# CS 312 Introduction to Programming Spring 2024



**Class Meets**

Sections 50720, 50725, 50730, 50735, : Monday, Wednesday and Friday, 9am - 10am, GDC 2.216

Sections 50740, 50745, 50750, 50755: Monday, Wednesday and Friday, 10am - 11am, GDC 2.216

**Instructor**

Professor Ramsey, she/her/hers

Office: GDC 6.318

Email: ramsey@cs.utexas.edu, Do NOT use the Canvas inbox to contact Professor Ramsey.

Office hours: TBD, and by appointment

# 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, algorithm development, problem decomposition and debugging strategies.
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.

### Flags

There are no flags for this course.

#

# How Will You Learn?

### Statement of Learning Success

Welcome to CS 312! I’m impressed by you already, because you have been accepted to UT, a seriously competitive university, and you are also enrolled in this course. CS 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. I encourage you to take advantage of the many resources available. I will share these with you throughout the course. If you start to struggle in the course, please come see me during office hours! We can develop strategies based on your needs to meet the requirements of the course. **Know that, if you made it to this class, you belong here and you can do this.**

### Teaching Modality

This course is an in-person course and lecture and section attendance are required. You will read and complete exercises in preparation for each lecture, to allow the lecture time to be more in-depth and effective. Lectures include a conceptual overview, coding examples and student activities. The student activities are completed in groups and on paper. They allow students to apply what they have just learned to a specific problem. This improves understanding and retention and provides me with immediate feedback on student learning. The professor and a TA will assist groups with the activities and then address common mistakes and suggestions for improvement. Because the student group activities are essential, attendance is required. In-Class Quizzes are part of each lecture, to measure attendance and learning, and to incorporate regular attendance and incremental progress into students’ course grades.

This course includes a one-hour, weekly Section (sometimes called Discussion) on Mondays. A Teaching Assistant (TA), a UT CS student who was in 312 just like you not too long ago, leads each Section. They provide a smaller group environment for practice and expansion of lecture content. Section activities are designed to bridge the gap between lectures and specific, complex programming assignments. Because of this, Section attendance is required. Students may periodically earn Quiz grades based on their attendance in Section.

### 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 on Monday, 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, they are generally awesome people. You should at least know their name and hometown! Here is the TA info for each Section. Refer to the [Help Schedule](https://docs.google.com/spreadsheets/d/1zpfO2Hlb_rC5IkmjCBoJzei7TQeF5knYQV9wQXmWRAY/edit?usp=sharing) for up-to-date Help Hour days and times.

**MWF 9am Lecture TAs**

| **Section** | **TA Name** | **Meeting Time** | **Meeting Location** | **TA Email****(grading issues only)** |
| --- | --- | --- | --- | --- |
| 50720 |  |  |  |  |
| 50725 |  |  |  |  |
| 50730 |  |  |  |  |
| 50735 |  |  |  |  |

**MWF 10am Lecture TAs**

| **Section** | **TA Name** | **Meeting Time** | **Meeting Location** | **TA Email****(grading issues only)** |
| --- | --- | --- | --- | --- |
| 50740 |  |  |  |  |
| 50745 |  |  |  |  |
| 50750 |  |  |  |  |
| 50755 |  |  |  |  |

OLD TAS

**MWF 9am Lecture TAs**

| **Section** | **TA Name** | **Meeting Time** | **Meeting Location** | **TA Email****(grading issues only)** |
| --- | --- | --- | --- | --- |
|  | Akhilesh Bitla |  |  | akhilesh.bitla@gmail.com |
|  | Tanay Garg |  |  | tanay.garg@utexas.edu |
|  | Tarun Somisetty |  |  | tarun.somisetty@utexas.edu |
|  | Silas (Minh-Triet) Pham |  |  | silas@utexas.edu |
|  | Minerva Sharma |  |  | minervasharma@utexas.edu |

**MWF 11am Lecture TAs**

| **Section** | **TA Name** | **Meeting Time** | **Meeting Location** | **TA Email****(grading issues only)** |
| --- | --- | --- | --- | --- |
|  | Kaylee Marquez |  |  | kaylee.marquez@utexas.edu |
|  | Siddh Shah |  |  | siddh4shah@gmail.com |
|  | Pranav Bhuvanagiri |  |  | pranavbhu@utexas.edu |
|  | Saul Gutierrez |  |  | saulgtz02@utexas.edu |
|  | Kevin Ayala |  |  | kevinayala@utexas.edu |
|  | Austin Teng |  |  | austinteng@utexas.edu |

