TEXAS
The University of Texas at Austin

# CS 312  Introduction to Programming Fall 2025

# Welcome to CS 312!

## I'm glad you are here!

I'm impressed by you already, because you have been admitted to UT and to the CS 312 course! In this course, you will explore complex problems, design an approach, then code and test a solution.

Programming is a unique skill because you can't memorize a finite number of facts to be successful. Given a problem, there isn't a formula or recipe that will lead to the one correct answer. But, you don't have to be a genius either. Each problem requires curiosity, knowledge (concepts and syntax) and persistence. Curiosity and persistence are essential. Practice, practice, and more practice is mandatory.

This course provides a series of learning and practice activities, with support along the way. Students arrive at this class with different levels of experience. The course is designed for all of y'all. I'm glad you are here. Let's get this done.

**Class Meets**

Unique IDs 54540, 54545, 54550, 54555: Monday, Wednesday, Friday, 9am - 10am

Unique IDs 54560, 54565, 54566, 54570: Monday, Wednesday, Friday, 10am - 11am

Unique IDs: 54575, 54580, 54585, 54590: Monday, Wednesday and Friday, 1pm - 2pm

Lab/Sections meet on Monday, time and location varies

**Instructor**

Professor Ramsey, she/her/hers

Office: GDC 6.318

Email: ramsey@cs.utexas.edu

Office Hours (one-on-one): Monday 11-noon, GDC 6.318, Sign-up at https://calendly.com/ramseytx/10min, other times by appointment

Help Hours: Wednesday 11:10 - 11:50, 3rd Floor Lab

# Course Description

## University Catalog Course Description

First part of a two-part sequence in programming. Fundamental concepts of structured programming; procedures and data structures with a focus on problem solving strategies and implementation; introduction to concepts of informal specification, informal reasoning about program behavior, debugging, and ad hoc testing.

## Pre-requisites

Credit with a grade of at least C- or registration for Mathematics 408C, 408K, or 408N.

## Learning Outcomes

1. Students will be able to design and write programs based on their understanding of the Java programming language, including design, implementation, and testing.
2. Students will be able to implement Java code to solve non-trivial problems using data types, variables, decisions, loops, parameters, files, arrays, 2D arrays, classes and objects.
3. Students will develop curiosity and persistence in their work.

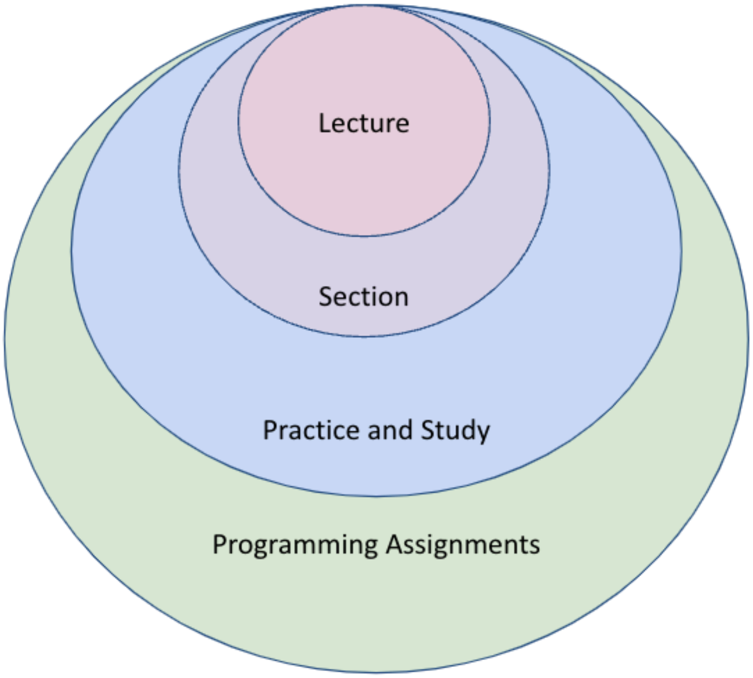
# How Will You Learn?

## Statement of Learning Success

Computer Science is unique, because students in the same class often have different levels of experience and confidence. Some of you haven't written any code and others may have written a smartphone app currently for sale in the App Store and Google Play. This class is for both of y'all and everyone in-between. This course is challenging, but many resources are provided to support your learning, fill in gaps in experience and build your confidence. This course will take significant time and effort. The professor and TA team will support your learning throughout the semester. Know that, if you made it to this class, you belong here and you can do this.

