Computer Science 1324
Introduction to Computer Programming for Non-Programmers
Fall 2025

Class Time: Monday, Wednesday, and Friday. 9-9:50 (Section 1) and 10-10:50 (Section 2)
Class Location: Section 1: Carson 438, Section 2: Devon 130
Exam Location: Dale Hall 206 (both sections and all three exams, including the final)
Instructor: Dr. Deborah Trytten Email: dtrytten@ou.edu Office: Devon 252

This course has three essential learning objectives:

1. Learning to write a few lines of code to solve given problems.
2. Learning to write complete computer programs to solve given problems.
3. Developing conceptual understanding of how computer programs work.

Students achieve these objectives through extensive practice in the Java programming language. This practice occurs in phases. First, students read about the lecture material before class. Then we discuss the material in class while working many examples together and checking understanding with a classroom response system (iClickers). After class, students practice with interactive programming tutors (Turingscraft, CodingBat problems), projects done in the laboratory, and homework assignments. This means that there are class deadlines every two or three days. Repeated practice leads to mastery.

Teaching Assistants (TAs):

Name	Email	Time	Laboratory Location
Mehdi Zaeifi	zaeifi@ou.edu	Section 11 (11-12:50)	M207 Sarkeys
Medhi Zaeifi		Section 12 (1-2:50)	M207 Sarkeys
Edward Wages	Edward.J.Wages-1@ou.edu	Section 13 (3-4:50)	M207 Sarkeys
Edward Wages		Section 14 (5-6:50)	M207 Sarkeys
Karan Prajapati	Karan.Suresh.Prajapati-1@ou.edu		
Ujwala Vasireddy	Ujwala.Vasireddy-1@ou.edu		

Office Hours: On the home page on Canvas. These sometimes change during the semester. Temporary changes will be announced through email. Permanent changes will be announced through email and made on Canvas.

Class Structure: This class is a synchronous in person class. You are expected to attend all class sessions and all laboratories in person at the scheduled time unless you are ill or have unanticipated caretaking responsibilities. Midterms and the final examination in the class will take place during the evening and must be done in person unless there are extenuating circumstances, usually arranged in advance.

Canvas Learning Management System: https://canvas.ou.edu

Log in with your OUNetID (usually the first 4 letters of your last name followed by a 4-digit number). All assignments, deadlines, grades, announcements, and course documents will be posted to the CS 1324 Canvas page. It is your responsibility to regularly check for updates. You can configure Canvas to email you notifications or send them through text messages. Please do this so you don't miss assignments.

Prerequisites:

1. Math 1523 Precalculus and Trigonometry, equivalent, or concurrent enrollment.
2. Basic computer literacy such as the ability to install software and navigate folder structures.
3. Little or no prior programming experience. If you've taken a programming course before, you must enroll in CS 1323 or CS 1321 instead.

Free Tutoring: In addition to faculty and TA office hours, the School of Computer Science through the William Kerber Foundation Teaching Scholars, and the Dean's Leadership Council (DLC) of the Gallogly College of Engineering. The course TAs, me, and these tutors should be the first place you go for help, since we are all trained specifically to support students in this course. The times, people, locations, and Zoom links will appear on Canvas under Pages as they are available (usually about 2-3 weeks into the semester).

Topics Covered: Programs, Java, identifiers, variables, assignment statements, constants, memory diagrams, primitive data types, operations on primitive data, conditional statements, input and output, repetition, methods, parameters, arguments, return values, passing by value, passing by sharing, nested control statements, one dimensional arrays, objects, user defined classes, and classes from the Java Application Programmers Interface (API) (including Arrays, ArrayList, Character, Collections, Double, Integer, Float, Math, Scanner, String, and StringBuilder), tracing program execution, debugging.

