

CS/CE 2336 Course Syllabus

COURSE INFORMATION:

Course Title: Computer Science II
Section: 001
Course Term: Spring 2025

INSTRUCTOR CONTACT INFORMATION:

Name: Brian Ricks
Office: ECSS 4.701
Email Address: bwr031000@utdallas.edu
Phone: (972) 883-2674

Communication: For course related questions, please email me from your UTD account with a subject that begins with: *CS2336.001*. I will give course announcements on eLearning.

Office Hours: Tuesday: TBA
Wednesday: TBA

Meetings outside of office hours can be arranged by appointment.

Do not procrastinate. I will not schedule meetings to assist students with assignments that are due within 24 hours.

GRADER CONTACT INFORMATION:

Will be announced on eLearning.

COURSE PREREQUISITES AND COREQUISITES:

Prerequisite: CS/CE 1337
Corequisite: CS/CE 2305, or MATH 3315

Corequisites can be taken previously. Computer Science / Computer Engineering majors take CS/CE 2305 as a corequisite or prerequisite. Data Science majors take MATH 3315 as a corequisite or prerequisite.

Credit cannot be received for both CS 2337 and CS/CE 2336.

COURSE 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.

Credit cannot be received for both CS 2337 and (CS 2336 or CE 2336). Prerequisite: (CE 1337 or CS 1337) with a grade of C or better. Prerequisite or Corequisite: (CE 2305 or CS 2305) with a grade of C or better or (Data Science major and MATH 3315). (3-0) S

STUDENT LEARNING OBJECTIVES/OUTCOMES:

1. Ability to implement recursive algorithms
 2. Ability to implement linked lists, stacks, and queues
 3. Ability to implement a binary tree
 4. Ability to use hash tables and graphs
 5. Ability to understand algorithmic analysis
 6. Ability to create a comprehensive programming project
 7. Ability to implement and use generics/templates
-

COURSE MODALITY:

This course is scheduled to be taught in the traditional (in-person) mode.

Should it become necessary to meet online, meetings will be held in MS Teams in the appropriate lecture channel. Check the course announcements and your UTD email account for updates.

Online meetings will be recorded and posted in MS Teams.

REQUIRED TEXTBOOKS AND MATERIALS:

Required Textbook: zyBook: CE/CS 2336: Computer Science II
ISBN: 979-8-203-92753-8

A zyBook is an online textbook, including interactive activities as you read. zyBooks works on a subscription-based model. To subscribe, please perform the following tasks:

1. Click on any zyBooks assignment link in eLearning (Do not go to the zyBooks website directly and create a new account)
2. Subscribe using your **@utdallas.edu** email and section number **001**

A subscription is \$89. Students may begin subscribing on Jan 08, 2025, and the cutoff to subscribe is Apr 27, 2025. Subscriptions will last until May 24, 2025.

IDE Information:

Use of laptop or other computer for completing programming assignments and exercises with software for creating and compiling Java programs. The computer lab in ECSS 2.104 has various IDEs you can use for writing programs. The hours of operation are posted at:

<https://engineering.utdallas.edu/about/facilities/open-access-labs/>

During lecture, we will primarily use IntelliJ IDEA: <https://www.jetbrains.com/idea/>

We may additionally use a browser-based IDE called OnlineGDB: <https://www.onlinegdb.com/>

While OnlineGDB can be used to practice coding skills, course projects may be better suited for a local IDE such as IntelliJ IDEA.

CLASS MATERIALS:

Other materials including the syllabus, sample programs, assignments, announcements, etc. will be posted in eLearning: <https://elearning.utdallas.edu>

COMPUTER SCIENCE MENTOR CENTER (CSMC):

The Computer Science Mentoring Center (CSMC) is a resource made freely available to all students taking this class, with costs supported by the CS Department. The CSMC provides assistance in many areas including:

- Understanding core concepts related to this class
- Developing a logical framework for a program
- Connecting programming constructs to the logic of the program
- Assisting in solving syntax and logical errors in your code

The mentors will meet with you 1-on-1 to address your specific problem areas. Their goal is to help you understand what is wrong and how to fix it, but they will not do the work for you. For more information about the CSMC, including location and hours of operation, please visit <http://csmc.utdallas.edu>

TENTATIVE COURSE CALENDAR:

