

Welcome to CSE 100: Advanced Data Structures!

This is a course in advanced data structure design, analysis, and implementation. You will use and implement of data structures like (un)balanced trees, graphs, priority queues, and hash tables. You will also learn how to make sophisticated use of memory management, pointers, and recursion in programs, and analyze them with theoretical and practical performance analysis.

### Instructor and Teaching Assistant (TA)

- **Debashis Sahoo** (Instructor)
  - Email: [dsahoo@ucsd.edu](mailto:dsahoo@ucsd.edu)
- **TBD** (TA)
- **TBD** (TA)
- **TBD** (TA)

### Grade Breakdown

- **Reading Quizzes (RQs):** 14%
- **Programming Assignments (PAs):** 36%
- **Projects:** 20%
- **Exams:** 30%
  - If you score **higher on the Midterm** than on the Final, **15% Midterm** and **15% Final**
  - If you score **higher on the Final** than on the Midterm, **5% Midterm** and **25% Final**
  - You must take and pass at least one of the two exams to pass the course

By default, we will use the standard scale of 90%—100% = A, 80%—89.9% = B, 70%—79.9% = C, 60%—69.9% = D, and <60% = F. These cutoffs may be lowered if need be, but they will never be raised. In other words, we may make it *easier* to get a certain letter grade, but never harder. **Pluses and minuses will be given at the instructor's discretion.**

### Course Components

This course will be taught as a "flipped classroom," meaning you will learn the content *before* class via the assigned instructional materials, and during class, we will focus on addressing and correcting your misconceptions via problem solving and discussion. Generally, any given day of the course schedule (let's use May 24 as an example), you will be expected to complete the following tasks:

- Complete the **topic videos** and **readings** *before* May 24
- Complete the **Reading Quiz (RQ)** *before* May 24
- Attend the **Problem Solving Session (PSS)** *on* May 24
- Attend the **Discussion** *after* May 24

- Work on the current **Programming Assignment (PA)** or **Project** (and complete it before its deadline)

#### *Before Class: Topic Videos and Readings*

Content in the course will be delivered asynchronously in the form of short **topic videos** (~5–10 minutes each). Each day of the course schedule will have a few assigned topic videos, and you are expected to watch the topic videos *before* the class on the corresponding day.

For students in China who may not have access to the Topic Videos via YouTube, you can download them from the following Baidu Pan folder:

- **URL:** [https://pan.baidu.com/s/1qthvEzk\\_H2Ag3c6zzqVW5A](https://pan.baidu.com/s/1qthvEzk_H2Ag3c6zzqVW5A)
- **Code:** vgfx

In addition to the topic videos, we will assign **readings** from this course's free online interactive textbook, *Data Structures*, to supplement your learning. This interactive textbook has embedded multiple choice, free response, and even coding challenges, so you may wish to refer to it for practice (though the challenges embedded throughout the textbook will not be factored into your course grade).

#### *Before Class: Reading Quizzes (RQs)*

Prior to a given day of the course schedule, after you have finished the topic videos (and potentially the optional readings) assigned for that day, you will complete a short **Reading Quiz (RQ)**, which will be due at **8:00 AM on that day**. All RQs are automatically graded instantly upon submission and will provide you feedback, and you have unlimited attempts until the deadline. **You are allowed to collaborate with your peers on the RQs.**

RQ points cannot be made up, and **late RQ submissions will not be accepted** in this course.

#### *During Class: Problem Solving Sessions (PSSs)*

Instead of holding traditional lecture-style sessions, we will be holding synchronous instructor-led **Problem Solving Sessions (PSSs)**. We expect that all students have at least been exposed to the topics for a given PSS *prior* to attending (via the corresponding topic videos, assigned readings, and RQ). Thus, rather than regurgitating the exact same information, these PSSs will focus on the following:

- **Asking Questions.** As you watch the topic videos or work on the RQs, you will likely be confused (which is a good thing!). During the PSSs, you will be able to ask any questions you may have about the topic, and we will work together to overcome the confusion.
- **Discussing Concepts.** The topic videos and readings are by no means comprehensive: they merely scratch the surface of a given concept. During the PSSs, we will explore the concepts more deeply as a group.
- **Solving Problems.** In this class, we will emphasize "learning-by-doing," and while the RQs will help you get a taste of the types of problems you may encounter, the PSSs will

allow us to think about more challenging problems, with an emphasis on real-world scenarios.

PSS attendance will be **completely optional**, and recordings of the PSSs will be made available for you to follow along (or re-watch) asynchronously. However, we strongly encourage as much participation as possible, and all students will be expected to either attend or watch the recording of **every** PSS.

#### *After Class: Discussion*

Each week, we will be holding synchronous TA-led **Discussion** sections, which will focus on the following:

- **Starting the PAs/Projects.** We understand that one of the most daunting steps when coding is the very first step: How do I even start? During the Discussion sections, we will provide an overview of the current PA/Project and provide some tips regarding how to approach it.
- **Asking Questions.** You will likely encounter the majority of your "learning breakdowns" as you work on the PAs and Projects, which may occur after the corresponding PSS. During the Discussion sections, you are welcome (and encouraged!) to ask any questions you may have.
- **Solving Problems.** You will have already attempted some problems while attempting the RQs and working on the PSSs, but those problems may not have been closely tied to the PAs/Projects. During the Discussion sections, we will work on problems more strongly resembling what you may encounter on the PAs/Projects.

