

Course number and name: **CS 06205: Computer Organization**
Credits and contact hours: 3 credits /3 contact hours
Faculty Coordinator: Mohammed Mansaray
Text book, title, author, and year: Computer Organization and Design: the Hardware/Software Interface, 5th Edition, David A. Patterson and John L. Hennessy, 2013.

Specific course information

Catalog description: This course provides an introduction to computer organization. Students are exposed to the register level architecture of a modern computer and its assembly language. The topics include machine level data representation, von Neumann architecture and instruction execution cycle, memory hierarchy, I/O and interrupts, instruction sets and types, addressing modes, instruction formats and translation.

Prerequisites: (MATH 03160 Discrete Structures or
MATH 03150 Discrete Mathematics) **and**
(CS 04103 Computer Science and Programming or
CS 04113 Introduction To Object Oriented Programming) **and**

Sophomore standing

Type of Course: Required Elective Selected Elective

Specific goals for the course

1. **arithmetic for computers.** The student has demonstrated understanding of the representation of numbers and arithmetic algorithms.
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
2. **computer instructions.** The student has demonstrated an understanding of the instruction set architecture of a contemporary computer
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
3. **CPU and datapath.** The student has demonstrated understanding of the design and operation of a simple single-cycle processor.
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

4. **memory hierarchy.** The student has demonstrated understanding of the design and operation of a cache memory system.
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

5. **low level programming.** The student has applied low level programming languages to implement complex programs such as internal operating system components and drivers to interface with and control hardware devices or to achieve other results (speed, size, efficiency, etc.).
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

 - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

6. **low level programming risks.** The student has explained the risks and rewards that result from using low level programming.
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

 - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

7. **Role of hardware and software in computing devices.** The student understands the big picture of computer organization, specifically how the hardware implements what is written in the software.
 - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

Required list of topics to be covered

1. Computer abstractions and technology
2. Hardware/software interfaces and interactions
3. Different types of systems programs (e.g., development environments, operating systems, utilities, networking functions, device drivers, storage frameworks, gaming engines)
4. The role of performance
5. Instructions: language of the computer
 - a. The MIPS assembly language
 - b. Stack and Heap: functions and recursion in MIPS
 - c. MIPS instruction representation
 - d. Memory addressing modes in MIPS
 - e. Translating and starting a program
 - i. Translation of high-level language to machine code
 - ii. Translation of machine code to High Level language.
 - f. Arrays vs. pointers
 - g. Modularization
6. Arithmetic for computers
 - a. Number system: integers to binary or hex
 - b. Signed and unsigned numbers
 - c. Addition and subtraction
 - d. Logical operations
 - e. Floating point number representation (IEEE 754 format)
 - i. Addition and subtraction
 - ii. Multiplication and division
7. CPU and Datapath
 - a. Logic design conventions: concept of combinational and sequential logic
 - b. Creating a simple datapath
 - c. Relationship between MIPS and datapath
 - d. Introduction to pipelining
 - e. Data and control hazards (stalls)
 - f. Overview of memory hierarchy
 - g. Basics of caches
 - h. Cache performance
 - i. Virtual memory
8. Providing and using Application Programming Interfaces (API's)

Optional list of topics that could be covered

1. Interfacing processors and peripherals
 - a. I/O performance and devices
 - b. Buses