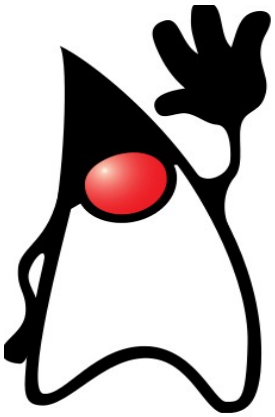
CS 280 Assembly Language & Machine Architecture
Imperial Valley College Fall 2013

Class Number: ***10380*** **(4 credit units)**
Room: 1705
TTh 5:15pm-8:15pm



Instructor:	Rick Castrapel
Phone:	(760) 355-6505
Office:	Room 2773
Email:	rick.castrapel@imperial.edu
Web Site:	http://www.imperial.edu/rick.castrapel
Textbook:	Principles of Computer Organization and Assembly Language Using the Java Virtual Machine; Juola Pearson/Prentice Hall 2007 ISBN: 0-13-148683-7
Office Hours:	MW 6:00-7:00pm,TR 4:00-5:00pm or by appointment

Course Description:	A course covering general concepts of internal organization of a computer a, machine and assembly language Topics include number system and data representation, primitive instructions and operations, program execution, addressing techniques, arrays, subroutines, macros, recursion, virtual memory, cache memory interrupt handling, and memory management.
Student Learning Outcomes:	1. Correctly describe the internal binary representation of base 10 integer and floating point values, character values, and logical values. 2. Describe nature and role of the system bus, channels, controllers, interfaces, direct memory access (DMA), cache memory, real memory and virtual memory. 3. Demonstrate an understanding of all steps involved in the assembly language programming process by writing/editing, assembling, compiling, linking and loading a simple assembly language program. 4. Demonstrate an understanding of addressing modes by writing/editing, assembling, compiling, linking and loading a simple assembly language program that employs a variety of addressing modes to reference stored scalar variables and array elements.
Prerequisites:	CS 220 and Math 90 or 91 or equivalent college course with "C" or better, or eligibility determined by math placement process.
Cell Phones:	Keep cell phones turned off during class. You may not use a cell phone as a calculator during tests.
Homework:	Homework will be assigned during some class periods and will be due at the next week.
Labs:	Most classes will have a lab component. These are a mandatory part of the course. You are encouraged to work on the labs in small groups, but each individual must turn in their own results.
Dropping:	You may be dropped from this class if you miss the first day or if you miss three or more class sessions total.The last day to drop this class is Nov 9. After that date, I must give you a letter grade. It is your responsibility to drop, not mine.
DSP&S:	Any student with a documented disability who may need educational accommodations should notify the Disabled Student Programs and Services (DSP&S) office as soon as possible. Room 2117 Health Sciences Building (760) 355-6312.
Grading:	There will be one midterm test worth 100 points. There will be a comprehensive final exam worth 100 points. Labs are 200 points total. Your homework is worth 100 points total.

Grading Policy

Midterm Test	100 points
Homework	100 points
Labs	200 points
Final Exam	100 points
Total	**500 points**

Grading Scale

90-100 %	A
80-89%	B
70-79 %	C
60-69 %	D
< 60 %	F

Academic Integrity is assumed and necessary. You will be treated as an adult professional and will be expected to behave accordingly. You must follow the Computer Sciences Code of Conduct. Programming assignments must be done individually. Failure to do so will result in a violation of the Academic Honor Code. The following cases will be considered as violations: identical code, and extremely similar code. Violations will be reported to the Office of Vice President of Student Services. Disruptive students will be required to leave the class for the day. Continued disruptive behavior, cheating, plagiarism or deliberate unsafe computer use may result in severe academic penalty. See the college bulletin for details.