Learning Objectives:

1	Select the appropriate type for data stored in a program from int, double, String, char, boolean, and identify constants.
2	Document code using comments, indentation, whitespace, selection of appropriate identifiers, Javadoc (reading, not writing), and standard Java code conventions.
3	Trace memory allocation of primitive data, arrays, and String and ArrayList objects in a single method.
4	Construct and trace statements using arithmetic, relational, and logic and assignment operators on data, considering both precedence and promotion.
5	Call class methods the Math, Integer, Character, Double, String, Arrays, and Collections classes.
6	Perform input from the console and from files using the Scanner class.
7	Perform output using the PrintStream (System.out) and PrintWriter classes
8	Construct and trace conditional and loop statements.
9	Select and nest control structures (sequence, conditional statements, loops, and methods) to meet requirements.
10	Construct and call methods to meet requirements when given the signature using parameter passing and return values.
11	Implement classes with static methods and no instance or class data to solve programming problems.

12 Predict the contents of memory following method calls using pass by sharing (reference), pass by value, and return values with primivite data, arrays, and String and ArrayList objects.
13 Solve programming problems using arrays of primitive data types and String.
14 Design method signatures that allow arrays that are entirely and partially filled with data to manipulated.
15 Trace selection and insertion sort on arrays of primitive data types and String.
16 Call instance methods in the ArrayList, String, Integer, Character, Double classes.
17 Solve programming problems using ArrayList of wrapper classes (Integer, Character, Double) and String.
18 Construct UML diagrams from API documentation and code.
19 Implement and call methods from simple student defined classes with instance and class data, accessors, and mutators from UML diagrams.
20 Find and fix errors in student written programs.

ABET: Students will increase their ability to meet the following ABET outcomes:
Outcome 1: Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.
Outcome 2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

Required Materials: Please purchase the following items **as soon as possible**. Each is required to complete a different type of assignment and counts for your grade. If you are unable to afford these items at the start of the semester, let me know so I can excuse your work for a little while or help you find another way to get access.

1. Laptop computer with network access for class and lab.
   - You are responsible for having a working laptop available for every class and laboratory. This includes finding a replacement with the necessary software installed if your laptop requires repairs during the semester. The TAs can help with this during office hours.
2. Zyante online textbook (ZyBook) with labs (zyLabs) and Coding Bat problems.
   - Click on the first Zybook assignment in Canvas, which has the title "Ch. 1: Introduction to Programming."
   - Click the link at the bottom of the assignment page to open the Zyante website in a new window.
   - Subscribe to the book.
3. Turing's Craft CodeLab
   - Click on the first interactive tutoring assignment in Canvas, which has the title "TC 1: Storing and Changing Primitive Data."
   - Click the link at the bottom of the assignment page to open the Turing's Craft website in a new window.
4. iClicker subscription

- You will receive an email that gives a link for registering for the iClicker. Be careful to use your official OU address (which looks like this: Deborah.A.Trytten-1@ou.edu, not your alias that looks like this: dtrytten@ou.edu).
- Install the mobile app on your device from the App Store or Google Play. You may use the web app if you prefer.
- iClickers include a free two-week trial. The access code needs to be purchased before two weeks ends. This can done from the campus bookstore or online. Go to http://iclicker.com (this cannot be done from the mobile app) to enter your access code.

Midterm Exams: We will have two midterm examinations: Monday October 6 from 7:30-9:30 pm and Monday November 10 from 7:30-9:30 in Dale Hall 206. These dates are on Canvas on the Calendar, and on the Assignments tab under Midterms.

Makeup exams are only available when required by University policy. Missing an exam without a previously approved excuse will generally result in a grade of zero unless there are extenuating circumstances. If a midterm is excused, your grade on the final examination will be used to replace the grade. In the unlikely event that the final exam is excused, the grade on the second midterm will be used to replace the grade.

Final Exam: 7:30-9:30 pm on Wednesday, December 17 in Dale 206. The College of Engineering requires this exam to be comprehensive. No final exam will be given early except as required by university policy.

Study Advice: In technical fields like computer science, the only way to master the subject matter is to practice. Learning to program is like learning to play a musical instrument. You can read a hundred books on playing the piano, but if you don't sit down in front of a keyboard and practice, you won't be able to produce any interesting music. Similarly, if you only attend lectures or read the textbook, you're going to find it difficult to write functioning computer programs of any complexity.

