

CS116 – Object-Oriented Programming II

Last Updated - 1/29/04

Course Manager - Matthew Bauer, Senior Lecturer

2 credit hours; required for CS & CPE (or CS201); 75 min. lecture & 75 min. lab each week

Catalog Description - Continuation of CS 115. Introduces more advanced elements of object-oriented programming – including dynamic data structures, recursion, searching and sorting, and advanced object-oriented programming techniques. For students in CS and CS related degree programs. Prerequisite: CS 115 (2-1-2)

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. 2 nd Object oriented Language (including compare and contrast), 2 nd Programming Environment, Arrays and Pointers, Functions	
2. Inheritance (subclasses, dynamic binding, abstract classes, and interfaces).	
3. Strings	
4. Introduction to recursion.	
5. Searching and sorting algorithms (linear and binary search, selection sort, insertion sort, and quick sort - introduced via recursive versions).	

6. Algorithm analysis.	
7. Problem Solving approaches (This section is dispersed appropriately throughout the semester to illustrate the above techniques.)	
8. Software Engineering – design, testing, debugging (This section is dispersed appropriately throughout the semester to illustrate the above techniques.)	
Exams	
Final Exam	