

CSC402: Syllabus

Contact Information

Instructor: Eric J. Fredericks

Home Page: <https://www.cdm.depaul.edu/Faculty-and-Staff/Pages/faculty-info.aspx?fid=1604>

Email: eric.fredericks@depaul.edu

Phone:

Address: School of Computing, DePaul University
14 East Jackson Boulevard
Chicago, IL 60604

Office: Classroom

Class Page: <https://reed.cs.depaul.edu/efredericks/ds1>

Class Hours: Mon 5:45pm-9:00pm

- **NOTE: There is another person at DePaul with my name. Be sure you are emailing the correct person if you email me!**
- **Office hours** are on [Zoom](#) on **Mondays**, from **4:45pm - 5:30pm** and again at **9:15pm - 10:00pm**. I will be available **in person** in the classroom before and after class, or virtually, on [Zoom](#).
- If you would like to visit me for office hours, try to join me in-person if you can. It's easier for me to draw diagrams and help you visualize concepts in person. However, Zoom is an option if you need it.
- If you would like to talk outside of office hours, please send me a message on Discord or email me -- I don't check voicemail often.
- You can expect that I will respond to email and Discord messages within 24 hours on business days.
- Note that I work a full-time job in addition to teaching at DePaul CDM, so I may not be able to answer questions during the workday. Note also that I have commitments on weekends that may keep me from my computer. You should plan to have all of your homework questions to me no later than Friday. I cannot guarantee that I can get back to you before your homework is due.
- You can expect grades to be posted within a week of the due date.
- This syllabus is subject to change as necessary during the quarter. If a change occurs, it will be thoroughly addressed during class, posted under Announcements in D2L and sent via email.

Course Homepage for Lectures

Course Homepage: <https://reed.cs.depaul.edu/efredericks/ds1>

- Course content, homework files, and most of the information you need will be available on this site.
- Lecture slides may not be available before class.
- Each week's content may change substantially before the in-person section meets. There may be old information that I have not yet updated, so do not rely on the course site for anything too far in the future.
- Homework assignments may not be accurate until they are presented and assigned in class.
- Lecture slides may change after class.

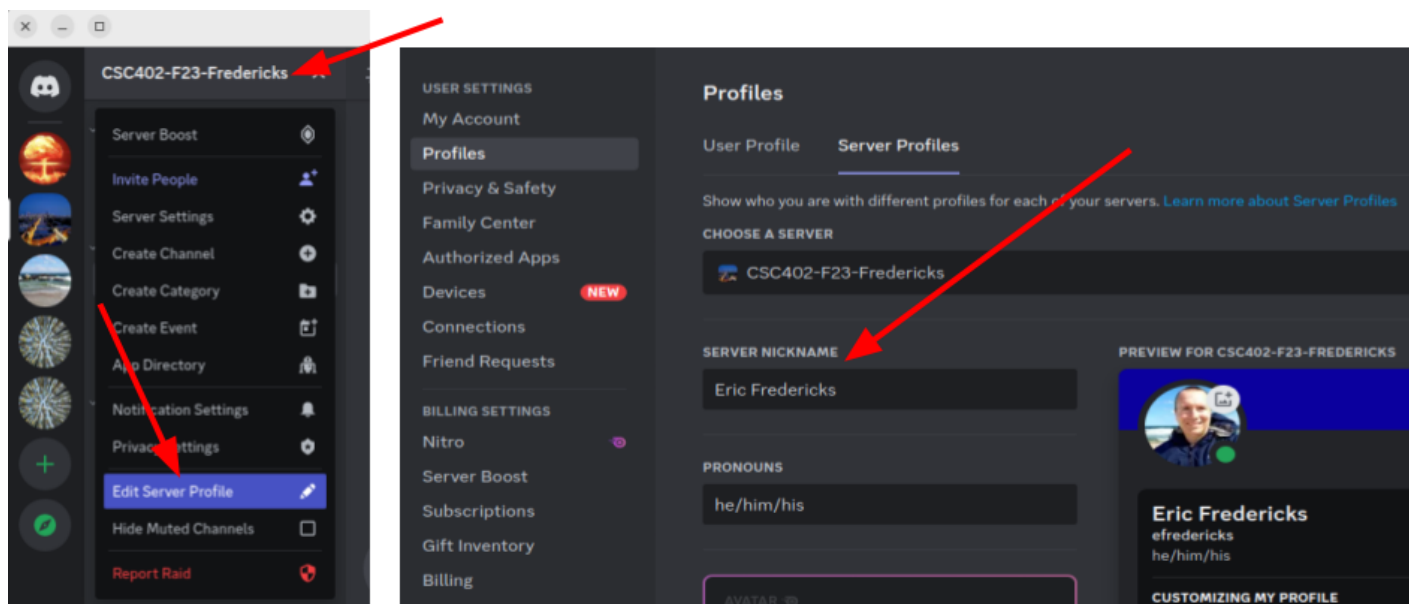
Course D2L Page: [D2L](#)