To facilitate your practice, this course has different types of assignments, which are described below. Start each assignment as early as you can and get help from me, a teaching assistant, or a tutor if you get stuck. Programming is very precise and can be challenging, which is why we have so much support available for this class. If you work hard on and understand the assignments, you should do well on the exams, earn a good grade in the class, and generally have a rewarding and fun semester.

Assignments: This course has 5 different assignment types. Each is designed to help you learn the material in a different way. All assignments are due at 11:59 PM on their posted due dates, except iClickers, which are due in class.

1. Zyante (zyBook): The online textbook will introduce you to new topics before I cover them in class.
   - Each section contains activities to reinforce the ideas in the text. Activities come in two types: participation and challenge. **You are only required to complete participation activities**, although you are welcome to complete challenge activities for

extra practice. Some of the challenge exercises are kind of weird, which is why I don't assign them.
- Each question can be attempted an unlimited number of times without a penalty. You earn 1 point for each question answered correctly before the deadline.
- Some sections are marked as optional. You are not required to complete these sections.
- Zybook assignments are due one day before we start discussing a topic in class, to prepare you for the discussion.

2. Participation: These assignments are given in class to keep you engaged and determine which topics need additional clarification.
   - Participation activities are given with iClickers
   - You earn 1 point for each class day where you answer half or more of the quiz questions correctly. You get one point for signing in during class. On each question you get half a point for answering and another half for answering correctly. This means that students who attend class and answer all the iClicker questions get full credit. I may use geolocation to prevent people who are not attending class from getting credit.
3. Turing's Craft (TC) and CodingBat (CB): Interactive tutoring assignments reinforce topics after they are discussed in class.
   - TC exercises and CB problems are answered in a web browser and provide immediate feedback on your code. Answers are often given in the form of code fragments (a few lines of code), rather than complete programs.
   - Each assignment is due about three days after the last class on the corresponding topic.
   - As with zyBook assignments, there is no penalty for incorrect answers. You earn 1 point for each TC exercise and 2 points for each CB problem answered correctly before the deadline.
   - We have been having some difficulty getting Turingscraft to synch to Canvas, requiring some manual work. This means that the grades on the Turingscraft site may not be available on Canvas immediately.
   - Turingscraft assignments are due usually two days after we complete a topic. CodingBat exercises are due three days after we finish a topic.
4. Homework: These assignments consist of questions that are like those on the exams.
   - Each homework will be posted as a PDF file with fillable fields. You can read and answer the questions using Acrobat Reader: https://get.adobe.com/reader. Do not use reader within a browser, you will need to use a desktop app to be able to save your changes.
   - Homework is submitted through Gradescope, which can be accessed from the link on the left side of the course Canvas page.
   - Homework is generally due five days after the last class on the corresponding topic.
5. Laboratory Projects: These assignments require you to solve a problem in a less structured environment by writing a complete program.

- Projects are assigned in lab each Monday and are submitted on zyLabs by the following Friday Some problems are due sooner due to scheduling constraints (midterms, the final exam).
- You and another student will be paired together to work as a team. You will perform pair programming, which has significant and important benefits: https://en.wikipedia.org/wiki/Pair_programming.
- Each exam will have a final problem that requires you to write most of a complete program. This question is typically worth 30–40% of the exam points. Projects are the assignments most like these questions. Thoroughly understanding and completing each project is thus an excellent way to prepare for exams.

All assignments other than projects must be completed *individually*.

Laboratory Sections: All labs are held on Mondays.

