



## Basic Course Information

Semester:	<b>Fall 2022</b>	Instructor Name:	<b>Octavio Ortiz</b>
Course Title & #:	<b>CS 281</b>	Email:	<b>octavio.ortiz@imperial.edu</b>
CRN #:	<b>10522</b>	Webpage (optional):	<b>Canvas Course</b>
Classroom:	<b>2724</b>	Office #:	<b>2767.1</b>
Class Dates:	<b>8/15 – 12/10</b>	Office Hours:	<b>Faculty Schedule</b>
Class Days:	<b>M/W</b>	Office Phone #:	<b>760-355-5706</b>
Class Times:	<b>11:30 – 2:10 PM</b>	Emergency Contact:	<b>Silvia Murray</b>
Units:	<b>3</b>	Class Format:	<b>In-Person</b>

## Course Description

Basics of machine architecture, machine language, assembly language and operating systems. Representations of data types and structures along with instruction representation and execution, addressing modes, subroutine calls and return mechanisms, fixed point systems, and basic organization of the von Neumann machine are included. (C-ID COMP 142) (CSU/UC)

## Course Prerequisite(s) and/or Corequisite(s)

CS 220 or CS 221 and MATH 091 or MATH 098 with a grade of C or better, or appropriate placement as defined by AB 705.

## 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. Describe how data is represented in computer memory.
2. Show how fundamental high-level programming constructs are implemented at the machine-language level.
3. Write simple assembly language program segments.

## Course Objectives

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

1. Describe how data are represented in the computer, including floating point numbers and arrays.
2. Code, execute and debug assembly language programs.
3. Demonstrate an understanding of computer architecture.
4. Demonstrate the function of CPU registers.
5. Use correct addressing modes and terminology.
6. Use integer arithmetic instructions.
7. Correctly use repetition constructs, macros and procedures in assembly language.
8. Program keyboard input and text screen output.
9. Use machine instructions involving data structure stacks to code selected algorithms.

## Textbooks & Other Resources or Links

### Assembly Language for x86 Processors

Author: Kip Irvine  
 Edition: 8th  
 ISBN: 978-0-13-538165-6  
 Copyright Year: 2020  
 Publisher: Pearson

## Course Requirements and Instructional Methods

Students will be exposed to various instructional methods. Lectures, both in person and through pre-recorded tutorial videos, will introduce students to fundamental programming concepts. Students will then apply what they learn in lectures to their own programming assignments and applications.

There will be weekly programming assignments where students will reflect on their learning and engage in previous, current and new programming topics. A comprehensive semester final exam or project will assess students' ability to read, debug and rationalize code segments that range in complexity.

## Course Grading Based on Course Objectives

ASSIGNMENT	POINTS
<b>Programming Exercises</b>	<b>15%</b>
Approximately 3-5 per week	
<b>Quizzes</b>	<b>50%</b>
3 Planned Quizzes	
<b>Projects/Final Exam</b>	<b>35%</b>
Midterm/Final project & comprehensive final	
<b>Total</b>	<b>100%</b>

Score	Letter Grade
<b>≥ 90%</b>	<b>A</b>
<b>≥ 80%</b>	<b>B</b>
<b>≥ 70%</b>	<b>C</b>
<b>≥ 60%</b>	<b>D</b>
<b>&lt; 60%</b>	<b>F</b>

## Course Policies

### Attendance:

Attendance is mandatory. Students are expected to attend every class meeting. Lectures will preview programming assignments, programming applications and future assessments.

- Although attendance is not explicitly factored into your grade, failing to complete programming assignments and assessments due to absences will negatively impact your grade.
- Students with excessive absences will be dropped from the course as outlined in AP 5075.

### Late Submissions:

Programming assignments are to be completed and submitted by the due date stated on Canvas. Late programming assignments will be accepted and penalized as follows:

- 90% maximum score if submitted within 24 hours past due date
- 80% maximum score if submitted within 48 hours past due date
- 70% maximum score if submitted within 72 hours past due date
- 50% maximum score if more than three days and less than a week past due date
- No credit will be given to assignments that submitted past the hard deadline (see calendar)

Programming applications/projects, quizzes and the final exam will NOT be accepted late.