### 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 a student who has done very well in the course. The leaders prepare for each session, providing practice and resources to reinforce understanding. This is a great place for:

* Asking questions - The sessions are small and the leaders are very understanding. If you aren't comfortable asking questions in lecture, then this is the place for you!
* Reducing stress - If you make a habit out of attending a weekly Study Session, you are going to be better prepared for assignments and tests, and get more sleep in the long run.
* Gaining experience - Sometimes, when learning a new concept, it makes sense in general, but you aren’t sure you can use it to solve new problems. Also, this course can be intimidating for students who don’t have a lot of programming experience. The way to fix these issues is to practice, practice, practice, to build up your experience and confidence. Group Tutoring offers practice, with an awesome coach to help.

There are two Free Group Tutoring Programs, Supplemental Instruction and Collaborative Study Sessions. They each hold 2 sessions per week for 4 sessions total. Each program’s two sessions during the same week are identical. Most students will choose one of the Group Tutoring Session times to attend consistently each week. When student leads, days and times are set, they will be announced in class and on Ed Discussion. They will also be added to the [Help Schedule](https://docs.google.com/spreadsheets/d/1zpfO2Hlb_rC5IkmjCBoJzei7TQeF5knYQV9wQXmWRAY/edit?usp=sharing).

### 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, there is a mysterious nature about you babe, I mean about CS, 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 breakfast. 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](https://deanofstudents.utexas.edu/emergency/), (512) 471-5017, studentemergency@austin.utexas.edu

# Course Requirements and Grading

### Required Materials

Building Java Programs: A Back to Basics Approach (5th Edition) online textbook,

$44.99 (expires in 6 months), $74.99 (never expires)

### Required Devices

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

You will need a smartphone in lecture to complete Daily Quizzes.

### High Level Schedule

| **Unit** | **Dates** | **Chapter/Topic** |
| --- | --- | --- |
| 1 | January 17 - February 7 | 1. Introduction2. Primitive Data and Definite Loops3. Parameters and Objects |
| 2 | February 9 - March 27 | 3G. Graphics4. Conditional Execution5. Program Logic and Indefinite Loops6. File Processing7. Arrays |
| 3 | March 29 - April 29  | 8. Classes9. Inheritance and Interfaces10. ArrayLists12. Recursion |

### Course Expectations

**Lecture** - Lectures are in person and synced with the textbook chapters. Before attending lecture, complete the textbook reading listed for the day in Canvas. You may also choose to watch videos, which are available for some lectures. (NOTE: The videos are supplemental and cover a small percentage of the content.) The first activity for most lectures is a Daily Quiz in Canvas. The quiz question(s) come from the after-class practice problems listed in Canvas for the previous lecture. You are expected to arrive on time to lecture, not leave early, get to know your classmates and participate in the activities. All devices must lay flat on your desk (a paper notebook or tablet, no open laptops). Actively engaging in the activities will improve your performance in the course. If you miss a lecture, watch the recorded session available on Lectures Online in Canvas. After lecture, complete the After-Class practice problems listed in Canvas. If you need more practice, then do more problems. **Most students who pass this class complete a great number of practice problems, even though they aren’t graded.** Doing this work will also ensure an awesome grade for the In-Class Quiz during the following lecture.

**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. Bring your laptop to Section. Attendance is required and may be the basis for periodic quiz grades. Section is a great place to find a study group to help study for unit exams.

**Office Hours (with Professor)** - I am available for one-on-one or group meetings on Zoom or in my office (GDC 6.318) to talk about your progress in the course, strategies to improve your learning, or for help with assignments. To provide students with dedicated time, you must sign up for Office Hours ahead of time using this [calendar](https://calendly.com/ramseytx/10min).

**Help Hours (with TAs and Professor) -** TAs and the Professor will be available in the GDC 3rd floor lab (in the GDC north tower, closest to Speedway) on a set schedule to provide help with course content and assignments. To make efficient use of available TA time and ensure all students are able to ask questions, it is important to do work on your own first, then ask TAs specific questions. Include what you know already, what you have tried already and what you learned from the debugger. Avoid general questions like, “Can you explain the assignment to me?” and “My code doesn’t work, can you fix it?”

**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 you write a new post, include the problem, what you expected to happen, what is happening, and what you have tried already. Complex or conceptual questions with long answers 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!

**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. 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**

* 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.
* If life circumstances are impacting your course work, contact the professor as soon as possible.

