Cerro Coso College

Course Outline of Record Report

10/13/2021

CSCIC257 : Computer Architecture and Organization

General Information

Author:	-
Course Code (CB01) :	CSCIC257
Course Title (CB02) :	Computer Architecture and Organization
Department:	Business Information Technolog
Proposal Start:	Fall 2013
TOP Code (CB03) :	(0706.00) Computer Science (transfer)
SAM Code (CB09) :	Non-occupational
Distance Education Approved:	Yes
Course Control Number (CB00) :	CCC000547180
Curriculum Committee Approval Date:	05/03/2013
Board of Trustees Approval Date:	06/13/2013
External Review Approval Date:	07/18/2013
Course Description:	This course covers basic hardware and software structure; I/O and main memory organization; internal representation of data; addressing methods; program control; microprocessors and multiprocessors, and RISC architectures. There is some assembly language programming.
Submission Type:	New Course
Author:	No value

Faculty Minimum Qualifications

Master Discipline Preferred:	• Computer Science
Alternate Master Discipline Preferred:	No value
Bachelors or Associates Discipline Preferred:	No value
Additional Bachelors or Associates Discipline Preferred:	No value

Course Development Options

Basic Skills Status (CB08)

Course is not a basic skills course.

Course Special Class Status (CB13)

Course is not a special class.

Grade Options

- Letter Grade Methods
- Pass/No Pass

☐ Allow Students to Gain Credit by Exam/Challenge

Allowed Number of Retakes

0

Course Prior To College Level (CB21)

Not applicable.

Rationale For Credit By Exam/Challenge

Retake Policy Description

☑ Allow Students To Audit Course

No value

Type:|Non-Repeatable Credit

Course Support Course Status (CB26)

No value

Associated Programs

☐ Course is part of a program (CB24)

Associated Program	Award Type	Active
No value	No value	

Transferability & Gen. Ed. Options

Course General Education Status (CB25)

No value

Transferability	Transferability Status
Transferable to both UC and CSU	Approved

Units and Hours:

Summary

Minimum Credit Units (CB07)	3
Maximum Credit Units (CB06)	3
Total Course In-Class (Contact) Hours	54
Total Course Out-of-Class Hours	108
Total Student Learning Hours	162
Faculty Load	0

Credit / Non-Credit Options

Course Credit Status (CB04)	Course Non Credit Category (CB22)	Non-Credit Characteristic
Credit - Degree Applicable	Credit Course.	No Value

Course Classification Status (CB11)	Funding Agency Category (CB23)	
Credit Course.	Not Applicable.	☐ Cooperative Work Experience Education Status (CB10)

☐ Variable Credit Course

Weekly Student Hours

	In Class	Out of Classs
Lecture Hours	3	6
Laboratory Hours	0	0
Activity Hours	0	0

Course Student Hours

Course Duration (Weeks)	18
Hours per unit divisor	0
Course In-Class (Contact) Hours	
Lecture	0
Laboratory	0
Activity	0
Total	54
Course Out-of-Class Hours	
Lecture	0
Laboratory	0
Activity	0
Total	108

Time Commitment Notes for Students

No value

Faculty Load

Extra Duties: 0

Faculty Load: 0

Units and Hours: - Weekly Specialty Hours

Activity Name	Type	In Class	Out of Class
No Value	No Value	No Value	No Value

Pre-requisites, Co-requisites, Anti-requisites and Advisories

Prerequisite

CSCIC252 - Introduction to Computer Science

To be successful in this course, students need to have basic programming skills and basic computer architecture knowledge such as logic gates. This is from the CSCI C252 course.

Entrance Skills

Entrance Skills	Description
No value	No value

Limitations on Enrollment

Limitations on Enrollment	Description
No value	No value

Specifications

Methods of Instruction

Methods of Instruction	Skills Development and Performance
Rationale	No value
Methods of Instruction	Problem Solving
Rationale	No value
Methods of Instruction	Project-based learning
Rationale	No value
Methods of Instruction	Laboratory
Rationale	No value
Methods of Instruction	Lecture
Rationale	No value
Methods of Instruction	Discussion
Rationale	No value

Methods of Instruction	Demonstration
Rationale	No value

Methods of Instruction	Computational Work
Rationale	No value

Assignments

A. Reading Text - Preparing for class by reading the chapters assigned
B. Programming Assignments - Assembly language assignments to complete
C. Homework Assignments - Problem sets as handouts or from the text to practice concepts, preparing for presenting topics on the course
D. Group work - Group work time for a group project

Methods of Evaluation	Rationale
Homework	Programming assignments demonstrating student's ability to design a application using an assembly language. Example: Create a program that uses addition to multiply to numbers
Research Paper	Essays demonstrating students' understanding of system performance. Example: Write a paper on specific system performance topic. Discuss how the performance can be improved or made worse.
Tests	Objective tests/quizzes demonstrating student's knowledge of computer components and organization. Example: Draw the computer components and how they interact.

