GAVILAN COLLEGE
5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 28 DIVISION: 50 ALSO LISTED AS:

TERM EFFECTIVE: Spring 2016 CURRICULUM APPROVAL DATE: 10/12/2015

SHORT TITLE: COMPUTER ARCHITECTURE

LONG TITLE: Computer Architecture and Organization

Units	Number of Weeks	Type	Contact Hours/Week	Total Contact Hours
3	18	Lecture:	3	54
		Lab:	0	0
		Other:	0	0
		Total:	3	54

COURSE DESCRIPTION:

Introduction to the organization and architecture of computer systems. 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 with an eye toward possible data representation errors and procedural errors. Throughout the course, students will write short assembly language programs that utilize the concepts being studied. (C-ID: COMP 142) ADVISORY: Some programming experience or programming coursework.

PREREQUISITES:

COREQUISITES:

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES

L - Standard Letter Grade

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:

02 - Lecture and/or discussion
05 - Hybrid
72 - Dist. Ed Internet Delayed

STUDENT LEARNING OUTCOMES:

1. Write simple Assembly language programs.

Measure: programming projects
PLO: 1, 2
ILO: 7, 3
GE-LO:
Year assessed or anticipated year of assessment: 2016

2. Discuss basic strategies and analyze design decisions of computer organization and design.

Measure: homework, exams
PLO:
ILO: 7, 3, 2
GE-LO:
Year assessed or anticipated year of assessment: 2016

3. Demonstrate the process whereby fundamental higher level programming constructs are implemented at the machine language level. Measure: programming projects, homework, exams
PLO: 1, 2
ILO: 7, 3, 2
GE-LO:
Year assessed or anticipated year of assessment: 2016

PROGRAM LEARNING OUTCOMES:
1) Code, debug, document, test, and run programs.
2) Write programs in at least three different programming languages, and compare and contrast the philosophies and comparative advantages of each these languages.
3) Demonstrate professional conduct by meeting project deadlines, and participating in self-managed teams.
4) Create algorithms to solve programming problems, and implement those algorithms.

CONTENT, STUDENT PERFORMANCE OBJECTIVES, OUT-OF-CLASS ASSIGNMENTS
Curriculum Approval Date: 10/12/2015
(6 hours lecture)
Introduction
Computer Subsystems – the Von Neumann model
How the Subsystems Interact
Data Storage Formats
Bits and Groups of Bits
Mathematical Equivalence of Binary and Decimal
Unsigned Decimal to Binary Conversion
Memory — A Place to Store Data (and Other Things)
Using C Programs to Explore Data Formats
Examining Memory With a Debugger
ASCII Character Code
write and read Functions
Exercises: Binary, Decimal, Hexadecimal
(3 hours lecture)
Computer Arithmetic
Addition and Subtraction
Arithmetic Errors — Unsigned Integers

Arithmetic Errors — Signed Integers

Overflow and Signed Decimal Integers

C/C++ Basic Data Types

Other Codes

Exercises: Two's complement, Binary Addition

(3 hours lecture)

Central Processing Unit

CPU Overview

CPU Registers

CPU Interaction with Memory and I/O

Program Execution in the CPU

Using a Debugger to View the CPU Registers

Exercises: Tracing a C program with a Debugger to Inspect the Registers

(6 hours lecture)

Programming in Assembly Language

Creating a New Program

Program Organization

Viewing Both the Assembly Language code and the Source Code

Assemblers and Linkers

Creating a Program in Assembly Language

Instructions Introduced Thus Far

Assembly Language Programming Assignments: Write a short C Program in Assembly Language, Write Assembly Language Functions that Return an Integer or Return a Character

(3 hours lecture)

Program Data – Input, Store, Output

Calling write in -bit Mode

Introduction to the Call Stack

Viewing the Call Stack

Local Variables on the Call Stack

Designing the Local Variable Portion of the Call Stack

Exercises: Use Debugger to Inspect the Stack, Inspect Argument Passing Registers..

Using syscall to Perform I/O

Calling Functions, -Bit Mode

Instructions Introduced Thus Far

Exercises: Read and Write Instructions

Modify a given so that the stack grows from lower numbered array elements to higher numbered ones. Write a program in assembly language that prompts the user to enter an integer, then displays its hexadecimal equivalent

(3 hours lecture)

Computer Operations

The Assignment Operator

Addition and Subtraction Operators

Introduction to Machine Code

Instructions Introduced Thus Far

Assembly Language Programming Assignments: Assemble mov and add instructions learned so far by hand (on paper). Check your work by entering and running the code. Disassembling a code snippet.

