Kutztown University
Kutztown, Pennsylvania

Computer Science Department
College of Liberal Arts and Sciences

I. Course Description: CSC 136: Computer 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

This course is a second year programming course for computer science majors. This course is required by all computer science majors to complete the first year concepts in programming. This course is a prerequisite for many other computer science courses as it provides the basic understanding of elementary data structures and object-oriented programming.

III. Course Objectives

The student will:

A. Incorporate object-oriented techniques into design and implementation of basic objects such as list containers.

B. Implement pointers and address issues related to their use.

C. Demonstrate the ability to debug programs.

D. Employ templates to create generic functions and object-types.

E. Explain basic recursive functions.

F. Describe 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
    1. basics
        a. data attributes
        b. member functions
            i. mutators (sets)
            ii. facilitators
            iii. inspectors (gets)

c. constructors
2. function overloading
3. overloading of operators
4. Declaration in header, implementation in .cpp, application/driver
a. Class Definition (header (.h) files)
b. ifndef for protection from multiple includes
c. public, private, protected – the information hiding principle
d. friend functions
e. Class Implementation (source (.cpp) files)
f. implementation of friends
5. Programming with classes
a. Interaction between classes and applications
b. makefiles

D. Pointers
1. to simple types
2. as data attributes in classes
a. copy constructor, = operator, destructor
b. shallow vs. deep copy
c. Array as a class

E. Debugging Tool
1. single file program
2. Multiple file projects

F. Templates
1. Functions
2. Classes
a. List Container
b. h file implementation
c. two-file implementation
i. Explicit Instantiation

G. Linked Lists
1. Nodes

H. Simple Recursion

I. Stacks
1. Abstract Data Type (ADT)
2. Implementations
3. Applications

J. Queues
1. ADT
2. Implementations
3. Applications

K. Additional Data Structures
1. STL vectors
2. Trees

VI. Instructional Resources

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

Dale, N., Weems, C., Richards, T., *C++ Plus Data Structures 6th Edition*, Jones & Bartlett, (2016).

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, 8th ed.* Boston : Addison-Wesley, (2015).

Grimes, R., *Beginning C++ Programming,* Packt Publishing, (2017).

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), 7th ed.* Cambridge, MA: Course Technology (2014)

Savitch, W. J., *Problem solving with* ***C++*** *9th ed*. Boston : Pearson/Addison-Wesley, (2014).

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