### Student 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. Responses will take longer on weekends. Be aware of this when using slip days for a Programming Assignment. Quick Ed Discussion replies are not likely and should not be expected on Friday night or Saturday.

**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.



### 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** (2-3 per week, on average) - Quizzes are implemented in Canvas. They provide frequent feedback on student progress.

* **At-Home Quizzes** are completed outside of lecture, and reflect participation in setup, surveys and program designs. Quiz timing and due dates are defined in Canvas modules. At-Home Quizzes have a 2-day grace period for each due date, with no penalty or slip days charged.
* **In-Class Quizzes** occur during lectures, to measure attendance and incremental learning. The quiz starts at the beginning of lecture and is completed on student smartphones. The questions are based on questions from the previous lecture’s after-class practice, defined in the Canvas modules. Additional questions may follow during the lecture, based on lecture content or in-class activities just completed. The quiz grade is based on attendance (60%) and correctness (40%). It is not possible to take an In-Class Quiz outside of the original in-person lecture.

**Programming Assignments** (About 1 per week) - Each Programming Assignments begins on a Friday and is due at 11pm the following Thursday. Each Assignment includes detailed instructions that must be followed to receive full credit. Plan to spend 20 hours per week programming. Each assignment is graded with automated test cases and TA manual grading. You must earn at least half of the autograded points to be eligible for manual grading points. 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** | **Due Date****Thursday at 11pm** |
| --- | --- | --- | --- |
| 1 | I Know an Old Lady | 1. Introduction | 1/25 |
| 2 | UT Tower | 2. Primitive Data and Definite Loops | 3/1 |
| 3 | Scintillation Grid | 3. Parameters and Objects3G. Graphics | 2/15 |
| 4 | Rock, Paper, Scissors | 4. Conditional Execution | 2/22 |
| 5 | Hangman | 5. Program Logic and Indefinite Loops | 2/29 |
| 6 | Home Field Advantage | 6. File Processing | 3/7 |
| 7 | Personality Quiz | 7. Arrays | 3/21 |
| 8 | Substitution Cipher | <no new chapter> | 4/4 |
| 9 | Connect 4 | <no new chapter> | 4/11 |
| 10 | Critters | 8. Classes9. Inheritance and Interfaces | 4/18 |
| 11 | Guitar Hero | <no new chapter> | 4/25 (no slip days) |

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 11pm. Each assignment is closed on Saturday at 11pm. 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, so students can see how many they have left. The student does not need to request slip days since they are automatically applied.

**Unit Exams -** This course has three major Exams, each in-person and on paper. The *tentative* Unit Exam dates are February 7, March 27 and the week of May 1. Exams may consist of short answer questions, multiple choice questions and hand-written code segments. Writing code by hand, without the help of online references, a color-coded IDE or a debugger, is challenging. This is why the lecture activities provide practice writing code by hand. The first two Exams are 2 hours long and held outside of lecture times, on Wednesday evenings. The last Unit Exam will take place during finals week and be 3 hours long. At the end of the semester, each students’ lowest exam score will be replaced by the average of their 3 exam scores. For example, if the unit exam grades are 70, 80 and 90, the 70 will be replaced by an 80. **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, she provides approval, and the issue is documented through Student Emergency Services, the student will be excused from the exam. In this case, the average of the other two unit exam grades will be used as the grade for the missing exam.

### Letter Grade

The grading categories will be adjusted and weighted as follows:

| **Category** | **Adjustment** | **Weight** |
| --- | --- | --- |
| Quizzes  | Drop lowest 5 grades | 10% |
| Programming Assignments | Drop lowest 1 grade | 20% |
| Exams | Lowest grade replaced by average of three exam grades | 70% |
| TOTAL | 100% |

Grade letter cutoffs are below. Percentages will not be rounded. For example, to earn an A, you must have a percent of 93.0 or above. A 92.99 is an A-.

| **Percent Min and Max** |
| --- |
| **A** | **A-** | **B+** | **B** | **B-** | **C+** | **C** | **C-** | **D+** | **D** | **D-** | **F** |
| >=93 | >=90 <92 | >=87<89 | >=83<86 | >=80<82 | >=77<79 | >=73<76 | >=70<72  | >=67<69 | >=63<66 | >=60<62 | <60 |

### 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:

* 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 5 lowest Quiz grades (5 total for both At-Home and In-Class 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. This flexibility covers both valid and less valid reasons. These are examples of cases when there are no extensions or makeups. This is not a comprehensive list.

