

Course Manual

CAS CS 320: *Concepts of Programming Languages*

Boston University

Fall 2026

Contents

1	Week 0 To-Do list	2
2	General Information	3
2.1	Prerequisites	3
2.2	Learning Objectives	3
2.3	Course Structure	4
2.4	Resources	5
3	Evaluation	6
3.1	Assignments	6
3.2	Final Project	7
3.3	Labs	7
3.4	Workshops	7
3.5	Quizzes	7
3.6	Midterm Exams	7
3.7	Final Exam	8
4	Policies	9
4.1	Diversity Statement	9
4.2	Disability Statement	9
4.3	Sexual Misconduct	9
4.4	Academic Integrity	9
4.5	Generative AI	10
4.6	Additional Attendance Policies	11
4.7	Additional Grading Policies	11
5	Closing Remarks	13
5.1	Course Agreement	13
5.2	University Resources	13

Course Code	CAS CS 320
Course Title	Concepts of Programming Languages
Semester	Spring 2026
Instructor	Ankush Das & Nathan Mull
Teaching Fellow	Jared Pincus
Teaching Assistant	Vivian Tvedt
Meeting Times	Tuesday and Thursday, 2:00PM-3:15PM (A2) Tuesday and Thursday, 3:30PM-4:45PM (A1)
Meeting Location	KCB 106 (A2) CDS B62 (A1)
Midterm Exams	October 15 (during lecture) November 12 (during lecture)
Final Exam	December 17 (3:00PM-5:00PM)
Grade Breakdown	15% Assignments 05% Final Project 10% Labs 10% Workshops 20% Quizzes 25% Midterm Exams 15% Final Exam

Figure 1: Course overview

Abstract

This is a manual for the course CAS CS 320: *Concepts of Programming Languages*. It contains a general overview of the course and its policies. It does *not* contain specifics about the material being covered in the course; this appears on the course webpage. Figure 1 contains an overview of the course.

Week 0 To-Do list

You should complete the following items within the first 48 hours of the start of the semester. Please reach out if you have concerns about any of the items listed.

- ☐ Verify that you have access to a laptop computer during the semester
- ☐ Verify that you know where lectures are held
- ☐ Verify that you know where the discussion section in which you're registered is held
- ☐ Join Piazza (Sign-up link)
- ☐ Join Gradescope (Sign-up link, Entry code: E632PP)
- ☐ Familiarize yourself with the course webpage
- ☐ (*Optional*) Add the Piazza, Gradescope, and course webpage as bookmarks in your Internet browser
- ☐ Review the course calendar and determine which office hours you're able to attend
- ☐ (*Optional*) Add the course calendar to your own calendar
- ☐ Follow the installation instructions on the course webpage to set up your personal machine
- ☐ Read this manual in its entirety
- ☐ Submit the assignment on Gradescope confirming that you've read this manual
- ☐ If necessary, review material from CAS CS 210: *Computer Systems* on how to work in the terminal

General Information

CAS CS 320: *Concepts of Programming Languages* is a course *about* programming languages, particularly the **design** and **implementation** of programming languages. In this course, we take up the programming language as an object of formal study. This course is *not* about how to program, though the principles we cover are generally useful for writing and reasoning about programs.

The first part of the course is on functional programming in OCaml, based on CS3110: *Data Structures and Functional Programming* at Cornell University. The topics include functional design, algebraic data types, higher-order functions, polymorphism, functional data structures, and modular programming. It's during this part that we learn to *think* functionally, to view programs not as sequences of commands manipulating global state, but as compositions of functions that deconstruct and reorient data. We also introduce in this part of the course fundamental concepts from the theory of programming languages—like inference systems, typing rules, semantic rules, and derivations—with an eye towards how these concepts apply to OCaml in particular.

In the second part of the course we implement several interpreters; that is, we write programs that *execute* other programs. The topics covered include formal grammar, parsing, operational semantics, variable scope and binding, closures, type checking, type safety, and type inference. By the end of the course, you'll be able to reason formally about what a programming language is, and what makes a programming languages good.