The following are answers to frequently asked questions about the labs:

- You must attend the lab you are enrolled in to receive credit on a project. Missing your lab will result in a grade of zero, even if you complete the project independently and submit it before the deadline.
- Remote attendance via zoom is permitted for laboratories for students only in extenuating circumstances and by prior arrangement with Dr. Trytten. The TAs do not have the authority to allow students to attend lab virtually.
- Because laboratory assignments are done with a partner, if you are late to lab by more than 10 minutes or leave before the project is finished, you will receive a zero on the project.
- Lab partners will be assigned by the instructor at least twice during the semester.
- Although you and your partner will collaborate on each project, you *both* must submit a copy of your source code. Make sure you each have a copy before leaving lab! Do not trust your partner to email it to you later.
- Projects are graded according to objectives included in each project handout. Review the objectives before submitting your code so you don't lose points.
- Incomplete projects can be submitted for partial credit, but source code that does not compile or fails immediately upon execution generally receives no credit.

Grading: Your final course grade is calculated using the weights in the table below. The weights applied to the Zyante, iClicker and Turingscraft & CodingBat grades are intentionally low. This allows you to learn from mistakes with only small penalties. Completing these assignments is how most students develop the conceptual understanding needed to do well on the homework, projects, and especially exams.

Assignment	Weight (%)
Zyante	5
iClicker	5
Turingscraft & CodingBat*	5
Homework	15
Lab Projects	20
Midterm Exams	30
Final Exam	20

* Zyante questions and Turing's Craft exercises are each worth 1 point, and CodingBat problems are each worth 2 points.

Letter Grades: Your course grade will be converted into a letter using a scale no higher than the following. The scale may be lowered at the end of the semester at my discretion.

Letter	Percentage
A	90+
B	80–89
C	70–79
D	60–69
F	Below 60

Borderline Grades: It would be nice if all course grades fell cleanly into the ranges shown above. Most semesters, however, a handful of letter grades are decided by only a few points. In these difficult cases, I will use the following algorithm: A course grade is considered a borderline grade if it is within three points of the next higher letter. For example, 87, 68 and 79 are borderline course grades, but 81 and 92 are not. For borderline grades, if the grade on the final exam is above the threshold for the higher letter, the higher letter will be given. Otherwise, the lower letter will be given.

Grade Checking: Canvas has a grade book that stores the raw data used to calculate your course grade. It is your responsibility to periodically check that your grades are recorded properly. If you find an error, email me as soon as possible, and I will correct it. **The grade summary on Canvas is not always accurate and cannot be made accurate** since Canvas does not allow the implementation of several course policies. In addition, Canvas sometimes excuses grades that are not actually excused. Treat Canvas grades with great suspicion and recalculate them by hand using the rules in this syllabus if you need an accurate calculation.

Delayed and Excused Assignments: This class has many assignments due. As a result, students often need to ask for a deadline to be extended a few times during the semester. This request is made using a form: https://tinyurl.com/CS1324DEFall2025. This link is also available on the class home page and under Pages. The use of this form is required. The form emails me, so there is no reason for you to email me when you submit the form.

Here are some guidelines:

- Most requests that meet the guidelines below will be approved. If the request is not approved, I will email you an explanation when I review it.
- Most students in this class have asked for less than three delayed or excused assignments during the semester. In fact, most students ask for none. Students who request more assignments than this are more likely to be declined unless there is a compelling reason.
- Requests for the same type of assignment to be delayed or excused (e.g. multiple homework assignments or projects) are more likely to be declined. Each assignment type in this class serves a particular purpose. Missing one type of assignment will make it more difficult to be successful.
- It is better to ask for an extended deadline rather than an excused assignment. Doing work is how people learn in this class, and excused assignments can interfere with the learning process.
- When the reason for an excused assignment is poor time management or assignments in other classes, assignments can generally be delayed only a few days (5 at the absolute most, but generally 3). This is because getting behind on assignments tends to make it harder to understand what we are doing in class and leads to difficulties learning new material.
- Once a solution to an assignment has been posted I cannot allow students to submit delayed assignments. I will, however, excuse them. In this case, I strongly recommend that students complete the assignment, even though it cannot be submitted for a grade.

Religious Observances: It is University policy to excuse absences that result from religious observances and to reschedule exams and assignment deadlines that fall on religious holidays. Please check the schedule and inform me of conflicts as soon as possible, especially for midterms and the final examination.

