

# Syllabus

## CSED 5133, Fundamentals of Programming

### Fall 2024

#### Course Information

**Instructor:** Dr. Ben Schafer

**Email:** ben.schafer@uni.edu

#### Student Hours:

I want to be available to you. To get started, I am available during the school day with appointments during the following times:

- MWF, 10-11 AM
- MWF, 12-1 PM

I am also available after school and in the evenings. The days and times may change as I find out what works best for you and juggle this with my family obligations.

To schedule an appointment use

- <https://bit.ly/SchaferCalendar>

Then at the appropriate time, you can log on to the following Zoom room:

- <https://uni.zoom.us/j/3192732187>

If those times don't work for you, please send me an email, and propose a specific alternative. I want to meet with you at a time/place that works for both of us, and I am very willing to work things out.

**Credit Hours:** Three (3). This course meets the Credit Hour Expectations outlined in the UNI Course Catalog. Students should expect to work approximately ~3 hours per week for every course credit hour. In other words, you should expect this course to require 9-10 hours of work in a typical week.

#### Class Websites:

- <https://www.cs.uni.edu/~schafer/cohort24/FOP/> (most lesson materials)
- Blackboard (Grades and Competency Demos)

#### Textbook

No single textbook adequately fits our needs. Instead, all required readings and other materials will be selected from legally available resources on the internet or from instructor produced materials.

#### Computer Use

This is an online course. All the learning materials and programming environments for this course are available from any computer with a web browser and internet access.

#### Course Description

Introduction to computer programming through a survey of programming environments used by teachers. Topics include structure of programming, study of several programming environments used by students at a variety of age/ability levels, and end-user programming for teachers.

## Course Learning Outcomes

By the end of this semester students taking this course should be able to meet the following:

**Programming Oriented Outcomes.** Students should be able to:

- trace a segment of code to determine the result produced or state achieved by given code
- modify a provided piece of code to accomplish a given task
- choose and sequence action statements to accomplish a given task
- develop and use selection statements (if-then, if-then-else, etc.) to control selection between actions
- develop and use iteration statements (for, while) to control repetition of actions
- explain the concepts of sequence, loops, parallelism, events, conditionals, operators, variables, and lists within the context of computer science.

**Teaching Oriented Outcomes.** Students should be able to:

- define the key concepts of computer science. These concepts include:
  - sequences
  - operations
  - variables
  - loops
  - conditionals
  - lists
  - parallelism
  - events
- discuss basic elements of instruction regarding the key concepts of computer science in the context of a K-12 classroom.

## How Student Performance Will Be Evaluated

### Course Activities

You will complete three different types of activities this semester.

- Ungraded "Practice Problems"
  - We will do a lot of activities, both in and out of class, that are a way to enhance your learning. In most cases, it is the **process** that I am after rather than the results.
    - To help you gauge your understanding of the material, you will frequently submit these to a system for automatic feedback.
  - Officially, these are not part of your final grade - you are not required to submit any of these for formal evaluation. However, I believe that both of us will be able to better understand your learning if you complete and submit these activities.
- Unit Problem Sets [Three Unit Problem Sets, each worth one final grade]
  - The course is divided into 3 units.
    - Each of these units will contain a series of "Problem Sets" which are graded programming assignments (like homework in other courses)
    - Each unit is graded independently based on activity specifications and an evaluation table published separately for each unit.
  - All in-unit activities [programming assignments] may be submitted multiple times up to the unit deadline.

- Competency Demos (CD) [Five CDs for a total of 6 grades]
  - You can think of these like exams.
  - Each will be administered in-class via Blackboard and/or Autolab (using Python)
  - CDs 1-4 are worth one grade each.
    - If you do not like the grade you earned on the original CD, you will be provided an opportunity to ask questions, re-study the material, and attempt a second CD to improve your grade. Specific details on this process will be provided when it becomes an option.
  - The Final CD will be broken into two separate grades.
    - Due to the nature of the Final CD, there is NOT an opportunity to retake and improve your score.

### Course Grading

I use a grading system drawn from the philosophies of "standards-based grading" and "equitable grading" (<https://gradingforequity.org/>). The main idea is that I WANT you to succeed in the course by giving you multiple opportunities for you to show me that you have learned the necessary material. As was noted previously, in many situations, if you can't do this the first time, you will be able to re-study and try again.

You will earn multiple "grades" in this course. Each of these is a category of understanding that, for simplicity, is recorded as a score from 1-4 with the following meaning:

Score	Meaning
1	UNASSESSABLE - You submitted the deliverables for the activity but what you submitted shows little understanding of the standards of the activity.
2	NEEDS WORK - You have made significant progress towards demonstrating competency but there are limited items that remain unsatisfied.
3	SATISFACTORY - You have "met" the standards of the activity.
4	EXCELLENT - You have "exceeded" the standards of the activity. [You have met the standards of competency and shown considerable understanding/knowledge of the material.]