- Use D2L for homework submission, registering for exam proctoring, and for other information.

Discussion Group

We will use Discord as a discussion forum for class. It is my intention to create a learning community on Discord, so you should feel free to ask questions and engage with classmates on the material there.

- [Download](#) the native Discord app for your computer and/or mobile device.
- Click here to join the Discord server for the class: [Discord Discussion Forum](#)
- Change your **Server Nickname** to be your first and last name. Use whatever firstname you prefer, but be sure that your last name is the same as it is in DePaul's Campus Connect.



- If you have a common last name, be sure to provide some indication of who you are so we can identify you.
- We have all types of students in class. Be sure to use appropriate language.
- Don't [troll](#).

Overview

This is the first course in a two-course sequence on data structures using Java. The course introduces basic Java programming, reviews recursion, introduces asymptotic notations, and focuses mainly on linear data structures including arrays, linked lists and their variants, stacks and queues, and data structures supporting disjoint-set operations. The implementation of the basic operations on each data structure are discussed and analyzed in terms of their efficiency. The applications covered highlight and exploit the unique characteristics of the data structures, and emphasize problem solving and recursive thinking.

Objectives

Programs are not just for computers: We use them to communicate to other people.

- Ability to code without typing (on paper/whiteboard)
- Understanding correctness of programming
- Understanding performance characteristics of programs
- Competence with iterative and recursive solutions to problems
- Basic object-oriented programming
- Use of debuggers
- Use of testing

Plus

- Understanding of linked lists and “resizable arrays”
- Understanding of stacks, queues, deques
- Understanding of heaps
- Understanding of union-find

A **data structure** is a concrete implementation of an abstract type.

- For example, linked lists and resizable arrays are data structures that could be used to represent the “list” type in python
- Which does python use? Why?

Lecture Plan

The following lecture plan is tentative and subject to change as the course progresses.

- **Class 1:** [2023/09/11] Arrays and Loops (1.1)
- **Class 2:** [2023/09/18] Java Memory, Debugging and Tracing (1.1, 1.2)
- **Class 3:** [2023/09/25] Abstract Data Types, Linked Structures (1.2, 1.3)
- **Class 4:** [2023/10/02] Stacks and Queues, Recursion (1.1, 1.3)
- **Class 5:** [2023/10/09] Counting and Intro to Analysis (1.3, 1.4)
- **Class 6:** [2023/10/16] Midterm, More Analysis (1.4)
- **Class 7:** [2023/10/23] Nested Loops, Union Find (1.4, 1.5)
- **Class 8:** [2023/10/30] Elementary Comparison-Based Sorting (1.2, 2.1, 2.5)
- **Class 9:** [2023/11/06] Priority Queues and Heaps (2.4, 6.1)

- **Class 10:** [2023/11/13] Review
- **Class 11:** [2023/11/20] Final exam

Lecture slides will be available after each lecture. They *may* be available before the lecture, but may change.

Prerequisites

A prior programming class.

Textbooks

If you are delayed in getting the texts, you can view them online at [O'Reilly](#).

Required Books

[Core Java SE 9 for the Impatient, 2nd Edition](#) [[Amazon](#), [Indiebound](#)]

by Cay Horstmann (Addison-Wesley, 2017)

Available as [Ebook](#)

([Online version](#))

[Companion site](#).

Older edition is fine.

