IMPERIAL VALLEY COLLEGE

Basic Course Information

Semester:	**Fall 2021**	Instructor Name:	**Octavio Ortiz**
Course Title & #:	**CS 281**	Email:	**octavio.ortiz@imperial.edu**
CRN #:	**10526**	Webpage (optional):	**Canvas**
Classroom:	**N/A**	Office #:	**2767.1**
Class Dates:	**8/16/21 – 12/11/21**	Office Hours:	**MW: 9:40 – 10:10 AM** **6:00 – 6:30 PM** **T/TR: 9:10 – 10:10 AM**
Class Days:	**N/A**	Office Phone #:	**760-355-5706**
Class Times:	**N/A**	Emergency Contact:	**Silvia Murray: 760-355-6201**
Units:	**3**	Class Format:	**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 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

zyBooks – Assembly Language and Machine Organization
Digital Interactive Book
Access through Canvas

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.

zyBooks assignments will consists of participation and challenge activities. Programming assignments will be relatively short and will assess a student's mastery of a particular programming skill, as well as a student's ability to problem solve.

There will be weekly discussion posts where students will reflect on their learning and engage in discussions regarding 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 Assignments	**40%**
Will vary throughout the semester	
zyBooks Participation Activities	**20%**
Will vary throughout the semester	
zyBooks Challenge Activities	**20%**
Will vary throughout the semester	
Final Exam	**20%**
Comprehensive final due **12/10/21**	
Total	**100%**

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

Course Policies

Attendance:

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.

Late Submissions:

Programming assignments and discussion posts are to be completed and submitted by the due date stated on Canvas. Late programming assignments and discussion posts 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 are one week or more past due

Programming applications/projects the final exam/project will NOT be accepted late.

Make-up Assignments:

There are no make-up assignments.

- Programming assignments that are more than a week past due will receive a score of 0 and cannot be made up.
- Programming applications/projects cannot be made up, however, if the material is presented again in future applications, then the failed assessment will be reevaluated.

Drop Policy

The instructor reserves the right to drop students who fail to complete the first assignment by the assigned due date.

Other Course Information

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.

Course 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	8/16	• **zyBooks – Information as Bits** ○ Sections 1.1-1.7 ○ Install VS and configure MASM assembler ○ Spin up Windows Instance in AWS (if needed)	
Week 2	8/23	• **zyBooks – Information as Bits** ○ Sections 1.8 – 1.13	
Week 3	8/30	• **zyBooks – MIPS Assembly Part 1** ○ Sections 2.1 – 2.5	
Week 4	9/6	• **zyBooks – MIPS Assembly Part 1** ○ Sections 2.6 – 2.10 ○ Programming Assignment MASM	
Week 5	9/13	• **zyBooks – MIPS Assembly Part 2** ○ Sections 3.1 – 3.5	
Week 6	9/20	• **zyBooks – MIPS Assembly Part 2** ○ Sections 3.6 – 3.10 ○ Programming Assignment MASM	
Week 7	9/27	• **zyBooks – C to Assembly** ○ Sections 4.1 – 4.5	
Week 8	10/4	• **zyBooks – C to Assembly** ○ Sections 4.6 – 4.9 ○ Programming Assignment MASM	
Week 9	10/11	• **zyBooks – MIPSzy Processor Design** ○ Sections 5.1 – 5.6	
Week 10	10/18	• **zyBooks – MIPSzy Processor Design** ○ Sections 5.7 – 5.13 ○ Programming Assignment MASM	
Week 11	10/25	• **zyBooks – Memory** ○ Sections 6.1 – 6.6	
Week 12	11/1	• **zyBooks – Memory** ○ Sections 6.7 – 6.13 ○ Programming Assignment MASM	
Week 13	11/8	• **zyBooks – Input/Output** ○ Sections 7.1 – 7.4	
Week 14	11/15	• **zyBooks – Input/Output** ○ Sections 7.5 – 7.8 ○ Programming Assignment MASM	
Thanksgiving Break			

Week	Date	Topic	Assignment
Week 15	11/29	• **Review**	
Week 16	12/6	• **Comprehensive Final Exam/Project (Due 12/10/21)**	

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