* 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
* Camping off the grid for a week in the middle of the semester
* Oversleeping
* Internet issues
* Computer wasn’t working
* A test 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.

### Grade Adjustments

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. A grade won’t be adjusted based on time, effort, impact on class average, or other reason. If you feel the rubric for a Programming Assignment was applied incorrectly, follow these steps:

* Quiz - Email the professor within 1 week of the grade posting on Canvas. Explain why you think the grade is incorrect.
* Programming Assignment - Email your TA within seven days of the date of the grade posting on Canvas. Provide details explaining why you believe the rubric was applied incorrectly.
* Unit Exam: Unit Exams are returned in Section. If you feel a Unit Exam rubric was applied incorrectly, you must tell the TA during the Section in which the exam was returned. You must return the exam to the TA and follow-up with an email explaining why you believe the rubric was applied incorrectly.

# Course Tools

The class will use the following tools. Refer to the Setup Instructions for information on downloading software and setting up accounts.



# Academic Integrity

In the real world, software is developed collaboratively by teams of developers who use code written by others all the time (sample code, open source code, third-party libraries, etc.). Not only are collaboration and code reuse both good for software, a developer who attempts to work in isolation won’t understand the context, dependencies or business environment in which her code needs to work. But, a university setting requires individual grades and each course needs to prepare individual students for future courses. What is the balance between the two realities?

The Computer Science department Code of Conduct states: “The University and the Department are committed to preserving the reputation of your degree. It means a lot to you. In order to guarantee that every degree means what it says it means, we must enforce a strict policy that guarantees that the work that you turn in is your own and that the grades you receive measure your personal achievements in your classes.”

Students who violate this policy will be subject to discipline including:

* One-on-one meeting with the professor to discuss the situation.
* Earning a 0 for the given work. This grade cannot be dropped as the lowest grade.
* Report to the Dean of Students for further disciplinary action.

For all graded assignments, you must submit work that is 100% your own.

Important ideas when working on Programming Assignments:

**The lines of code that you submit come from your own mind.**

**You should understand how they work and be able to explain them to someone else.**

**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.**

**The best way to avoid the temptation to cheat is to start early and finish early.**

**Most cheating happens on Thursday and Saturday nights.**

**Helping someone else cheat is just as bad, and has the same consequences, as cheating yourself.**

Here are more specific guidelines:

| **Activity** | **OK** | **Not OK** |
| --- | --- | --- |
| Working with a TA  | Showing the TA your high level design and asking for help with specific questions.Showing the TA your code and asking specific code or debugging questions.  | Asking the TA to do the design work for you by asking general questions like, “How should I solve this problem?”Asking the TA to debug your program for you, before you have tried different things and run the debugger. For example, “Here is my code, can you debug it for me?” |
| Working with others writing code | Talking with other current students to discuss Programming Assignments at a high level.  | Working with another classmate, a previous CS 312 student, your Mom, a tutor or any other person on code-level details. Any time you are thinking of specific Java statements, you should be working on your own. Don’t let other people read your code and don’t read someone else’s code. It is too easy, even if you don’t mean too, to type code that is too similar after seeing one solution. Don’t type code into someone else’s file, even if you are trying to be helpful. |
| Working with others on debugging code  | You can describe a debugging issue at a high level to another person, asking for help. This includes the error message, what you expected and what you tried already.  | You cannot show another person your code or look at another student’s code, working together to fix a specific bug. |
| Using GeeksForGeeks, Stack Overflow or other online coding resource | Using examples of syntax and generic algorithms also available in the textbook as a reference, for example, a Java for loop.Searching for information about a specific program error message. | Using an example of a specific assignment, for example, Connect 4 or Hangman. |
| Using Chat GPT or other AI tool | You may use Chat GPT to get examples of syntax and standard algorithms that are also found in your textbook, or example, a Java for loop. | You may not use Chat GPT for your Programming Assignments. It is too tempting to ask Chat GPT to fine tune a generic algorithm specific to the Programming Assignment to a degree that the plagiarism checkers will be triggered. Know that the TAs create several Chat GPT solutions to each assignment, they contain identifiable characteristics, and these are compared with student submissions.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. |
| 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. |

Know that I wouldn’t spend the time to write this much about Academic Integrity if it wasn’t important and if I wasn’t sad each semester when students fall short. My one-on-one meetings with students who have cheated are the saddest part of my job.

