CSC 221

THIS IS A HYBRID COURSE.
1/3 of the coursework is delivered through traditional lecture and 2/3 is delivered on-line using Blackboard and "asi.csudh.edu/Computer_Science/CSC221/" webserver.
All exams will be taken in class during regular meeting time.
Prerequisites: CSC 121


References Guidance on Secure Software/ Federal

(1) http://www.us-cert.gov/swa
(2) Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
And Software Security Testing
See https://buildsecurityin.us-cert.gov/swa/downloads/software_security_testing.pdf

(3) http://www.cert.org/work/software_assurance.html
(4) http://www.cert.org/secure-coding/
Course goals and objectives: The goal of the course is to provide introduction to the concepts of assembly language programming and machine organization specific to Intel processor architecture. The focus is on the principles that should underline the development of computer assembly language programs, and on relevant implementation techniques. The specific topics covered include: hardware and software architecture, instructions groups, addressing, procedures and macros, low level I/O and interrupts. The students will be given programming projects to cover these areas. The objective of the course is to provide the knowledge needed to efficiently utilize computer hardware, and to give background for compiler constructions and embedded systems design. Guidance on Secure Coding concepts and practices is provided.

Learning outcomes: Successful students will learn how to write, compile, and execute programs (including macros and subroutines) in Microsoft Assembly Language (MASM) for Intel type computers. and a basic understanding of secure coding best practices.

Requirements: There will be two midterm exams and an in-class final exam. You will be required to submit 4 programming projects.

Grading:

The weights of the various assignments are given below:

- 2 Midterm exams 50%
- Final exam 30%
- Programming projects 20%

Completion of all exams and all projects is required to pass the course.

The following grading table will be used:

<table>
<thead>
<tr>
<th>Score</th>
<th>94-100</th>
<th>91-93</th>
<th>88-90</th>
<th>84-87</th>
<th>81-83</th>
<th>78-80</th>
<th>74-77</th>
<th>71-73</th>
<th>68-70</th>
<th>64-67</th>
<th>61-63</th>
<th>0-60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade</td>
<td>A</td>
<td>A-</td>
<td>B+</td>
<td>B</td>
<td>B-</td>
<td>C+</td>
<td>C</td>
<td>C-</td>
<td>D+</td>
<td>D</td>
<td>D-</td>
<td>F</td>
</tr>
</tbody>
</table>

Academic Integrity: Academic integrity is of central importance in this and every other course at CSUDH. You are obliged to consult appropriate sections of University Catalog and obey all rules and regulations imposed by the university relevant to its lawful missions, processes, and functions.
**Programming Assignments:** Unless stated otherwise, all projects are individual assignments and are expected to be the student's own work. Projects must be handed in by "class's meeting time" on their assigned due date. Projects must be handed in the form of "printout of the source code + printout of the screenshot of program execution + the file containing the source code (e-mailed as attachment or delivered on USB drive)". Late projects will be penalized 5% per 24 hour period of lateness, up to 50%. No projects will be accepted 10 days after the due date.

**All work turned in is to be typed.**

**Attendance and drop policy:** Students are expected to attend lectures, study text, and contribute to discussions. It is the student's responsibility to contact the instructor in the event a midterm exam is missed and to make arrangements before returning to class. Drops after (Consult CSUDH Class Schedule) will reflect the students average. The student must initiate all withdrawal procedures. **Non-attendance does not constitute withdrawal and could result in an unearned F.**

**You will be notified in class of any and all changes to this syllabus**

---

**Tentative weekly schedule:**

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Reading</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction and review of references on Secure Coding practices and other resources</td>
<td>Ch 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Hardware and Software Architecture</td>
<td>Ch 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Hardware and Software Architecture</td>
<td>Ch 2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Assembly Language Fundamentals</td>
<td>Ch 3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data Transfer</td>
<td>Ch 4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Using Assembler</td>
<td>Appendix A</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Addressing, Arithmetic</td>
<td>Ch 4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Procedures</td>
<td>Ch 5</td>
<td>Midterm_1</td>
</tr>
<tr>
<td>9</td>
<td>Conditional Processing</td>
<td>Ch 6</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Integer Arithmetic</td>
<td>Ch 7</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Advanced Procedures</td>
<td>Ch 8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>String and Arrays</td>
<td>Ch 9</td>
<td>Midterm_2</td>
</tr>
<tr>
<td>13</td>
<td>Structures and Macros</td>
<td>Ch 10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>File Processing</td>
<td>Ch 11</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Storage and DOS Programming</td>
<td>Ch 14-15</td>
<td></td>
</tr>
</tbody>
</table>

**FINAL EXAM Thursday, May 19, 2011 @ 5:30PM - 7:30PM**
Assignment 0 (not graded)
Due: week 2
Create addition and multiplication tables for octal and hexadecimal codes.

Assignment 1
Due: week 6
Write, assemble, and test a program to calculate and display the sum
\[ S = (\text{there will be individual formulas for every student}) \]
Read N from keyboard. Use LOOP instruction

Assignment 2
Due: week 8
Write, assemble, and test a program to read N characters from the keyboard and store those characters in an array myarray.
After that process an array in the following fashion: (there will be individual algorithm for every student).

Assignment 3
Due: week 10
Write, assemble, and test a program that calculates the sum of all "divisible by (value will be given individually to every student)" numbers from the range [n,m].

For this exercise you are required to use subroutine. Calculate sum/product in the main procedure;
Check if the number is "divisible by x" in the subroutine.

Assignment 4
Due: week 12
Write a program that reads a paragraph from the keyboard and processes it in the following fashion:
(there will be individual algorithm for every student).
Terminate the paragraph with special character (individual for every student).
Display results on screen.