Accommodation of Disabilities: The University of Oklahoma and I are both fully committed to providing reasonable accommodation(s) for all students with disabilities. If you require accommodations, you must register with the Accessibility and Disability Resource Center: https://www.ou.edu/adrc. If you have accommodation(s) and some element of the class isn't working for you, I invite you to stop by the office and talk to me about how we can meet your needs.

Title IX Resources and Reporting Requirement: Anyone who has been impacted by gender-based violence, including dating violence, domestic violence, stalking, harassment, and sexual assault, deserves access to resources so that they are supported personally and academically. The University of Oklahoma is committed to offering resources to those impacted, including: speaking with someone confidentially about your options, medical attention, counseling, reporting, academic support, and safety plans. If you would like to speak with someone confidentially, please contact OU Advocates (available 24/7 at 405-615-0013) or another confidential resource (see "Can I make an anonymous report?"). You may also choose to report gender-based violence and discrimination through other means, including by contacting the Institutional Equity Office (ieo@ou.edu, 405-325-3546) or police (911). Because the University of Oklahoma is committed to the safety of you and other students, I, as well as other faculty,

Graduate Assistants, and Teaching Assistants, are mandatory reporters. This means that we are obligated to report gender-based violence that has been disclosed to us to the Institutional Equity Office. This includes disclosures that occur in: class discussion, writing assignments, discussion boards, emails and during Student/Office Hours. For more information, please visit the Institutional Equity Office.

Adjustments for Pregnancy/Childbirth Related Issues: Should you need modifications or adjustments to your course requirements because of documented pregnancy-related or childbirth-related issues, please contact your professor or the Accessibility and Disability Resource Center at 405/325-3852 as soon as possible. Also, see the Institutional Equity Office FAQ on Pregnant and Parenting Students' Rights for answers to commonly asked questions.

Children in Class: I ask that all students work with me to create a welcoming environment that is respectful of all forms of diversity, including diversity in parenting status.

- Breastfeeding babies are welcome in class or laboratory whenever necessary.
- For older children and babies, I understand that unforeseen disruptions in childcare often put parents in the position of having to miss class to stay home with a child. While this is not meant to be a long-term childcare solution, occasionally bringing a child to class or lab to cover gaps in care is acceptable. Sick children should not be brought to class or lab. I will excuse grades for in-class work of any type for any parent who needs to stay home with a sick child.
- In all cases where babies and children come to class, I ask that you sit close to the door so that if your child needs special attention and is disrupting learning for other students, you may step outside until their need has been met.

Electronic Devices in Class: You may use laptops, tablets, cell phones, and other electronic devices in class in ways that enhance your learning. These devices may not be used in ways that distract other students (e.g., playing games, texting friends, watching videos, or making noise). Your cell phone should generally be off during class time unless you are using it for class-related work or have a good reason to have it on (e.g. family responsibilities). My cell phone is on every day because I have family responsibilities.

Academic Integrity: The Student's Guide to Academic Integrity defines academic misconduct as "any act that improperly affects the evaluation of a student's academic performance or achievement," including cheating on exams, improper collaboration on assignments, and plagiarism (https://www.ou.edu/integrity/students).

The most common violation in this course is plagiarism, usually on homework and projects. Plagiarism is "an act or instance of using or closely imitating the language and thoughts of another author and the representation of that author's work as one's own" (https://www.dictionary.com/browse/plagiarism). When completing assignments in this class, please keep the following in mind:

- Solutions should not be copied from internet sources, including websites, artificial intelligence software (like ChatGPT), and paid professional programmers.

- Do not show, give, or email another student a copy of your work before the submission deadline. Do not trust other students to not plagiarize your work. Every semester I have several students who submit work with another student's name still on it.
- Do not use artificial intelligence systems like ChatGPT to complete your work. Copying from an automated source is plagiarism, just like copying from a human and for exactly the same reasons.
- The penalties for permitting your work to be copied are usually the same as the penalties for copying someone else's work, since it is not possible to distinguish the person who copied from the person who allowed their work to be copied. If I can determine who created the work, the person copying the work will receive a harsher penalty.
- Submitting iClicker responses when not in the classroom is academic misconduct.

