

CS 161: Foundations of Computer Science I

Course Syllabus

Winter 2007

Professor: Richard Croft, Ph.D. Badgely Hall 109 962-3695
e-mail: rcroft@eou.edu
web: www.eou.edu/~rcroft

Office Hours: M W F 2:00— 3:00
T Th 11:00— 12:30
and by appointment.

Note that I may be in BH 123 or LH 235 during office hours.

Catalog Description

Includes structured programming, computer organization, social and ethical issues in computer science.

About this Course

This class introduces students to the use of computers as problem solving tools.

Outcomes

Students enrolled in this class will learn basic conceptual knowledge necessary to develop computer programs, and how to use fundamental control structures in a representative programming language. When they have completed this course students will be able to:

1. describe the conceptual organization of a computer;
2. describe how a processor loads a program and carries out the instructions;
3. define problem solutions using simple commands and control structures;
4. design algorithms;
5. write, compile, and debug simple programs;
6. use subroutines to modularize program solutions; and
7. implement simple solutions using arrays.

Prerequisites: MATH 111

Textbooks and Materials

Zelle, J. (2004). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates. (ISBN 1-887902-99-6 (Required))

Means of Assessment:

Quizzes and two exams will be used to assess student mastery of conceptual material, and exercises and projects will be used to assess student ability to apply concepts. Projects will be a series of increasingly-complex programs requiring use of concepts introduced in class.

Course Activities

This course includes regularly-scheduled class meetings, reading assignments, written exercises, programming assignments, quizzes, and two exams. Class meetings will be spent in lectures and class discussions of topics and approaches to solving problems. Once a week the class will meet in the programming lab for guided activities..

Students will complete assignments and reading outside class. The programming lab will be available for students to do programming work, and the instructor will be available for consultation during office hours and by appointment. Please allow yourself time to complete work on time—a common rule of thumb for university classes is allow two to three hours of time outside class each week for each credit hour the course carries.

Policies

Your continued enrollment in this class signifies that you understand and accept these policies.

Attendance, while not a factor in grade computation, is nevertheless strongly encouraged. Students who miss class are responsible for all lecture notes, assignments, and assignment revisions distributed in the missed session. Chronic absences will very likely be reflected in poor performance on quizzes, exams and programming assignments.

Assignments. When each exercise or project is assigned, the due date will be posted on the class web site. Ample time will be allowed for completion (provided you don't procrastinate too much).

Save all returned work. Keep a back-up copy of any work you turn in.

Late Assignments. Any work turned in after the due date and time may incur a 20% penalty for each school day it is late. Within two or three days of the due date for programming I will post a sample solution. Programs turned in after this time will receive no credit.

Reading Assignments will be in advance of the material being covered in lecture. Students are advised to complete readings *before* the class discussion and then to *re-read* the material afterward.

Programming is a critical skill for computer science students and increasingly important in the fields of multimedia development, computer animation, and graphics as well. Learning to program well is time consuming and requires great attention to detail. As we discuss conceptual material in class you should anticipate the sorts of problems you may need to solve for the next program and begin thinking about them. When an assignment is made, you should immediately begin working on the design for the solution.

Since one cannot memorize the solution to every problem, concentrate on understanding *how* your solution works—the ability to solve new problems is the most important skill a programmer can develop.

Good programming style and use of documentation are also important, and consequently will be included in consideration for grading. Appropriate documentation and style will be gradually introduced throughout the course.

Pop quizzes will provide feedback to let you determine if you are assimilating enough detail in course topics, and may cover lectures, discussion and assigned reading. One in four quiz scores may be dropped (*i.e.*, if we have four quizzes the lowest won't count; if we have 8, the lowest two won't count.) Quizzes may be made up only if I deem the documented excuse valid.

Understandable communication is essential in any profession. Grammar, spelling, and clarity of expression will affect evaluation of any written work you produce outside the classroom. This and neatness apply to program listings as well as written assignments.

If you have any questions, comments, concerns, or suggestions, please feel free to write them on a slip of paper and leave it on the lectern (or hand it to me) when the class breaks. Your feedback may help improve the course.

Academic Misconduct

Eastern Oregon University places a high value upon the integrity of its student scholars. Any student found guilty of an act of academic misconduct (including, but not limited to, cheating, plagiarism, or theft of an examination or supplies) may be subject to having his or her grade reduced in the course in question, being placed on probation or suspended from the university, or being expelled from the university—or a combination of these. Please see Section II of the *2002-2003 Student Handbook and Planning Calendar: Campus Citizenship (Academic)*, p. 32ff; *Campus Citizenship (Behavior)*, p.41ff.

Plagiarism is copying, in part or as a whole, the work of another person and submitting it as your own, and is expressly forbidden by the University's Academic Honesty Code. This includes using someone else's program as a template and revising identifiers and documentation to create your program. I will randomly interview students and ask them to explain how their programs work. Solutions that students cannot explain may be subject to severe penalties (see above!), as will any solutions that have clearly been copied from other sources.

Classroom Decorum. Please try to get to class on time. Arriving late is a distraction, and not courteous to other students. If you are on time, please leave the desks nearest the door vacant for students who arrive after class starts.

Unless the class is engaged in a discussion, refrain from conversation with other members of the class. Unnecessary conversation is a distraction to the students and to me. If you have a question please address it to me.

Please turn off cell phones and pagers while in class.

Students with Disabilities

If you have a documented disability or suspect that you have a learning problem and need reasonable accommodations, please contact the Disability Services Program in Loso Hall 234 (telephone 962-3081) **before** the end of the second week of classes.

Grading

Your final grade for this course will depend on your completion of the assigned exercises and programming assignments, quizzes, and exams. Note that the weight given each program will depend on the time it requires. Credit is distributed as follows:

Exercises:	5 percent
Programming Assignments:	35 percent
Quizzes:	20 percent
Midterm Exam:	15 percent
Final Exam:	25 percent

Grade cutoffs will be no *higher* than 92 for A, 84 for B, 75 for C and 65 for D, but may be lower if analysis of the distribution of scores indicates they should be.

Course Schedule (Tentative*)

Week	Main Topics (Additional special topics may be announced)	Reading
1	Course introduction; Review of basic computer hardware and types of computer languages	Chapter 1
2	Literal and variable data Fundamental statements The String data type (introduction) First program due	2.1—2.5 4.1—4.2
3	Counter-controlled iteration; Numeric data types, numeric computation;	2.6—2.7 Chapter 3
4	Designing programs; pseudo code and flow charts; Control structures: Selection Second program due	7.1—7.3 7.5
5	Control structures: sentinel controlled iteration	8.1—8.3.2
6	Midterm exam Objects and Graphics Third program due	5.1—5.5
7	Subroutines (functions); Parameter passing	Chapter 6
8	Lists (arrays) and Strings Fourth program due	4.3; 11.2
9	Approaches to program design	Chapter 9
10	Special Topics Review for Exam Last program due	

*Schedule is subject to change depending on class dynamics.