

COURSE DESCRIPTION

Course No.	CSC261	Course Title	Computer Organization and Assembly Language
Course Type	Required		
Sem. Hours	3	Coordinator	Mark Armstrong

Current Catalog Description:

Introduction to the organization of the computer through the use of Assembly Language programming. Data representation, parts of the computer system, Assembly Language fundamentals, instruction sets, memory, and floating-point operations.

Textbook:

Introduction to 80x86 Assembly Language and Computer Architecture, Richard C. Detmer, Jones and Bartlett, 2010.

References:

Logic and Computer Design Fundamentals, M. Morris Mano and Charles R. Kime, Pearson, 2008.

Computer Organization and Design, David A. Patterson and John L. Hennessy, Morgan Kaufman, 2014.

Course Outcomes:

Upon successful completion of this course, a student should be able to:

1. devise and use multiple levels of abstraction in computation, including the use of models and simulations (abstraction)
2. analyze data representation through decimal base conversions, decimal and floating-point data storage and manipulation (data)
3. create and evaluate Assembly Language programs to implement algorithms utilizing elementary data structures (programming)
4. design an elementary RISC architecture utilizing both combinational and sequential circuits (design)

Activities Enabling Program Outcomes (POx refers to program student outcome x)

Instruction: The fundamental areas of this course explore computer organization through instruction in assembly language programming and computer organization and design. Students are first introduced to different numbering systems, including examples of where these systems are utilized within the system. Students explore conversions between bases and ASCII representation of characters (POa). Instruction then proceeds to basic Intel Assembly Language programming including basic syntax, control structures, elementary data structures, procedures and macros, logical instructions, and string operations. Several programs are created to demonstrate each concept (POa,b,c,i,j).

Instruction then switches to basic computer organization and design utilizing a RISC architecture (currently SPARC) by comparing and contrasting the Intel and SPARC architecture. Students are introduced to basic circuits through study of Boolean algebra including minterm canonical forms, optimizations including Boolean algebra properties and Karnaugh maps, and logic gates (POa). Combinational and sequential circuit design, including programmable logic devices, error detection and correction, is then introduced. Several examples of circuit implementations are demonstrated including decoders, multiplexers, counters, latches and flip-flops, memory, and sequence recognizers (POb, c, j). Finally datapath and controls for the RISC architecture are introduced and demonstrated (POa).

Student Activities and Assessment: This course has no activities identified for data collection in program outcome assessment.

Prerequisites by Topic:

Students must have a grade of at least C (2.0) in CSC 230 (Elementary Data Structures) and CSC 250 (Foundations of Computer Science I), or permission of instructor.

Major Topics Covered in the Course:

- Data Representation
- The Computer System
- Elements of Assembly Language
- Input/Output
- Branches
- Loops
- Arrays
- Procedures
- Bit Manipulation
- String Operations
- Floating Point Operations
- Boolean Algebra and Gates
- Combinational Devices
- Arithmetic Functions and Circuits
- Sequential Circuit Analysis & Design
- Registers, Counters, and Memory
- Programmable Logic Devices
- Error Detecting and Correcting Codes
- Datapath
- Controls

Estimated Curriculum Category Content (Semester hours):

<i>Area</i>	<i>Core</i>	<i>Advanced</i>	<i>Area</i>	<i>Core</i>	<i>Advanced</i>
Algorithms	.25	0	Software design	.25	0
Data structures	.25	0	Prog. Languages	.5	0
Comp Org & Arch	1.75	0			