

CSET 1100 Introduction to CSET (3 semester credit hours)

CSET Required
IT 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.

Textbook:

“Python Programming for the Absolute Beginner,” Third Edition, Michael Dawson, Course Technology PTR, 2010.

References:

Course web pages: <http://cset.sp.utoledo.edu/cset1100py/>

Related Program Outcomes:

CSET Program (a, b, c, e and i) – See attached table

IT Program (a, b, c, e and i) – See attached table

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

Topic	Lecture Hours
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.

Course Coordinator

Allen Rioux (allen.rioux@utoledo.edu)

2/26/07

Dan Solarek (daniel.solarek@utoledo.edu)

2/28/2011

Syllabus: CSET 1100

	Student Outcomes: CSET Program	Course Outcomes	Assessment Methods
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]	<ul style="list-style-type: none"> • Create technical and simple game programs using the Python programming language. • Debug and test programs. • Describe and implement the data structures available in the Python programming language. • Design simple algorithm using pseudocode. • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by the ability to create and document a procedure, translate the procedure into Python code, execute the Python code in a both a UNIX environment and a Windows environment
b	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.	<ul style="list-style-type: none"> • Design simple algorithm using pseudocode. 	as evidenced by the ability to create and document a procedure
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]	<ul style="list-style-type: none"> • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by the ability to create and document a procedure, translate the procedure into Python code, execute the Python code in a both a UNIX environment and a Windows environment
d	an ability to function effectively as a member or leader on technical teams to accomplish a common goal.		
e	an understanding of professional, ethical, legal, security and social issues and responsibilities including a respect for diversity.	<ul style="list-style-type: none"> • 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. 	As evidenced by the ability to remotely connect to a UNIX-like server using ssh and the tradeoffs with respect to telnet connections
f	an ability to communicate effectively with a range of audiences using a range of modalities including written, oral and graphical.		
g	an ability to analyze the local and global impact of computing on individuals, organizations, and society.		
h	recognition and understanding of the need for and an ability to engage in self-directed continuing professional development.		
i	an ability to select and apply current techniques, skills, and tools necessary for computing practice.	<ul style="list-style-type: none"> • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by successful navigation of multiple similar UNIX-like operating systems
j	an ability to conduct standard tests and measurements; to conduct, analyze, and interpret experiments; and to apply experimental results to improve processes.		
k	a commitment to quality, timeliness, and continuous improvement.		

Syllabus: CSET 1100

	Student Outcomes: IT Program	Course Outcomes	Assessment Methods
a	an ability to select and apply knowledge of computing and mathematics appropriate to the discipline. Specifically, an ability to use and apply current technical concepts and practices in the core information technologies. [IT-j]	<ul style="list-style-type: none"> • Create technical and simple game programs using the Python programming language. • Debug and test programs. • Describe and implement the data structures available in the Python programming language. • Design simple algorithm using pseudocode. • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by the ability to create and document a procedure, translate the procedure into Python code, execute the Python code in a both a UNIX environment and a Windows environment
b	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.	<ul style="list-style-type: none"> • Design simple algorithm using pseudocode. 	as evidenced by the ability to create and document a procedure
c	an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. And, an ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems. [IT-k]	<ul style="list-style-type: none"> • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by the ability to create and document a procedure, translate the procedure into Python code, execute the Python code in a both a UNIX environment and a Windows environment
d	an ability to function effectively as a member or leader on technical teams to accomplish a common goal.		
e	an understanding of professional, ethical, legal, security and social issues and responsibilities including a respect for diversity.	<ul style="list-style-type: none"> • 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. 	As evidenced by the ability to remotely connect to a UNIX-like server using ssh and the tradeoffs with respect to telnet connections
f	an ability to communicate effectively with a range of audiences using a range of modalities including written, oral and graphical.		
g	an ability to analyze the local and global impact of computing on individuals, organizations, and society.		
h	recognition and understanding of the need for and an ability to engage in self-directed continuing professional development.		
i	an ability to select and apply current techniques, skills, and tools necessary for computing practice. And an ability to effectively integrate IT-based solutions into the user environment. [IT-l]	<ul style="list-style-type: none"> • Create programs that implement a variety of common algorithms using the Python programming language. 	as evidenced by successful navigation of multiple similar UNIX-like operating systems
j	an understanding of best practices and their application. [IT-m]		
k	an ability to assist in the creation of an effective project plan. [IT-n]		