2.1 Prerequisites

The formal prerequisites for this course are:

- CAS CS 111 & 112: *Introduction to Computer Science*
- CAS CS 131: *Combinatoric Structures*
- CAS CS 210: *Computer Systems*

You'll get the most out of this course (and you'll have the best time of it) if you've completed the 200 level major requirements for the computer science degree at BU. Experience with a high-level programming language (like Python or Java) is a must. The dependence on CS131 is a level of mathematical maturity and exposure to the principle of induction. We don't depend explicitly on the material of CS210, but it will be useful in appreciating the second half of the course. We also assume general comfort in the terminal. If you don't have experience with this, then you'll be have to self-learn it within the first week of the course.

2.2 Learning Objectives

From this course we hope that you will:

- Learn the rudiments of OCaml, a functional programming language which is likely very different from other languages you’ve learned. In particular, we’d like you to learn to “think functionally” and understand the benefits of doing so.
- Learn to read and write formal specifications for programming languages.
- Gain an understanding of the theoretical basis for what makes a “good” programming language.
- Gain an appreciation for what goes into the development of the programming languages we use everyday, particularly by implementing interpreters.

CS320 fulfills a single unit in the BU Hub area **Creativity/Innovation**. This deserves some explanation. Programming language design is a (perhaps surprisingly) creative endeavor, in which personal aesthetics play a large role in what features are included (or not) in a language. A fair amount of creativity also goes into discovering new features that aid the safety, productivity, or ergonomics of a new language.¹ This is in part to say that what we cover is based on our notion of what a “good” programming language is, which won’t necessarily align with your own views or the views of other programming language designers. This also must be understood in spite of the fact that much of what we do in the course will not feel very creative. All I can say on this point: you have to know what came first before you can change the state-of-the-art.

2.3 Course Structure

Lectures

We hold lectures each week on Tuesday and Thursday (see the registrar and the course webpage for details). During lecture, we cover the material that’s presented in the reading, do live coding examples, and provide practice problems. The material used in lecture is made available on the course webpage before the lecture meeting. Barring technical difficulties, recordings of the lecture will be made available.

The first half of the semester will be taught by Ankush Das and the second half by Nathan Mull. All course staff members will be available throughout the semester for office hours and workshops.

We won’t take attendance during lecture (except during Workshops—see below for details) but it’s highly recommended that you attend, and refer you to the BU Attendance policy. You’ll be expected to participate in lectures by working on practice problems and occasionally discussing topics with the people sitting around you.

Workshops

Five lectures during the semester will be run as workshops. This means you’ll be expected to do some at-home learning beforehand; we’ll spend a majority of the lecture time working on an in-class activity in groups. We’ll take attendance insofar as you’ll have to submit the required material by the end of lecture.

Discussion Sections

We hold discussion sections each week on Wednesday (see the registrar and the course webpage for details). The discussion sections have two purposes. On some weeks they will be centered around a group activity. On other weeks there will be proctored quizzes (see below for details). As with lectures, we won’t take attendance during discussion sections, but it is highly recommended that you attend.

¹Rust is a great example of a modern language with a fair amount of creativity behind its design.

2.4 Resources

Material

The first half of the course uses the textbook OCaml Programming: Correct + Efficient + Beautiful. All other course material will be made available in the course GitHub repository and on the course webpage. If you are unfamiliar with git and GitHub, see the GitHub documentation for information and tutorials. Please check the course webpage and repository frequently. We'll also be testing out some new notes for this course. These will be made available on the course webpage.

Programming

The programming in this course is done in OCaml. You're required to set this up on your personal machine or on a machine that you'll have access to throughout the semester. You'll have an opportunity to go through this set-up during the first lab of this course. Please attend office hours and use Piazza if you need help troubleshooting. If you're worried about access to technology, please contact me as soon as possible and we can see what we can do (though I cannot make any guarantees).

Course Communication

Course announcements and discussions will happen on Piazza. If you're unfamiliar with Piazza, see their support page for information and tutorials. Some policies regarding the use of Piazza:

- *Don't ask homework questions directly.* Formulate a question which will aid in your understanding, and will hopefully help others as well.
- *Don't give homework solutions directly.*
- *Piazza is as useful as it is active.* The course staff will be answering questions on Piazza, but don't hesitate to answer questions yourself.

Make sure to set notifications correctly so you can keep up with updates regarding the course. "I didn't see the Piazza post about it" is never a valid excuse for missing a piece of information.

Submission

We'll be using Gradescope for assignment and lab submission. If you're unfamiliar with Gradescope, see their Get Started page for information and tutorials.

