

Kutztown University
Kutztown, Pennsylvania

Computer Science Department
College of Liberal Arts and Sciences

I. Course Description: CSC 136: Computer and Information Science II

This course extends the topics developed in CSC 135. Also covered are concepts of data abstraction and encapsulation as part of the object-oriented paradigm, pointers, recursion, and beginning data structures such as stacks and queues.

3 s.h. 3 c.h. Prerequisite: CSC 135 or equivalent with a C or better.

II. Course Rationale

It is a 2nd year programming course for Computer Science majors. It is required by all CS majors to complete the first year concepts in programming. It is a prerequisite for almost all other CS courses as it provides the basic understanding of elementary data structures and object-oriented programming.

III. Course Objectives

The student will:

1. Incorporate object-oriented techniques into design and implementation of basic objects such as list containers.
2. Demonstrate an understanding of language syntax, including function signatures.
3. Demonstrate an understanding of pointers and issues related to their use
4. Demonstrate an understanding of templates to create generic functions and object-types.
5. Implement of an abstract data type

IV. Course Assessment

The course assessment will be a subset of tests, projects, homework assignments, and final exam.

V. Course Outline

- A. Review of CSC 135: arrays, files, search/sort (as necessary)
- B. Structs (brief)
- C. Classes
 - ◆ basics
 - data attributes
 - member functions
 - ⇒ mutators (sets)
 - ⇒ facilitators
 - ⇒ inspectors (gets)
 - constructors

- ◆ function overloading
- ◆ overloading of operators
- ◆ Declaration in header, implementation in .cpp, application/driver
 - Class Definition (header (.h) files)
 - ifndef for protection from multiple includes
 - public, private, protected – the information hiding principle
 - friend functions
 - Class Implementation (source (.cpp) files)
 - implementation of friends
- ◆ Programming with classes
 - Interaction between classes and applications
 - makefiles
- D. Pointers
 - ◆ to simple types
 - ◆ as data attributes in classes
 - copy constructor, = operator, destructor
 - shallow vs. deep copy
 - Array as a class (Array ADT)
- E. Debugging Tool
 - single file program
 - Multiple file projects
- F. Templates
 - ◆ Functions
 - ◆ Classes
 - Array ADT
 - h file implementation
 - two-file implementation
 - ⇒ Explicit Instantiation
- G. Linked Lists
 - ◆ Nodes
- H. Simple Recursion
- I. Stacks
 - ◆ Implementations
 - ◆ Applications
- J. Queues
 - ◆ Implementations
 - ◆ Applications
- K. Other Topics as Time Permits
 - ◆ STL vectors
 - ◆ trees

VI. Instructional Resources

Childs, J. D. C++ : classes and data structures, Harlow : Prentice Hall, (2007).

Dahl, O. J., Dijkstra, E. W., and Hoare, C. A. R. *Structured Programming*. San Diego, CA: Academic Press, (1972).

Friedman, F. L., Koffman, E. B., *Problem solving, abstraction, and design using C++ / 6th ed.* Boston : Pearson Addison-Wesley, (2010).

Gaddis, T., Walters, J., and Muganda, G., *Starting Out With C++, 4th Edition*, El Granada, CA: Scott-Jones, (2004).

Gaddis, T., *Starting out with C++ : From control structures through objects, Brief Edition, 7th ed.* Boston : Addison-Wesley, (2011).

Headington, M. R. and Riley, D. D. *Data Abstration and Structures Using C++*. Lexington, MA: D. C. Heath and Company (1994).

Kernighan, B. S. and Plauger, P. J. *The Elements of Programming Style*. Highstown, NJ: McGraw-Hill, Inc. (1974).

Knuth, D. E. *The Art of Computer Programming - Fundamental Algorithms. Vol. 1. (2nd edition)* Reading, MA: Addison-Wesley (1973).

Liang, Y. D., *Introduction to programming with C++ : comprehensive version*, Upper Saddle River, NJ: Pearson/Prentice Hall, (2007).

Main, M, *Data Structures and Other Objects in C++, 4th ed.* Reading, MA: Addison- Wesley, (2010).

Malik, D.S., *C++ Programming: From Problem Analysis to Program Design (Introduction to Programming), 5th ed.* Cambridge, MA: Course Technology (2010)

Mitchell, E., *Object-oriented programming from square one* Carmel, IN : Que, (1993).

Savitch, W. J., *Problem solving with C++ 8th ed.* Boston : Pearson/Addison-Wesley, (2011).

Stroustrup, B., *The C++ Programming Language*, Reading, Mass. : Addison-Wesley, (1986)

VanThesel, D. *Program Style, Design, Efficiency, Debugging, and Testing*. Englewood Cliffs, NJ: Prentice-Hall, Inc. (1978).

Wirth, N. *Algorithms + Data Structures = Programs*. Englewood Cliffs, NJ: Prentice Hall (1976).

Periodicals:

- Journal of the ACM
- Communications of the ACM
- Computing Reviews
- Computing Surveys
- Computerworld
- Byte
- On-Computing
- Popular Computing