## Teaching Approach

To learn complex, conceptual material, it is important to layer different types of learning that build on each other. Lecture leads you through activities that build the foundation for new topics. A weekly Section builds on this foundation, providing small-group coding experiences with more complex problems. Readings and practice problems provide another mode for learning that reinforces the knowledge and skills necessary to solve new problems. The Programming Assignments let students apply what they have learned to a large and complex problem. Other learning opportunities include TA Help Hours, professor Office Hours and Free Group Tutoring.



## Learning Supports

**Section -** Section activities are designed to bridge the gap between lectures and complex programming assignments. For most students, the Programming Assignments will be very difficult without attending Section. Students will work in small groups to complete code of similar difficulty as Programming Assignments and Unit Exam questions, with the help of a TA.

**Office Hours (with Professor) -** I am available for one-on-one meetings in my office (GDC 6.318) to talk about your progress in the course, strategies to improve your learning, or other issues. To provide students with dedicated time, you must sign up for an Office Hour time using this calendar.

**Help Hours (with TAs and Professor) -** TAs and I will be available in the GDC 3rd floor lab (in the GDC north tower, closest to Speedway) on a set schedule to help with course content and assignments. To talk with the TA, first submit an Ed Discussion post for the Assignments category, completing the template. Sometimes, when you complete the template, you will realize the answer on your own! After submitting the post, write your name on the white board. The TA will call your name when it is your turn. They will read your Ed Post first, then address the questions. You may choose to write the Ed Discussion post before Help Hours start.

**Ed Discussion (On-line Discussion Tool) -** Check Ed Discussion at least once a day for important announcements.  Even if you are a programming genius who doesn't need help with assignments, you must check Ed Discussion daily. If you aren't a genius yet, and you get stuck when studying or working on a program assignment, you can post a question to Ed Discussion. Before you post a question on Ed Discussion, search to see if the question and response already exists. When posting, fill out the template to provide the TA with enough information. Complex or conceptual questions are better suited for Help Hours. If you want to include a code segment in your post, be sure you make the post visible only to instructors (this includes the TAs and professor), so other students aren't tempted to copy your code. Read other students' posts. You can learn a lot from questions and answers from other students. Answer questions when you can!

**Teaching Assistants-** This course is supported by Teaching Assistants (TAs). Each TA has taken this course and done very well.  They are also very interested in helping you learn! They lead Section, provide Help Hours in the lab, and answer questions on Ed Discussion. They will also provide meaningful feedback on graded assignments, so you can improve on future assignments. It's a good idea to get to know your TA. You should at least know their name and hometown! Here is the TA info for each Section. Refer to the Section and TA Info Canvas page for TA info. Refer to the Help Schedule for TA Help Hour day and times.

**Free Group Tutoring -** One way to learn complex material is to learn in layers from multiple sources. You might understand 10% of the content during lecture, then understand 30% after working on practice problems, and 60% after attending Group tutoring, etc. Group tutoring sessions are led by experienced CS students. The leaders prepare for each session, providing practice and resources to reinforce understanding. There are two

Free Group Tutoring sessions per week Each session is identical, so students typically attend only one. When days and times are set, they will be added to the Help Schedule.

**Sanger Learning Center -** The Sanger Learning Center provides free individual tutoring, online and in person, by appointment and drop-in. CS 312 is one of their official supported courses. Find more info here.

## If You Feel Anxious

Computer Science courses are challenging. Anxiety is especially prevalent in Computer Science, since students arrive at UT with very different levels of experience and confidence. Also, CS can be mysterious, since you can't memorize a fixed number of facts. You need to solve new problems and there isn't a direct path to finding the answer. Here are some ideas:

- You are not alone in feeling anxiety. Almost all students feel the same anxiety you are feeling at some point.
- It's OK to feel anxious some of the time. This is your body recognizing that you are learning something new, that you aren't content to stay home and watch Netflix all the time, you are out in the world and making things happen. It's uncomfortable to feel that you don't know the answer or how to find it, but that's the normal feeling to have before you find the answer.
- One best way to reduce anxiety about the course is to learn some CS each day. Come to class each day, bring questions to Help Hours, do 5 practice problems each morning before coffee. Get closer to figuring it out every day, so you build layers of understanding over time. This is less exciting than pulling all-nighters, but it's also less stressful and more likely to earn the grade you want.

