Course Description

MET CS300
Foundations of Modern Computing

This course introduces basic concepts in discrete mathematics, computer systems and programming that are necessary for modern computing systems. It also develops analytic and logical thinking and prepares students to take graduate-level courses in software development degree. This course first reviews the basic concepts in discrete mathematics including logic, sets, functions, relations and combinatorics. Then it discusses the fundamental concepts in computer systems such as computer organization, basic OS concepts, CPU scheduling, memory management, process management and synchronization. Concurrently with the above mathematics and systems studies, programming concepts are introduced and practiced throughout the whole course using Python.

Learning Objectives

The course is designed to prepare students without a technical background in computer science to succeed in graduate courses in the Master of Science in Software Development and the MS in Computer Information Systems.

For students coming from other programs, this course is a technically oriented introductory survey of modern computing system.

Course Objectives

Upon completion of this class, students are expected to:

For Discrete Math:

- Explain the basic concepts of logic, and solve mathematical problems that involve the laws of logic and the rules of inference.
- Explain the basic concepts of sets, and solve mathematical problems that involve sets and subsets, set operations and laws of set theory
- Explain the basic concepts of combinatorics, and solve mathematical problems that involve permutations and combinations.
- Explain the basic concepts of functions and relations, and solve mathematical problems that involve one-to-one, onto functions and function composition, the product sets and relations, equivalence relations and partitions.

For Computer Systems:

- Convert numbers between binary, decimal and hexadecimal formats, and explain how basic data types (such as integer, real number, character) are represented in the machine.
- Explain how source program are compiled/interpreted, interpreted executed, including the instruction execution cycle and the role of interrupts.
- Identify the he basic components in a computer system. Explain how multiprocessor/multicore CPU affect the system performance. Explain the storage hierarchy and how caching is applied between different layers in the hierarchy.
- Explain basic concepts of an OS. Explain OS kernel and basic protection mechanisms implemented in an OS.

- Describe the process model and basic operations.
- Explain basic CPU scheduling algorithms in an OS.
- Identify the synchronization issues and describe basic solutions.
- Explain virtual address space and the paging scheme.
- Describe basic components and features of distributed systems and networks.

For Programming:

- Describe the basic concepts of programming.
- Explain how algorithms are developed and implemented in high level languages.
- Design, write, and debug python programs that use sequence, selection and repetition statements, primitive data types, strings, lists, tuples and that do I/O and file manipulation.
- Decompose the program into functions and modules
- Explain and apply basic object oriented concepts including classes, objects and inheritance.
- Design, write and debug basic GUI programs.

Course Organization

This course is 7 weeks long: 6 weeks of content, and 1 week for the final exam. Each of 6 weeks includes:

- reading assignments
- online content
- review questions
- graded assignments
- self-test or quiz
- Possibly one discussion question as extra credit

Each assignment is due in 1 week.

The study guide, which precedes each module, lists specific due dates. Assignments and quizzes are due at 6am ET each Tuesday. Review questions and math assignments are optional, but strongly encouraged. The review questions are very similar to the quiz questions. Review questions may be answered as many times as you like, while quizzes are timed and may be taken only once. Math assignment solutions are provided in videos labeled “echo 360” Solutions to information technology assignments and quizzes will be provided after they are graded.

You will see “blocks” of content in the online material that are labeled “Advanced Content.” We have found that some students like additional material beyond what is formally part of the course. Hence, we are in the process of adding such content. You are not responsible for advanced content on the quizzes, assignments, or final exam.

Course Outline

Week 1

Math 1: Logic

- Basic connectives, Truth tables

- The laws of logic
- Rules of inference

Programming 1: What is Programming?

- Using the Python Interpreter; Invoking the Interpreter;
- The Interpreter and Its Environment

Week 2

Math 2: Combinatorics

- Combinatorics: permutations and combinations
- Sets and subsets
- Set operations, the laws of set theory

Programming 2: An Informal Introduction to Python

- Using Python as a Calculator;
- Numbers;
- Strings;
- Lists;
- First Steps Towards Programming

Week 3

Math 3: Relations and Functions

- Cartesian products and relations
- Functions
- One-on-one and into functions

Programming 3: Control Flow

- *if* Statements;
- *else* Clauses for Statements;
- *while* Statements;
- the *range*() function;
- *break* and *continue* statements

Week 4

System 1: Data/Program Presentation and Computer Organization

- Data Presentation: bit, bytes, binary, decimal, hexadecimal, basic data types
- Program representation and execution: source code, target code, virtual machine, compiler process
- Computer organization: CPU (multicore, superscale), storage hierarchy (cache, memory, disks), data and address

Programming 4: Defining Functions

- Default Argument Values
- Keyword Arguments

Week 5

System 2: Intro to OS, Process and Synchronization

- Basic OS concepts: protection, kernel, system calls vs. API, OS structures
- Process Management, CPU Scheduling
- Synchronization

Programming 5: Coding Style

- Naming
- Specification
- Debugging

Week 6

System 3: Memory Management and Computer Networks

- Virtual Memory
- Distributed systems and computer networks

Programming 6: Modules

- Packages; Importing * From a Package;
- Intra-package References
- File Input and Output

Week 7 - Final Exam