**COURSE OUTLINE OF RECORD**

**Number:** CS G242  
**TITLE:** Computer Architecture and Organization

**ORIGINATOR:** Cristian Racataian  
**EFF TERM:** Fall 2014

**CROSS LISTED COURSE:**  
**TOP NO:** 0706.00  
**CID:** COMP 142

**SEMESTER UNITS:** 3.0  
**HRS LEC:** 54.0  
**HRS LAB:** 0.0  
**HRS OTHER:** 0.0

**CONTACT HRS TOTAL:** 54.0  
**STUDY NON-CONTACT HRS RECOMMENDED:** 108.0

**CATALOG DESCRIPTION:**

This course will cover the organization and behavior of real computer systems at the assembly-language level. The mapping of statements and constructs in a high-level language onto sequences of machine instructions is studied, as well as the internal representation of simple data types and structures. Numerical computation is examined, noting the various data representation errors and potential procedural errors.

**JUSTIFICATION FOR COURSE:**

Compliance with C-ID. AS-T core course

**PREREQUISITES:**

**COREQUISITES:**

**ADVISORIES:**

- CS G175: C++ Programming
- or
- CS G153: Java Programming, Introduction

**ASSIGNED DISCIPLINES:**

Computer science

**MATERIAL FEE:** Yes [ ] No [X] Amount: $0.00

**CREDIT STATUS:** Noncredit [ ] Credit - Degree Applicable [X] Credit - Not Degree Applicable [ ]

**GRADING POLICY:** Pass/No Pass [ ] Standard Letter [X] Not Graded [ ] Satisfactory Progress [ ]

**OPEN ENTRY/OPEN EXIT:** Yes [ ] No [X]

**TRANSFER STATUS:** CSU Transferable[ ] UC/CSU Transferable[X] Not Transferable[ ]

**BASIC SKILLS STATUS:** Yes [ ] No [X]  
**LEVELS BELOW TRANSFER:** Not Applicable

**CALIFORNIA CLASSIFICATION CODES:** Y - Not Applicable

**NON CREDIT COURSE CATEGORY:** Y - Not applicable, Credit Course

**OCCUPATIONAL (SAM) CODE:** E

**REPEATABLE ACCORDING TO STATE GUIDELINES:** No [X] Yes [ ] NUMBER REPEATS:

**REQUIRED FOR DEGREE OR CERTIFICATE:** No [ ] Yes [X]

Computer Science(Associate in Science for Transfer)

**GE AND TRANSFER REQUIREMENTS MET:**

CSU Transfer Course

A. Transfers to CSU  
AS-T Computer Science
COURSE LEVEL STUDENT LEARNING OUTCOME(S) Supported by this course:

1. Diagram the relationship between machine-level architecture & organization and high-level abstractions such as programming languages.
2. Identify the fundamental components, both hardware and software, in the architectural and organizational design of a computer system.
3. Solve problems involving operations of computer arithmetic and identify errors arising from binary representation.
4. Create assembly language segments with the correct data structure.

COURSE OBJECTIVES:

1. Compare and contrast the impact on system performance of varying computer system architectures.
2. Describe the manner in which the architectural and organization components of computer system work, individually and collectively.
3. Diagram the relationship between machine-level architecture & organization and high-level abstractions such as programming languages.
4. Identify the fundamental components, both hardware and software, in the architectural and organizational design of a computer system.
5. Solve problems involving operations of computer arithmetic and identify errors arising from binary representation.
6. Write the software code, mathematical formulas/expressions, and algorithms in the Assembly language.
7. Use different memory addressing modes.
8. Interact with input, output devices via the interrupts.

COURSE CONTENT:

LECTURE CONTENT:

1. Bits, bytes, and words
2. Numeric data representation and number bases
3. Fixed- and floating-point systems
4. Signed and twos-complement representations
5. Representation of nonnumeric data (character codes, graphical data)
6. Representation of records and arrays
7. Basic organization of the von Neumann machine
8. Control unit; instruction fetch, decode, and execution
9. Instruction sets and types (data manipulation, control, I/O)
10. Assembly/machine language programming
11. Instruction formats
12. Addressing modes
13. Subroutine call and return mechanisms
14. I/O and interrupts

METHODS OF INSTRUCTION:

A. Lecture:
B. Dist. Ed – Delayed Interaction:
C. Online:
D. Independent Study:
INSTRUCTIONAL TECHNIQUES:
Lecture and/or discussion for specific content topics using sample code, projector and white board. Hands-on programming and hardware projects during lecture.

COURSE ASSIGNMENTS:
Reading Assignments
Text book and instructor provided handouts.

Out-of-class Assignments
Students will create Assembly solutions for problems assigned in class.

Writing Assignments
Diagrams and reports on procedures for computer components assemble and utilize.

METHODS OF STUDENT EVALUATION:
Midterm Exam
Final Exam
Short Quizzes
Written Assignments
Objective Examinations
Projects (ind/group)
Problem Solving Exercises
Oral Presentations
Skills Demonstration

Demonstration of Critical Thinking:
Students will be demonstrating their projects. Optional research papers and classroom presentations will further demonstrate their ability in critical thinking and problem solving.

Required Writing, Problem Solving, Skills Demonstration:
Students will be required to complete Assembly projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their hardware projects.

TEXTS, READINGS, AND RESOURCES:
TextBooks:

LIBRARY:
Adequate library resources include: Print Materials

Comments:

Attachments:
Attached Files