## If It's Really Not Going Well

If you are struggling or feeling overwhelmed, because of academics or personal issues, you aren't alone. College and young adulthood brings new responsibilities and massive change, which can be very stressful. Please don't feel you need to handle this on your own. Here are some ways to get help:

- Meet with me during Office Hours or email me to find another time to meet.
- Meet with the computer science advisors. They can help with more than just your class schedule! (512) 471-9509, under-info@cs.utexas.edu
- Contact Student Emergency Services. They can help with crisis and non-crisis situations such as family issues, housing, money, stress, and more. They can also communicate with your professors to find a way forward. website, (512) 471-5017, studentemergency@austin.utexas.edu

# Course Requirements and Grading

## Required Materials and Devices

Building Java Programs: A Back to Basics Approach (5th Edition) online textbook. The cost is typically around $50 for a 6 month subscription and $80 for an unlimited subscription.

Powered, working, and configured smartphone to upload quizzes during each lecture and section.

You can use a lab computer in GDC (see me for more information) or your own computer.

## High Level Schedule

| Unit | Chapter/Topic |
|---|---|
| 1 | 1. Introduction<br>2. Primitive Data and Definite Loops<br>3. Parameters and Objects |
| 2 | 3G. Graphics<br>4. Conditional Execution<br>5. Program Logic and Indefinite Loops<br>6. File Processing<br>7. Arrays |
| 3 | 8. Classes<br>9. Inheritance and Interfaces<br>10. ArrayLists<br>12. Recursion |

## Student Expectations

**Lecture** - Lectures are in person and synced with the textbook chapters. Because CS 312 is a challenging course and students arrive at CS 312 with different levels of experience,  the lectures have been carefully designed in three parts:

- Lecture Prep - Read the textbook section(s) or watch the textbook video before class. Learn what you can on your own. Think of questions about the content. Build an early understanding of the concepts.
- Lecture - Lecture will learn about concepts, see live coding and complete hands-on activities with a partner to apply the ideas just learned. Try to absorb as much as you can. Also, arrive on time, don't

leave early, get to know your classmates and participate in the activities. Name tents should be on your desk at all times. All devices must lay flat on your desk (a paper notebook or tablet, no open laptops). You will turn in the activity paper for a quiz grade at the end of lecture. You must be present and have a powered, working and configured smart phone with you to earn this grade. If you miss a lecture, you can view it on Lectures Online, from the left side Canvas navigation.

- Lecture Follow-up - Complete the Problem Set associated with the lecture. This reinforcement and repetition will prepare you to solve a new problem on the exam. Ask yourself, "Could I understand this material enough to teach it to another student?" If not: read the textbook reading and complete more practice problems. **Most students who pass this class complete a great number of practice problems, even though they aren't graded.**

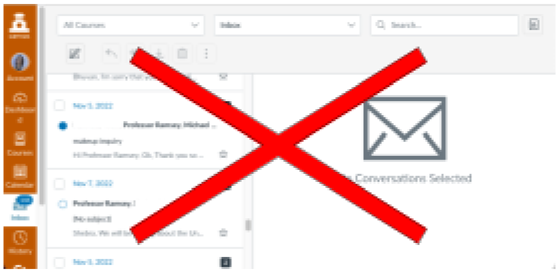
**Section -** Section is designed to strengthen problem solving skills to improve Exam performance. Bring your laptop to Section. You will complete an In-Section Quiz during each section. You must be present and have a powered, working and configured smart phone with you to earn this grade. Section is also a great place to find a study group to help study for unit exams.

**Office Hours (with Professor)** - If you have questions about the course or are struggling, sign up for Office Hours with Professor Ramsey with this link.

**Help Hours (with TAs and Professor) -** When you think, "Oh no, what does that mean?" or "How would I do that?" go to Help Hours right away. You can figure it out before you get behind. You will spend less time waiting on a TA if you attend on Mondays and Tuesdays.

**Ed Discussion (On-line Discussion Tool) -** Check Ed Discussion at least once a day for important announcements. When you get stuck when studying or working on a program assignment, you can post a question to Ed Discussion. Read other students' posts. You can learn a lot from questions and answers from other students. Answer questions when you can!

**Email -** Check your email each day. The professor and TAs may send emails about individual issues. You may send an email to the professor about the course. You may send a TA an email about a grading issue. Include in your email: your first and last name, EID and which course you are taking. Limit questions about content and programming assignments to Help Hours, Office Hours or Ed Discussion (not email). **Do not use Canvas Inbox for email.** The professor and TAs do not respond to messages sent by Canvas Inbox.



