CSC 300: Data Structures I

Kenny Davila Castellanos

AY 2023-2024 - Spring Term

1 General Information

Instructor:	Kenny Davila Castellanos
Term:	Spring 2023-2024
Sections:	604 - (Class number 36403) - In Person 613 - (Class number 36404) - Online Async (Sync-Option)
Time:	MW 3:10PM - 4:40PM
Location:	CDM - 222 Loop

2 Course Description

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.

PREREQUISITE(S): CSC 242 or CSC 243

3 Course Evaluation

- **Weekly Coding Assignments:** 20%
- **Mini-Projects:** 20%
- **Exams:** 60%

3.1 Weekly Coding assignments

To help you practice and master the weekly materials, this class comes with a total of 9 weekly assignments. They will become available a few hours before or after the first weekly lecture, and they will be due by 11:59pm of the day before the first lecture of the following week. In this case, the assignments should be released every Monday, and they will be due before midnight on Sunday of the same week.

3.1.1 Assignment Format

For each weekly assignment, you will be provided with a PDF document containing a detailed description about the assignment requirements. Each student is expected to read the PDF document carefully. It is your responsibility to make sure that you understand what you are being required to do. If something is unclear, please ask the instructor about it as early as possible. The instructor might be unable to help if you ask the same day the assignment is due.

The assignments are generally divided into smaller sequential steps. These have been carefully crafted to help you complete the required work by implementing smaller units or functions which are easier to test in isolation. You should aim to complete each individual step before moving to the next one. You should never implement later steps if earlier steps are crashing or not compiling.

Each assignment also comes with an initial skeleton code you will have to modify to complete the work. This will be provided in a .zip file containing a full compiling project for IntelliJ. The code might come with supplementary instructions and reminders about the requirements for the homework, but make sure to always verify this against the provided PDF document. Make sure to contact the instructor if anything seems unclear or if the instructions in the code seem different to what was specified in the PDF document.

3.1.2 Requirements and Approaches

Every homework is carefully designed to help you master the specific content studied in the class. While you will have to study some of the Java programming on your own, no assignment is meant to require you to study Java features never mentioned in the lectures. For example, we are going to study basic Java syntax including iteration such as for and while loops. We will not be covering more advanced syntax that enables functional programming in Java such as streams and lambda functions. It is okay to study these, but make sure that you have fully master the basic techniques first.

Under the same motivation, some homework will specify very concrete approaches that should be used to solve a particular problem. These approached are typically based on the weekly content of the class. In a few cases, there might be more advanced data structures that could be used in such context and might lead to better results. However, keep in mind that the main goal is to evaluate if you are able to understand and use the weekly content. In that sense, any programs that do something very different from what is being asked are not eligible for partial credit. For example, if you are supposed to use lists, but instead you use a different data structure, specially **data structures not officially covered in this course**, you will automatically receive 0, no partial credit even if the program solves the task.

3.1.3 Deadlines and Late Submissions

Be aware that you are given close to a **full week** to complete each assignment. It means that each assignment will be challenging enough to keep you busy for a few days. If you are new to Java, you can expect to need to spend a significant amount of time reading the official Java documentation, the supplementary Java books, and other online resources that might teach you specifics about the syntax of Java and its standard library that we might not be covering in the lectures due to time constrains. Therefore, you should plan accordingly. It will be very hard for you complete each assignment if you do not start early. Each student is expected to at least **read the assignment description** within the first **48 hours** and ask me immediately if anything is unclear. The assignments are divided in smaller steps, so you should plan to complete one or two steps per day.

The assignments will be due at 11:59pm. Anything submitted after the due date and time is considered a **late submission**. Keep in mind that you are given a **week** to complete the work, and last-minute delays and inconveniences are not acceptable excuses. Still, I will accept **at most 2 late submissions per student** at the following multipliers: 100% if submitted within the first 3 hours, 75% if submitted after 3 hours but before 24 hours, and 50% if submitted after 24 hours and before 48 hours. If the homework is submitted after 48 hours, or if it is your third time submitting a late assignment, it will receive a score of 0. Any exceptions to these rules will only apply to students who have properly completed the required paperwork through the Dean of Students Office (see the Section Absence Notification).

3.1.4 NO Compilation = 0

All programming assignments that do not compile will receive zero points. You will always start with a working skeleton code, which should compile on a properly configured working environment. Students should modify this skeleton piece by piece making sure that it compiles every so often. Do not keep writing new code if your previous code does not compile. Keep in mind that I will give you a score of 0 if I cannot run your program as submitted. Therefore, it is better to have 1 or 2 steps that compile and work properly than all steps if the code does not even compile. Students should make sure that their code compiles before making a submission.

