CS/CE 2336.007 - Computer Science II

Tuesdays & Thursdays 2:30 - 3:45 pm ECSS 2.201

Websites: elearning.utdallas.edu (assignment submissions, grades, etc.)

utdallas.box.com/v/cs2336-007-spr25

Instructor	Dr. Jey Veerasamy *Online Office hours* (MS Teams utd.link/jey): TBD *Communication:* MS Teams chat (preferred) or email jeyv@utdallas.edu
TA	TBD

Catalog Description

CS/CE 2336 - Computer Science II (3 semester credit hours) Further applications of programming techniques, introducing the fundamental concepts of data structures and algorithms. Topics include recursion, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and algorithmic analysis. Includes comprehensive programming projects. Programming language of choice is Java.

Course Pre-requisite: CS/CE 1337 with a grade of C or better. Prerequisite or Corequisite: CS/CE 2305 with a grade of C or better or (Data Science major and MATH 3315). Credit cannot be received for both CS 2337 and CS/CE 2336.

Student Learning Objectives

Students will be able to implement different data structures using the Java programming language. They will be able use different data structures to program solutions to solve real problems. It will also help them understand algorithmic analysis and complexities. After successful completion of this course, you should be able to:

- Ability to implement recursive algorithms
- Ability to implement linked lists, stacks, and queues
- Ability to implement a binary tree

- Ability to use hash tables and graphs
- Ability to understand algorithmic analysis
- Ability to create a comprehensive programming project
- Ability to implement and use generics/templates

Teaching Philosophy Statement

My goal is to make each class as enjoyable as possible! Since most of my courses are entry-level programming courses, I use project-based learning approach to teach. In other words, I work on a series of real-world scenario-based problems. After discussing each problem in high level, I expect the students to code along with me in the class. If you attend all the classes & be engaged with the content in each session, you will do very well in my courses!

Required Textbooks

Y. Daniel Liang, "Introduction to Java Programming and Data Structures, Comprehensive Version", 12th edition, ISBN 978-0136520238 Pearson Publishing (Earlier editions or 13th edition are OK too)

Book's videos: https://yongdanielliang.github.io/JavaVideos.html

We will be using codio platform to submit & auto-grade the assignments and most activities. Your submissions will be tested against several testcases, similar to ZyLab platform you might have used in CS 1136/1336 courses. You will be prompted to purchase a license ($40 fee) as part of the first assignment/activity we will do in that platform.

Software Options:

- Windows & Mac: jetbrains.com/idea
- Windows: VS Code
- Mac: use pre-installed XCode
- Codio: click on My Projects, then New Project, select Java, ...
- Web based: replit.com/new/java10 this popular site supports several languages!

- Web based: Online GDB compiler onlinegdb.com/online_java_compiler (includes debugging!)
- There are several more: Visual Studio, Eclipse, NetBeans, ...

Additional Resources

- Intro to Java: http://javabeginnerstutorial.com/core-java
- Java questions: http://stackoverflow.com/questions/tagged/java
- Online Java tutorial: http://introcs.cs.princeton.edu/java/10elements
- Java Docs: https://docs.oracle.com/en/java/index.html
- Another tutorial: https://www.programiz.com/java-programming
- JetBrains Academy: https://academy.jetbrains.com/?tag=Java

Grading Policy

Letter grades will be assigned as follows. I do not plan to the curve the final grades. You have to earn your grade with your work.

A+ >= 97	97 > A >= 94	94 > 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
Below 60: F		

Tentative Course Schedule

Week	Class Activity	Chapters
1	Java Basics • Java program structure • Input/output (console and file) • Functions • Strings • Arrays	1-8
2	Recursion	18

3	• Algorithm analysis
	  • Big O notation
	  • Calculating Big O
		22

4	Sorting Algorithms	23
5	Classes, Object-Oriented Design	9-10

6	• Inheritance
	  • Calling specific base class constructor
	  • Overridden methods
	  • equals
	  • toString
		11

7	• Polymorphism
	  • instanceof
	• Interfaces
	  • Comparable
		13

8	**Test 1 (In-class - Date TBD)**	20
9	Linked Lists • Insert	

	• Remove • Traverse	
10	Stacks, Queues	
11	Exception Handling Generics, Arraylists	12 19

12	• Binary search trees
	  • Insert
	  • Remove
	  • Search
	  • Traversals
	  • Depth-first
	  • Breadth-first
		25

13	• Graphs
	  • Implementation
	  • Vertex container
	  • Adjacency matrix
	  • Adjacency list
	  • Traversal
	  • Depth-first
	  • Breadth-first
		28,29

14	• Hash tables
	  • Hash codes
	  • Compression
	  • Collision handling
	  • Open addressing
	  • Linear probing
	  • Double probing
		27

	• Double hashing • Simple chaining • Rehashing	
15	Review	
	Test 2 (In class - Date TBD)	

Course & Instructor Policies

Course credit is given only for the work assigned in the course schedule. There will be no opportunities for any extra credit in this course. The final grade will be computed as follows.

