COURSE DESCRIPTION

Dept., Number	CSci 223	Course Title	Computer Organization and Assembly Language
Semester hours	3	Course Coordinator	Philip J. Rhodes, Assistant Professor

Current Catalog Description

Introduction to computer architecture of computer systems. The topics include processor and external device structures and operation, machine operations and instructions, assembly language concepts, and assembly language programming.

Textbook

Randal E. Bryant and David R. O'Hallaron. *Computer Systems: A Programmer's Perspective*, Prentice Hall, 2003.

References

Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, Prentice Hall, 1988.

Course Outcomes

Upon successful completion of this course, the students can:

1. explain the representation and design tradeoffs of signed and unsigned integers, floats, characters and strings;
2. read, write, and view binary files;
3. explain and use *pointers* in C and similar languages;
4. explain how high-level language features such as recursion, function calls and control structures are implemented in a machine language;
5. write programs in assembly language;
6. explain the *memory hierarchy* and its effect on program performance;
7. explain processor pipelines and superscalar architecture;
8. use a *debugger* to aid in program development;
9. use a *profiler* to discover performance bottlenecks.

Relationship between Course Outcomes and Program Outcomes

The course outcomes contribute to the program outcomes as follows: (1) to (a,j), (1,2) to (b), (1,2,3,4,5,6) to (c), (1,2,3) to (e), (1 through 9) to (i), (4) to (k)

Prerequisites by Topic

1. Fundamental programming skill (CSci 112)
2. Basic data structures and algorithms (CSci 112)

Major Topics Covered in the Course

1. Data representation and manipulation
    a. Binary and hexadecimal number systems
    b. Representation of unsigned integers
    c. Binary addition
    d. 2's complement representation of signed integers
    e. Bitwise operators
    f. Binary multiplication
    g. IEEE 754 floating point numbers
    h. ASCII character codes
    i. Strings
    j. Text files
    k. Binary files
    l. Pointers
2. Assembly Language
    a. Data movement instructions and operand formats
    b. Logical and arithmetic instructions
    c. Processor status register
    d. Conditional and unconditional jumps
    e. Function calls and parameter passing by value and by reference
3. Architecture
    a. The memory hierarchy
    b. Fetch/decode/execute cycle
    c. Historical factors
    d. RISC and CISC
    e. Prefetching and pipelining
    f. Superscalar architecture
    g. Stalls and hazards
    h. Branch prediction and out-of-order execution
    i. Instructionl-level parallelism
    j. Code optimization

Assessment Plan for the Course

A comprehensive, 30-question exam constructed by a faculty committee is administered to each offering of CSci 223. Student performance is analyzed question-by-question to identify needed adjustments in the textbook, lectures, or assignments. Faculty who regularly teach the class and the classes that follow participate in the evaluation, in the selection of textbooks, and in formulating a response appropriate to the assessment results.

How Data in the Course are Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

The standard exam administered in CSci 223 (see the previous item) is included in the curriculum–wide outcome assessment described in Chapters 2, 3, and 4 of the Self–Study

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	1.5		Software design	0.5	
Data structures	0.5		Concepts of programming languages	0.5	