**Fall 2024 Course Syllabus**

**CSC 240-0C1**

**C++ Data Structures**

**Instructor: Frank Alvino**

**Office Hours:** Fridays 1pm-2pm or by appointment.

**Course Dates:** 8/18/2025 to 12/10/2025

**Office Phone:** 224-698-7470

**Text Messaging:** 224-698-7470**;**

**Email:** [**falvino@oakton.edu**](mailto:falvino@oakton.edu)

Please include your name and course (CSC 240) on all email and text messages.

**D2L login site:** [**https://d2l.oakton.edu**](https://d2l.oakton.edu)

Your D2L login credentials are the same as your Oakton login credentials.

**Source Code:** see **Source Code** link in **Course Files and Resource Links** under **Content** in D2L. Some of the Labs and Exercises will have supporting files in the same D2L module.

**I. Course Prefix Course Number Course Name Credit Lecture Lab**

**CSC 240 C++ Data Structures 3 3 1**

**II. Prerequisite**

CSC 155 or CSC 170 and CSC 171 with minimum of grades C.

**III. Course Description**

A second course in computer programming using the C++ programming language that provides a survey of data structures including files, lists, sets, trees, tables, queues, stacks, graphs, and other classes. Content covers: utilization of object-oriented design techniques to implement large-scale problems; program verification and complexity analysis; dynamic concepts; sorting and searching algorithms; abstract data types; recursion.

**IV. Learning Objectives**

1. Use stream I/O: files and console for basic interaction within the program interface. 2. Use object-oriented programming skills to design maintainable classes in separate class files and employ multiple file compilations.
2. Design and develop simple class hierarchies with the use of templates and abstract classes.
3. Apply the concepts of dynamic binding, polymorphism, inheritance, and method overloading.
4. Write expressions that use reference and static variables as data members and method arguments, and class methods for object assignment and copy constructors.
5. Explain the concepts of information hiding, including the difference between public and private data members and methods.
6. Explain separating program implementation from the programming interface.
7. Analyze various algorithms for memory optimization, time complexity, and correctness in order to understand efficiency.
8. Implement various sort and search algorithms throughout the various fundamental data structures studied.
9. Apply recursive programming techniques when appropriate and contrast the iterative versions.
10. Use dynamic and static memory to write data structures such as linked lists, stacks, queues, binary trees, and tables.
11. Write and use basic functionality of data structures such as insert, remove and iterative schemes.

**V. Academic Integrity and Student Conduct**

Students and employees at Oakton College are required to demonstrate academic integrity and follow Oakton’s Code of Academic Conduct. This code prohibits:

* cheating,
* plagiarism (turning in work not written by you, or lacking proper citation),
* falsification and fabrication (lying or distorting the truth),
* helping others to cheat,
* unauthorized changes on official documents,
* pretending to be someone else or having someone else pretend to be you,
* making or accepting bribes, special favors, or threats, and
* any other behavior that violates academic integrity.

There are serious consequences to violations of the academic integrity policy. Oakton’s policies and procedures provide students a fair hearing if a complaint is made against you. If you are found to have violated the policy, the minimum penalty is failure on the assignment and, a disciplinary record will be established and kept on file in the office of the Vice President for Student Affairs for a period of 3 years.

Please review the Code of Academic Conduct **and** the Code of Student Conduct, both located online at [***www.oakton.edu/studentlife/student-handbook.pdf***](http://www.oakton.edu/studentlife/student-handbook.pdf)***.***

**What happens if I cheat?**

A grade of 0 will be given to a test or assignment where there is evidence of academic misconduct. This includes any form of plagiarism. All programs must be completed individually by the student. Copying code from any other source other than our textbook source code is considered plagiarism and will not be accepted. Do not attempt to copy/use another students’ code. Do not try to use a “tutoring” website for solutions to your assignments, e.g. chegg.com or chatgpt.com. No matter what the circumstance is, if there is any matching code with another students’ code, it will be considered cheating. The instructor has over a decade of recognizing copied code—do not attempt to use any code other than your own.