Examples of students who have been caught cheating in my courses:

* Students genuinely intending to be helpful, letting another student look at their code or sending another student their code, when the person says they just want to look at it for one thing for one minute.
* Students working together, helping each other, but one or both students start looking at the other student’s code for extended periods of time. Then, segments of their code use the same very-specific solution.
* Students viewing solutions to the given assignment from GitHub, Course Hero or another “homework help” website and:
	+ Copy and pasting the solution and submitting as-is.
	+ Copy and pasting the solution, doing search-and-replace changes (variable names, method names, etc.) and formatting/comment changes (more blanks lines, different size indents, editing comments, etc.). The plagiarism checkers are smart enough to recognize these types of changes as superficial, and the code similarities are still flagged.
	+ Only using the code as a reference and doing their best to write their own solution. But, once they have seen a solution, they end up writing code that is too similar.

It might seem that, given the same Programming Assignment that requires the same output, two students might organically arrive at the same solution. But, given the complexity of the Programming Assignments and the hundreds of micro-decisions required to develop a solution, it is extraordinarily unlikely that two students will arrive at the same solution. In cases when a violation isn’t clear, you will be asked to explain the code you have written to demonstrate your understanding.

# Course Policies and Disclosures

**University Code of Conduct**

The core values of the University of Texas at Austin are learning, discovery, freedom, leadership, individual opportunity, and responsibility. Each member of the University is expected to uphold these values through integrity, honesty, trust, fairness, and respect toward peers and community.

**University Student Honor Code**

I pledge to be honest about what I create and to acknowledge what I use that belongs to others.

I pledge to value the process of learning in addition to the outcome, while celebrating and learning from mistakes.

This code encompasses all of the academic and scholarly endeavors of the university community.

**UT Computer Science Code of Conduct**

We in the UTCS community will act with:

Respect: We will respect others’ viewpoints and experiences.

We will speak with kindness, lead with empathy,

and approach discussions with our colleagues with trust.

Integrity: We will maintain integrity in our actions,

including those related to our academic, professional, and personal selves.

Awareness: We will seek to be aware of how our actions affect others,

including the larger social impact of everything we create or to which we contribute.

We will be aware that everyone travels different paths, leading to different experiences.

### Covid

Behavior Concerns and COVID-19 Advice Line, [website](https://safety.utexas.edu/behavior-concerns-advice-line), 512-232-5050

If you are feeling sick, please stay home. If you are worried about how this will affect your work or grade in class, please email me. The [Protect Texas Together](https://protect.utexas.edu/) website has the latest information about Covid, including the local threat level and how to get vaccinated and tested. If you may have been exposed to Covid, refer to the university’s [Exposure Action Chart](https://healthyhorns.utexas.edu/coronavirus_exposure_action_chart.html) for next steps. Refer to the [university policy](https://healthyhorns.utexas.edu/coronavirus.html) on COVID for more information and resources. Masks are recommended, but not required on campus. Getting vaccinated and getting tested when you feel sick are highly recommended. Also, if you have Covid concerns, call the Behavior Concerns and COVID-10 Advice Line.

### Title IX Reporting

[Submit a Report](https://utexas-gme-advocate.symplicity.com/titleix_report/index.php/pid105743), [website](https://titleix.utexas.edu/), titleix@austin.utexas.edu, 512-471-0419

If you have been a victim or if you know about an incident of sexual harassment, sexual assault, interpersonal violence or stalking, the UT Title IX office wants to help. It’s important that all students have the opportunity to learn while feeling safe. You can submit a report online, by email or by phone. You will be in control of the process, making decisions that are right for you. Also know that, if you share information about an incident of this nature, I am required to file a report with the Title IX Office. This won’t mean that you are in trouble and you won’t be required to work with them. Just know that they are here to help. Case managers can provide support, resources, or academic accommodations. Case managers can also provide support, resources, and accommodations for pregnant, nursing, and parenting students. Methods to contact them or submit a report are above.

### Names and Pronouns

Class rosters are provided to the instructor with the student’s legal name, unless they have added a chosen name with the registrar’s office. If you have not yet done so, I will gladly honor your request to address you with the name and pronouns that you prefer for me to use for you. It is helpful to advise me of any changes or needs regarding your name and pronouns early in the semester so that I may make appropriate updates to my records and be informed about how to support you in this class.

