COLLEGE OF THE DESERT

Course Code CS-008

Course Outline of Record

1. Course Code: CS-008
2. a. Long Course Title: Computer Architecture and Organization
   b. Short Course Title: COMPUTER ARCH
3. a. Catalog Course Description:
   The organization and behavior of computer systems at the assembly-language level. The translation of statements and constructs in a high-level language into 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. Digital electronics with the Boolean algebra of logic gates is studied.
   b. Class Schedule Course Description:
   The organization and behavior of real computer systems and data structures at the assembly-language level. Digital electronics and numerical computation is examined.
   c. Semester Cycle (*if applicable*): *N/A*
   d. Name of Approved Program(s):
   - COMPUTER SCIENCE AS Degree and Transfer Preparation
4. Total Units: 3.00 Total Semester Hrs: 90.00
   Lecture Units: 2 Semester Lecture Hrs: 36.00
   Lab Units: 1 Semester Lab Hrs: 54.00
   Class Size Maximum: 32 Allow Audit: No
   Repeatability No Repeats Allowed
   Justification 0
5. Prerequisite or Corequisite Courses or Advisories:
   Course with requisite(s) and/or advisory is required to complete Content Review Matrix (CCForm1-A)
   Prerequisite: CS 007A with a minimum grade of C
6. Textbooks, Required Reading or Software: *(List in APA or MLA format.)*
   a. Plantz, Robert (2012). *Introduction to Computer Organization with x86-64 Assembly Language and GNU/Linux* (1/e). Lulu.
   College Level: Yes
   Flesch-Kincaid reading level: 12
7. Entrance Skills: *Before entering the course students must be able:*

a.

Locate and operate the basic components in the structure of a computer system

- CS 007A - Locate and operate the basic components in the structure of a computer system.

b.

Demonstrate the grammar, punctuation, and vocabulary of a programming language by composing original programs

- CS 007A - Demonstrate the grammar, punctuation, and vocabulary of a programming language by composing original programs.

c.

Demonstrate good program design principles to do problem solving, creating programs that are correct, easy to write, read, modify, and repair

- CS 007A - Demonstrate good program design principles to do problem solving, creating programs that are correct, easy to write, read, modify, and repair.

d.

Describe the sequence of steps to go through, in designing, writing, testing and debugging a program

- CS 007A - Describe the sequence of steps to go through, in designing, writing, testing and debugging a program.

8. Course Content and Scope:

Lecture:

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. Logic Gates and Logic Circuits
9. Control unit; instruction fetch, decode, and execution
10. Instruction sets and types (data manipulation, control, I/O)
11. Assembly/machine language programming
12. Instruction formats
13. Addressing modes
14. Subroutine call and return mechanisms
15. I/O and interrupts
16. History of computers and computer software

Lab: *(if the "Lab Hours" is greater than zero this is required)*

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. Logic Gates and Logic Circuits
9. Control unit; instruction fetch, decode, and execution
10. Instruction sets and types (data manipulation, control, I/O)
11. Assembly/machine language programming
12. Instruction formats
13. Addressing modes
14. Subroutine call and return mechanisms
15. I/O and interrupts

9. Course Student Learning Outcomes:

1.

Describe the low-level storage of data, including the concepts of bits, bytes, words, fixed and floating point systems, signed and twos-complement representations, representation of non-numeric data, and representations of records and arrays.

2.

Discuss the history of computers and computer software

3.

Discuss the low-level architecture of computers, including the von-Neumann machine; components and data flow within the CPU; data flow to memory, registers and I/O devices; and Instruction Set Architecture

4.
Write simple assembly language program segments

5.
Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level

6.
Discuss the basics of operating systems design

10. Course Objectives: *Upon completion of this course, students will be able to:*
a. Describe Bits, bytes, and words
b. Discuss numeric data representation and calculate in different number bases
c. Explain Fixed- and floating-point systems
d. Determine representations of numbers in signed and twos-complement representations
e. Describe representation of nonnumeric data (character codes, graphical data)
f. Describe representation of records and arrays
g. Discuss basic organization of the von Neumann machine
h. Describe the different Logic Gates and analyze Logic Circuits
i. Describe the control unit; instruction fetch, decode, and execution
j. Discuss instruction sets and types (data manipulation, control, I/O)
k. Build and debug assembly/machine language programming
l. Discuss different instruction formats and Addressing modes
m. Describe subroutine calls and return mechanisms, I/O and interrupts
n. Discuss the basic history of computers and computer software