While it might be tempting to translate these categories to the scale we use when we calculate GPASS (which is also a 4-point scale) that is not the way they are used or interpreted in this course. See the discussion about final grade assignment for more on this idea.

Notice that there are a total of 9 grades earned during this course (3 formally graded "problem sets" and 6 grades associated with Competency Demos). Thus, it is possible to earn up to 36 points in the course.

Final course grades will be determined using the following evaluation criteria.

Grade Earned	Score	Additional Conditions
A	32 – 36	All scores 3-4
A-	32 – 36	All scores 2-4
B	27 – 31	All scores 3-4
B-	27 – 31	All scores 1-4
C	23 – 36	All scores 2-4
C- *	23 – 36	All scores 1-4
D *	19 – 22	All scores 2-4
D- *	19 – 22	All scores 1-4
F *	0 - 18	

#### NOTES:

- Because I allow – and encourage – retakes, most students do just fine in this course. It is VERY rare that a student who has been an active participant in the class is unable to earn at least a C for a final grade.
- If I feel there are specific and individual circumstances where "mathematically" you earned a grade slightly lower than I feel your overall competence has demonstrated than I reserve the right to raise your grade one level from that published above.
- I reserve the right to "adjust" your course grade if your performance on the Final CD deviates significantly from your overall performance in the course.
  - If your Final CD grades are lower than your overall course grade, I reserve the right to lower your grade one level from that published above.
  - If your Final CD grades are higher than your overall course grade, I reserve the right to raise your grade one level from that published above.

Please note, to be responsive to your needs I reserve the right to modify the structure of this course as we are in progress. If there is significant deviation from the policies described above, this new policy will be clearly discussed with you and in a timeframe that gives you a time to plan accordingly.

#### Final Thoughts

If you are having trouble with a topic in the class, please reach out to me **early**. Do not wait until the situation is out of control. I am very willing to help. However, I have to know you need and want that help.

## Additional Policies and Statements

### Scholastic Conduct

You are responsible for being familiar with the University's Academic Ethics Policies:

<https://www.uni.edu/policies/301>

Copying from other students is expressly forbidden. Doing so on CDs will be penalized every time it is discovered. The penalty can vary from zero credit for the copied items (first offense) up to a failing grade for the course. If an assignment makes you realize you do not understand the material, ask a fellow student a question designed to improve your understanding, *not* one designed to get the assignment done. Your final submission for assignments should be **individual, original** work unless otherwise specified. Any substantive contribution to your solution by another person or taken from a publication should be properly acknowledged in writing. Failure to do so is plagiarism and will necessitate disciplinary action. In addition to the activities we can all agree are cheating (plagiarism, bringing notes to a closed book exam, etc.), assisting or collaborating on cheating is cheating. Cheating can result in failing the course and/or more severe disciplinary actions. Remember: Discussing assignments is fine and even encouraged. Copying code or answers is not.

### Artificial Intelligence Statement

Artificial Intelligence (AI) is rapidly changing the face of software development. AI tools already exists that can write everything from a single line of code in the middle of your program (co-pilots built into IDEs) all the way up to full programs with an adequately formed prompt (e.g., ChatGPT). These tools hold great promise to increase the productivity of well-trained programmers. I have no doubt in my mind that these tools will become a very accepted addition (notice my choice of that word) to a programmer's toolkit in the very near future. But these tools come with a real "cost." AI tools are still incredibly fallible. They work exceptionally well when they are right. But when they are wrong, they can fail in a truly "spectacular" manner. And I don't see this situation changing. EVER.

Programmers are tasked with the incredibly important job of making sure that the code they use, or produce is correct, free from errors, and does not introduce security issues. To properly use these AI tools, we first need to understand how code works, inside and out. Only once we understand how to code can we read, test, and properly evaluate code generated by AI tools. Therefore, the purpose of this class is to teach you how to fully understand the foundations of coding by creating introductory-level programs to solve problems **without** the use of AI.

There is a common quote attributed to John Dewey who said, "All genuine learning comes through experience." Unfortunately, the best experience – in the long run – comes from failure and figuring out how to overcome that failure. It will probably be very tempting to ask an AI to write your homework solutions for you. However, doing this robs you of learning experiences and – I hate to say this – the failures that often come with that. But remember, those failures help you learn.

More than 50% of the points earned for the final grade in this class come during CDs which are given in situations where you will be strictly forbidden from using AI. Thus, if you haven't struggled and

overcome problems during the homework phase of the course, it is highly likely that you will be unable to generate and read code during the CD phases of the course. You really must know your stuff.

Because of this...