Date	Lecture Material	zyBook Chapter
January 22	Syllabus Review and zyBooks	
January 27	Basic Elements of Java	Chapters 3 - 7
January 29	I/O and Control Structures	
February 3	User-Defined Functions, Arrays, Strings	Chapters 8 - 10
February 5	User-Defined Functions, Arrays, Strings	
February 10	Recursion	Chapter 11
February 12	Objects and Classes	Chapter 12
February 17	Objects and Classes	
February 19	Inheritance and Polymorphism	Chapter 13
February 24	Inheritance and Polymorphism	
February 26	Abstract Classes and Interfaces	Chapter 14
March 3	Linked Lists	Chapter 15
March 5	Linked Lists	
March 10	Stacks and Queues	Chapter 16
March 12	No class – midterm exam study	
March 13 - 14	Midterm Exam – In the testing center, we will <i>not</i> have lecture on March 12	
March 17	No class – Spring Break	
March 19	No class – Spring Break	
March 24	Exception Handling	Chapter 17
March 26	Generics	Chapter 18
March 31	Generics	
April 2	Arraylists	Chapter 19
April 7	Binary Search Trees	Chapter 20
April 9	Binary Search Trees	
April 14	Binary Search Trees	
April 16	Graphs	Chapter 21
April 21	Graphs	
April 23	Hashing	Chapter 22
April 28	Hashing	
April 30	Hashing	
May 5	Algorithmic Complexity	Chapter 23
May 7	Algorithmic Complexity	
May 12 - 14	Final Exam – In the testing center	

This calendar is tentative and may change as the semester progresses. Please check eLearning for announcements regarding changes to the calendar.

INSTRUCTOR POLICIES:

Grading Policy:

Your course average will be calculated as follows:

Midterm Exam – 20%

Final Exam – 25%

Programming Assignments (x3) – 15%

Programming Projects (x2) – 30%

Reading Assignments – 10%

The instructor intends to assign letter grades as shown below. An average with a fractional portion of five tenths or above will be rounded up to the next whole number for determining the letter grade.

Averages	Letter grade
97+	A+
93-96	A
90-92	A-
87- 89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
67-69	D+
63-66	D
60-62	D-
Below 60	F

Grading Concerns:

If you think there is a mistake in the grading of your *project*, you must notify *both* the grader and the instructor by email *within two weeks* after the date the grade is posted in the gradebook on eLearning. Keep in mind that a regrade may result in an increase *or* in a reduction of the original grade.

Most deductions are made because students did not fully read the project instructions, disregarded the instructions, did not adequately test their programs, or did not follow the style guidelines provided. You may *not* change the problem to suit your purposes. To get maximum credit you **MUST** read the instructions carefully and test your programs thoroughly.

If you think there is a mistake in the grading of your *exam* and would like to request that it be regraded, you must notify the instructor of this by email *within two weeks* after the date the grade is posted in the grade book on eLearning. Your request for any regrade must describe in detail what you perceive as the problem with the grading. Keep in mind that a regrade may result in an increase *or* in a reduction of the original grade.

Student Responsibilities:

- You are responsible for all the material in the assigned reading in the required course textbook.
- You are responsible for all the material in the slides.
- You are responsible for all material discussed during course lectures.
- You are responsible for all material supplied on eLearning (including announcements and discussion postings).
- Students are expected to be respectful of each other and of the course instructor. Disruptive behavior will not be tolerated.

Exams:

Course Exams will be administered in the testing center. *You must register for each exam in advance of the exam date.*

Exam registration is located here: <https://www.registerblast.com/utdallas/Exam/List>

Make-up exams will be administered *only for well-documented emergencies*, and you must make every attempt possible, via email, MS Teams private message, or phone, to notify the instructor *prior* to the scheduled date and time or *immediately thereafter*. If notification is not received in a timely manner, no make-up exam will be given.

Programming Projects:

You will be given two programming projects, comprising design, programming, and testing. Each project will be given at least two weeks to complete. You may submit a project up to 24 hours after the due date and time with a 10-point penalty. No excuses will be entertained for late project submissions.

Projects are individual endeavors and students are not to work in groups on any project. Students are permitted (and I openly encourage students) to discuss the project. Feel free to share ideas on the logic, but **DO NOT SHOW YOUR CODE TO OTHER STUDENTS**. When discussing logic, try to keep it general. If you give out every little piece of logic you have, there is a good chance the person you are helping will have very similar code as yours and may be flagged for being too similar. Be careful of posting your code online. Another student could use your code without your knowledge and could involve you in a code plagiarism referral.

Students should avoid using resources like GitHub, Chegg, ChatGPT, or Grok for help on projects. Copying code from a web site or AI source is considered plagiarism and will be treated as such. If you find code on a web site, it is highly likely another student will find it as well which may cause both submissions to be flagged for similarity. Submitting project and assignment details to web sites for outside help (e.g. Chegg) is also considered academic dishonesty by UTD.

Any projects that are approximate or identical copies will be reported to the Office of Community Standards and Conduct, and I will accept their decision regarding the grade if they believe that academic dishonesty has occurred.

Programming projects will be graded on a 100-point basis. Not only will your project be graded on proper execution, but also things like efficiency, implementation and code comments. Keep in mind that you always want to write code that is easy to understand and is also easy to maintain. Fewer lines do not necessarily mean a better program. Code commenting is also very important.

You are responsible for testing your project thoroughly before submission. I will not give you the exact test cases that will be used for grading before the project is due. As a computer scientist, you must be able to identify all possible input and make sure that your code produces proper output and does not crash.

Programming Assignments:

You will be given three programming assignments. Each programming assignment will be submitted to zyBooks, with technical aspects of the code automatically graded according to test cases. Code structure and code commenting will be manually graded and added to the auto grade to comprise the total grade per assignment.