3.2 Mini-projects

These are longer and more challenging than the weekly assignments. You will be given at least 2 weeks to complete each mini-project, but these will typically overlap with the weekly assignments, so plan accordingly. There will be a mini-project due before the midterm and another on the second half of the class due before the final.

Unlike weekly coding assignments, the goal of the mini-projects is to analyze the performance of your code beyond simple task completion. That is, your main task is to actually run a few experiments using your code. You will typically need to run your code several times using different data or parameters, in order to understand how the program behavior changes as a function of its inputs. You might also need to compare different algorithms for the same task, Finally, you will have to write a report to describe your code, experimental results, and analysis. In a way, the report is more important than the code itself, but you will not be able to write the report if your code does not work properly.

3.3 Exams

We will have a midterm on Week 6, and the final in Week 11. Both exams will be given on paper, 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 below.)

- Students in the in-person section must take the exam on the chosen day and time, as announced on the course schedule.
- Students in online sections (synchronous or asynchronous) must register with a proctor for both exams. In order to register, use the CDM Proctored Exams section of the course homepage on D2L. You must register at least a week before the exam. Please do it ASAP. For further information, see `https://www.cdm.depaul.edu/onlinelearning/pages/exams.aspx`

You must pass the final exam in order to pass the course. Exams will include multiple problems of similar difficulty to the homework, and you need to be able to solve them within just a few minutes. **Remember that you will be working alone, on paper, without any external resources**. Therefore, it is your responsibility to ensure that you can complete any given individual functions from the homework code in about five minute under similar conditions. To solve the weekly homework, you may find it useful to consult external resources at first. That's fine as long as you do not fall into plagiarism. However, it is then incumbent on you to repeat those problems until you can do them yourself, starting from scratch, in a reasonable amount of time (about five minutes).

4 Java Environment

We will be using JAVA SE 21 (latest LTS version). You might use newer editions, but I should be able to compile your code using this version of Java. If you decide to use a newer version of Java, consult the official documentation of each new function you use to make sure that it is backwards compatible with JAVA SE 21.

The main IDE for the class will be: IntelliJ IDEA Community Edition . I will not be providing support for any other IDEs. You can certainly work your code using other tools/IDE that you prefer, but your final submission must run and compile easily using IDEA. Submissions that I cannot run, or that would require significant changes to make them run in my environment will receive a 0 even if they work perfectly on your computer.

5 Discussion Forum

For in-person students, live discussions can happen regularly during lecture time. For online students who opt to use the synchronous option, they will be able to join these discussions using zoom. To join a meeting, use the link listed on the D2L calendar of events.

We will use Discord as a discussion forum for class. Join the class server at: `https://discord.gg/KZBGPt2Gvm`.

- Change your 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.
- It is recommended to download the native Discord app for your computer and/or mobile device. Alternatively, use the web browser client.
- Always be polite.

- Use appropriate language.
- No trolling.
- Only students enrolled in the class will be allowed in the server.

The discussion forum is an extension of our time in class. This is particularly great for students that cannot attend the live lectures. As we move forward in content, multiple students are likely to share similar questions. The discussion allows to answer these common questions for everyone at once. Make sure to check the group regularly.

6 Course Homepage for Lectures

I will be using D2L as the primary website for posting lecture slides and provide links for any additional resources. Slides might not be available before class, and when they are, they might be subject to changes after the class.

7 Contact Information

- **Instructor:** Kenny Davila Castellanos
- **Email:** kdavilac@depaul.edu
- **Address:**
  School of Computing, DePaul University
  243 South Wabash Avenue
  Chicago, IL 60604-2301
- **Office:** CDM 702
- **Class Hours:** M W 3:10pm-4:40pm

A few things to note:

- You can expect that I will respond to email and Discord messages within 24 hours on business days.
- If you would like to talk, email me first to make an appointment.
- Generally, you can expect grades to be posted within a week of the due date.

8 Textbooks

If you are delayed in getting the texts, you can view them online at O'Reilly: `https://go.oreilly.com/depaul/`.

8.1 Required Books

- 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.
- 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.

8.2 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.
Available as Ebook

8.3 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!
See also here.

