

Course Syllabus

CS151 - Fall Semester - 2003-04

Course Information

Course Number: CS151

Prerequisite: CS150

Meeting Times: 11:00-12:15

Meeting Days: TR

Classroom: CH 7

Instructor Information

Instructor: Frank Young

E-mail: youngf@hawaii.edu

Office: CH 2E, Telephone: 933-3189

Office 2-3:30pm (TR) or by appointment.

Hours: Send me email or call me at home when I am not on campus.

The phone number is 886-2436. Please call before 9:00 pm.

Please leave a message on the answering machine if I am not at home.

Course Description

This course is the continuation of CS 150. This course introduces advanced programming techniques such as recursion and memory management. Simple data structures like stacks and queues are also examined. This course is an introduction to powerful abstraction methods, methods that empower the user. This course will change the way you think about data and algorithms and prepare you for further study of computer science.

Prerequisites

You must have successfully completed CS 150 before taking this course. Successful completion does not mean that you received a D in that course. I will assume that you have at least a C level knowledge of the content of CS150.

Textbook

The textbook for this course is *Problem Solving with C++: The Object of Programming, Third Edition*, by Walter Savitch. (Addison-Wesley, 2001, ISBN 0-201-70390-4).

There may be some other useful documents on the *links* page of the class web site.

Educational Objectives and Student Outcomes

1. Students will understand recursion and be able to apply it effectively when developing computer software.
2. Students will understand basic data structures (arrays, stacks, queues, lists, strings, and linked data) and will be able to use them when developing computer software.
3. Students will understand basic abstract data structures and their various implementations. They will be able to assess the impact of data structure implementation on the run-time behavior of software.
4. Students will understand standard techniques of pointer manipulation and dynamic memory management. They will be able to apply this knowledge when designing and implementing classes for basic data structures. They will also be able to apply this knowledge to the software development process.
5. Students will understand and be able to use advanced C++ concepts and capabilities such as templates, inheritance, polymorphism and exception handling. Students will be able to use and understand C++ documentation.
6. Students be aware of standard class libraries and the issues involved in their use. Students will be able to use standard libraries correctly and effectively.
7. Students will be familiar with object-oriented design techniques and be able to apply them to software design.
8. Students will be aware of most of the stages of software development. They will be able to complete most of a project lifecycle, beginning with specification and design and concluding with implementation and testing.

Grading and Evaluation

The course grade will be based on a series of individual programming assignments, three quizzes and a participation grade. The weighting of these items will be as follows:

Quizzes: 45 % (15% each)

Programming Assignments: 35 %

Group Projects: 20 %

These areas are described in more detail below.

Examinations and Quizzes

There will be three quizzes. *Tentative* dates for the quizzes are shown on the *class calendar* and will be verified in class. Test material will be drawn from the lecture and readings.

Programming Assignments

There will be approximately 7 individual programming assignments during this course. You will generally have one week to complete each assignment. Details of the assignments can be found under the *Assignments* link as they become available.

Programming assignments are graded on a 25-point scale with marks assigned to the following five areas:

Design (5 points)

The design of the program is appropriate and can be easily understood.

Documentation (5 points)

There are appropriate comments to explain the program and code.

Code Style (5 points)

The program is written in a consistent, readable form.

Execution (5 points)

The program compiles without errors or warnings and executes correctly.

Testing (5 points)

The program passes a series of acceptance tests.

Programming is fun, but it can be very time-consuming. To pass the class you will need to spend several hours each week in the lab outside of lecture working on the programming assignments. You will find that you will learn much of the material while you do the programming assignments. So the time you spend on programming assignments should be considered study time.

Group Projects

To further develop programming and problem solving skills, there will be two group projects, each worth 10% of the course grade. These projects will involve all phases of the software lifecycle from requirements analysis to testing. Details of the projects will be made available under the *Assignments* link.

Groups will be assigned by the instructor. Project groups will be responsible for arranging meetings outside of scheduled class time if necessary.