Projects are done in the laboratory with one assigned partner. Sharing your code with this person is not only permitted but required. Sharing code with anyone else is a violation of academic integrity.

Proper and Improper Collaboration: When you pass this class with a grade of C or better, I am certifying that you are a competent Java programmer. I cannot make this certification without seeing work that you complete on your own. Interactive programming tutors, homework, and examinations should be the work of a single individual, not their friends, not an artificial intelligence system, and not their tutor. It is permissible and even advisable to talk to other students in the class for help completing or improving your work, however, this help must not interfere with my ability to evaluate the quality and quantity of your understanding of computer programming. To understand this distinction, please review the examples in the table below. This is not a comprehensive list of all the ways in which academic integrity can or cannot be violated.

Situation	**Integrity Violation?**
Students A and B meet and work on their homework together. Neither student prepared anything in advance.	**Yes**
Students A and B create drafts of their homework assignment independently and meet to compare answers and discuss their understanding of the material. Each student decides independently whether to make changes that are discussed.	No, and this is a great idea
Students A and B agree to prepare drafts of their homework assignment independently, but only Student A does. Student A shares his draft with Student B who reviews it and offers suggestions for improvement.	**Yes**
Students A and B agree that student A will work the even problems and student B will work the odd problems. They share their work.	**Yes**
Student A has completed a project and is helping student B complete the same project. Student A explains to student B what student B's code does, which is different than what student B thinks the code does. Student B determines how to modify the code independently.	No
Student A has completed a project and is helping student B complete the same project. Student B is having trouble getting one part of the program to work, so student A texts student B three lines of their solution.	**Yes**

Student A has completed a project and is helping student B complete the same project. Student B is having difficulty getting the program to work, so student A tells student B exactly what to type for several lines.	**Yes**
Student A has completed a project and is helping student B complete the same project. Student B is having difficulty getting the program to work, so student A suggests that student B use a specific debugging strategy (e.g., "Trace these lines of the code").	No
Student A has completed a project and is helping student B complete the same project. Student A shows student B an example program in the online textbook that will be helpful in figuring out the solution to the problem.	No
Students A and B work on a project together but are not laboratory partners. After they have finished it, student A takes the code and modifies it so the programs do not appear to be identical.*	**Yes**

*I can easily detect these changes using automated software. Don't bother doing this.

Chegg and Other Online Tutoring Sources: There are a wide variety of tutoring resources available through paid websites. Many of these sites have students upload assignments and solutions and surreptitiously provide these documents to other students. What appears to be a session with a tutor may be, behind the scenes, the tutor doing a search of their company database of solutions to share. By using these sites you risk being charged with academic misconduct, either by supplying other students with answers they did not author or by receiving someone else's answer that you did not author. Since these companies are not open with students about their practices, you cannot know whether a tutor is providing meaningful support (for example, identifying misunderstandings of content and explaining them like our teaching assistants and I would) or simply feeding you someone else's solution. The tutor's actions can result in different students submitting answers that are identical, which may be flagged as academic misconduct during grading, especially when the solution is incorrect (which it often is since their tutors may lack expertise). See the table below for specific examples. **There is no way to use these sites without risking being charged, and even committing, academic misconduct at this time.**

Scenario	**May Be Charged With Academic Misconduct?**	**Guilty of Academic Misconduct**
You use the website to receive help. During the process, the assignment and all or part of your solution are stored on a company computer. The assignment or solution are subsequently delivered by the company to another student that you do not know.	Yes, because another student may submit an assignment that is identical to yours.	No, but you're going to have to prove it since two students with identical solutions is usually considered strong evidence of academic misconduct, especially if the solution is not correct.
You use the website to receive help. The assignment has already been uploaded, and your tutor provides you	Yes, because you committed academic misconduct.	Yes, and you may not even be aware that the tutor was providing someone else's solution instead of teaching.

with the uploaded solution. You submit all or part of this solution as your work.		This is academic misconduct because you are copying from a tutor.