CS 280 Fall 2013 Tentative Schedule

Date	Text	Event	Notes
08/20/11	Ch 1	**Computation and Representation**	
08/22/11			Also Appendices A & E
08/27/11			
08/29/11	Ch 2	**Arithmetic Expressions**	
09/03/11			
09/05/11	Ch 3	**Assembly Language Programming In *Jasmin***	
09/10/11			
09/12/11			Also Appendices B & C
09/17/11			
09/19/11	Ch 4	**Control Structures In *Jasmin***	
09/24/11			
09/26/11			
10/01/11			
10/03/11	**MidTerm**	**Midterm Exam**	
10/08/11	Ch 5	**General Architecture Issues: Real Computers**	
10/10/11	Ch 6	**Intel 8088**	
10/15/11			
10/17/11			
10/22/11	Ch 8	**Intel Pentium**	
10/24/11			
10/29/11			
10/31/11			
11/05/11	No Text	**Microcontrollers: The PIC**	
11/07/11			
11/12/11			
11/14/11			
11/19/11			
11/21/11			
11/26/11			
11/28/11		**HOLIDAY**	Thanksgiving
12/03/11		**Review**	Review for Final Exam
12/05/11	**Final Exam**	**Comprehensive Final**	

CS 220

STUDENT LEARNING OUTCOMES:

Upon course completion, the successful student will have acquired new skills, knowledge, and or attitudes as demonstrated by being able to:

1. Correctly describe the internal binary representation of base 10 integer and floating point values, character values, and logical values.(ILO1, ILO2, ILO4)
2. Describe nature and role of the system bus, channels, controllers, interfaces, direct memory access (DMA), cache memory, real memory and virtual memory.(ILO1, ILO2, ILO4)
3. Demonstrate an understanding of all steps involved in the assembly language programming process by writing/editing, assembling, compiling, linking and loading a simple assembly language program.(ILO1, ILO2, ILO4)
4. Demonstrate an understanding of addressing modes by writing/editing, assembling, compiling, linking and loading a simple assembly language program that employs a variety of addressing modes to reference stored scalar variables and array elements.(ILO1, ILO2, ILO4)

MEASURABLE COURSE OBJECTIVES AND MINIMUM STANDARDS FOR GRADE OF "C":

Upon satisfactory completion of the course, students will be able to:

1. Correctly describe the internal binary representation of base I0 integer and real values, character values, and logical values.
2. Describe nature and role of the system bus, channels, controllers, interfaces, direct memory access (DMA), cache memory, real memory and virtual memory.
3. Demonstrate an understanding of all steps involved in the assembly language programming process by writing/editing, assembling, compiling, linking and loading a simple assembly language program.
4. Demonstrate an understanding of addressing modes by writing/editing, assembling, compiling, linking and loading a simple assembly language program that employs a variety of addressing modes to reference stored scalar variables and array elements.
5. Demonstrate an understanding of assembly language flow control concepts by writing/editing, assembling, compiling, linking and loading a simple assembly language program that contains branching and looping constructs.
6. Demonstrate an understanding of assembly language subroutines and macros by writing/editing, assembling, compiling, linking and loading a simple assembly language program that contains and invokes procedures, functions, and macros.
7. Demonstrate an understanding of recursion by writing/editing, assembling, compiling, linking and loading a simple assembly language program that implements a recursive algorithm using one or more stacks.
8. Demonstrate an understanding of interrupts and interrupt handling by writing/editing, assembling, compiling, linking and loading a simple assembly language program that detects and services an interrupt.

Imperial Valley College Computer Sciences Code of Conduct

We believe that everyone has a right to work in an environment where people treat one another honestly and fairly. Because academic dishonesty can threaten this environment we will pursue abuses of the policies outlined below aggressively.

When you submit any piece of work for grading or other evaluation, the reader will assume that you are the sole author of all aspects of it. The expectation is that you are the originator of every idea and author of every sentence in an essay, help file, or other document, that you wrote every line of code, that you designed every data structure and created every piece of data. In practice, you will often have good reason to use other people's work or to collaborate with others in creating a work that you will submit. In these cases, it is your responsibility to make the reader clearly aware of what has come from other sources. If a reasonable reader would assume, on reading your work, that some part was created by you alone when in fact it was created by someone else or by you in partnership with someone else, that reader has been misled. It is your responsibility to prevent such misimpressions, and the department will hold students accountable both for intentionally misleading readers and for failing to prevent reasonable misimpressions.

