

University of Toledo

Computer Science & Engineering Technology

Master Syllabus

Course Title: Introduction to Computer Science & Engineering Technology
Course Code & Number: CSET-1100

Credit Hour Total: 4 **Weekly Contact Hours Lecture:** 3 **Lab Hours:** 1

Prerequisite(s): MATH 1330

Text: Introduction to Programming Using Python, 1st Edition, Y. Daniel Liang, 2012
ISBN: 10: 0132747189

Software: None

Course Coordinator: Jared Oluoch (Jared.Oluoch@utoledo.edu) CSET Required

Current Catalog Description:

This three semester hour course is the first course in computer hardware and software for CSET majors. Single and multi-user operating systems, command-line processing, program planning and creation and simple Internet tools are covered.

Related Program Outcomes:

CSET Program (a, b, c, e and i)

IT Program (a, b, c, e and i)

- a.) An ability to select and apply knowledge of computing and mathematics appropriate to the discipline. Specifically, an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer—based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CAC---j]
- b.) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- c.) An ability to design, implement, and evaluate a computer--based system, process, component, or program to meet desired needs. Specifically, an ability to apply design and development principles in the construction of software systems of varying complexity. [CAC---k]
- e.) An understanding of professional, ethical, legal, security and social issues and responsibilities including a respect for diversity.
- i.) An ability to select and apply current techniques, skills, and tools necessary for computing practice.

Course Objectives:

After successful completion of this course, students will be able to:

- Describe the fundamentals of computer history, hardware, software and OS.
- Connect from a remote location to a UNIX server on which they have an established shell account.
- Transfer files to and from a PC or Mac to their account on a UNIX server.
- Use the UNIX “man” pages to determine the proper syntax for user-level UNIX commands.
- Manage their file system on a UNIX server.

- Create technical and simple game programs using the Python programming language.
- Debug and test programs.
- Describe issues related to user interfaces and user friendliness.
- Describe and implement the data structures available in the Python programming language.
- Design simple algorithm using pseudocode and/or a flowchart.
- Create programs that implement a variety of common algorithms using the Python programming language.

Major Topics Covered in the Course

Topic	Lecture Hours
History of Computing	1.0
Overview of OS (Unix as example)	1.5
Introduction to net centric computing (FTP,SSH etc)	1.5
Machine level representation of data	1.5
Assembly level machine organization	1.5
Overview of Programming Languages	1.5
Procedure Development: Pseudocode, Flowcharting & Control Structures	1.5
The process and mechanism of creating a program	1.5
Program environment, tools	1.5
Intro to Python: Basic Programming structure	1.5
Basic Syntax, Variable declaration, I/O	1.5
Basic Operators, Expression evaluation, Priority, casting	1.5
Introduction to the gcc for compiling C program code	1.5
Block statement, Python Flow Control: Branching	1.5
Looping, break, continue, sentinel	1.5
Abbreviations, Characters conversion	1.5
Debugging and testing, error and handling	1.5
Library, Function	1.5
Scope of variables, parameter passing	1.5
One-Dimensional Arrays, searching, summing, etc.	1.5
String and operations	1.5
Properties of good software design, user friendliness	1.5
Higher-Dimensional Arrays	1.5
Typedef, structures	1.5
Text File Processing	1.5
Software requirements, validation and verification	1.5
Problem solving using algorithm, simple algorithm	1.5
Sequential and binary search	1.5
Quadratic Sorting algorithms	1.5
O(N log N) sorting	1.5
Totals	44.5

Laboratory Projects:

- Telnet, ftp and Unix file system
- Program creation in response to various homework assignments

Oral and Written Communications

Not part of the course.

Social and Ethical Issues

Briefly discussed the issues involved with collaborating on programming assignments.

Theoretical Content

Number systems, binary arithmetic, number system conversions, data abstraction

Problem Analysis

This course emphasizes problem analysis in the areas of program development. Students learn how to create pseudo-code to describe a solution to a problem, select appropriate data types, and test resulting programs.

Solution Design

This course requires students to produce a number of programs that lead to a design solution for a programming problem. Students use structures to read, write and modify binary records.