Academic Integrity Process: Upon the first documented occurrence of academic misconduct, I will report the violation to the Office of Academic Integrity Programs. If you are found guilty by this process, the penalty may be a failing grade in the class and being suspended from OU for a semester. If you have committed academic misconduct previously, the sanctions can be more severe, including expulsion from OU. The grade sanction that I usually request for plagiarism on a single assignment is a zero on the assignment. The grade sanction that I usually request for a student accused of cheating on an examination is an F in the course. The procedure to be followed is documented in the University of Oklahoma Academic Misconduct Code. If I elect to admonish you, the appeals process is described here: https://www.ou.edu/integrity/students. The highest penalty for an admonition is a zero on the assignment.

Tutors and Academic Integrity: Before you hire a private tutor, please take advantage of the many people who are paid by OU to support this class. These people are trained to tutor properly. Private tutors can be a source of support if you are struggling in the class, but only if the tutor is aware of the distinction between teaching you the material so that you can do your own work and completing assignments for you. Tutors who simply complete your assignments are not only failing to help you learn, they are also abetting academic misconduct. Each of the situations listed above in the table of collaboration scenarios applies when student A is a tutor.

Use of Generative AI in Class (e.g. Chat GPT): This course focuses on how people learn to produce, understand, and fix computer programs. While AI is becoming skilled at creating computer programs, there is no evidence that allowing students to use AI to complete programming assignments develops the necessary conceptual understanding of programming to allow students to succeed as computer scientists. At this time, synthesizing code from memory is a method that has been proven to create successful computer scientists. Therefore, using AI to perform assignments will be permitted in this class only when specified. There are ways to demonstrate that code was generated by AI and these methods are rapidly improving. Do not assume that copying from AI cannot be detected or proven.

Students who copy generated AI results into assignments are committing plagiarism, just as if they had copied from another student.

- Use of generative AI tools is only allowed when the assignment specifically permits it. If the assignment is silent on the use of generative AI tools, they are not allowed.
- If there is a question about whether a student completed work independently or with the use of these tools, Dr. Trytten will invite the student to her office to explain the assignment solution in detail. Students who choose not to come within 48 hours or who cannot explain the work they submitted will be charged with an academic integrity violation.

Mental Health Support Services: If you are experiencing any mental health issues that are impacting your academic performance, counseling is available at the University Counseling Center (UCC). The Center is located on the second floor of the Goddard Health Center, at 620

Elm Rm. 201, Norman, OK 73019. To schedule an appointment call (405) 325-2911. For more information, please visit http://www.ou.edu/ucc.

Copyright Statement for In-Person or Online Courses: Sessions of this course could be recorded and/or live-streamed. These recordings are the intellectual property of Dr. Deborah Trytten and may not be shared or reproduced without the explicit, written consent of Dr. Trytten. In addition, privacy rights of others such as students, guest lecturers, and providers of copyrighted material displayed in the recording may be of concern. Students may not share any course recordings with individuals not enrolled in the class or upload them to any other online environment.

Ownership of Course Materials: All original content used in this course is owned by Dr. Deborah Trytten. This includes but is not limited to exams, lectures, quizzes, handouts, protocols, electronic documents, and syllabi. Original or transcribed content may not be copied, recorded, retransmitted, posted online, or sold without her and/or her expressed, written consent.

Schedule: The schedule below represents the initial plan for this semester. Due dates will be modified as the semester goes along in Canvas. **The dates in Canvas are the official dates.** Seriously, just look in Canvas to find out when things are due.
Here is a key to understandng this chart.
Z stands for Zyante. TC stands for Turingscraft. CB stands for Coding Bat. HW stands for homework.
The highlight colors are used as follows. Red indicates days when class will not be held. Yellow indicates laboratory assignments. Blue indicates days related to the examinations. The dates on the left are the dates that the topics are covered in class. The dates on the right are the dates that assignments are due.