[Algorithms 4e](#) [[Amazon](#), [Indiebound](#)]

by Robert Sedgewick and Kevin Wayne (Addison-Wesley, 2011)

Available as [Ebook](#)

([Online version](#))

([Author videos](#)) These are also for sale as an [Ebook](#)

[Companion site](#).

Do not get an older edition. They are completely different books.

Recommended Books

[Schaum's Outline of Data Structures with Java 2e](#) [[Amazon](#), [Indiebound](#)]

by John Hubbard (Schuams, 2009)

This book is a good source of example problems with solutions.

More Books

[How to Think Like a Computer Scientist](#)

by Allen B. Downey.

Free!

An good introduction to Java.

Skip the GridWorld chapters, which are intended to help with the AP exam in CS.

See also these [lecture notes from MIT](#). The first three lectures are particularly useful.

[Java for Python Programmers](#)

by Brad Miller.

Free!

[Introduction to Programming in Java \(Chapter 1\)](#)

by Robert Sedgewick and Kevin Wayne

Free!

This is the first chapter of the introductory text written by the authors of our primary textbook.

It presents the same material as section 1.1 of the primary text, but at a slower pace.

[Effective Java 3e](#) [[Amazon](#), [Indiebound](#)]

by Joshua Bloch (Addison-Wesley, 2008)

Available as [Ebook](#)

([Online version](#))

The algorithms text describes all of the Java that is required for the class. The discussion is terse, making it an excellent reference. If you would like a longer discussion of Java, you might want a supplementary text. In this case, you might consider one of the following.

- Kathy Sierra and Bert Bates's "Head First Java" ([Online version](#))
- Bruce Eckel's [Thinking in Java, 4th Edition](#)
- David Flanagan's "Java in a Nutshell" ([Online version](#))
- Cay Horstman's "Big Java" ([Online version](#))

Expectations

We will discuss new concepts each week.

You will have weekly assignments, including a quiz, each week. Some assignments are programming assignments and some assignments are written assignments.

We will review homeworks and quizzes in class. We will go over concepts that students find particularly difficult.

- You do not have to do it alone, but you must be self-motivated.
- You can ask me and other members of the class for help. Discord is great for getting help quickly!
- You must start assignments early. I will do my best to help you whenever I can, but I cannot guarantee that I can respond to questions over the weekend.
- You must complete the weekly quizzes on D2L.
- You must submit all homeworks to the *correct* folder on D2L. I cannot be responsible for finding your homework if you submit it to the wrong folder.

Getting the homework correct is not enough. **Simply getting programming homework to work once will not help you communicate to me that you understand the concepts you need to learn.** You need to learn how to draw pictures diagramming concepts we are covering, and to write code on paper, without a compiler, notes, Google, etc.

Repetition and iteration are key to learning! I recommend that when you repeat homeworks on your own, you get used to *writing Java code on paper*. Write your code out on paper and then type it into IntelliJ IDEA. Compile and run the tests, and fix any problems. Spend no more than 20 minutes on any one problem per day. Keep doing this every day, and you will find that you will be able to write good clean code on paper that is intelligible to me when I'm grading your exams. If you write code on exams that indicates you have not internalized Java syntax and semantics, your exam grade will reflect this!

Deadlines are generally not individually negotiable. First of all, this course moves very fast and covers a lot of ground, so you will want to keep up with the material. Secondly, I want to get you feedback on your assignments quickly, so you can use that feedback as you study. And finally, I cannot keep up with grading multiple students' late assignments - it becomes very time consuming to context switch back to previous assignments.

Since I drop one quiz and at least one homework assignment, you are covered if you miss an assignment due to unforeseen circumstances. **Under no circumstances can deadlines be extended without working with the Dean of Students' office** (see *Absence Notifications* below).

Attendance

If you are in Section 701, DePaul University, and I, expect you to attend class!

• Incomplete Grades

An incomplete grade is defined in the Student Handbook as follows (note that the policy in the undergraduate student handbook applies to both undergraduate and graduate students): A temporary grade indicating that the student has a satisfactory record in work completed, but for unusual or unforeseeable circumstances not encountered by other students in the class and acceptable to the instructor is prevented from completing the course requirements by the end of the term. Please see <https://www.cdm.depaul.edu/Current%20Students/Pages/Grading-Policies.aspx> for additional information.