* For instructions on how to add your pronouns to Canvas, visit [this site](https://utexas.instructure.com/courses/633028/pages/profile-pronouns).
* If you would like to update your chosen name with the registrar’s office, you can do so [here](https://enterprise.login.utexas.edu/idp/profile/SAML2/Redirect/SSO?execution=e1s2), and reference [this guide](https://docs.google.com/document/d/17uzmcD7oGE5JPMueJN7CsBlgE7SICUYu7ysmrFgc8cM/edit).
* For additional guidelines prepared by the Gender and Sexuality Center for changing your name on various campus systems, see the Resources page under UT Resources [here](https://diversity.utexas.edu/genderandsexuality/publications-and-resources/).

### Disability and Access

Disability and Access [Website](https://diversity.utexas.edu/disability/), ssd@austin.utexas.edu, 512-471-6259 (voice) , 866-329-3986 (video phone)

Please let me know if you experience any barriers to learning so I can work with you to ensure you have equal opportunity to participate fully in this course. If you have a disability, or think you might have a disability, please contact Disability and Access to learn about how you can fully participate in your courses. If you are already registered, I will receive your accommodation letter at the beginning of the semester. We will meet early in the semester to discuss your accommodations.

### Inclusion and Equity

Office of Inclusion & Equity, [Submit Concern form](https://utexas.qualtrics.com/jfe/form/SV_bIriVPRhsfP3UCW), [website](https://equity.utexas.edu/), equity@utexas.edu, 512-471-1849

My goal is to create a welcoming and productive environment for all students. If you don’t feel welcome or if you experience a barrier to learning, please let me know through email or office hours. I was one of two female CS students at UT in the 90s and I didn’t always feel welcome. This seems trivial to those who haven’t experienced it, but this feeling can be draining and make it harder to do your best work. I want to know if you feel this way and work to find ways to fully welcome you into the class. You can also submit a concern about this class, or another UT experience, to the Office of Inclusion and Equity.

### Religious Holy Days

Office of Inclusion and Equity, [website](https://equity.utexas.edu/), equity@utexas.edu, 512-471-1849

If you need to miss a class or Assignment due to a religious holy day, you will be able to complete the missed work within a reasonable amount of time after the absence. Please let me know fourteen days before the absence, so we can make arrangements. If you have questions about religious accommodations, contact the Office of Inclusion and Equity.

### Sharing of Course Materials is Prohibited

No materials used in this class, including, but not limited to, lecture slides, activity handouts, recorded lectures, videos, assessments (quizzes, exams, papers, projects, homework assignments), reference materials, review sheets, and additional problem sets, may be shared online or with anyone outside of the class without my explicit, written permission. Unauthorized sharing of materials may facilitate cheating. The University is aware of the sites used for sharing materials, and any materials found online that are associated with you, or any suspected unauthorized sharing of materials, will be reported to [Student Conduct and Academic Integrity](https://deanofstudents.utexas.edu/conduct/) in the Office of the Dean of Students. These reports can result in initiation of the student conduct process and include charge(s) for academic misconduct, potentially resulting in sanctions, including a grade impact.

### Campus Carry

[Campus Carry Policy](https://www.utexas.edu/campus-carry)

Students who carry a weapon on campus should be familiar with related policies. Note that I prohibit the carrying of a weapon in my personal office.

### Behavior Concerns

Behavior Concerns and COVID-19 Advice Line, [website](https://safety.utexas.edu/behavior-concerns-advice-line) , 512-232-5050

If you have concerns about the safety or behavior of fellow students, TAs or professors, contact the Behavior Concerns and COVID-19 Advice Line. If something doesn't seem right it doesn’t hurt to call. They can help evaluate the situation and you can remain anonymous in most cases.

### UT Safety

UT IT, [Email update](https://utdirect.utexas.edu/utdirect/bio/address_change.WBX), [Emergency contact update](https://utdirect.utexas.edu/apps/student/emergency_contact/)

Be sure that the university has your current email and emergency contact information. Remember to check your email daily.

### University Resources

For a list of university resources that may be helpful to you as you engage with and navigate your courses and the university, see the [University Resources Students Canvas page](https://utexas.instructure.com/enroll/TP964H).

### Land Acknowledgement

I would like to acknowledge that we are meeting on the Indigenous lands of Turtle Island, the ancestral name for what is now called North America. Moreover, I would like to acknowledge the Alabama-Coushatta, Caddo, Carrizo/Comecrudo, Coahuiltecan, Comanche, Kickapoo, Lipan Apache, Tonkawa and Yslete Del Sur Pueblo, and all the American Indian and Indigenous Peoples and communities who have been or have become part of these lands and territories in Texas.