- 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" (3e available for free for download online and fine for our purposes)
- David Flanagan's "Java in a Nutshell" (Online version)
- Cay Horstman's "Big Java" (Online version)

9 General Expectations

The theoretical foundations and some code examples will be covered in the lectures. However, to master Java, you will have to practice outside of the classroom. We will not spent too much lecture time to cover the very basics of Java, you will have to study these from the recommended textbooks and other online resources.

You will have weekly programming assignments. You should always start them early, you are given a week to complete them for a reason. You do not have to do it alone, but you must be self-motivated. You can ask me and other members of the class. However, keep in mind that the goal is that by the end of the class, you are able to solve these problems on your own. Do not simply try to get programs that will produce the right output, focus on understanding deeply each solution and think of similar problems that could be solved using these methods.

10 Lecture Recordings for Asynchronous Students

Synchronous lectures are recorded for access after class. You will find the lectures on D2L.

11 Course-specific Policies

11.1 General

You must attend class!

11.1.1 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/Student-Resources/Pages/Grading-Policies.aspx` for additional information.

11.1.2 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/Student-Resources/Pages/Enrollment-Policies.aspx` for additional information.

11.1.3 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/about/departments/Pages/dos.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.

11.2 Supplemental Instruction Review Sessions

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, go.depaul.edu/si. If you are unable to attend a session, you will be able to access the recordings of online sessions.

11.3 AI, Cheating and Plagiarism

Keep in mind that cheating in programming courses can happen in multiple ways. The following clauses are by no means exhaustive, but they are meant to provide a general notion of what is considered cheating and/or plagiarism in this course. When in doubt, always ask the instructor first.

In general, you are not allowed to share your code with anyone. Giving copies of your code to other people will be considered a form of cheating. It does not matter if you do it intentionally or unintentionally, directly or indirectly. It is always your responsibility to protect the privacy of your code (e.g., do not use public code repositories). If you share code with others, you will receive a 0 on the corresponding assignment/project and a report of an academic integrity violation will be filed for this action.

Similarly, receiving or copying code from other students is not allowed. You are also not allowed to copy code from external sources. You can seek for help from external tutors who might help you figure out how to solve a given problem, but you should never pay anyone to complete a homework or project on your behalf. In general, submitting code written by other people as if it was your own code constitutes a form of plagiarism, and it will receive 0 and a report of an academic integrity violation will be filed for this action.

The usage of code generator based on AI (Chat-GPT, Auto-pilot, etc.) is not allowed for this class. The objective of the course is for the students to learn the basic theory about data structures and to develop the required coding skills to use them in practice without requiring the assistance of external tools. Using AI to complete assignments is both a form of cheating (e.g., you are breaking the class rules) and plagiarism (e.g. you are submitting code that you did not write). If I determine that your code was very likely written by AI, you might also receive a 0 on the assignment, and report of an academic integrity violation will be filed for this action.

12 School Policies

12.1 Changes to Syllabus

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.

12.2 Online Course Evaluations

Evaluations are a way for students to provide valuable feedback regarding their instructor and the course. Detailed feedback will enable the instructor to continuously tailor teaching methods and course content to meet the learning goals of the course and the academic needs of the students. They are a requirement of the course and are key to continue to provide you with the highest quality of teaching. The evaluations are anonymous; the instructor and administration do not track who entered what responses. A program is used to check if the student completed the evaluations, but the evaluation is completely separate from the student's identity. Since 100% participation is our goal, students are sent periodic reminders over three weeks. Students do not receive reminders once they complete the evaluation. Students complete the evaluation online in CampusConnect.

12.3 Academic Integrity and Plagiarism

This course will be subject to the university's academic integrity policy. More information can be found at http://academicintegrity.depaul.edu/ If you have any questions be sure to consult with your professor.

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.

12.4 Academic Policies

All students are required to manage their class schedules each term in accordance with the deadlines for enrolling and withdrawing as indicated in the University Academic Calendar. Information on enrollment, withdrawal, grading and incompletes can be found at `https://www.cdm.depaul.edu/Student-Resources/Pages/PoliciesandProcedures.aspx`.

12.5 Students with Disabilities

Students who feel they may need an accommodation based on the impact of a disability should contact the instructor privately to discuss their specific needs. All discussions will remain confidential. To ensure that you receive the most appropriate accommodation based on your needs, contact the instructor as early as possible in the quarter (preferably within the first week of class), and make sure that you have contacted the Center for Students with Disabilities (CSD) at:

- Lewis Center 1420, 25 East Jackson Blvd.
  Phone number: (312)362-8002
  Fax: (312)362-6544
  TTY: (773)325.7296