If copied code is submitted, you will receive a grade of 0 on the assignment (with no opportunity for a replacement assignment) and an Academic Integrity Violation form will be filed with the Office of Student Affairs.

If and when you are caught cheating, you automatically lose the opportunity to earn any extra credit points for the course—even completed extra credit assignments will not be honored for points; and, the instructor will not write you a letter of recommendation—ever.

If a tutor is involved with any code a student acquired assistance with to complete their assignments, the code for the assignment must be written by the student and all of that code must include the tutor’s name(s) and contact information in a commented section preceding it.

Each student is responsible for understanding their own code and will be held accountable for their programs’ content. If a student cannot complete the assignments on their own, a Withdrawal from the course is recommended.

Students have been caught in the past and have received a score of 0 on the copied code submitted and have been found guilty by the Student Affairs Office for cheating.

Don’t cheat—you are only cheating yourself.

**VI. Outline of Topics**

**A. Program Development**

1. Complexity analysis of efficiency

a. Big-O notation

2. Design strategies

a. Modularity

b. Recursion

c. Object-oriented class relationships

1. Inheritance and encapsulation
2. Abstract templates
3. Dynamic binding

3. Program verification

**B. Abstract data design**

1. Lists

a. Stacks and queues

b. Variations

2. Trees

a. Binary search trees

b. Tree traversals

3. Tables

a. Priority queues/heaps

4. Graphs

5. Hashing

a. Hash tables

b. Hash maps

**C. Algorithms**

1. Sorting

a. Selection and insertion sorts

b. Merge, heap, and quick sorts

2. Searching

a. Binary and sequential searches

b. Hash functions

c. Breadth and depth first searches

**VII. Methods of Instruction**

This course is an asynchronous 8-week (half semester) online course and all lectures and supporting content for the course can be found in D2L.

There will be various assignments in the form of programming Labs and Exercises, Quizzes, Exams, and Discussions.

All of the Labs/Exercises to be submitted for a grade are uploaded into their appropriate Assignment folder via D2L.

All Discussion assignments are found within D2L.

The online virtual meeting hours can be accessed via D2L under Content at Virtual Meeting.

**VIII. Course Practices Required**

1. Complete Lab/Exercise assignments with the ability to demonstrate the proper use of the algorithms and logic identified.

2. Engage in the Discussion assignments, as identified in the discussion guidelines and rubric in D2L.

3. Reading the textbook is necessary to understand the course materials. Reading the textbook will be required for reference to the language syntax.

4. Exams are taken at home.

5. Each student is required to communicate via their preferred Oakton email address only. **Please give a subject and identify which course you are emailing about.**

6. When using the text messaging number, please use it for course related messages only. **Please give a subject and identify which course you are texting about.**

**IX. Instructional Materials**

***Required Textbook:***C++ Plus Data Structures, 6th Edition, by Dale, 2018, Jones & Bartlett.

ISBN: 9781284089189

Access to D2L for online Content.

Note: Current textbook information for each course and section is available on

Oakton’s Schedule of Classes.

If a Direct Digital Access (DDA) fee has been charged as part of tuition and fees,

viewable on myOakton, students have the option to opt-out of the e-book(s) and

fee(s). Please email ddaoptout@oakton.edu for more information.

**X. Methods of Evaluating Student Progress**

Grading Scale (as percent):

100-90 A

89-80 B

79-70 C

69-60 D

Below 60 F

4 Quizzes each worth 10 points each.

15 Discussions Posts each worth 5 points.

15 Assignment each worth 20 points consisting of (labs, exercises and problems).

Midterm and final exams each worth 50 points each.

**The course is worth 515 total points.**

**Exams:**

Exam topics are cumulative with the preceding Lab assignments and text readings.

All Exams are open book (hard copy or eText/digital/online) including any D2L course documentation (e.g. ppt files, or lab documents) and open IDE (your choice).