Discussion attendance will be **completely optional**, and recordings of the Discussion sections will be made available for you to follow along (or re-watch) asynchronously. However, we strongly encourage as much participation as possible, and all students will be expected to either attend or watch the recording of **every** PSS.

#### *Programming Assignments (PAs) and Projects*

The **Programming Assignments (PAs)** will be the core form of assessment in this course. Typically, we will release a PA each **Wednesday**, which will be due at **10:00 PM** the following **Tuesday**.

Each PA will focus on the content covered in the 3 lectures during the week the PA was released. For example, if a PA is released on the Wednesday of Week 5, it will focus on content covered during the Monday, Wednesday, and Friday lectures of Week 5. All PAs are automatically graded instantly upon submission and will provide you feedback, and you have unlimited attempts until the deadline.

In addition to the PAs, which are generally split into smaller chunks, we will have two larger **Projects** towards the end of the course. These will combine data structures knowledge you acquire from class and homework with the pragmatics of writing a significant application that demands high performance.

For both, the grade you will receive is the grade of your highest-scoring on-time submission. PAs and Projects will only be graded based on correctness: we will not grade for programming style (e.g. comments, indentation, etc.). However, we strongly urge you to maintain good programming style as you work on the PAs and Projects, as clean code is easier to read, interpret, and ultimately debug. Also, **tutors will skip your ticket in the labs if they are unable to read your code easily.**

While we encourage discussing higher-level conceptual aspects of the PAs and Projects with your peers, **all work must be yours alone.** Sharing code (or anything easily translated into code) is a violation of UCSD's Academic Honesty policy and is not allowed.

**Late PA and Project submissions will not be accepted in this course.** Start early, start often.

### *Exams*

You will have two exams in this course: a Midterm and a Final. The exams will be administered asynchronously: you will have a larger window of time within which you can choose to start the exam at any time, and from the moment you start the exam, you will have a limited amount of time to complete the exam (Midterm: 50 minutes; Final: 180 minutes). The time will be shown via a countdown. **The dates of both exams can be found in the Course Calendar.**

### **Academic Integrity**

All students **must** submit the Academic Integrity Form at the start of the course. **You will not receive any course credit until you submit the form.** By submitting the form, you are agreeing to its terms, so be sure to read it carefully. We encourage you to study together and discuss concepts from this class, and you may work on the RQs collaboratively, but all PAs and Projects must be written completely independently. No sharing code or anything that can be trivially converted to code! If you are found to have violated the course's Academic Integrity Policy, you will receive an **automatic F in the course**, and you may face even stricter sanctions from the University. In short, **do not cheat!**

### **Getting Help**

We expect that *all* students will need help at some point in this course. If you find yourself needing help, this is not cause for embarrassment: it is completely expected, and our goal is to ensure that you are able to receive the help you need. Please be sure to seek help early and often through any (or all!) of the following resources:

- **Your Study Group:** Building a support system of friends with whom you can struggle and work through the challenges you encounter is one of the best ways to seek help. You will quickly learn how much you can figure out working together!
- **Tutor Lab Hours:** There are many, many hours in which tutors are willing and available to help you with any questions you might have. Lab hours will be posted on the Course Calendar. For these sessions, you should submit a ticket on [Autograder](#) to request help.
  - There is an extra step since we are remote: along with your ticket, you should submit a link to a Zoom meeting that you create and start.

- The tutor will join that Zoom meeting, chat with you to help you with your issue, and then leave the meeting.
- You can, for example, start a screen share with your code before the tutor arrives to help them immediately see your issue and present what you're seeing.
- Please refer to the UCSD [Zoom: User Guide](#) for detailed functionality instructions, including how to [Share Screen](#) and [Annotate Shared Content](#).
- Note that tutors are **not** going to debug your code for you! **You** are the only person responsible for debugging your code, and the tutors are only here to help *guide* you in your debugging efforts. **You are expected to know how to debug C++ code using gdb, valgrind, etc.**
- **Discussion Board:** Please use this Discussion Board for any questions related to the RQs, PAs, Projects, course logistics, etc.
  - When you create a new post, you will have the ability to specify the specific category that best fits your post.
  - The Discussion Board is **not** for thorough/lengthy code-specific help: if you need help debugging your code, please seek help from the Tutor Lab Hours!
- **Office Hours:** The instructors and the Teaching Assistant (TA) are always willing to help you during our office hours. Ideally, office hours should be reserved for higher-level conceptual questions: coding-specific questions are best asked of the tutors during lab hours.
  - The TA's office hours can be found on the Course Calendar.
  - The instructor's office hours will be one-on-one by appointment (email the instructor to schedule).

Thank you for reading the course syllabus!