If I notice code submitted for formal evaluation for a course grade (programming sets or CDs) that doesn't look like the type of thing we have covered in class – either in structure or in use of concepts that we haven't covered – I will ask you to meet with me to conduct a code walkthrough before assigning the grade. This will consist of a short (10-15 minute) meeting with me within a week of the situation. During that time, I will ask you to explain your code and where the use of any advanced structure/concept comes from. It may be that you do, in fact, know your stuff. It may be that your code looks the way it does because of some prior programming knowledge or experience. That's great! During our code walkthrough you should be able to explain what is happening and I will gladly give you an appropriate grade for that code. However, if you fail to demonstrate you understand the code you turned in or otherwise do not meet with me within the week requested, I'll replace the marker in the gradebook with a grade of 0 for the activity.

In summary, the purpose of AI tools is not to replace programmers. Instead, they are meant to speed up the productivity of programmers once they can read the code produced to verify it is correct, free from errors, and does not introduce security issues. This class is to help you with that first step of knowing how code works. At this level, this means knowing how to create code entirely by yourself.

## **TLDR**

- Do not use AI tools to generate programs for this class.
  - AI tools will have their place in programming. An introductory programming course is not that place.
- Using AI tools will put you at risk of a code review meeting with me. If I see solutions using coding that was not introduced in class, I will send you an email to meet with me within a week.
  - If you have advanced programming knowledge and can explain the code you submitted, that's great! I'll learn a bit more about your previous experience and put you on a list not to bug again.
  - If you cannot explain your code or do not meet with me within a week, the code will turn into a grade of 0.

## **Accessibility**

The University of Northern Iowa (UNI) complies with the Americans with Disabilities Act Amendments Act of 2008 (ADAAA), Section 504 of the Rehabilitation Act of 1973, the Fair Housing Act, and other applicable federal and state laws and regulations that prohibit discrimination on the basis of disability. Students with disabilities experiencing a barrier to access should connect with Student Accessibility Services (SAS) to request accommodations. For more information about the accommodation process, please contact SAS at (319) 273-2677 Relay 711, [accessibilityservices@uni.edu](mailto:accessibilityservices@uni.edu), or GIL 118. Additional information is also available at [sas.uni.edu](http://sas.uni.edu).

## **The Learning Center**

The Learning Center @ Rod Library provides free tutoring for a variety of different areas (i.e. writing, math, science, business, Spanish, college reading and learning strategies). The Learning Center @ Rod Library is open for walk-in assistance Monday-Thursday 10am-10pm and is free of charge for all UNI

students. If you are unavailable during normal tutoring hours, online tutoring is also available through Smarthinking. You will need your CATID and passphrase to gain access. To access the Smarthinking platform go to <https://tlc.uni.edu/online>. For more information, go to <https://tlc.uni.edu>, email [TheLearningCenter@uni.edu](mailto:TheLearningCenter@uni.edu) , call 319-273-6023, or visit the TLC desk located on the main floor of Rod Library.

### **Free Speech**

The University of Northern Iowa supports and upholds the First Amendment protection of freedom of speech and the principles of academic and artistic freedom. We encourage the free and responsible exchange of diverse ideas on our campus. The University is committed to open inquiry and the spirited and thoughtful debate of such ideas.

## **Office of Compliance and Equity Management**

### **Non-discrimination in Employment or Education**

Content in this class has the potential to be disturbing to some individuals based on life experiences. If you ever feel the need to step out of the classroom or decline participation in an activity, please request an alternative learning experience.

UNI Policy 13.02 Discrimination, Harassment, and Sexual Misconduct states: "The University is committed to providing a workplace and educational environment, as well as other benefits, programs, and activities, that are free from discrimination and harassment based on a protected class, as well as retaliation."

Policy 13.02 outlines prohibited conduct and reporting processes. All University employees who are aware of or witness discrimination, harassment, sexual misconduct, or retaliation are required to promptly report to the Title IX Officer or Title IX Deputy Coordinator.

- Title IX Officer Leah Gutknecht, Assistant to the President for Compliance and Equity Management, 117 Gilchrist, 319.273.2846, leah.gutknecht@uni.edu
- Title IX deputy coordinator: Christina Roybal, Sr. Associate Athletic Director Athletics Administration, North DOME 319.273.2556, christina.roybal@uni.edu

If you or someone you know has been harassed or assaulted, you can find the appropriate resources at [safety.uni.edu](http://safety.uni.edu) and [equity.uni.edu](http://equity.uni.edu). Resources that provide free, confidential counseling are also detailed at [safety.uni.edu](http://safety.uni.edu).

For additional information, contact the Office of Compliance and Equity Management, 117 Gilchrist Hall, 273-2846, [equity@uni.edu](mailto:equity@uni.edu).