Tests	50%	There will be 2 tests (Test1: 25% & Test2: 25%) You will take each test in a computer in our classroom or in another designated room. Each test will contain a few multiple choice questions, fill-in-the-blank questions & one or two coding questions. There may be a few bonus questions to reduce your stress 🙂 Any make-up tests will be arranged and scheduled during the same week at the discretion of the instructor. There should be a valid reason (like Dr note, official off-site game participation, etc.) for scheduling make-up tests & you need to coordinate with the instructor in advance. Makeup test due to other scenarios (including missing to book the slot in the testing center) will result in 20% penalty. Unlike many other courses, best way to prepare for the test in programming courses is to attend each class & be engaged, complete the activities and weekly assignments & get your doubts clarified in timely manner. Then, you do not really need to prepare separately for the test! Just relax, put your thinking cap ON and focus...
Assignments	30%	There will be weekly assignments, most of them due on Fridays & they will have weightage proportional to their points – all of them together will contribute 30% to the final weighted grade. Right way to approach any

		programming assignment is to start right away & ask for help when you get stuck (you can approach the instructor or TA for help during office hours - we will help you "just enough" to get the past glitch and enable you to complete the remaining stuff on your own). Do not waste hours trying to fix a small glitch or do not use ChatGPT or web resources either! In simple words, your approach will determine whether the programming assignments provide an enjoyable learning experience or end up as painful & seemingly useless activities. One more thing: The more you complete the assignments on your own, the better prepared you'll be for the exams! Plagiarism has no place in the college education. UTD policies require all the professors to forward all suspicious cases to academic disciplinary committee. If the instructor believes a student has committed an act of plagiarism, student will be referred to UTD administration directly - Review utdallas.edu/conduct/integrity & utdallas.edu/conduct/manage-dishonesty for details. So, do not copy the code from others or use ChatGPT or web resources too much - do not share your code with others either - you can utilize ChatGPT-like resources when you go to the real job! Complexity level of each assignment will vary – each assignment may take several hours to complete. You are expected to start working on them as soon as they are posted so that you have "enough" time to work through the glitches, get the necessary help & still manage to submit on time. Do not expect me or the TA to rescue you at the 11th hour! Late submissions will be accepted with 10% penalty for 3 additional days (until Monday night). If you cannot complete an assignment due to medical condition, send the Doctor note to the professor using MS Teams chat. You will be given a few additional days to complete the assignment.
Activities	20%	There will be 2 to 5 activities every week to ensure that you are keeping up with the class content (work on a problem in the class with my assistance, take-home quiz, review the slides etc.) - so bring your laptop to every class - all of them will have proportional weightage & together they will account for 20% of your final grade. Activities ensure that you are keeping up with the course - they cannot be made up after the deadline! In other words, late submissions will NOT be accepted, however 3 lowest scores will be dropped when computing the final grade, to cover the common issues like car-trouble, overslept, etc. You should be able to earn full 20% if you attend all the sessions & submit all the activities. If you cannot complete an activity due to medical condition,

		send the Doctor note to the professor using MS Teams chat. If you had too many excused absences, follow-up with the professor to adjust the activity score when the semester ends.

Course & Instructor Policies

You are required to bring a laptop to each class. I expect you to work on one or more coding problems in every class. I expect you to be physically and mentally present in the class. You are NOT allowed to do anything in the laptop or phone unrelated to the class work.

In addition to meeting the instructor before or after the class, you can also visit the instructor or TA during respective virtual office hours to get help with the assignments. MS Teams chat or emails are NOT recommended to get help for the assignments. Outside the office hours, you can reach out to Computer Science Mentor Center at https://csmc.utdallas.edu.

Class Recordings

Instructor will try to record all the sessions in MS Teams so that all students can use them for future reference. If the instructor or a UTD school/department/office plans any other uses for the recordings, consent of the students identifiable in the recordings is required prior to such use unless an exception is allowed by law.

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same: “As a Comet, I pledge honesty, integrity, and service in all that I do.”

Academic Support Resources

The information contained in the following link lists the University’s academic support resources for all students. Please see http://go.utdallas.edu/academic-support-resources.

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus. Please go to go.utdallas.edu/syllabus-policies for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.