Course number and name: **CS 01205: Computer Laboratory Techniques**

Credits and contact hours: 3 credits / 3 contact hours

Faculty Coordinator: Professor Darren Provine

Text book, title, author, and year: Unix in a Nutshell, by Arnold Robbins, 2005, et al; and The C Programming Language, by Kernighan & Richie, 1989

Specific Course Information

Catalog description: A practical introduction to the hardware, software and networks used by the Computer Science Department. A foundation in programming using the language or languages required for intermediate and advanced computer science courses will be included.

Prerequisites: CS 04113 Introduction To Object Oriented Programming or
CS 04103 Computer Science And Programming; and
Sophomore Standing

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Specific Goals for Course

1. Student will demonstrate ability to manage UNIX/BSD/Linux files and directories; creating, moving, renaming, and deleting files/directories., setting access permissions on files/directories, and using tools such as find to locate files in a directory hierarchy.

2. Students will demonstrate ability to manage multiple revisions of files using tools such as RRCS/CVS or other revision management software.

3. Students will demonstrate understanding of C pointers and arrays.

4. Students will demonstrate understanding of C bitwise operations.

5. Students will demonstrate ability to manage compilation of C programs broken across multiple files, including builds software such as Make.

6. Students will demonstrate ability to use symbolic debugger programs such as gdb, examining data inside a running program and tracing its execution.

7. Students will demonstrate ability to use a Unix shell, such a bash or csh, interactively to manage day-to-day tasks of software development, and creating shell scripts to solve simple administrative or file-management tasks.

8. Students will demonstrate ability to use standard Unix filters, such as grep, sed, tr, sort, head, and tail, to search inside files and to modify program output when needed.

Required list of topics to be covered:

1. Implementation of basic security concepts including permissions, bounds checking, input validation, type checking and parameter validation

2. Regular expressions as used in (C/Linux); Standard Unix text filters, such as grep, sed, tr, and sort.

3. Data structures and algorithms in C.

4. Basic Boolean logic/operations in C.

5. Linux BASH Scripting

6. Programming constructs and concepts in C, including:
   a. Variables and types
   b. Strings, arrays, structures
   c. Sequential and parallel execution
   d. Assignments
   e. Decisions and branching
   f. Loops
   g. Functions, procedures and calls
   h. Debugging techniques
   i. Arrays, pointers, and memory access
   j. File access

7. Model/View/Controller architecture, including callback functions activated by outside events.

8. Appropriate and secure use of library functions

9. Defensive programming techniques

10. Command line interfaces