(3 hours lecture)

Program Flow Constructs

Repetition

Binary Decisions

Instructions Introduced Thus Far

Assembly Language Programming Assignments: Write a program in assembly language that displays all the printable characters that are neither numerals nor letters on the screen, one character at a time. Write a program in assembly language that prompts the user to enter a text string, reads the user's input into a char array, echoes the user's input string, modifies the character in a specified way, and displays the modified string.

(3 hours lecture)

Writing Your Own Functions

Overview of Passing Arguments

More Than Six Arguments, -Bit Mode

Interface Between Functions, -Bit Mode

Instructions Introduced Thus Far

Assembly Language Programming Assignments: writeStr, ReadLn in Assembly Language, examining different levels of compiler optimization.

(3 hours lecture)

Bit Operations; Multiplication and Division

Logical Operators

Shifting Bits

Multiplication

Division

Negating Signed ints

Instructions Introduced Thus Far

Assembly Language Programming Assignments:

Write a program in assembly language that allows the user to enter a decimal integer then displays it in binary. Write a function, mul16, in assembly language that takes two 16-bit integers as arguments and returns the 32-bit product of the argument.

(3 hours lecture)

Data Structures

Arrays

structs (Records)

structs as Function Arguments

Structs as C++ Objects

Instructions Introduced Thus Far

Assembly Language Programming Assignments: Write a program that allows the user to maintain an address book. Write a program that allows the user to set up and maintain two bank accounts. Each account should have a unique account name. The user should be able to credit or debit the account.

(6 hours lecture)

Fractional Numbers

Fractions in Binary

Fixed Point ints

Floating Point Format

IEEE

Floating Point Hardware

Comments About Numerical Accuracy

Instructions Introduced Thus Far

Assembly Language Programming Assignments

(6 hours lecture)

Interrupts and Exceptions

Hardware Interrupts

Exceptions
Software Interrupts
CPU Response to an Interrupt or Exception
Return from Interrupt/Exception
The syscall and sysret Instructions
Summary
Instructions Introduced Thus Far
Assembly Language Programming Assignments
(3 hours lecture)
Input/Output
Memory Timing
I/O Device Timing
Bus Timing
I/O Interfacing
I/O Ports
Programming Issues
Interrupt-Driven I/O
I/O Instructions
Assembly Language Programming Assignments
(1 hour) Review
(2 hours) Final Exam

METHODS OF INSTRUCTION:
Lecture, guided discovery, video tutorials, demonstration.

METHODS OF EVALUATION:
Category 1 - The types of writing assignments required:
Percent range of total grade: 0 % to 20 %
Written Homework

If this is a degree applicable course, but substantial writing assignments are NOT appropriate, indicate reason:
Course is primarily computational
Course primarily involves skill demonstration or problem solving

Category 2 - The problem-solving assignments required:
Percent range of total grade: 20 % to 90 %
Homework Problems
Quizzes

Exams

Category 3 - The types of skill demonstrations required:
Percent range of total grade: % to %

Category 4 - The types of objective examinations used in the course:

Percent range of total grade: 0 % to 10 %
Multiple Choice
True/False
Matching Items
Completion

REPRESENTATIVE TEXTBOOKS:

Required:

Robert G. Plantz. Introduction to Computer Organization with x86-64 Assembly Language & GNU/Linux. http://bob.cs.sonoma.edu: lulu.com, May 2015. Or other appropriate college level text.

Reading level of text, Grade: 12+ Verified by: ev - MS Word

Other textbooks or materials to be purchased by the student: none

ARTICULATION and CERTIFICATE INFORMATION

Associate Degree:

CSU GE:

IGETC:

CSU TRANSFER:

Transferable CSU, effective 201570

UC TRANSFER:

Not Transferable

SUPPLEMENTAL DATA:

Basic Skills: N
Classification: Y
Noncredit Category: Y
Cooperative Education:
Program Status: 1 Program Applicable
Special Class Status: N
CAN:
CAN Sequence:
CSU Crosswalk Course Department: CSIS
CSU Crosswalk Course Number: 28
Prior to College Level: Y
Non Credit Enhanced Funding: N
Funding Agency Code: Y
In-Service: N
Occupational Course: C
Maximum Hours: 3
Minimum Hours: 3
Course Control Number: CCC000559304
Sports/Physical Education Course: N
Taxonomy of Program: 070600