**General**

- Scan the provided bar code when you attend lecture, section, Help Hours or other. This will help us understand how different course components are working.
- Understand that students in this course have different levels of experience and confidence. Be generous and brave with your classmates.
- Use professional languages and tone of voice with your professor, TAs and classmates. Act as you would with your boss and colleagues at a job.
- If your laptop stops working, this is a big deal! Stop everything! Get your laptop to the shop! If the repair will take more than a few days, contact the Professor about using the lab computers for your work. No extensions are given for computer issues.
- If you lose access to the internet, this is a big deal! Call your provider right away! Work at the library! Use a personal hotspot! No extensions are given for internet access issues.
- If life circumstances are impacting your course work, contact the professor as soon as you are able.

## Course Expectations

**Office and Help Hours -** The Help Schedule will be posted and followed. Any changes will be made at least 24 hours in advance and will be posted to Ed Discussion.

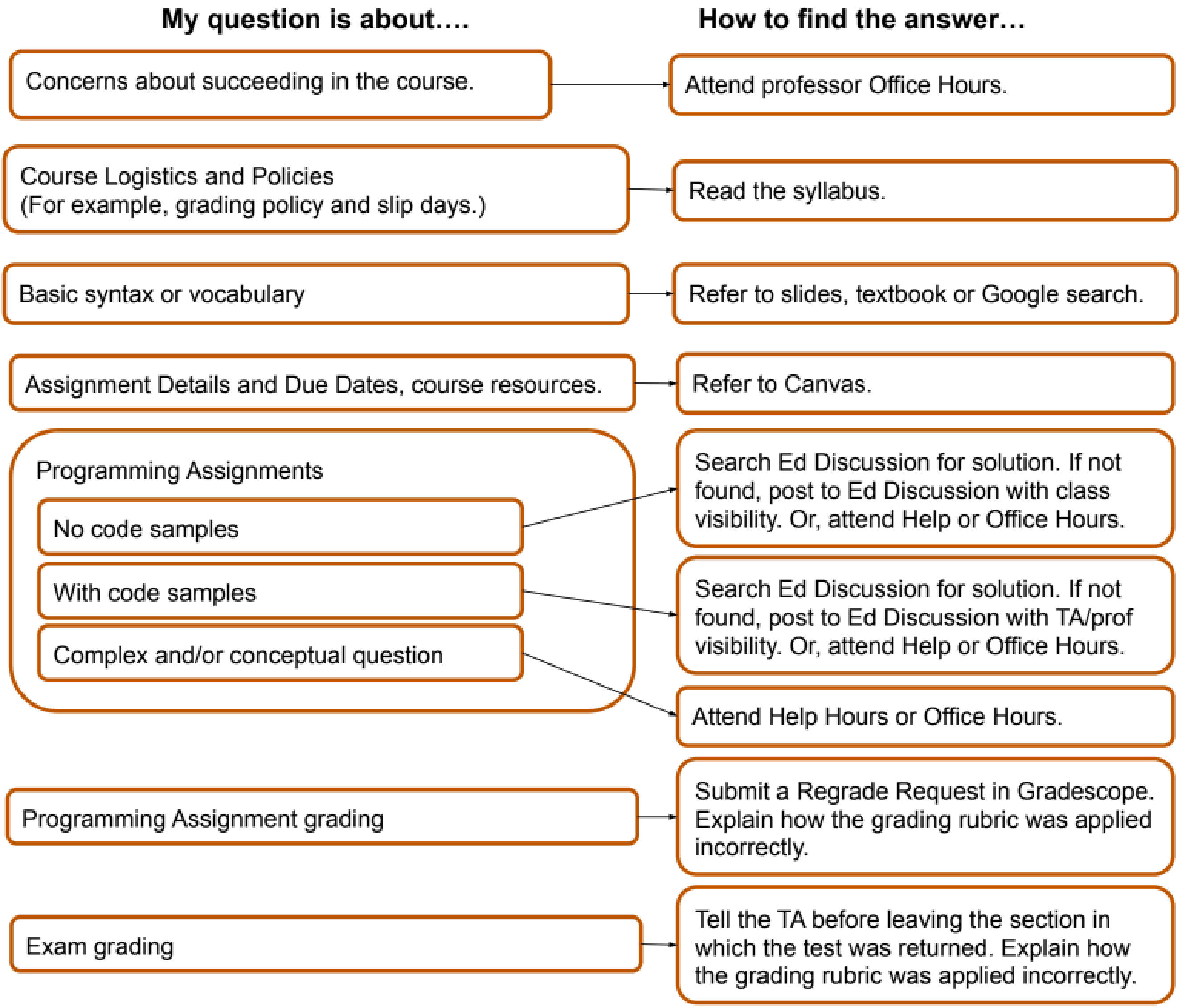
**Ed Discussion -** TAs will do their best to reply to Ed Discussion posts within 24 hours on weekdays (Sunday night - Friday afternoon). Responses should not be expected on weekends. *Be aware of this when using slip days for a Programming Assignment.*

