

CSC 340 Programming Methodology

Prerequisite: *grades of C or better in CSC 220 and CSC230.*

Catalog Description: This course explores advanced data structures and algorithms for manipulating them in C++. Emphasis is placed on design and implementation of those structures and a variety of practical applications. Algorithm coverage will include sorting and searching, and graph algorithms. Students will solve a series of problems to enhance their problem-solving skills.

Course Topics:

C++ Topics – transitioning from Java

STL, pointers, namespaces, inheritance, polymorphism, parameter passing, dynamic memory allocation

Searching and sorting algorithms

Sorting: quick sort, bubble sort, binary sort, mergesort, heapsort and insertion sort; runtimes of these algorithms will be considered.

Graph algorithms

Programming Exercises:

There will be a number of intensive hands-on exercises. Work must be done on an individual basis.

Programming assignments will involve using the various data structures presented during the term. A moderate-sized course project will be assigned which will utilize the various data structures and algorithms introduced during the term.

Method of Evaluation:

Student learning will be evaluated on the basis of

- Completeness and quality of programming assignments.
- Grade on midterm and final examination.

The weight assigned to each element of evaluation will be determined by the instructor of the course on the first day of the class.

Course Objectives:

The objectives of this course include:

- Introduction and advanced concepts in C++
- Students will develop several medium sized programs in C++
- Cover widely-used searching and sorting algorithms
- Cover commonly used graph algorithms
- Enhance the students' programming skills

Learning Outcomes:

At the end of this course students will

- Be able to write medium-sized C++ programs utilizing STL and an integrated development environment
- Determine which of the common sorting and searching algorithms to utilize for given problems

- Be able to apply and implement graph algorithms in practice

Missed Exams:

Generally, there will be no make-up exams and no incomplete grades given. If you miss an exam, you must notify me before the exam or, if physically impossible, soon after. If any of the scheduled exam dates are in conflict with your religious observances, you must notify me, in writing, at least two weeks in advance of the exam. If you have an acceptable, documented excuse, you may be given a make-up exam or be given the average score of other exam(s) at the discretion of the instructor. Note that a make-up exam will consist of questions/exercises that might have a degree of difficulty that does not match those on the original exam.

Students with disabilities:

Students with disabilities who need reasonable accommodations are encouraged to contact the instructor. The Disability Programs and Resource Center (DPRC) is available to facilitate the reasonable accommodations process. The DPRC is located in the Student Service Building and can be reached by telephone (voice/TTY 415-338-2472) or by email (dprc@sfsu.edu).

Attendance:

“Students are expected to attend classes regularly because classroom work is one of the necessary and important means of learning and of attaining the educational objectives of the institution.” (SFSU Bulletin) To this end, attendance will be taken at different times through the term. Students missing class on a day of attendance will lose attendance points.

Required Textbook:

C++ for Java Programmers (Paperback) by Mark Allen Weiss , Prentice Hall; United States edition (November 7, 2003)

Recommended References:

Data Structures and Abstractions with Java, 2nd Edition, by Frank M. Carrano

Created and approved: November 2009