Computer Science 1320 Course Syllabus

Jeffrey D. Oldham

1999 Aug 26

Course

Course: Problem Solving and Algorithm Design I

Prerequisites: none

This course is the first course for computer science majors, following the guidelines established by the Association for Computing Machinery. This course also partially satisfies the requirements for *Understanding the World Through Science* of the common curriculum. The course content will include learning about block structured strongly typed programming languages as well as conceptual information including beginning data structures, computer arithmetic, computer organization, operating systems, programming languages, sorting, and searching. Our study will include data types, arrays, strings, structures, files, recursion, decisions, and loops.

Course Goals and Objectives

The objectives of this course include, but are not limited to, the following:

- learning fundamental problem solving methodology
- applying problem solving techniques to algorithm design
- implementing algorithms in a suitable programming language
- development and analysis of algorithms
- introduction to the basic topics in data structures
- introduction to sorting and searching algorithms

Instructor

Instructor: Jeffrey D. Oldham

Email: oldham@cs.trinity.edu

WWW: http://www.cs.trinity.edu/˜joldham/1320/

Telephones: 210.999.8139 (office), 210.832.9879 (home)

Office: Halsell 201J

Office Hours: Tuesdays and Thursdays 1:30–4:30pm, Wednesdays 10:00am–2:00pm

Text

Text: *Problem Solving with C++: The Object of Programming* by Walter Savitch, Addison-Wesley, Second Edition, 1999.

These additional references graciously provided by Maury Eggen.

- Cormen, Leiserson and Rivest, *Introduction to Algorithms*, McGraw Hill, 1990.
- Eggen and Eggen, *Introduction to Computer Science using C*, PWS Publishers, 1996.
- Hanly, Koffman, and Friedman, *Problem Solving and Program Design in C*, Addison Wesley, 1993.
- Kelley and Pohl, *C by Dissection: The Essentials of C Programming*, Benjamin Cummings, 1992, Second Edition
- Kernighan and Ritchie, *The C Programming Language*, Prentice Hall, 1988, Second Edition.
- King, *C Programming: A Modern Approach*, Norton Publishers, 1996.
- Schildt, *C: The Complete Reference*, McGraw Hill, 1990, Second Edition.
- Sobell, *A Practical Guide to the Unix System*, Benjamin Cummings, Third Edition, 1995.

Grades

The grades in this course will be determined by the results of

- three exams, approximately equally weighted,
- several homework assignments, weighted according to difficulty, and
- periodic quizzes.

Averages will be calculated as a simple percentage, i.e., points earned divided by points possible. Letter grades will be assigned according to the following scale:

90–100	A
80–89	B
70–79	C
60–69	D

with plus and minus grades assigned in marginal cases. Keeping track of your scores will indicate where you stand in the class at any time.

Exams

Exams are comprehensive but will emphasize the most recent material. They are scheduled as follows. Please plan accordingly.

Examination 1	30 September
Examination 2	04 November
Reading Days	December 9–10
Final Examination	December 15 8:30am for CS1320-4 December 11 6:30pm for CS1320-5

Quizzes

Expect a quiz once a week. Quizzes will usually be given on Tuesdays and will cover the previous week's work. Quizzes will be taken when administered.

Homeworks

Several homework assignments will be required for successful completion of this class. Each assignment will be due at the *beginning of the period* on the day assigned.

Most homeworks will be laboratory problems, which will be coded in a suitable programming language on the departmental UNIX network. Solutions turned in for grading must represent your own individual effort. Each problem submission must include a printed listing of the programs as well as documentation for the programs submitted. Detailed requirements for problem submission will be given in class.

Discussion of the homework assignments between students is encouraged, but you must submit your own work. You are cheating yourself if you get the program from some source other than creating your own. Programs that are identical beyond coincidence are in violation of the Academic Integrity policy of the university and will result in disciplinary action, including, but not limited to a failing grade on that assignment for all parties involved. You are responsible for the security of your work, both electronic and hard copy.

Attendance

Regular class attendance is strongly encouraged, and attendance may be taken.