IMPERIAL VALLEY COLLEGE

Basic Course Information

Semester:	**Fall 2025**	Instructor Name:	**Octavio Ortiz**
Course Title & #:	**CS 281**	Email:	**octavio.ortiz@imperial.edu**
CRN #:	**10522**	Webpage (optional):	**Canvas Course**
Classroom:	**801**	Office #:	**2766**
Class Dates:	**8/11 – 12/6**	Office Hours:	**Faculty Schedule**
Class Days:	**T/R**	Office Phone #:	**760-355-5706**
Class Times:	**11:20 – 12:45 PM**	Emergency Contact:	**Silvia Murray**
Units:	**3**	Class Format:	**Hybrid (In-Person/Online)**

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 (integers, real numbers, characters, colors and sounds) is represented in computer memory.
2. Show how fundamental high-level programming constructs are implemented at the assembly-language level.
3. Write a programming application in assembly language.

Course Objectives

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

1. Identify and describe the internal representation of non-numeric data.
2. Interpret and convert numerical data from one base format to another.
3. Define and demonstrate the conversion of fundamental high-level programming constructs to both the assembly and machine language level.
4. Describe how data are represented in the computer, including floating point numbers and arrays.
5. Code, execute and debug assembly language programs.
6. Demonstrate an understanding of computer architecture.
7. Demonstrate the function of CPU registers.
8. Use correct addressing modes and terminology.
9. Use integer arithmetic instructions.
10. Correctly use repetition constructs, macros and procedures in assembly language.
11. Program keyboard input and text screen output.

12. Use machine instructions involving data structure stacks to code selected algorithms.
13. Write a simple interrupt handler.

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
Programming Exercises	**10%**
Approximately 10-15	
Exams	**60%**
Three planned exams	
Projects/Final Exam	**30%**
Midterm/Final project & comprehensive final	
Total	**100%**

Score	Letter Grade
≥ 90%	**A**
≥ 80%	**B**
≥ 70%	**C**
≥ 60%	**D**
< 60%	**F**

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

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	8/12	• **Syllabus & Course Policies** ○ Modules, collaborative notes, programming assignments, etc…	
	8/14	• **Basic Concepts** ○ Data representation (Binary, hex, addition, subtraction)	
Week 2	8/19	• **Basic Concepts** ○ Data Types	
	8/21	• **x86 Processor Architecture** ○ Components of a typical x86 Computer ○ 32-BIT x86 Processors	
Week 3	8/26	• **Assembly Language Fundamentals** ○ First assembly language program ○ Adding & subtracting Integers	
	8/28	• **Assembly Language Fundamentals** ○ Defining data ○ Symbolic constants	
Week 4	9/2	• **Assembly Language Fundamentals** ○ Determining appropriate data types	
	9/4	• **Assembly Language Fundamentals** ○ Determining appropriate data types • **Exam 1 (Chapters 1-3)** ○ Deadline to submit late assignments (Ch.1-3) for 50% credit.	
Week 5	9/9	• Data Transfers, Addressing & Arithmetic ○ Data transfer instructions	
	9/11	• Data Transfers, Addressing & Arithmetic ○ Addition & Subtraction ○ Data-related operators & directives	
Week 6	9/16	• Data Transfers, Addressing & Arithmetic ○ JMP & LOOP Instructions	
	9/18	• Data Transfers, Addressing & Arithmetic ○ Arrays	
Week 7	9/23	• Procedures ○ Stack Operations ○ Defining and Using Procedures	
	9/25	• Procedures ○ Linking to an External Library ○ Irvine32 Library	
Week 8	9/30	**Midterm Project**	