Day	Date	Topic	Lab	Z	TC	JB
Monday	25-Aug	Introduction to Programming		29-Aug		
	25-Aug	No Laboratory				
Wednesday	27-Aug	Storing Primitive Data		29-Aug		
Friday	29-Aug	Project 1	9/5/2025			
Monday	1-Sep	Labor Day				
	1-Sep	No Laboratory				
Wednesday	3-Sep	Storing Primitive Data			6-Sep	
Friday	5-Sep	Arithmetic Operations		4-Sep		
Monday	8-Sep	Arithmetic Operations				
	8-Sep	Project 2	9/12/2025			
Wednesday	10-Sep	Arithmetic Operations			13-Sep	
Friday	12-Sep	User Interaction		11-Sep		
Monday	15-Sep	User Interactions			18-Sep	

	15-Sep	Project 3	9/19/2025			
Wednesday	17-Sep	Conditional Statements		16-Sep		
Friday	19-Sep	Conditional Statements			22-Sep	
Monday	22-Sep	Logical Operators		21-Sep		
	22-Sep	Project 4	9/26/2025			
Wednesday	24-Sep	Logical Operators			27-Sep	
Friday	26-Sep	While Statements		25-Sep		
Monday	29-Sep	While Statements			2-Oct	
	29-Sep	Project 5	10/3/2025			
Wednesday	1-Oct	Methods--part 1		30-Sep		
Friday	3-Oct	Review for Midterm				
Monday	6-Oct	Methods--part 1				
	6-Oct	Midterm 1--No Laboratory				
Wednesday	8-Oct	Methods--Part 2		7-Oct		
Friday	10-Oct	OU Texas Holiday				
Monday	13-Oct	Methods--Part 2			16-Oct	17-Oct
	13-Oct	Project 6	10/17/2025			
Wednesday	15-Oct	For Loops		14-Oct	18-Oct	
Friday	17-Oct	Arrays of Primitive Data		16-Oct		
Monday	20-Oct	Arrays of Primitive Data			22-Oct	
	20-Oct	Project 7	10/24/2025			
Wednesday	22-Oct	Methods and References		21-Oct		
Friday	24-Oct	Methods and References			27-Oct	
Monday	27-Oct	Perfect and oversize arrays		26-Oct		
	27-Oct	Project 8	10/31/2025			
Wednesday	29-Oct	Perfect and oversize arrays				1-Nov
Friday	31-Oct	Nested Loops		30-Oct		
Monday	3-Nov	Nested Loops			6-Nov	7-Nov
	3-Nov	Project 9	11/7/2025			

Wednesday	5-Nov	Sorting Algorithms		4-Nov		
Friday	7-Nov	Review for Midterm 2				
Monday	10-Nov	Sorting Algorithms				
	10-Nov	Midterm 2--No Laboratory				
Wednesday	12-Nov	Using objects from the API		11-Nov		
Friday	14-Nov	Using objects from the API			17-Nov	18-Nov
Monday	17-Nov	Classes with Generics		16-Nov		
	17-Nov	Project 10	11/21/2025			
Wednesday	19-Nov	Classes with Generics			22-Nov	
Friday	21-Nov	Building new Classes		20-Nov		
Monday	24-Nov	Building new Classes				
	24-Nov	Project 11	11/30/2025			
Wednesday	26-Nov	Thanksgiving				
Friday	28-Nov	Thanksgiving				
Monday	1-Dec	Building new Classes			4-Dec	
	1-Dec	Project 12	12/5/2025			
Wednesday	3-Dec	Access Modification		2-Dec		
Friday	5-Dec	Access Modification			8-Dec	
Monday	8-Dec	Aggregation		7-Dec		
	8-Dec	Project 13	12/10/2025			
Wednesday	10-Dec	Aggregation				
Friday	12-Dec	Review for Final				
Wednesday	17-Dec	Final Exam 7:30-9:30 PM				