CIC Approval: 12/13/2018
BOT APPROVAL:
STATE APPROVAL:
EFFECTIVE TERM: Fall 2020

SAN DIEGO COMMUNITY COLLEGE DISTRICT
MESA, AND MIRAMAR COLLEGES
ASSOCIATE DEGREE COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER: Computer and Information Sciences 211

COURSE TITLE:	**Units:**
Computer Organization and Assembly Language	4 Grade Only

CATALOG COURSE DESCRIPTION:

This course is an introduction to the organization of modern digital computers and assembly language programming. Topics include language syntax; instruction set mnemonics; and segment, index, pointer, general purpose, and flag registers. A variety of memory addressing techniques are covered, as well as stack operations, particularly those associated with passing parameters to subroutine calls. Additional topics include machine architecture; memory addressing; input/output; interrupts; control structures; compiling; linking; and printer and disk interfaces. This course is intended for students majoring in computer and information sciences.

REQUISITES:

Advisory:
ENGL 047A with a grade of "C" or better, or equivalent or Milestone R50/W50
or
ENGL 048 with a grade of "C" or better, or equivalent or Milestone R50
&
ENGL 049 with a grade of "C" or better, or equivalent or Milestone W50
and
MATH 038 with a grade of "C" or better, or equivalent or Milestone M30

FIELD TRIP REQUIREMENTS:
May be required

TRANSFER APPLICABILITY:
Associate Degree Credit & transfer to CSU UC Transfer Course List

CID:
COMP 142

TOTAL LECTURE HOURS:
48 - 54

TOTAL LAB HOURS:
48 - 54

TOTAL CONTACT HOURS:
96 - 108

OUTSIDE-OF-CLASS HOURS:

96 - 108

TOTAL STUDENT LEARNING HOURS:
192 - 216

STUDENT LEARNING OBJECTIVES:
Upon successful completion of the course the student will be able to:

1. Describe the organization and architecture of a computer system.
2. Employ the structure and syntax of assembly language in programming.
3. Differentiate and utilize the hexadecimal, binary, and octal number systems.
4. Apply binary logic operators to assembly language programs.
5. Utilize memory management and memory addressing methods.
6. Use the Disk Operating System (DOS) and Basic Input/Output System (BIOS) services to interface with devices such as the keyboard, disk drive, monitor, and printer.
7. Write and test assembly language programs to perform a variety of tasks, including storing and retrieving data from memory; accepting keyboard input and displaying a message to the screen; implementing a system and interrupt; reading or writing data to a register; implementing a loop; and processing a data array.
8. Compile and link an assembly language program into executable code.

SECTION II

1. COURSE OUTLINE AND SCOPE:

A. **Outline Of Topics:**
The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

I. Data representation
  A. Numbering systems and bases
    1. Binary
    2. Decimal
    3. Hexadecimal
  B. Data organization
    1. Bits
    2. Bytes
    3. Words
    4. Double words
  C. Logical operations
    1. AND
    2. OR
    3. XOR
    4. NOT
  D. Bitwise operations
  E. Signed numbering system (two-s complement)
  F. Packed data, shifts, and rotates
II. Boolean algebra
  A. Truth tables and boolean expressions
  B. Canonical forms
  C. Simplification
  D. Combinatorial and sequential circuits
III. System organization
  A. System components
    1. Central Processing Unit (CPU)
    2. Memory
    3. Bus
  B. System timing, memory access, wait states, and cache memory
  C. Piplines, superscalar operation, and stalls/hazards