### Make-up Assignments:

There are no make-up assignments.

- Programming applications/projects and quizzes cannot be made up, however, if the material is presented again in future applications or quizzes, then the failed assessment will be reevaluated.

### Drop Policy

The instructor reserves the right to drop students who fail to attend the first-class session or fail to complete the first assignment by the assigned due date.

## Other Course Information

### Resources:

<https://asmirvine.com/> - Publisher resources

<https://cs.lmu.edu/~ray/notes/x86assembly/> - x86 Assembly Language Programming

## IVC Student Resources

IVC wants you to be successful in all aspects of your education. For help, resources, services, and an explanation of policies, visit <http://www.imperial.edu/studentresources> or click the heart icon in Canvas.

## Anticipated Class Schedule/Calendar

The semester calendar is meant to provide an overview of the topics that will be covered throughout the semester. Every effort will be made to adhere to the calendar; however, changes might be necessary.

Week	Date	Topic	Assignment
Week 1	8/15	<ul style="list-style-type: none"> <li>• <b>Syllabus &amp; Course Policies</b> <ul style="list-style-type: none"> <li>○ Modules, collaborative notes, programming assignments, etc...</li> </ul> </li> </ul>	
	8/17	<ul style="list-style-type: none"> <li>• <b>Basic Concepts</b> <ul style="list-style-type: none"> <li>○ Data representation (Binary, hex, addition, subtraction)</li> </ul> </li> </ul>	
Week 2	8/22	<ul style="list-style-type: none"> <li>• <b>Basic Concepts</b> <ul style="list-style-type: none"> <li>○ Data Types</li> </ul> </li> </ul>	
	8/24	<ul style="list-style-type: none"> <li>• <b>x86 Processor Architecture</b> <ul style="list-style-type: none"> <li>○ Components of a typical x86 Computer</li> <li>○ 32-BIT x86 Processors</li> </ul> </li> </ul>	
Week 3	8/29	<ul style="list-style-type: none"> <li>• Assembly Language Fundamentals <ul style="list-style-type: none"> <li>○ First assembly language program</li> <li>○ Adding &amp; subtracting Integers</li> </ul> </li> </ul>	
	8/31	<ul style="list-style-type: none"> <li>• Assembly Language Fundamentals <ul style="list-style-type: none"> <li>○ Defining data</li> <li>○ Symbolic constants</li> </ul> </li> </ul>	
Week 4	9/5	Labor Day (no class)	
	9/7	<ul style="list-style-type: none"> <li>• Assembly Language Fundamentals <ul style="list-style-type: none"> <li>○ Determining appropriate data types</li> </ul> </li> </ul>	
Week 5	9/12	<ul style="list-style-type: none"> <li>• <b>Quiz 1 (Chapters 1-3)</b> <ul style="list-style-type: none"> <li>○ <b>Deadline to submit late assignments (Ch.1-4) for 50% credit.</b></li> </ul> </li> <li>• Data Transfers, Addressing &amp; Arithmetic <ul style="list-style-type: none"> <li>○ Data transfer instructions</li> </ul> </li> </ul>	
	9/14	<ul style="list-style-type: none"> <li>• Data Transfers, Addressing &amp; Arithmetic <ul style="list-style-type: none"> <li>○ Addition &amp; Subtraction</li> <li>○ Data-related operators &amp; directives</li> </ul> </li> </ul>	
Week 6	9/19	<ul style="list-style-type: none"> <li>• Data Transfers, Addressing &amp; Arithmetic <ul style="list-style-type: none"> <li>○ JMP &amp; LOOP Instructions</li> </ul> </li> </ul>	
	9/21	<ul style="list-style-type: none"> <li>• Data Transfers, Addressing &amp; Arithmetic <ul style="list-style-type: none"> <li>○ Arrays</li> </ul> </li> </ul>	
Week 7	9/26	<ul style="list-style-type: none"> <li>• Procedures <ul style="list-style-type: none"> <li>○ Stack Operations</li> <li>○ Defining and Using Procedures</li> </ul> </li> </ul>	
	9/28	<ul style="list-style-type: none"> <li>• Procedures <ul style="list-style-type: none"> <li>○ Linking to an External Library</li> <li>○ Irvine32 Library</li> </ul> </li> </ul>	
Week 8	10/3	<b>Midterm Project</b>	
	10/5		