• Retro-Active Withdrawal

CDM understands certain extenuating circumstances can hinder one's ability for academic success and completion of course work. Please see <https://www.cdm.depaul.edu/Current%20Students/Pages/Enrollment-Policies.aspx> for additional information.

• Absence Notifications

In order to petition for an excused absence, students who miss class due to illness or significant personal circumstances should complete the Absence Notification process through the Dean of Students office. The form can be accessed at <https://offices.depaul.edu/student-affairs/support-services/academic/Pages/default.aspx>. Students must submit supporting documentation alongside the form. The professor reserves the sole right whether to offer an excused absence and/or academic accommodations for an excused absence.

Assessment

Grades will be determined as follows.

- **20% Quizzes**
- **25% Programming assignments**
 - To solve the weekly homeworks and quizzes, you may find it useful to consult external resources at first. That is acceptable for this class. It is not an **academic integrity violation**.
 - However, it is your responsibility to **repeat each problem** until you can do it yourself, starting from scratch, in a reasonable amount of time (about five minutes).
 - Once you have ensured that you can complete the problem using the IDE, you must then ensure that you can do it **on paper**. This is a crucial part of the homework!
 - If you cannot produce the solution, on paper, without assistance, then you have not understood it. You will not do well on exams or job interviews.
 - Programming assignments that do not compile will receive **zero points**. You must type in the homework and ensure that your code runs. If you cannot complete one problem of the homework, comment out your failed solution and restore the stub

function that was provided. You can find the original assignments in the [course code archive](#).

- Unit tests are provided in most homework files. If it's obvious that you didn't run or test your code, you will receive zero points for that assignment.
- Always due at 5:45pm -- not a second later -- watch the clock!
- Make sure you submit to the correct folder -- no credit for other folders!
- No late homework will be accepted unless there are extreme circumstances substantiated by the Dean of Students' office.
- You can submit as many times as you like -- I will always grade the last submission.

- **55% Exams**

- On exams, you must work alone, **without any external resources**.
- Exams will be given on paper. You will be asked to write **code on paper**. You must also be able to **draw diagrams** just as we will do on the whiteboard during lectures.
- Students in the in-person section must take the exam on the chosen day and time, as announced on the course schedule.
- Students in the online section must register with a proctor for both exams. Students that require special accommodations for exams must also register ahead of time. See [here](#).
- Exams will be given in person. **There are no online exams**. You must appear physically in front of a proctor (either the instructor or someone else, as described in the policy linked above.)

I will drop one quiz and at least one homework assignment from your final grade.

[DePaul's academic integrity policy](#)

All students are expected to abide by the University's Academic Integrity Policy which prohibits cheating and other misconduct in student coursework. Publicly sharing or posting online any prior or current materials from this course (including exam questions or answers), is considered to be providing unauthorized assistance prohibited by the policy. Both students who share/post and students who access or use such materials are considered to be cheating under the Policy and will be subject to sanctions for violations of Academic Integrity.

Supplemental Instruction

This course was selected to participate in DePaul's Supplemental Instruction (SI) program. SI is a series of peer-led review sessions based on an internationally-recognized evidence-based active learning model for students taking historically difficult courses. SI is free and for all students who want to strengthen their understanding of course material and improve their grades.

At each session you will be guided through collaborative learning strategies by your SI Leader, a DePaul student who has previously taken the course and done well. SI sessions offer a chance to work together with classmates to compare notes, practice important concepts, develop effective study methods, and test yourselves before quizzes and exams.

Your SI Leader will facilitate either (3) 1-hour SI sessions or (2) 1.5 hour SI Sessions per week between weeks 2-10. Sessions are offered on-campus and online. Please complete the poll your SI Leader will send to you the first week to help them plan the most accommodating session days/times and modality for you.

Session times can be found on the [Supplemental Instruction website](#) as well as the course D2L site by Monday Week 2. If you are unable to attend a session, you will be able to access the recordings of online sessions.

Your SI Leader: Albert Luna | aluna27@depaul.edu