Preface

This document provides a brief description on the objectives of each of the Labs included

Dr. Cho,
The following concepts and skills were covered in lab assignments (CS278 “C” Burris Spring 2010) unless otherwise noted. If more detailed information is required, please let me know.

David Burris

Labs: “Screen Shots” are not considered professional and not allowed.

Labs

Lab 1:

A) Basic payroll application including calculation of gross pay for hourly employees with FICA calculation.
B) Required creating a script and telnet/ftp to an external Linux server. Use of environment variables, chmod, path, formatting output.
C) Quantitative/Qualitative Thought problems (examples):

State a position and explain why it is the best choice in light of all three management positions stated below. Consider the situation where the FICA rate and limit change frequently as well as the situation where these numbers are not subject to frequent change.

Group 1: Many employers allow employees to place numeric values for the FICA rate and FICA limit directly in their code (hard code). Is this a good or bad practice? Consider the effects of program enhancement (the tax law changes and these values must be changed, not the method for performing the calculations. Assume the state enacts a tax law using the same rate and limit as the federal government. Two years latter one government entity changes one or more values and the other government entity does not. What are the potential dangers encountered during program modification?

Group 2: Using the discussion of group one, group 2 believes company policy (SOP – standard operating policy) should be to never allow the use of numeric literals in the code. They believe the FICA rate and FICA limit should be defined as “constants” and only the constant referenced within the code. Any programmer using numeric literal should be given an opportunity to travel and meet new people at their own expense. When a tax law changes, the maintenance programmer should modify the source code and re-build the project. Each constant should only be defined one time in the project.

Group 3: The policy position espoused by group 1 and group 2 only allow programmers to specify the numeric values for the required computations. Group 3 supporters believe all values used in the computations should be placed in a file the user can edit. When the program starts, it reads the file to obtain values such as the FICA rate and FICA limit. Hence the program will not have to be modified, rebuilt, and checked for potential errors introduced to the code during the modification process. Explain the pros and cons of this approach.

Lab 2:

Payroll problem requiring all input and output using files. Multiple control breaks were required including department totals, page breaks, report totals, report headers, and report footers. Flow charting was used as a design tool. Program modules had to be in separate directories with a script used to complete the translation process for a "C." In addition to the script, to receive a "B" or above, they had to use" make" the utility.

Lab 3:

The primary purpose of this lab was the creation and use of libraries. Each grading option required completion of all preceding grading options. The library modules involved numerical analysis concepts with respect to infinite series to determine results provided the user. The "C" option concentrated on static libraries. The "B" option highlighted shared libraries. The "A" option required implementation of a dynamic library. This lab provided an opportunity to cover many concepts associated with multi-user systems and packaging/shipping final products.

Lab 4:

The "D" option goal was to manipulate subscripts using an array of structs. The "C" option required them to sort the material a second time modifying the "D" option algorithm to use a pointer sort with traditional subscript notation. The "B" option required the same sorting algorithm using relative addressing in addition to the "C" option solution. The "B" option required the sort functions to be placed in either a static, shared, or dynamic library. The "A" option required the use of a "template" to implement at least one of the sort routines. All options "D" through "B" had to be completed as part of the "A" option.

Lab 5:

Recursion was emphasized to implement functions for multiplication, division, greatest common divisor, exponentiation, etceteras for use with a limited architecture computer (only addition implemented in the hardware). Again, projects involving multiple files and directories were emphasized. Scripts and the make utility were required for the translation process to create libraries.

Lab 6:

Inter-process communication was emphasized using anonymous pipes, named pipes, and sockets. Race conditions and deadlock were introduced.

Object Orient Programming including single/multiple inheritance and polymorphism were covered. Additional topics included inline code, re-entrant code, public, protected, and private were discussed. There was insufficient time to assign labs. All concepts included extensive coded examples of implementation.

The 8086 interrupt architecture was covered in detail with code to manipulate the system clock.