# BUTTE COLLEGE
## COURSE OUTLINE

I. **CATALOG DESCRIPTION**

**CSCI 10 - Computer Architecture and Organization** **3 Unit(s)**

**Prerequisite(s):** CSCI 20
**Recommended Prep:** NONE
**Transfer Status:** CSU/UC
34 hours Lecture
51 hours Lab

This course is an introduction to the organization and behavior of modern computer systems at the assembly language level. Topics include numerical computation, the internal representation of simple data types and structures, data representation errors, and procedural errors. Students will learn how to map statements and constructs of high-level languages onto sequences of machine instructions. (C-ID COMP 142).

II. **OBJECTIVES**

Upon successful completion of this course, the student will be able to:

A. Identify and describe the organization of modern computer systems, including processor architecture and behavior and the representation of data types and data structures.
B. Write assembly language instructions that map fundamental high-level language constructs onto machine language.
C. Write simple assembly language program segments, including subroutines, to perform Input/Output (I/O), arithmetic, logic, and other basic operations.

III. **COURSE CONTENT**

A. **Unit Titles/Suggested Time Schedule**

Lecture

| Topics | | Hours |
|---|---|---|
| 1. | Bits, bytes, and words | 2.00 |
| 2. | Numeric data representation and number bases | 3.00 |
| 3. | Fixed and floating point systems | 2.00 |
| 4. | Signed and two's complement representations | 2.00 |
| 5. | Representation of non-numeric data | 2.00 |
| 6. | Representation of records and arrays | 3.00 |
| 7. | Basic organization of the von Neumann architecture | 2.00 |
| 8. | Control Unit and the fetch-decode-execute cycle | 1.00 |
| 9. | Instruction sets and instruction types | 2.00 |
| 10. | Assembly language programming | 6.00 |
| 11. | Instruction formats | 1.00 |
| 12. | Addressing modes | 4.00 |
| 13. | Subroutine call and return mechanisms | 2.00 |
| 14. | I/O and interrupts | 2.00 |
| Total Hours | | 34.00 |

Lab

| Topics | | Hours |
|---|---|---|
| 1. | Bits, bytes, and words | 3.00 |
| 2. | Numeric data representation and number bases | 3.00 |
| 3. | Fixed and floating point systems | 3.00 |
| 4. | Representation of non-numeric data | 3.00 |
| 5. | Representation of records and arrays | 6.00 |
| 6. | Assembly language programming | 18.00 |
| 7. | Addressing modes | 6.00 |
| 8. | Subroutine call and return mechanisms | 3.00 |
| 9. | I/O and interrupts | 6.00 |
| Total Hours | | 51.00 |

## IV. **METHODS OF INSTRUCTION**

A. Lecture
B. Collaborative Group Work
C. Homework: Students are required to complete two hours of outside-of-class homework for each hour of lecture
D. Demonstrations
E. Multimedia Presentations

## V. **METHODS OF EVALUATION**

A. Quizzes
B. Homework
C. Lab Projects
D. Mid-term and final examinations

## VI. **EXAMPLES OF ASSIGNMENTS**

A. Reading Assignments

1. Read the chapter in your text on arithmetic with unsigned integers. Write a brief description of the condition(s) that would cause an error when adding two unsigned integer values. Provide two examples of unsigned integer values that, when added together, would cause an error.
2. Read the chapter about instruction format in the "Intel 64 and IA-32 Architectures Software Developer's Manual," which can be found online. Draw a simple diagram showing the instruction code format for Intel 64 and IA-32 Architectures.

B. Writing Assignments

1. Write a short assembly language program that reads in an unsigned 32-bit integer from user input, then displays the input value in binary, decimal, and hexadecimal formats. Then, test the input value to see if it is a prime number, and output a string ("PRIME" or "NOT PRIME") based on the result. Be sure to include proper header documentation and thorough inline commenting to describe your algorithm.
2. Write truth tables for the bitwise logic instructions AND, OR, XOR, and NOT. Include an example with each table and a description of one possible use of each instruction in a program.

C. Out-of-Class Assignments

1. Use the Internet to find information about the von Neumann and Harvard processor architectures. Write a brief summary of the fundamental components of both architectures.

What is the key difference between the two architectures?

2. Prepare a short programming exercise to be completed by one of your classmates. Your exercise must require that your classmate use at least two different addressing modes, and should take no more than 150 lines of code to complete.

## VII. RECOMMENDED MATERIALS OF INSTRUCTION

Textbooks:

A. Hyde, Randall. The Art of Assembly Language. 2nd Edition. No Starch Press, 2010.

B. Plantz, Robert G. Introduction to Computer Organization with x86-64 Assembly Language & GNU/Linux. 2015 Edition. Robert G. Plantz, 2015.

**Created/Revised by:** John Trolinger
**Date:** 03/07/2016