Week	Date	Topic	Assignment
	10/2	• **Exam 2 (Chapters 4-5)** ○ Deadline to submit late assignments (Ch.4-5) for 50% credit.	
Week 9	10/7	• Conditional Processing ○ Boolean & Comparison Instructions ○ Conditional Jumps	
	10/9	• Conditional Processing ○ Conditional Loop Instructions ○ Conditional Structures	
Week 10	10/14	• Integer Arithmetic ○ Shift & Rotate Instructions ○ Shift & Rotate Applications	
	10/16	• Integer Arithmetic ○ Multiplication & Division Instructions ○ Extended Addition & Subtraction ○ ASCII and Unpacked Decimal Arithmetic	
Week 11	10/21	• Advanced Procedures ○ Stack Frames ○ Recursion	
	10/23	• Advanced Procedures ○ INVOKE, ADDR, PROC, and PROTO	
Week 12	10/28	• Advanced Procedures ○ Creating Multimode Programs	
	10/30	• **Exam 3 (Chapters 6-8)** ○ Deadline to submit late assignments (Ch.6-8) for 50% credit.	
Week 13	11/4	• Strings & Arrays ○ String Primitive Instructions ○ Selected String Procedures ○ Two-Dimensional Arrays ○ Searching & Sorting Integer Arrays	
	11/6	• Structures & Macros ○ Structures ○ Macros ○ Conditional-Assembly Directive	
Week 14	11/11	**Veteran's Day – No Class**	
	11/13	• Structures & Macros ○ Defining Repeat Blocks • MS-Windows Programming ○ Win32 Console Programming	
Week 15	11/18	• **Final Project** • MS-Windows Programming ○ Writing a Graphical Windows Application	
	11/20	• MS-Windows Programming	

Week	Date	Topic	Assignment
THANKSGIVING BREAK			
Week 16	12/2	• Final Project	
	12/4	• Final Project Due • Comprehensive Final Exam	

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

IMPERIAL VALLEY COLLEGE

OCTAVIO ORTIZ

IVC Instructor's Schedule of Classes and Office Hours

Science, Math and Engineering Department

Fall 2025

		7:00 AM	8:00 AM	9:00 AM	10:00 AM	11:00 AM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM
M	COURSE		CS 221 - HYBRID		ENGR 100			Office Hours				
	CRN		LEC/LAB 10521		LEC/LAB 10929			Zoom				
	LEC/LAB		8:00 - 9:25 AM		10:15 - 12:45 PM			1:00 - 2:00 PM				
	FACULTY		RM #801		RM #4300			RM #2766				
T	COURSE		Office Hours	ENGR 210		CS 281 - HYBRID			CS 221			
	CRN		In-Person	LEC 10603		LEC/LAB 10522			LEC/LAB 10962			
	LEC/LAB		8:30 - 9:30 AM	9:40 - 11:05 AM		11:20 - 12:45 PM			2:00 - 4:30 PM			
	FACULTY		RM #2766	RM #2721		RM #801			RM #801			
W	COURSE		CS 221 - HYBRID		ENGR 100			Office Hours				
	CRN		LEC/LAB 10521		LEC/LAB 10929			In-Person				
	LEC/LAB		8:00 - 9:25 AM		10:15 - 12:45 PM			1:00 - 2:00 PM				
	FACULTY		RM #801		RM #4300			RM #2766				
R	COURSE		Office Hours	ENGR 210		CS 281 - HYBRID			CS 221			
	CRN		Zoom	LEC 10603		LEC/LAB 10522			LEC/LAB 10962			
	LEC/LAB		8:30 - 9:30 AM	9:40 - 11:05 AM		11:20 - 12:45 PM			2:00 - 4:30 PM			
	FACULTY		RM #2766	RM #2721		RM #801			RM #801			
F	COURSE											
	CRN											
	LEC/LAB											
	FACULTY											

Course No	Sections	Lec Hrs	Lab Hrs
ENGR 100	1	2	3
ENGR 210	1	3	0
CS 221	2	2	3
CS 281	1	2	3
Total		11	12

Instructor's Name	Octavio Ortiz
Phone	760-355-5706
Office No.	2766
Email	octavio.ortiz@imperial.edu