**Program Assignment Grading -** Programming assignments will be graded by Wednesday at 6pm, following the Thursday due date. This allows time to review the feedback from the last assignment and make any related updates before submitting the next assignment. If this isn't possible, the TA will notify the students and provide a date when the grading will be complete. The TA will provide manually written, meaningful feedback to most programming assignments.

**Unit Exam Grading -** Unit Exams will be graded within 2 weeks of the Unit Exam date. A solution key and rubric will be provided.

## I Have a Question!

This graphic provides a summary of where to go with specific questions.



| My question is about.... | How to find the answer... |
|---|---|
| Concerns about succeeding in the course. | Attend professor Office Hours. |
| Course Logistics and Policies<br>(For example, grading policy and slip days.) | Read the syllabus. |
| Basic syntax or vocabulary | Refer to slides, textbook or Google search. |
| Assignment Details and Due Dates, course resources. | Refer to Canvas. |
| Programming Assignments | |
| No code samples | Search Ed Discussion for solution. If not found, post to Ed Discussion with class visibility. Or, attend Help or Office Hours. |
| With code samples | Search Ed Discussion for solution. If not found, post to Ed Discussion with TA/prof visibility. Or, attend Help or Office Hours. |
| Complex and/or conceptual question | Attend Help Hours or Office Hours. |
| Programming Assignment grading | Submit a Regrade Request in Gradescope. Explain how the grading rubric was applied incorrectly. |
| Exam grading | Tell the TA before leaving the section in which the test was returned. Explain how the grading rubric was applied incorrectly. |

## Assignments and Grading

Assignments and grades are tracked in Canvas. Note that the grade average in Canvas may vary slightly from the official grade due to dropped assignments and final calculation details. No additional points can be earned outside of the work described in this syllabus.

**Quizzes** (8 per week, on average) - Quizzes are implemented in Canvas. They provide frequent feedback on student progress. At the end of the semester, the lowest 8 quiz grades will be automatically dropped.

- **At-Home Quizzes** are completed outside of  lecture and section. Quiz timing and due dates are set in Canvas modules. At-Home Quizzes have a 2-day grace period for each due date, with no penalty or slip days charged. Types of At-Home Quizzes:
    - Setup and Surveys - Beginning of semester setup activities and surveys at the beginning and end of the semester.
    - Program Design - Each Programming Assignment has a corresponding Design Quiz to guide the design process and provide TAs with information on student understanding of the assignment.
    - Content - To allow lecture to address more complex programming concepts, more straight forward content is covered in an At-Home Quiz. This type of quiz includes slides, a video, sample code and a Problem Set. When you complete the Problem Set, you will take a screen shot of the green check marks and submit this for the quiz grade.
- **In-Lecture Quizzes** occur during lectures, to measure attendance and incremental learning. A paper Quiz will be passed out during class to students displaying their name tent. Students will work in pairs, applying the new information to a specific problem. If a student arrives in class after the Quizzes are handed out, a max of 50% can be earned. If a student is not displaying their name tent, a max of 50% can be earned.  At the end of class, each individual student will submit a photo of the work to their Canvas Quiz assignment. Students must be present in the given lecture and have a powered, working and configured smart phone with them, to earn this grade. If a student has a technical issue with the Canvas quiz upload (other than not having a working phone), they need to let the professor know before they leave class, to address the issue. If you accidentally submit the wrong file, or the wrong number of files, you will be able to resubmit until the lecture ends. It is not possible to submit the quiz after lecture ends.
- **In-Section Quizzes** occur during section, to measure attendance and incremental learning. Each student will answer design questions and write code on paper, often working in groups, and with the help of the TA leading section. At the end of Section, each individual student will submit a photo of their work to their Canvas Quiz assignment. Students must be present in the given section and have a powered, working and configured smart phone with them, to earn this grade. If a student has a technical issue with the Canvas quiz upload (other than not having a working phone) during section, they need to let the TA know before they leave section, to address the issue. It is not possible to submit the quiz after section ends.