Programming assignments will be graded on a 100-point basis. You may submit a programming assignment up to 24 hours after the due date and time with a 10-point penalty. No excuses will be entertained for late programming assignment submissions. The same rules on cheating apply for programming assignments as per the policy above regarding programming projects.

Note that you *must* access the programming assignments through the eLearning links, otherwise your grade may not update to eLearning. Please be vigilant in checking eLearning to make sure that your programming assignment grades are correct.

Reading Assignments:

All reading assignments contained within zyBook chapters that check your knowledge as you read. These include participation activities and challenge activities. Your score per assignment is the total number of activities completed within the assignment.

You will be given at least one week to complete each reading assignment, with *no extension past the due date*. Each reading assignment grade will display in eLearning as a percentage, with 100% representing completion.

Note that you *must* access the reading assignments through the eLearning links, otherwise your grade may not update to eLearning. Please be vigilant in checking eLearning to make sure that your reading assignment grades are correct.

The instructor will drop your lowest reading assignment grade.

What you need to do to be successful in this course:

- Attend every course lecture and pay close attention.
- Read the appropriate zyBook chapters *before* the lecture. Reading assignments can be completed after lecture, and can themselves be used to reinforce what you previously read.
- Dedicate 9-10 hours per week outside of course lectures for reading, practicing writing code, completing projects, and studying for exams.
- Do all your work and do it individually. Students who get too much help from others: the instructor, mentors, peers, and the internet may not acquire the knowledge they need to successfully complete the exams.
- Do not procrastinate, especially on the programming projects. Start early so that you have time to try alternate approaches, ask questions, and test your program. A program is not necessarily complete when it compiles. Your program must produce correct results under various conditions, as given in the exercise or assignment. You must design test cases in addition to designing your code.

- Learn to debug your programs yourself. Add print statements that print the values of inputs and the results of intermediate calculations. Add print statements to indicate that sections of code have been reached during execution. Learn to use the integrated debugger.
 - Do not wait till the end of the semester to seek help. If you wait until late in the semester, it may be difficult to catch up, as the course is constantly moving forward.
 - If you do not understand a concept: reread the text and complete as many activities as you can (including those which are not part of reading assignments). Arrange a meeting with your instructor and come to the meeting with a list of specific questions. Visit the Computer Science Mentor Center (CSMC), prepared with specific questions.
 - Write more programs than are required. The more programs you practice with, the better you will do in this course. You learn to program by doing – coding, testing, and fixing (debugging). This course is like a mathematics course – you need to work *many* problems.
 - Enter the sample programs from the text. Experiment by making small changes. Note how the changes affect the program translation and/or execution. Participation activities are a good way to experiment.
 - Pick some of the challenge activities which aren't used in reading assignments and complete them.
-

MORE FROM UTD:

Class Materials

The instructor may provide class materials that will be made available to all students registered for this class as they are intended to supplement the classroom experience. These materials may be downloaded during the course, however, these materials are for registered students' use only. Classroom materials may not be reproduced or shared with those not in class, or uploaded to other online environments except to implement an approved Office of Student AccessAbility accommodation. Failure to comply with these University requirements is a violation of the [Student Code of Conduct](#).

Classroom Conduct Requirements Related to Public Health Measures

UT Dallas will follow the public health and safety guidelines put forth by the Centers for Disease Control and Prevention (CDC), the Texas Department of State Health Services (DSHS), and local public health agencies that are in effect at that time during the Fall 2021 semester.

Class Attendance

The University's attendance policy requirement is that individual faculty set their course attendance requirements. Regular and punctual class attendance is expected. Students who fail to attend class regularly are inviting scholastic difficulty. In some courses, instructors may have special attendance requirements; these should be made known to students during the first week of classes.

Class Participation

Regular class participation is expected. Students who fail to participate in class regularly are inviting scholastic difficulty. A portion of the grade for this course is directly tied to your participation in this class. It also includes engaging in group or other activities during class that solicit your feedback on homework assignments, readings, or materials covered in the lectures (and/or labs). Class participation is documented by faculty. Successful participation is defined as consistently adhering to University requirements, as presented in this syllabus. Failure to comply with these University requirements is a violation of the [Student Code of Conduct](#).

Class Recordings

Students are expected to follow appropriate University policies and maintain the security of passwords used to access recorded lectures. Unless the Office of Student Accessibility has approved the student to record the instruction, students are expressly prohibited from recording any part of this course. Recordings may not be published, reproduced, or shared with those not in the class, or uploaded to other online environments except to implement an approved Office of Student Accessibility accommodation. Failure to comply with these University requirements is a violation of the [Student Code of Conduct](#).

The instructor may record meetings of this course. These recordings will be made available to all students registered for this class if the intent is to supplement the classroom experience. 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 review the catalog sections regarding the [credit/no credit](#) or [pass/fail](#) grading option and withdrawal from class.

Please go to <http://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.