

Bergen Community College
Computer Science
Course Syllabus

Instructor: _____

Phone: _____

Email: _____

Office hours _____

Course Title: CIS-271 Computer Organization and Assembly Language

Prerequisites: CIS-165 (C++ Programming I) or CIS-265 (C++ Programming II)

Credits/Hours: 3 Credits 3 hours

Gen'l Ed. Course: No

Course Description:

Computer Organization and Assembly Language is a study of the interactions between hardware and software necessary for understanding the organization and applications of computer systems. Topics considered include data representation, Boolean algebra and computer logic, the central processing unit and program execution, main memory, classes of machine language instructions, addressing formats, addressing modes, and the fundamentals of assembly language programming.

General Objectives: this course has been designed to:

1. Introduce fundamental concepts of computer organization necessary for understanding the structure and applications of computer systems.
2. Provide an understanding of the internal data representation and memory allocation.
3. Enable the student to comprehend the steps involved in program execution.
4. Emphasize the importance of the mathematical foundation in Computer Science with the introduction of the relationship between Boolean algebra and logic gates.
5. Provide the student with the basic concepts of the Assembly Programming Language.

STUDENT OUTCOMES: upon satisfactory completion of the course, the student will

- Be able to name the five major components of a computer system, and explain the function of each component.
- Know how to trace the content of the appropriate special purpose registers in the CPU as assembly language instructions go through the instruction execution cycle.
- Understand basic logic circuits, and understand how full adders are used to perform binary addition.
- Know the different memory management techniques used by the operating system and the advantages and disadvantages of each.
- Understanding how character data and real numbers are internally represented, and the ability to represent signed integers two's complement and in packed format.
- Be able to identify the different addressing formats and addressing modes discussed in class, and will be able to write short code segments that use these addressing modes.
- Be able to identify the parts of an Assembly Language program.

Course Grade:

The student will be evaluated using a variety of methods, which may include, but are not limited to, some or all of the following: quizzes, exams, written assignments, programming assignments, and projects

Textbook:

Computer Systems, Fourth Edition; J. Stanley Warford; Jones and Bartlett Publishers, Inc, 2010
ISBN:098- 0-7637-7144-7

Course Content

1. Computer Systems Overview
 - Abstraction
 - Hardware
 - Software
2. High Level Languages
 - Machine dependency
 - Flow of control
 - The Run Time Stack
 - Concept of Recursion
3. Machine Level
 - Internal Representation of Data
 - Human Representation
 - Base Conversions (Decimal, Binary, Hexadecimal)
 - Binary Math
 - Hexadecimal Math
4. Computer Architecture
 - History of Computing
 - von Neumann Concepts
 - Main memory
 - Registers
 - Instruction Formats
 - Addressing modes
 - Program Execution Cycle
5. Assembly Language
 - Assembly Language Instructions
 - Language Format
 - Addressing
 - Branching
 - Pseudo Operations
 - Machine Language Instructions
6. The Operating System
 - Process Management
 - The Loader
 - Interrupts
 - Processes
 - Storage Management
 - Memory space allocation techniques
 - Virtual memory
 - File Management
 - Operations
 - Logical vs. Physical
 - Allocation Techniques
7. Logic Gates
 - Boolean Algebra
 - Truth Tables
 - Circuits
8. Combinational Circuits
 - Combinational Networks
 - Half Adder
 - Full Adder
9. Sequential Circuits
 - Feedback
 - Flip-Flops

Please turn off the sound to all electronic devices before entering any lab or classroom! You will be asked to leave if this disruption occurs, this will be counted as an absence.