

CS 161: Foundations of Computer Science I

Course Syllabus Fall 2011

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

Office Hours: M—F 11:00—11:50
and by appointment.

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

Catalog Description

Introduces basic data representation, branching and iteration, memory management, computer architecture, and the analysis and design of problem solutions.

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 the use of literal and variable data;
3. precisely define problem solutions in English;
4. design computer algorithms using fundamental control structures (sequence, iteration, and selection);
5. write, and debug programs to implement algorithms;
6. design and write modular programs using functions; and
7. design programs that use list data.

Prerequisites: MATH 111 (or consent of instructor)

Textbooks and Materials

Gaddis, T. (2010). *Starting out with program logic & design*. Pearson/ Addison Wesley. (ISBN 0-13-53711-4 (Required Earlier editions are acceptable))

Means of Assessment:

Quizzes and two exams will be used to assess student mastery of conceptual material (outcomes 1, 2, 3 & 6), and exercises and projects will be used to assess student ability to apply concepts (outcomes 4, 5, 6, & 7). 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, lab sessions, 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

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). Begin working on solutions to programming problems as soon as you can.

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 (or portion thereof) it is late. Within two or three class days of the due date for programming I will post a sample solution. Programs turned in after this time will receive no credit.

You should at least scan **Assigned Readings** in advance of the material being covered in lecture. Re-read assignments after the material has been covered in class.

Once a week (usually on Friday) the class will meet in the programming lab (BH 123) for lab work. Each week there will be a lab activity for you to complete. Thoughtful completion of these assignments will clarify concepts covered in lecture, help you succeed on programming assignments, and give you an idea of what some quiz and exam questions will be. On some days lab will begin with a drill to give you practice in reading and typing program code.

Programming is a critical skill for computer science students and increasingly important in many other disciplines including multimedia development, computer graphics, mathematics, and the sciences. 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 and how you arrived at it—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.

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 the relevant section of the student handbook on-line at: <http://www.eou.edu/saffairs/handbook/honest.html>

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.

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.

Turn off cell phones and pagers while in class.

Each student will receive a **Lab Account** for the programming lab. Your user ID and password are *yours*—do not share them with anyone else (the same applies to your EOU e-mail and Webster accounts). If more than one person knows the password for a computer account then it is not secure. Incidentally, with this in mind, no faculty member or administrator keeps a copy of your lab account information—don't lose it.

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 ^o |
|------|---|----------------------|
| 1 | Course introduction; Review of basic computer hardware and types of computer languages; Using the programming lab | Chapter 1 |
| 2 | Literal and variable data; Fundamental statements Designing programs; pseudo code and flow charts; The String data type (introduction) First program due | Chapter 2 |
| 3 | Counter-controlled iteration; Numeric data types, numeric computation; | 5.1, 5.3, 5.4 |
| 4 | Control structures: Selection Second program due | 4.1—4.4 4.6—4.7 |
| 5 | Control structures: sentinel controlled iteration | 5.2, 5.5, 5.6 |
| 6 | Midterm exam Modular programming and top-down design Third program due | Chapter 3 |
| 7 | Subroutines (functions); Parameter passing | Chapter 6 |
| 8 | Lists (arrays) and Strings Fourth program due | 8.1—8.3 |
| 9 | Manipulating data in lists Thanksgiving Break Wed—Fri | 8.4 |
| 10 | Special Topics Review for Exam Last program due | |

*Schedule is subject to change depending on class dynamics.