**Programming Assignments** (About 1 per week) - Each Assignment includes detailed instructions that must be followed to receive full credit. Plan to spend 20 hours per week programming. Each Programming Assignments begins on a Friday and the submission in Gradescope is due at 11:59pm the following Thursday. The latest submission in Gradescope is the only one that will be graded. Each assignment is graded with automated test cases and TA manual grading. The code must pass at least 50% of the automated test cases to be eligible for manual grading.  *Code submitted for a programming assignment must include only Java language* *elements addressed in the textbook sections that have been covered in class up to that point in the course. Any exceptions will lead to a 0 grade for the assignment that cannot be dropped. In addition, this is an indication of possible cheating and will lead to more detailed analysis of the assignment*. At the end of the semester, the lowest Programming Assignment grade will be dropped. Refer to the Academic Integrity Section of this syllabus for important information on completing your own work. The tentative schedule for assignments follows.

| # | Assignment | Chapter and Topic |
|---|---|---|
| 1 | I Know an Old Lady | 1. Introduction |
| 2 | UT Tower | 2. Primitive Data and Definite Loops |
| 3 | Scintillation Grid | 3. Parameters and Objects<br>3G. Graphics |
| 4 | Rock, Paper, Scissors | 4. Conditional Execution |
| 5 | Hangman | 5. Program Logic and Indefinite Loops |
| 6 | Home Field Advantage | 6. File Processing |
| 7 | Personality Quiz | 7. Arrays |
| 8 | Substitution Cipher | <no new chapter> |
| 9 | Connect 4 | <no new chapter> |
| 10 | Critters | 8. Classes<br>9. Inheritance and Interfaces |

Each student has 8 slip days for Programming Assignments. These days can be used when life circumstances cause a student to miss the deadline such as sickness, university event, car trouble, family trouble, laptop trouble, internet trouble, etc. Avoid submitting your code on Thursday or Saturday after 10pm. Canvas can get overloaded if too many students try to submit at the same time, causing the submission to fail. You will not be permitted to submit your assignment for grading if this is the case, and you will receive a 0 for the assignment. Notes about slip days:

- A maximum of 2 slip days can be used for one Assignment.
- No slip days can be used for the last Assignment.

- The last possible time to submit an assignment, using slip days, is 48 hours after the original due date, Saturday at 11:59pm. Each assignment is closed on Saturday at 11:59pm. After this time, no submissions are possible. There are no extensions. Assignments not submitted, or submitted after the original due date without enough slip days remaining, will receive a 0.
- TAs will begin grading after the Assignment has closed in Canvas. If the most recent submission date/time is passed the due date/time, they will deduct 1 or 2 slip days from the student's available slip days. (Slip days work in units of days. Partial days will not be considered.) The remaining number of slip days is tracked in the Slip Days assignment in Canvas. The student does not need to request slip days since they are automatically applied.

**Unit Exams -** *Tentative exam dates*. Any changes will be posted to Canvas modules and Ed Discussion.

- Unit 1 - Wednesday, September 17, 6:30-8:30pm
- Unit 2 - Wednesday, October 22, 6:30-8:30pm
- Unit 3/Final - During finals week, December 11-15. The day and time will be set by the university.

Exams may consist of short answer questions, multiple choice questions and hand-written code segments. Code segments will be written without the help of online references, a color-coded IDE or a debugger.

- **Exam Time Conflict -** Because Unit Exams are outside of the regular class time, they sometimes conflict with another UT exam. If this is the case, email the professor at least two weeks ahead of the exam day, including a screenshot of your course schedule from the UT registration system and a screenshot of the syllabus that includes the conflicting exam day and time. I will reply with an approval to take the alternate time exam. No one can take the alternate time exam without previous approval. Approvals are given only for a conflicting UT exam time.
- **Missing an Exam -** If you have an issue that will prevent you from taking an exam, it is important to email the professor right away and document the situation through Student Emergency Services. If a student emails the professor before the exam day, the issue is documented through Student Emergency Services, and the professor provides approval, the student will be excused from the exam. In this case, the average of the other two exam grades will be used as the grade for the missing exam. Note that this situation only applies in extreme circumstances and is very rare.