Each Exam is to be taken during the week(s) identified in the Course Schedule.

Exams are taken via D2L

The two Exams are mostly T/F, multiple choice, or short fill-in-the-blank questions.

Each Exam may have a written coding portion, which will require the writing of C++ code. The topics for each Exam are listed in the Course Schedule.

Exams are graded manually by the instructor to rectify any discrepancies in the auto-grading of the Exams. After initially completing an Exam, your score will typically be lower than what is expected. Please be patient with the instructor and allow for some time for manual grading to be administered—your final score will likely be higher than the initial score. You will be able to review the incorrect answers on the Exams once they are manually graded by the instructor.

**Lab/Exercise/Problem Assignments Submission (**watch D2L video under Content → Course Files and Resource Links → Eclipse and Creating C++ Projects for Assignments**):**

Upload the completed Labs/Exercises into the appropriate D2L Assignment folder. Only upload the .cpp and .h files and/or any necessary input text files (e.g. .txt). If a Lab/Exercise requires more than one file, zip **all of the necessary files required for that particular assignment** into one folder—this includes any input files.

Problems will be shorter assignments focusing on the topic of the week. Many of the problems will be taken from the text book.

The .zip file should use a naming convention similar to the following which includes your last name and the name of the assignment:

**Lab<Number>\_<Last Name>.zip.**

**For example: Lab1\_Alvino.zip**

It is not necessary to zip a single file, if submitting a **single** file into the Assignment folder. In this case, use the same naming convention as above for the file name.

For example: **Lab1\_Alvino.cpp**

Labs/Exercises must be submitted into their appropriate Assignment folder if the student wants to receive points towards their calculated grade. Only the most recently submitted Lab/Exercise file(s) are graded.

**Incomplete Course Grade of “I”:**

Only given in extreme circumstances.

Incomplete grades are only given to students who have shown prior work and have earned at least a “C” on their midterm and have a “C” average for all of the completed assignments (Labs and Exercises).

**Incomplete Labs/Exercises:**

Lab/Exercise will receive a score of 0 if any of the following apply:

* Compilation errors
* Runtime errors
* Multiple unzipped files are submitted
* Input files are missing
* Missing files necessary for compilation

**Lab/Exercise Completion:**

Lab/Exercise will earn full points if it:

* compiles with NO errors
* follows the specifications
* contains proper comments
* contains a good variable naming convention
* contains proper scoping/indentation of code blocks
* produces NO runtime errors

Partial credit on Labs/Exercises can be earned. Typically, a Lab/Exercise submitted receives full points if all six steps above are covered.

**Rubric for coding style:**

The coding style of the programming assignment is worth 10% of the points earned for the assignment. Use the Coding Conventions at Oakton College C++ document found in Extra Files under Content in D2L as a reference.

* 40% proper program documentation (main and functions)
* 30% proper indentation of code blocks
* 20% appropriate variable and function naming
* 10% correct output

**Programming Assignment Redo:**

If you do not receive full credit for a submitted assignment, you can redo it.

If an assignment does not receive full points, you can resubmit it for full points.

**All assignments including Extra Credit must be resubmitted by 12/11/2024 (11:59 pm) for points.**

**A driver program is the main code block of your code.**

**Here is an example of a driver program:**

**Lab1\_Temesvari.cpp**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Author: D.S. Malik**

**// This program illustrates how to use the STL class queue in a**

**// program.**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**#include <iostream>**

**#include <queue>**

**using namespace std;**

**int main()**

**{**

**queue<int> intQueue;**

**intQueue.push(26);**

**intQueue.push(18);**

**intQueue.push(50);**

**intQueue.push(33);**

**cout << "Line 11: The front element of intQueue: "**

**<< intQueue.front() << endl;**

**cout << "Line 12: The last element of intQueue: "**

**<< intQueue.back() << endl;**

**intQueue.pop();**

**cout << "Line 14: After the pop operation, the "**

**<< "front element of intQueue: "**

**<< intQueue.front() << endl;**

**cout << "Line 15: intQueue elements: ";**

**while (!intQueue.empty())**

**{**

**cout << intQueue.front() << " ";**

**intQueue.pop();**

**}**

**cout << endl;**

**return 0;**

**}**

**XI. Other Course Information**

Tuesday, November 17th—Last day to Withdraw with a “W” from 8-week (2nd half) courses