Week	Date	Topic	Assignment
Week 9	10/10	<ul style="list-style-type: none"> <li>• <b>Quiz 2 (Chapters 4-5)</b> <ul style="list-style-type: none"> <li>○ Deadline to submit late assignments (Ch.4-5) for 50% credit.</li> </ul> </li> <li>• Conditional Processing               <ul style="list-style-type: none"> <li>○ Boolean &amp; Comparison Instructions</li> <li>○ Conditional Jumps</li> </ul> </li> </ul>	
	10/12	<ul style="list-style-type: none"> <li>• Conditional Processing               <ul style="list-style-type: none"> <li>○ Conditional Loop Instructions</li> <li>○ Conditional Structures</li> </ul> </li> </ul>	
Week 10	10/17	<ul style="list-style-type: none"> <li>• Integer Arithmetic               <ul style="list-style-type: none"> <li>○ Shift &amp; Rotate Instructions</li> <li>○ Shift &amp; Rotate Applications</li> </ul> </li> </ul>	
	10/19	<ul style="list-style-type: none"> <li>• Integer Arithmetic               <ul style="list-style-type: none"> <li>○ Multiplication &amp; Division Instructions</li> <li>○ Extended Addition &amp; Subtraction</li> <li>○ ASCII and Unpacked Decimal Arithmetic</li> </ul> </li> </ul>	
Week 11	10/24	<ul style="list-style-type: none"> <li>• Advanced Procedures               <ul style="list-style-type: none"> <li>○ Stack Frames</li> <li>○ Recursion</li> </ul> </li> </ul>	
	10/26	<ul style="list-style-type: none"> <li>• Advanced Procedures               <ul style="list-style-type: none"> <li>○ INVOKE, ADDR, PROC, and PROTO</li> <li>○ Creating Multimode Programs</li> </ul> </li> </ul>	
Week 12	10/31	<ul style="list-style-type: none"> <li>• Strings &amp; Arrays               <ul style="list-style-type: none"> <li>○ String Primitive Instructions</li> <li>○ Selected String Procedures</li> </ul> </li> </ul>	
	11/2	<ul style="list-style-type: none"> <li>• Strings &amp; Arrays               <ul style="list-style-type: none"> <li>○ Two-Dimensional Arrays</li> <li>○ Searching &amp; Sorting Integer Arrays</li> </ul> </li> </ul>	
Week 13	11/7	<ul style="list-style-type: none"> <li>• <b>Quiz 3 (Chapters 6-9)</b> <ul style="list-style-type: none"> <li>○ Deadline to submit late assignments (Ch.4-5) for 50% credit.</li> </ul> </li> <li>• Structures &amp; Macros               <ul style="list-style-type: none"> <li>○ Structures</li> <li>○ Macros</li> </ul> </li> </ul>	
	11/9	<ul style="list-style-type: none"> <li>• Structures &amp; Macros               <ul style="list-style-type: none"> <li>○ Conditional-Assembly Directive</li> <li>○ Defining Repeat Blocks</li> </ul> </li> </ul>	
Week 14	11/14	<ul style="list-style-type: none"> <li>• MS-Windows Programming               <ul style="list-style-type: none"> <li>○ Win32 Console Programming</li> </ul> </li> </ul>	
	11/16	<ul style="list-style-type: none"> <li>• MS-Windows Programming               <ul style="list-style-type: none"> <li>○ Writing a Graphical Windows Application</li> </ul> </li> </ul>	
THANKSGIVING BREAK			
Week 15	11/28	<ul style="list-style-type: none"> <li>• MS-Windows Programming</li> </ul>	



Week	Date	Topic	Assignment
	11/30	<ul style="list-style-type: none"><li>MS-Windows Programming</li><li><b>Final Project</b></li></ul>	
Week 16	12/5	<ul style="list-style-type: none"><li><b>Final Project</b></li></ul>	
	12/7	<ul style="list-style-type: none"><li><b>Final Project Due</b></li><li><b>Comprehensive Final Exam</b></li></ul>	

**\*\*\*Subject to change without prior notice\*\*\***