11. Methods of Instruction: *(Integration: Elements should validate parallel course outline elements)*
a. Activity
b. Lecture
c. Technology-based instruction

12. Assignments: *(List samples of specific activities/assignments students are expected to complete both in and outside of class.)*
In Class Hours: 108.00
Outside Class Hours: 108.00
a. In-class Assignments
Programming Projects
Simple digital electronic circuits
b. Out-of-class Assignments
Written assignments
Programming Projects

13. Methods of Evaluating Student Progress: *The student will demonstrate proficiency by:*
- Written homework
  Programming Projects
- Laboratory projects
  In-class Programming Projects
- Mid-term and final evaluations

14. Methods of Evaluating: Additional Assessment Information:

15. Need/Purpose/Rationale -- *All courses must meet one or more CCC missions.*

PO-GE C4.b - Language & Rationality (Communication & Analytical Thinking)

Raise questions and problems, formulating them clearly and precisely.

Express solutions to complex problems using language and logic.

Apply logical and critical thinking to solve problems; explain conclusions; and evaluate, support, or critique the thinking of others.

PO-BS Critical Thinking

Assess relevant information and come to thought-out conclusions and solutions.

Communicate meaningfully with others.

PO-BS Problem Solving

Use a variety of solution methods and techniques, for example, making a sketch, systematic listing, using the solution of a simpler (but related) problem.

Recognize the importance of checking a proposed solution to verify that it satisfies the requirements of a problem.

Recognize that a solution may not be possible, given limits of time, money, or other finite resources.

Restate (formulate) a problem mathematically.

Use background information in a subject to understand the nature of a problem and transfer information to new problems accordingly.

Identify what isn’t known, but needs to be known in order to solve a problem (depending on the problem domain, reading and/or mathematical skills are helpful).

IO - Scientific Inquiry

Analyze quantitative and qualitative information to make decisions, judgments, and pose questions.

IO - Critical Thinking and Communication

Apply principles of logic to problem solve and reason with a fair and open mind.

Apply standard conventions in grammar, mechanics, usage and punctuation.

16. Comparable Transfer Course

University System	Campus	Course Number	Course Title	Catalog Year
CSU	California Polytechnic University, San Luis Obispo	CSC 225	Introduction to Computer Organization	2013
CSU	California Polytechnic University, Pomona	CS 264	Computer Organization and Assembly Programming	2013
CSU	CSU Fullerton	CPSC 240	Computer Organization and Assembly Language	2013

17. Special Materials and/or Equipment Required of Students:

18. Materials Fees: □ Required Material?

Material or Item	Cost Per Unit	Total Cost

19. Provide Reasons for the Substantial Modifications or New Course:

ADT requirement

20. a. Cross-Listed Course *(Enter Course Code)*: *N/A*

b. Replacement Course *(Enter original Course Code)*: *N/A*

21. Grading Method *(choose one)*: Letter Grade Only

22. MIS Course Data Elements

a. Course Control Number [CB00]: CCC000578623

b. T.O.P. Code [CB03]: 70600.00 - Computer Science (Transfe
c. Credit Status [CB04]: D - Credit - Degree Applicable
d. Course Transfer Status [CB05]: A = Transfer to UC, CSU
e. Basic Skills Status [CB08]: 2N = Not basic skills course
f. Vocational Status [CB09]: Not Occupational
g. Course Classification [CB11]: Y - Credit Course
h. Special Class Status [CB13]: N - Not Special
i. Course CAN Code [CB14]: *N/A*
j. Course Prior to College Level [CB21]: Y = Not Applicable
k. Course Noncredit Category [CB22]: Y - Not Applicable
l. Funding Agency Category [CB23]: Y = Not Applicable
m. Program Status [CB24]: 1 = Program Applicable

Name of Approved Program *(if program-applicable)*: *N/A*
Attach listings of Degree and/or Certificate Programs showing this course as a required or a restricted elective.)

23. Enrollment - Estimate Enrollment
First Year: 30
Third Year: 40

24. Resources - Faculty - Discipline and Other Qualifications:
a. Sufficient Faculty Resources: Yes
b. If No, list number of FTE needed to offer this course: *N/A*

25. Additional Equipment and/or Supplies Needed and Source of Funding.
N/A

26. Additional Construction or Modification of Existing Classroom Space Needed. *(Explain:)*
N/A

27. FOR NEW OR SUBSTANTIALLY MODIFIED COURSES
Library and/or Learning Resources Present in the Collection are Sufficient to Meet the Need of the Students Enrolled in the Course: Yes

28. Originator Geoffrey Hagopian Origination Date 09/05/16