

CS201 - Accelerated Introduction to Computer Science

Last Updated - 9/1/03

Course Manager - Matthew Bauer, Senior Lecturer

4 credit hours; required for CS & CPE (or CS115/CS116); 150 min. lecture & 100 min. lab each week

Catalog Description - Problem-solving and design using an object-oriented programming language. Introduces a variety of problem solving techniques, algorithms, and data structures in object-oriented programming.

Prerequisites: CS105 or CS 115 or experience using any programming language . (3-2-4)

Course Goals - Students should be able to:

- Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers
- Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers
- Break a problem into logical pieces that can be solved (programmed) independently.
- Develop, and analyze, algorithms for solving simple problems.
- Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
- Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, structures, and class libraries including strings and vectors
- Explain the basics of the concept of recursion.
- Write, test, and debug simple recursive functions and procedures.
- Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding, sub-classing, inheritance, templates
- Use a development environment to design, code, test, and debug simple programs, including multi-file source projects, in an object-oriented programming language.
- Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
- Determine the time complexity of simple algorithms.

Major Topics Covered in Course

1. Fundamental data storage and manipulation (types and variables, statements and expressions)	5 hours
2. Functions	3 hours
3. Classes (classes and objects, instance variables and instance methods, and encapsulation).	5 hours
4. Flow of control (Boolean expressions, conditional statements, and loops).	10 hours
5. Vectors	5 hours
6. 2 nd Object oriented Language (including compare and contrast), 2 nd Programming Environment, Arrays and Pointers, Functions	10 hours
7. Inheritance (subclasses, dynamic binding, abstract classes, and interfaces).	5 hours
8. Strings	3 hours
9. Introduction to recursion.	3 hours
10. Searching and sorting algorithms (linear and binary search, selection sort, insertion sort, and quick sort - introduced via recursive versions).	5 hours
11. Algorithm analysis.	2 hours
12. Problem Solving approaches (This section is dispersed appropriately throughout the semester to illustrate the above techniques.)	5 hours
13. Software Engineering – design, testing, debugging (This section is dispersed appropriately throughout the semester to illustrate the above techniques.)	10 hours
Exams	4 hours
Final Exam	-

