CSCI 110 LAB OUTLINE

FUNDAMENTALS OF COMPUTER SCIENCE

TEXT: Experiments in Computer Science, C version 4th Edition
Jung & Brookshear Addison/Wesley

Approved: **Effective: Fall 2007**

MATERIAL TO BE COVERED	SESSION	TIME LINE
Introduction: general hardware notions and computer and memory organization processor capabilities. Gaining access to the machine, your local resources. Your LAN account on MS Windows 2000 and accounts on UNIX subnet. Using the local file system and server resources. Using text editors (IDE, MS or Borland, notepad, wordpad, VI, emacs). Data representation, algorithms and problem solving. Practice problems for binary and hexadecimal number systems, conversionbetween bases, representation of negatives using two's complement, binary arithmetic.	1	2.5 Hours
Operating systems, gaining access to remote resources on server, using the global NIS student accounts on UNIX, using telnet, ftp and rlogin, a first look into computer networking and networking protocols (character mode). Getting started with programming, programming fundamentals, understanding C. Writing your first C program, using the preprocessor, linker, loader, building your code. Understanding the first error messages and debugging. Using variables, assignments and repetition.	2	1.825 Hours
Understanding the primitive data types: int, float, double, char and representing floating point (IEEE754). A first look at pointers, addressing and dereferencing. Understanding computer arithmetic. Input and output with scanf(), getc(), gets(), printf(). Understanding strings and string functions. First examples with arrays and structures.	3	6.125 Hours
Understanding selection (simple if, if else, nested ifs, switch) and repetition (for, while, do while and nested loops). Looking at two important applications: binary search, selection (and bubble) sort.	5	2.5 Hours
Understanding pointers. Functions, the building blocks of C language. Understanding and contrasting call by value, call by reference. How to pass arrays as arguments to functions. Global versus local variables and static variables. Conditional compilation. Understanding the principles of software engineering applied to software design and structured programming. How to use modularity to enhance debugging. Creating software documentation. A systematic approach to software testing.	7-Jun	3.75 Hours
Introduction to recursion and examples. Looking at the operating system support to recursion: buffer frames or activation records. A first example of using command line arguments, the GCD program. Understanding the scope of a variable. Recursive binary search and quicksort.	8	1.825 Hours

CSCI 110 LAB OUTLINE

FUNDAMENTALS OF COMPUTER SCIENCE

TEXT: Experiments in Computer Science, C version 4th Edition
Jung & Brookshear Addison/Wesley

Approved: **Effective: Fall 2007**

MATERIAL TO BE COVERED	SESSION	TIME LINE
Managing dynamic memory allocation with: malloc(), calloc(), realloc() and eallocation with free(). How to create ragged arrays in C using arrays of pointers. Applications to dynamic data structures: linked lists and binary trees (optional).	12-Oct	1.825 Hours
Creating and working with files. Comparing sequentail files and random access files. Using files for the merge sort.	13	2 Hours

*** One hour = One hours of face time. ****This outline allows for 3 hours of review, testing and project presentation.
16 Week Term: 1 week = 3.75 hours (face time) 6 Week Term: 1 week = 10 Hours (face time)

NOTES:

There are 5 lab handouts that will direct student work through the 10 chapters we cover from the lab book (except sessions 4, 14, 15 and 16 from the lab book that will NOT be covered). The lab handouts require that you show your work, will be collected and graded. There are also 4 projects that will be graded. One project will require that you write a report and make a class presentation. The lab and projects grade will represent up to 35% of the course grade. A few questions on the comprehensive Final Exam will be on specific problems form the last two projects.

Submitted by: Pop