Equipment

No Value

Textbooks

Author	Title	Publisher	Date	ISBN
	Tanenbaum, A. S., Austin, T.. (2013) Structured Computer Organization, , Pearson			

Other Instructional Materials

No Value

Materials Fee

No

Learning Outcomes and Objectives

Course Objectives

No value

CSLOs

Compare and contrast the impact on system performance of varying computer system architectures. Expected SLO Performance: 70.0

Identify the fundamental components in the architecture and organization of a computer system. Expected SLO Performance: 70.0

Describe the manner in which the architectural and organization components of computer system work, individually and collectively. Expected SLO Performance: 70.0

Create a program using assembly language. Expected SLO Performance: 70.0

Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level. Expected SLO Performance: 70.0

Outline

Course Outline

A. Overview of computer organization and architecture
a. Reasons; strategies; strengths and weaknesses
b. Inherent in different computing architectures
B. Digital logic and digital systems
a. Fundamental building blocks
i. Logic gates
ii. Flip-flops
iii. Registers
b. Logic expressions; minimization; sum of product forms.
c. Physical considerations
C. Machine level representation of data
a. Bits; bytes and words
b. Numeric data representation and number bases
c. Fixed- and floating-point systems
d. Signed and twos-complement representations
e. Represenation of nonnumeric data
f. Representation of records and arrays
g. Limitations of computer arithmetic
D. Computer Architecture
a. Von Neumann architecture
b. Control unit
c. Instruction sets and types
d. Machine level operation
e. High-level programming constructs at the machine language level
f. Assembly language programming
E. Memory system organization and architecture
a. Types of memory technology
b. Memory latency and bandwidth
c. Memory hierarchy
d. Memory management and caching
e. Errors in memory systems

f. Coding; data compression and data integrity
g. Virtual memory
F. Interfacing and communication
a. Interrupts for I/O control and data transfers
b. I/O drivers using assembly language
c. Buses in computer system
d. Data access for storage devices
e. Differences in operating systems
f. Buses
i. Bus protocols
ii. Arbitration
iii. Direct-memory access
g. RAID architecture
G. Functional Organization
a. Implementation of simple datapaths
b. Control unit
c. Instruction pipelining
d. Introduction to instruction level parallelism
H. Multiprocessing and alternative architecture
a. Introduction to SIMD; MIMD; VLIW; EPIC
b. Systolic architecture
c. Interconnection networks
d. Shared memory systems
e. Cache coherence
f. Memory models and memory consistency
A. Digital logic and digital systems
a. Fundamental building blocks
i. Logic gates
ii. Flip-flops
iii. Registers
b. Logic expressions; minimization; sum of product forms.
c. Physical considerations
B. Machine level representation of data
a. Bits; bytes and words
b. Numeric data representation and number bases
c. Fixed- and floating-point systems
d. Signed and twos-complement representations
e. Representation of nonnumeric data
f. Representation of records and arrays
C. Computer Architecture
a. Assembly language programming
i. Opcodes
ii. Storing Data
iii. Mathematical operations
iv. Decisions
v. Looping
vi. Functions
D. Interfacing and communication
a. Interrupts for I/O control and data transfers
b. I/O drivers using assembly language
E. Functional Organization
a. Instruction pipelining

Delivery Methods and Distance Education

Delivery Method: Please list all that apply -Face to face -Online (purely online no face-to-face contact) -Online with some required face-to-face meetings (“Hybrid”) -Online course with on ground testing -iTV – Interactive video = Face to face course with significant required activities in a distance modality -Other

Face 2 Face
Online

Hybrid
Interactive

Rigor Statement: Assignments and evaluations should be of the same rigor as those used in the on-ground course. If they are not the same as those noted in the COR on the Methods of Evaluation and out-of-class assignments pages, indicate what the differences are and why they are being used. For instance, if labs, field trips, or site visits are required in the face to face section of this course, how will these requirements be met with the same rigor in the Distance Education section?

No Value

Effective Student-Instructor Contact: Good practice requires both asynchronous and synchronous contact for effective contact. List the methods expected of all instructors teaching the course. -Learning Management System -Discussion Forums -Moodle Message -Other Contact -Chat/Instant Messaging -E-mail -Face-to-face meeting(s) -Newsgroup/Discussion Board -Proctored Exam -Telephone -iTV -Interactive Video -Other (specify)

contact_moodle_forums
contact_moodle_message
contact_email
contact_phone
contact_itv
contact_other

Software and Equipment: What additional software or hardware, if any, is required for this course purely because of its delivery mode? How is technical support to be provided?

No Value

Accessibility: Section 508 of the Rehabilitation Act requires access to the Federal government's electronic and information technology. The law covers all types of electronic and information technology in the Federal sector and is not limited to assistive technologies used by people with disabilities. It applies to all Federal agencies when they develop, procure, maintain, or use such technology. Federal agencies must ensure that this technology is accessible to employees and the public to the extent it does not pose an "undue burden". I am using -iTV—Interactive Video only -Learning management system -Publisher course with learning management system interface.

s508_itv
s508_moodle
s508_publisher

Class Size: Good practice is that section size should be no greater in distance ed modes than in regular face-to-face versions of the course. Will the recommended section size be lower than in on-ground sections? If so, explain why.

No Value