Code Plagiarism.

Computer science is a discipline where it is difficult to draw a precise line between acceptable and unacceptable collaboration. On the one hand we want to encourage you to try out other peoples' code; code reuse is an area of active research within computer science. On the other hand you will learn to write code only if you do it yourself. You are not learning and have crossed the line of acceptable behavior if you do not understand the solution you have submitted. We have the right to ask students to explain the code they submit. If you have "reused" someone else's code to an extent that you feel a need to change variable names or slightly rearrange the order of statements, then you have also violated the honor code. We also reserve the right to use electronic tools to check code for plagiarism. By submitting code for grading in any computer science course, you grant the instructor a license to send a copy of that code for plagiarism analysis to a research service, such as MOSS. The instructor, or their service, may compare your code against other students' code, or compare their code to yours. Give credit to someone else's ideas with a citation rather than turning in their work as your own.

Text Plagiarism.

When you hand in an essay or other writing assignment, you must give credit to your sources. You must provide a reference for any idea, conclusion, information or data that you got from another source (such as a book, an article on the Net, or a person). If you use someone's words, you must show that you are quoting them (use quotation marks or indent long quotes) and your reference should show your exact source (such as the page number of the article or book). If you quote someone, you must quote them accurately, word for word. To avoid plagiarizing, you might find the following articles useful:

- Cheating and Plagiarism in Regulations section of IVC General Catalog
- How Not to Plagiarize at `http://www.utoronto.ca/writing/plagsep.html`.
- Citing Sources and Avoiding Plagiarism at `http://www.lib.duke.edu/libguide/cite/works_cited.htm`.

By submitting a writing assignment for grading in any computer science course, you grant the instructor a license to send a copy of that assignment for plagiarism analysis to a research service, such as TurnItIn. The instructor, or their service, may compare your paper against other students' papers, or compare their papers to yours.

Social Responsibility.

Many people use our machines: students, faculty, staff, and outside visitors. Our machines affect other machines on and off campus and they affect the users of these machines. It is not hard to abuse others by mailing ``spam," ``flaming" to newsgroups, being a ``cracker," displaying digital pornography, bogging down the CPU with processes, or hogging the printer. We expect your use of computer resources will be based on the Golden Rule: do unto others as you would have them do unto you. Poor social responsibility because you are new is one thing, but malicious practices are another matter and will not be tolerated. Do not use BitTorrent or messenger services on IVC Computer Science computers, as these are a common source of computer viruses.

Right to Privacy.

You are encouraged to store *electronic property* on computers provided for your use by Computer Sciences, and you have a privacy right to this information. Others also have a right of privacy to the property they store on our computers. You should not search other's file systems, read their mail, scan or remove their files, try to crack their password, login as someone else, intercept other's network traffic, install viruses, or otherwise violate the right to privacy of others. We will not intentionally abuse your right to privacy. However, to administer our machines we may need to do things you should not, for example, we may need to try to crack your password to verify that it is secure, or kill your processes, or remove your files, or read your email, or otherwise invade your privacy when we suspect you are an abuser of our systems.

Discrimination:

It is the policy of the university that all students, faculty, staff, and guests enjoy an environment free from all forms of discrimination, including ethnic, racial, religious, and sexual harassment.

Disclaimers.

The Imperial Valley College Catalog, and the Student Handbook have additional guidelines on campus standards, behavior, discipline, complaint resolution, etc. The Computer Science Honor Code does not replace or supersede these polices. Faculty teaching computer science courses may establish other *honor* criteria for their classes.

As our machines are part of a larger international network, we assume certain responsibilities as a member of a growing electronic community. Exercising this responsibility may require us to search for suspected abusers of our or others computers. If you suspect that someone has violated your rights as a user of our machines, inform the systems administrator; do not attempt to track them down yourself.

Ideas for this code of honor have been collected from other universities, most notably, Stanford University and the University of Florida.