## Letter Grade

The letter grade for the course will be determined by the lowest average in the following categories. This approach requires that students perform well in all three types of assessments.

- Quizzes
- Programming Assignments
- Exams

Grade letter cutoffs are below. Percentages will not be rounded.

| Percent Min and Max | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | **A-** | **B+** | **B** | **B-** | **C+** | **C** | **C-** | **D+** | **D** | **D-** | **F** |
| >=93 | >=90<br><93 | >=87<br><90 | >=83<br><87 | >=80<br><83 | >=77<br><80 | >=73<br><77 | >=70<br><73 | >=67<br><70 | >=63<br><67 | >=60<br><63 | <60 |

Here are several examples:

| Student | Quiz Average | Programming Assignment Average | Exam Average | Final Average | Final Letter Grade | Notes |
|---|---|---|---|---|---|---|
| 1 | 85.78 | 85.12 | 95.67 | **85** | **B** | lowest average is 85 |
| 2 | 79.99 | 93.12 | 98.78 | **79** | **C+** | lowest average is 79, averages are not rounded |
| 3 | 87.01 | 89.99 | 89.11 | **87** | **B+** | lowest average is 87 |

## Extensions and Make-ups

A significant amount of flexibility is built into the grading policy. This reduces the impact of conflicts, like university band or athletics, and also unforeseen situations, like getting sick. The flexibility includes:

- In-Lecture and In-Section Quizzes have no extension or make-ups. At-Home Quizzes have a 2-day grace period. You may turn in any At-Home quiz up to 2 days late, with no penalty. You don't need to request this flexibility. Each At-Home Quiz is open in Canvas for 2 days past the due date.
- At the end of the semester, the 8 lowest Quiz grades (8 total for At-Home, In-Lecture Quizzes, and In-Section Quizzes) will be dropped automatically.
- At the end of the semester, the 1 lowest Programming Assignment will be dropped automatically, as long as the assignment grade had not been adjusted due to a violation of academic integrity.

Because of this flexibility, there are no extensions or makeups for Quizzes or Programming Assignments for any reason, even when a student has a documented sickness or official university excuse. These are examples of reasons for missing class that are covered by the flexible policy and do not get extensions or other adjustments.

- Sick, including Covid or other documented sickness

- Sickness or death in the family
- University-sanctioned events such as an athletic or band event, even OU weekend
- Family commitment
- Religious holiday
- Internet, computer or phone issues
- Test(s) in another class on the same day

The only exceptions to grading policies in this syllabus are documented cases when a student is not able to attend school for more than 2 weeks, for example, a long-term hospitalization. If you are in a situation in which you feel you will be unable to attend class for a period of time, contact Student Emergency Services and your professors as soon as you are able. If you wait to contact the professor about this type of situation until the end of the semester, adjustments will not be possible.

## Regrades

Programming Assignments and Unit Exams are graded based on automated testing and formal rubrics that support objective and consistent grading. Grade adjustments will happen only if the rubric has been applied incorrectly. Adjustments are not made for judgement calls, minor variations in grading or for disagreements about the rubric. If you feel a mistake was made in applying the rubric for a programming assignment or an exam, submit a Regrade Request from Gradescope within 3 days of the grade being posted. Provide details explaining why you believe the rubric was applied incorrectly. Regrade requests without this information will not be considered. *Know that, based on applying the rubric again, the adjusted grade may lower the original grade*. Do not submit a regrade request “just to see” if you can get more points. If you submit too many regrade requests that do not meet these guidelines, you will lose the ability to submit regrade requests. Regrades are not available for Quizzes.

## Adding The Course Late

If you add the class late:

- Watch the lecturers that you missed. Note that In-Lecture Quizzes during the first week of class were practice runs that are not counted in your grade. The same is true for any In-Section Quiz during the first week of class.
- It is important to catch up on the At-Home Quizzes as soon as possible, since they include setup work that needs to be done before other work can be started. While there are At Home Quizzes due the first week, the final due date is not until the second week, with no late penalty. For the Scavenger Hunt Quiz, write a post on Ed Discussion asking to join a group. The TAs will help you find a group.

- If you did not add the class until the 5th or 6th class day (not the 5th or 6 lecture, but the 5th or 6th day that UT classes are held), email Ramsey at ramsey@cs.utexas.edu. She will provide info about the first programming assignment.

