

CS-230 Data Structures Syllabus

Course Information

Class Days: Monday, Wednesday, Friday
Class Time: 1:00 – 1:50 p.m.
Class Room: KT-217
Instructor: Dr. Iyad A. Ajwa
E-mail: iajwa@ashland.edu
Office: PTC-203
Office Phone: 419.289.5798

Office Hours: Monday – Friday: 8:15 a.m. – 9:00 a.m. and 12:00 noon – 1:00 p.m.
Monday, Wednesday, Friday: 11:00 a.m. – 12:00 p.m. and 2:00 p.m. – 3:00 p.m.
and by appointment.

Course Materials

- **Required Textbook:** Data Structures Using C++, Second Edition, by D. S. Malik, ISBN-10: 0324782012, ISBN-13: 9780324782011, ©2010 by Course Technology.

Course Description

Formal specification of abstract data types using the C++ programming language with examples of their uses in computer science: records, stacks, queues, linked lists, hash tables, heaps, and trees.

Student Outcomes

Data Structures is an introduction to computer data representation from an object-oriented perspective. Each of the major data structures (stacks, queues, linked lists, trees, and graphs) is developed and realized as a C++ class. Each collection class is then tested and used in an application such as a case study or series of complete programs. The main goals of this course are as follows:

1. Students get an integrated presentation of fundamental data structures and accompanying object-oriented programming methodology.
2. Students understanding of the structures and object-oriented methodology are reinforced by learning how to create reusable data structures with class templates and inheritance.
3. Students be able to create and manipulate dynamic data structures and continue to learn and truly utilize the power of C++.
4. Students understand various important applications of data structures and learn that data structure use is simplified by utilizing the powerful constructs of the C++ programming language.

Student Assessment Criteria

Overall student performance will be assessed using programming assignments (35%), class tests (45%), and the final examination (20%).

- **Programming Assignments:** The production of quality software requires, among other things, that it be timely. With this view in mind, you should expect ten (10) computer-use assignments that will be posted on ANGEL. Programming assignments vary in length and complexity. They should be done by the deadlines announced. Late assignments will be penalized 20% for the first day and 5% per calendar day after that. Each programming assignment must be submitted in the

drop box designated for that assignment on ANGEL. No other means of submission, including email, will be accepted.

- **Class Tests:** Three (3) in-class tests will be given during the course of the semester. Make-ups will be given if the student misses a test on the scheduled date for a legitimate reason **and** prior arrangements with the instructor are made. A make-up exam must be taken before the graded exam is returned to class. Below is a schedule for all class tests. Please note that dates of class tests are subject to change. Any changes will be announced in class.

TEST I:	Friday	September 23, 2011
TEST II:	Wednesday	October 19, 2011
TEST III:	Wednesday	December 7, 2011

- **The Final Examination** will be comprehensive. The following date and time of the Final Examination is announced by the Registrar's Office and cannot be changed except as outlined in the Student Handbook. Do not make any plans that would prevent you from taking the final exam at the scheduled time. All students must take the final exam.

Monday December 14, 2011 (8:00 a.m. – 10:00 a.m.)

- **Letter grades** will be assigned according to the following grading scale:
(100 ≥ A ≥ 93) (93 > A- ≥ 90) (90 > B+ ≥ 87) (87 > B ≥ 84) (84 > B- ≥ 80) (80 > C+ ≥ 77)
(77 > C ≥ 74) (74 > C- ≥ 70) (70 > D+ ≥ 67) (67 > D ≥ 64) (64 > D- ≥ 60) (60 > F ≥ 0)

Course Guidelines

- **Teaching Methodology:** Material will be presented in class through lectures and assigned readings in the textbook. Microcomputer software, including *Microsoft® Visual Studio 2008*, will be used extensively for developing programs. Class notes and miscellaneous material will be made available on Angel as appropriate. Students are encouraged to actively participate in class discussions. Programming assignments will be used to reinforce your understanding of the methodologies presented. Students will work individually to complete programming assignments.
- **Academic Integrity** is an important element of effective learning. All students are expected to read, understand, and follow the Academic Integrity Policy specified in the Ashland University Student Handbook. For exams, this means that students may not give or receive assistance on the exam (except from me). For graded programming assignments, students are permitted to work in groups and share information freely. However, copying another student's program on individual assignments is a violation of the policy (and defeats the purpose of doing the assignments) and is not permitted. If you do not understand how to do an assignment, contact me by telephone or email or come by my office, and I will be happy to assist you in a way that will increase your understanding of the assignment. Forms of academic dishonesty will be handled in accordance with University policy. The minimum punishment for plagiarism, cheating or other academic integrity violations is a zero on the exam or assignment on which the violation occurred. The maximum punishment is a grade of "F" for the course.
- **Student Accommodations:** Students with documented disabilities who require academic adjustments for this class are requested to contact me to discuss accommodations. While not required, it is in the best interest of the student to have this conversation early in the semester. In order to receive academic adjustments, paperwork from Disability Services must be provided to document this need. Disability Services is located in 105 Amstutz, extension 5953.

- **Course Outline:** The following is a tentative course schedule. Dates are subject to change.

Week	Topic	Reading Assignment	Week	Topic	Reading Assignment
1	Software Engineering Principles and C++ Classes	Chapter 1	8	Queues	Chapter 8
2	Object-Oriented Design (OOD) and C++	Chapter 2	9	Searching and Hashing Algorithms	Chapter 9
3	Pointers and Array-Based Lists	Chapter 3	10 & 11	Sorting Algorithms	Chapter 10
4	Standard Template Library (STL) I	Chapter 4	12	Binary Trees and B-Trees	Chapter 11
5 & 6	Linked Lists	Chapter 5	13	Graphs	Chapter 12
7	Stacks	Chapter 7	14	Standard Template Library (STL) II	Chapter 13