You will generally have two or three weeks to do each project.

Programming Guidelines

Naturally, programming assignments are to represent your own work unless they are specifically assigned as a group project. In this respect, they do not differ from examinations. Since your programs are to be your own, neither the logical design nor the actual code is to be the product of a group effort or of anyone other than yourself (with the exception of code given out to the class or available from the textbook). Violations will be viewed as academic dishonesty and will be treated as such. Since nontrivial programs produced independently are invariably different, strikingly similar programs submitted for an assignment will be suspect. Should more than coincidentally similar programs be turned in, the students involved will be required to submit evidence that their programs are their own work. Both handwritten evidence and computer code will be considered. The lack of such documentation will be taken to indicate that the assignment was not your own work. An inability to explain the design and operation of the program will also be taken as an indication that the work is not your own. For your own protection (and because it represents good programming practice), you should save all files, printouts, and other documents used to produce your programs until after you have received a final grade in the course. Failure to do so is at your own risk.

Late Policy for Assignments

There is a very strict late policy to keep you from getting behind on the assignments. As you can see from the table below, after the **late deadline it is not worth turning in the**

assignment, because *it is worthless*. It is much better to start the next assignment than to turn an old one in late.

When turned in	% Possible
Before the deadline	100
Before the late deadline	50
After the late deadline	0

The reason for this strict late policy is simple. If some students have not turned in an assignment then we cannot discuss it in class. This delays and hinders the education of the rest of the class. This is unacceptable. Hence the strict policy.

PLEASE NOTE:

The programming assignments are not trivial! If you wait to start the assignment until the day before it is due you will almost certainly not finish it on time, or do a good job on it. Start early!

Department Policies Concerning Course Administration

Honesty (or lack thereof):

All course work, including examinations, homework, and programming assignments, must be the individual work of the student who submits it unless specifically assigned as a group project. It is dishonest to submit as your own work anything that is copied, reproduced mechanically (such as disk file copying) or in any other way not done completely by yourself. Incidents of dishonesty will be handled according to the University procedures governing cheating and plagiarism. The following penalties may be assessed:

- An "F" in the course for dishonesty in any assignment, examination or paper.
- Any students who are involved in group action which makes possible cheating or plagiarism shall be subject to the same penalties as if they themselves had cheated or plagiarized.

The grade of "W" (Withdrawals):

Students who feel that they cannot complete a course and receive a grade that reflects their knowledge of the material may withdraw from that course before the published withdrawal deadline. The student must obtain a drop slip, process it properly, and submit it to the appropriate office. Upon successfully dropping a course the student will receive the grade of W. Do not stop attending a class without formally dropping it; this will result in the grade of F.

The grade of "I" (Incomplete):

The grade of "I" is given only when a student is unable to complete a *small* but important part of the course because of illness or other conditions beyond the control of the student. *Negligence, indifference, or a failing grade are never acceptable reasons for giving an "I"*.

In order to apply for a grade of "I", the student must present to the course instructor a signed statement including the reason for the incomplete, a detailed description of the work that is to be completed, and a deadline for completion, which must not exceed the date specified in the Academic Calendar. Then the appropriate form will be completed by the student and the instructor to issue the grade of "I".

Special Assistance:

UH Hilo will provide reasonable and appropriate accommodations to students with documented disabilities. Any student with a documented disability who would like to request accommodations should contact the University Disability Services Office (933-0816 (Voice), or 933-3334 (TTY), shirachi@hawaii.edu, Campus Center Room 311), as early in the semester as possible.

Advising:

Advising is a very important resource designed to help students complete the requirements of the University and their individual majors. Students should consult with their advisor at least once a semester to decide on courses, check progress towards graduation, and discuss career options and other educational opportunities provided by UH-Hilo. Advising is a shared responsibility, but students have final responsibility for meeting degree requirements.

[*Frank Young*](#)

Last modified: August 22, 2003