### CS 314 to CS 312 Switch

Students may start a semester in CS 314 and switch to CS 312 after 314's Unit 1 Exam. For these students, 312 grades for the time before they switched courses will be as follows:

- The average 314 quiz grade will replace each quiz grade already completed in 312.
- The average 314 programming assignment grade will replace each 312 programming assignment grade.
- The 312 Unit 1 exam grade will be the average of the 312 Unit 2 and Unit 3 exam grades. The CS 314 Unit 1 exam grade will not be used.

## Academic Integrity

For all graded assignments, you must submit work that is 100% your own. This section provides additional information about programming assignments.

### Guidelines

In business, programs are written by teams who work together on code. The team members use online tools, 3rd-party resources and, depending on the context, generative AI tools. The academic setting is different. The purpose of CS 312 is for you to learn how to independently write computer programs to solve nontrivial problems. The only way to learn how to program is to program. The more lines of code you write, the more comfortable you will become with the anxiety that you don’t know the answer and you don’t know how to get there. Each time you struggle, then figure it out, you are building the stamina and skills required to be a better programmer. You won’t do well in the course or at a job if you don’t accept and (somewhat) enjoy the little, seemingly endless, struggles and victories of the programming process.

For this foundational CS course, the lines of code that you submit must come from your own mind. You should understand how they work and be able to explain them to someone else. **Generative AI is not permitted for programming assignments.** Detailed guidelines:

| **Activity** | **OK** | **Not OK** |
|---|---|---|
| Working with others (another student, friend, Mom, tutor, programmer hired on the internet, or other) | Discussing the assignment at a high level and sharing design ideas. | Having another person:<br>● Internalize the instructions or create a design for you.<br>● Look at your code. Don't let someone look at your code and don't look at theirs. This can lead to solutions that are too similar and get flagged for plagiarism.<br>● Help you write lines of code.<br>● Debug your code. |
| Working with a TA | You can share your code with a current CS 312 TA to get help after you read the instructions, try what you know and run the debugger. | Don't ask the TA to explain the instructions before you have read them or to tell you how to code a feature before you have done a design. Do all the work you know how to do first. |
| Working with online resources such as Oracle's Java documentation, GeeksForGeeks, Stack Overflow. | Looking up syntax and generic algorithms also available in the textbook as a reference, for example, a Java for loop.<br>Asking conceptual questions, such as "What is an instance variable?"<br>Searching for information about a specific program error message. | Using a code example of a specific assignment, for example, Connect 4 or Hangman. |
| Using generative AI tool | Same as above | Using AI for any part of a programming assignment. |
| Using language translation tool | NONE | You may not use a tool to translate code you wrote yourself in one language into Java. You must write the Java code yourself. |
| Using Homework Helper sites like Course Hero | NONE | Searching, downloading or looking on someone else's computer at a solution to a programming assignment from a homework help website, even if you are just trying to get an idea for one tiny thing. Looking at a solution to the assignment is always cheating. |

| Activity | OK | Not OK |
|---|---|---|
| Protecting your work | It is your responsibility to be sure your course work is stored in a secure location and not available to others during or after the course. | When the course is over, students are tempted to make their code publicly available to “help” future students or to provide coding samples to potential employers. Do not make your solutions publically available on GitHub, a Homework Help site or by any other mechanism. The course TAs search for publicly available solutions and include them in the plagiarism check process. If your course solutions are found online, the matter will be forwarded to UT for disciplinary action. |

### Process

During the grading process, programs are flagged for possible cheating using tools and manual reviews. If an assignment of yours has been flagged for possible cheating, you will receive an email with “Grading Irregularity” in the subject line. You will need to schedule a time to meet with me during Office Hours using this link. The meeting must happen within 7 days of receiving the email. **You may not drop the course after receiving the email, and before the case has been resolved.** If you drop, and it is determined that cheating has occurred, the drop will be reversed by the registrar and you will be required to complete the course with no adjustments to due dates.

When we meet, I will ask you to describe how you wrote your program and explain how it works. I will ask other questions, based on the flags identified. If I determine that cheating has occurred, I will refer the case to the Student Conduct and Academic Integrity in the Office of the Dean of Students. They will investigate the case and determine penalties, including, but not limited to, receiving a 0 for the assignment that cannot be dropped.

## University Policies and Resources for Students

This Canvas page is a supplement to all UT syllabi and contains University policies and resources that you can refer to as you engage with and navigate your courses and the university.