Goals (to be added to syllabus handed out to students)

- Distinguish various intermediate forms of a C program including preprocessor output, assembly code, object code, and executable.
- Specify the memory layout of C's basic types, numeric, string, array, and structure.
- Allocate memory and use pointers to manipulate dynamic data structures and parameters.
- Use system routines to start, stop, and program communication between processes.
- Organize a complex program as a set of source and header files and use a tool such as make to build it.

CSS Degree Student Learning Outcomes that this course contributes to (to be added to syllabus handed out to students)

a. an ability to apply knowledge of computing and mathematics appropriate to the discipline;

b. an ability to analyze a problem, identify and define the computing requirements appropriate to its solution;

c. an ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs

UWT Student Learning Goals that this course contributes to (to be added to syllabus handed out to students)

Inquiry and Critical Thinking
Students will acquire skills and familiarity with modes of inquiry and examination from diverse

disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

Topics covered

- Bit representations for numeric types, including two's complement and floating point format
- Arrays and strings
- Pointers and structures
- Input / Output: Streams, Pipes, Files, Devices
- Program organization using source and header files
- System calls
- Libraries
- Use of scripting languages (if time permits)
- Common Unix shell commands

Additional Information

A likely textbook would be:

> *System Programming with C and Unix,* by Adam Hoover Addison Wesley 2010

The instructor may wish to explore the use of C in implementing some of the object oriented features of C++.