Wednesday, December 12th—LAST DAY OF SEMESTER

1. Oakton College values equity and the diversity of learners and community members we serve. We strive to make all learning experiences as accessible as possible. If you encounter a course, program or activity that is not accessible, or you wish to request disability-related accommodations, please contact the Access and Disability Resource Center at [accessdisability@oakton.edu](mailto:accessdisability@oakton.edu) or 847-635-1489.

2. Oakton College is committed to maintaining a campus environment emphasizing the dignity and worth of all members of the community, and complies with all federal and state Title IX requirements.

Oakton College is committed to combatting sexual misconduct. As a result, college faculty and staff members are required to report any instances of sexual misconduct, including harassment and sexual violence, to the Title IX Coordinator so that all parties involved may be provided appropriate resources and support options. There are two important exceptions to this requirement:

1. A list of the college’s Confidential Advisors who, as counselors and medical professionals, do not have this reporting responsibility and can maintain confidentiality, can be found here:

[*https://www.oakton.edu/studentservices/counseling/contact/*](https://www.oakton.edu/studentservices/counseling/contact/)

1. An important exception to the reporting requirement exists for academic work. Disclosures about sexual misconduct that are shared as part of an academic project, classroom discussion, or course assignment, are not required to be disclosed to the college’s Title IX office.

Students who have been subjected to any form of sexual misconduct, are encouraged to access these resources:

* Office of Student Affairs: 847-635-1745
* SHARE @ the Wellness Center: 847-635-1885 (8:30 AM TO 5 PM M-F) and 847-635-1880 (after business hours to reach the Counselor on Call)
* Oakton’s Title IX webpage: [*https://www.oakton.edu/about/title\_ix/index.php*](https://www.oakton.edu/about/title_ix/index.php%20)

If there is a safety concern, please contact Oakton Police Department, 847-635-1880.

Other reporting information is available here: [*https://cm.maxient.com/reportingform.php?OaktonCC&layout\_id=2*](https://cm.maxient.com/reportingform.php?OaktonCC&layout_id=2)

Resources and support for:

* Pregnancy-related and parenting accommodations can be found at: [*https://www.oakton.edu/about/title\_ix/pregnancy\_parenting.php*](https://www.oakton.edu/about/title_ix/pregnancy_parenting.php)
* Resources and support for LGBTQ+ students can be found at: [*https://www.oakton.edu/life-at-oakton/diversity-at-oakton/lgbtq+.php*](http://www.oakton.edu/life-at-oakton/diversity-at-oakton/lgbtq+.php)

1. Electronic video and/or audio recording is not permitted during class unless the student obtains written permission from the instructor. In cases where recordings are allowed, such content is restricted to personal use only. Any distribution of such recordings is strictly prohibited. Personal use is defined as use by an individual student for the purpose of studying or completing course assignments.

For students who have been approved for audio and/or video recording of lectures and other classroom activities as a reasonable accommodation by Oakton’s Access Disabilities Resource Center (ADRC), applicable federal law requires instructors to permit those recordings. Such recordings are also limited to personal use. Any distribution of such recordings is strictly prohibited.

Violation of this policy will result in disciplinary action through the Code of Student Conduct.

1. Oakton Emergency Building Closures. When an Oakton building is closed due to an emergency (e.g., weather-related closings), that doesn't mean instruction stops. Students should check for communications from their instructor via D2L, other learning environments such as MyMathLab, or via email or text. If you don't receive communication from your instructor within 24 hours of building closure, please contact them.