

CS 330 Computer Organization

&

Machine-Oriented Programming

Fall semester 2005

class meets: 5:30 - 6:45PM Mondays and Wednesdays
15th Street Classroom Building room 102

Instructor: Dr. Robert Hyatt (hyatt@cis.uab.edu)
Room 136a Campbell Hall 934-2213

Textbooks: Computer Organization & Architecture, 6ed
William Stallings

80x86 Assembly Language and Computer
Architecture Richard C. Detmer

Office Hours: Monday-Wednesday 3:30pm-5:30pm

Class Attendance: Class attendance is your option, but if you are going to attend class, you must be present when the class begins. Arriving late disrupts the class.

The last day to withdraw with a "W" grade is October 19.

There will be three in-class examinations that will count equally. Program assignments will count 20% of your final grade. The third exam (final exam) will be given Wednesday, December 14, 4:15-6:45.

Grades "A" : $\geq 90\%$ total credit "B" : $\geq 80\%$ total credit
"C" : $\geq 70\%$ total credit "D" : $\geq 60\%$ total credit
"F" : less than 60% total credit

make-up exams: Makeup exams will only be given with advance notice. Exams will be announced at least one week in advance of the date they are given. If you have to be out of town, or are sick, you must notify me or the departmental secretary prior to the time an exam is given, otherwise you will be given a grade of zero.

Cheating: Cheating is not tolerated in this course. If you are involved in cheating, either during the exams or with lab assignments, you will receive a grade of "F". Note that cheating includes copying some- else's work or helping someone else. For more information, see the UAB web site under "academic misconduct."

This course is going to use two books. The primary book will be the book by Stallings, but we are going to interleave parts of the Detmer book into the course throughout the semester, to introduce assembly language programming.

Programming projects: The following programming assignments must be completed and turned in by the due date given in class. Each assignment will have a separate due date, if the assignment is turned in late, the grade for that assignment will be penalized.

Assignment 1.

You are to read in two integer numbers, A and B. You are to use those two integers to compute the following expression: $\text{answer} = A * B - (A + B) / (A - B)$. You should produce an error message should A-B be zero, as that will cause the division to fail.

Assignment 2.

You are to read in N integer values and compute the average. You are to display how many numbers were read and their average. The numbers are to be 32 bit integers, positive or negative. A value of zero (0) indicates there is no more data, and the zero is not counted or used in the computation.

Assignment 3.

You are to read in N integers, terminated by a value of zero, which will not be used. You are to sort these numbers into ascending order and print them out. You must produce an error message if either no data is entered prior to a value of zero, or if too much data is entered and would overflow your array. You should plan on handling at least 100 integer values to properly size your array.

Assignment 4.

You are to read in a 32 bit integer. Numbering the bits from 0 (LSB, or rightmost bit) to 31 (MSB, or leftmost bit) you are to compute three results for this number: 1. The number of the least significant 1 bit set; 2. The number of the most significant 1 bit set; 3. The total number of 1 bits set.

For example, for the constant 00ffff0h, the LSB set is 4, the MSB set is 23, and there are 20 bits set total.

Assignment 5.

For the equation $AX^2 + BX + C = 0$, use the quadratic formula to compute the value of X, given A, B, and C. You must use floating point math, and you have to use the integer output macro DTOA to correctly display the floating point results including the fractional part. Note that all calculations must be done using the floating point instructions, even though you are initially entering A, B and C as plain positive/negative integer values.

Assignment 6.

Read in a string containing up to 256 characters. Determine if the character string is a palindrome. That is, the string reads exactly the same backward and forward. For example, ABBA, and ABCBA are both palindromes, while ABAB is not. You should print a message indicating whether the string is a palindrome or not.