IV. Memory organization
   A. Introduction to the 80x86 CPU
   B. 80x86 physical memory
   C. 80x86 segments
   D. 80x86 addressing modes
V. Variables and simple data structures in memory
   A. Declaring variables in an assembly language program
   B. Byte, word, dword, and other variable sizes
   C. Creating data types
   D. Pointers
   E. Composite data types
      1. Arrays
      2. Structures
      3. Pointers
   F. Accessing elements of arrays: row/column major ordering
   G. Dereferencing pointers
VI. The 80x86 instruction set
VII. Microsoft Assembler (MASM)=s directives, pseudo opcodes, macros, and other facilities
VIII. Arithmetic and logical operations
   A. Converting high-level languages (HLL) arithmetic expressions to assembly language
   B. Logical (boolean) expressions
   C. Multiprecision operations
   D. Masking operations
IX. Control structures
   A. Simulating control structures in assembly language
      1. IF...THEN...ELSE
      2. CASE
      3. WHILE
      4. REPEAT...UNTIL
      5. LOOP...ENDLOOP
      6. FOR
   B. Efficiency considerations
X. Procedures and functions
   A. Near and far procedures
   B. Passign parameters
   C. By value
   D. By reference
   E. By /result
   F. By result
   G. By name
   H. By lazy evaluation
   I. Passing parameters in registers
   J. In global variables
   K. On the stack
   L. In the code stream
   M. In a parameter block
   N. Returning function results
   O. Local (automatic) variables
   P. Recursion

B. **Reading Assignments:**
Reading assignments are required and may include, but are not limited to, the following:

I. Course textbook(s).
II. Articles from professional or trade journals such as Code Magazine or Science of Computer Programming.
III. Instructor handouts.
IV. Articles and guides on websites related to course material.

C. **Writing Assignments:**

Writing assignments are required and may include, but are not limited to, the following:

I. Written assembly language programs.
II. Program testing procedures.
III. Program operation narratives that demonstrate data flow.

D. **Appropriate Outside Assignments:**
Outside assignments may include, but are not limited to, the following:

I. Completing reading and writing assignments.
II. Evaluating and analyzing assembly language programming requirements.
III. Creating assembly language programs.

E. **Appropriate Assignments that Demonstrate Critical Thinking:**
Critical thinking assignments are required and may include, but are not limited to, the following:

I. Writing and testing an assembly language program.
II. Assessing computer applications to deductively and logically debug and correct errors.
III. Recommending programming improvements.

2. METHODS OF EVALUATION:

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

I. Applied in-class and out of class assignments.
II. In-class objective and/or essay question quizzes and/or examinations.
III. Written programs using a specific programming language.
IV. One-on-one program testing and operations.
V. Class discussion and participation.
VI. Program documentation.

3. METHODS OF INSTRUCTION:

Methods of instruction may include, but are not limited to, the following:

* Audio-Visual
* Collaborative Learning
* Distance Education (Fully online)
* Lecture-Lab Combination
* Other (Specify)
* A. In-class, applied computer practice of concepts and techniques included in course objectives.
* B. Interactive group activities including analysis, evaluation, and modification of programs.

4. REQUIRED TEXTS AND SUPPLIES:

Textbooks may include, but are not limited to:

TEXTBOOKS:
1. Harris, Sarah L. and David Money Harris. Digital Design and Computer Architecture: ARM Edition, 1st ed. Morgan Kaufmann, 2016, ISBN: 9780128000564
2. Hohl, William and Christopher Hinds. ARM Assembly Language: Fundamentals and Techniques, 2nd ed. CRC Press, 2015, ISBN: 9781482229851
3. Irvine, Kip R. Assembly Language for x86 Processors, 7th ed. Pearson, 2014, ISBN: 9780133769401
4. Paul, Richard. SPARC Architecture, Assembly Language Programming, and C, 2nd ed. Pearson, 1999, ISBN: 9780130255969
5. Stallings, William. Computer Organization and Architecture, 10th ed. Pearson, 2016, ISBN: 9780134101613

MANUALS:
1. Plantz, Robert G. Introduction to Computer Organization with x86-64 Assembly Language & GNU/Linux, Lulu or accessible online at bob.cs.sonoma.edu, 07-13-2016

PERIODICALS:

SOFTWARE:

SUPPLIES:

ORIGINATOR: Tasha Frankie

CO-CONTRIBUTOR(S)
DATE: 10/11/2018