Evaluation

The grading breakdown for this course is given in Figure 3.1. The sites of evaluation are detailed in below. Your raw percentage will be determined according to this breakdown and your final letter grade is guaranteed to be at least what is determined by Wheelock College’s Grading Scale.¹ But, to borrow a phrase from Professor Mark Bun: “to correct for the possibility of [quizzes] and exams being more difficult than anticipated, letter grades may be (significantly) increased above these guarantees.” Specifically, we may retroactively curve exam and quiz grades using a linear scale.²

3.1 Assignments

Assignments are released weekly on Thursdays and are due a week later on the following Thursday by 8:00PM. See the calendar on the course webpage for details. Assignments consist of written problems that are to be submitted as a pdf file via Gradescope, as well as programming problems that will be autograded. There are 9 assignments total. We drop your lowest assignment grade, so only 8 assignments count towards your final grade in the course. We don’t accept late assignments under any circumstances.

You’ll notice that, despite the fact that there are many assignments, they account for a very small portion of your final grade. This is, in part, necessitated by the recent advances in AI, which can easily solve problems we ask on homework assignments, but it’s also a conscious decision we’ve made to ensure that students are engaging with the material beyond what can be reasonably asked in a homework assignment. You should think of the assignments in this course as *accountability checks* in that they require to engage with the material each week. But the more effort you put into learning and internalizing the material in the assignments the more successful you’ll be in the other evaluation sites of the course like the quizzes and exams.³

¹Formally we’re part of the College of Arts and Sciences (CAS), but this grading scale is standard.

²See this article for details if you’re interested.

³We understand that this approach to evaluation is not universally liked, and that in-person evaluation can be challenging. We take this into consideration when we calibrate the difficulty of the material and the workload we expect.

15%	Assignments (9 total, 1 dropped)
5%	Final Project
10%	Labs (5 total, 1 dropped)
10%	Workshops (5 total, 1 dropped)
20%	Quizzes (5 total, 1 dropped)
25%	Midterm Exams
15%	Final Exam

Figure 3.1: Grade breakdown

Assignments are graded on participation. You're required to submit hand-written assignments (i.e., no typed solutions, barring cases of disability accommodations) and your handwriting must be exceptionally clear. For each assignment, you'll be required to submit only a handful of the given problems. The course staff will provide detailed feedback on these problems. Because assignments are graded on participation, *you'll get the most out of the submitted problems if you actually do them yourselves. You will not be penalized for your mistakes, the feedback is there for your edification, and for you to prepare for the quizzes and exams.*

3.2 Final Project

There will be one final project consisting of building a more featured interpreter. It should be thought of as being about 2 assignments worth of work. All notes above about assignments also apply to the final project. It cannot be dropped.

3.3 Labs

Each lab will have either have an in-person activity that you'll be required to submit at the end of the discussion section, or a proctored quiz (see below for details). There are 5 lab activities in total. We drop your lowest lab grade, so only 4 labs count toward your final grade in the course.

3.4 Workshops

Each workshop will have an in-class activity that you will be required to submit at the end of the lecture period. There are 5 workshops total. We drop your lowest workshop grade, so only 4 workshops count towards your final grade.

3.5 Quizzes

Quizzes occur roughly every other week on Wednesdays during discussion sections. They consist of 1-3 problems that are similar to the problems that have appeared on previously submitted assignments. If you've completed the assignments and practiced the concepts therein, you're expected to do well on the quizzes. There are 5 quizzes total. We drop your lowest quiz grade, so only 4 quizzes count towards your final grade in the course.

Quizzes are 30 minutes long unless you have disability accommodations which allow for more time.

3.6 Midterm Exams

The midterm exams will be held on October 15 and November 12 during lecture. They are a closed-note written exams. You should expect the midterm exams to be more difficult than the quizzes. The grades of the midterm exams will be weighted as follows:

$$17\% \times \max(s_1, s_2) + 8\% \times \min(s_1, s_2)$$

where s_1 is the (normalized) score received on the first midterm and s_2 is the (normalized) score received on the second. In other words, we offer a small amount of grade forgiveness by weighting your better performance more favorably.

3.7 Final Exam

The final exam will be (tentatively) held December 17, 3:00PM-5:00PM (see the registrar and the course webpage for details). It's a written exam and is meant to verify that you've internalized the basic concepts of the course, and can apply them to solve novel problems. You should expect the final exam to be the same difficulty as the midterm exams.

Policies

There are a number of policies associated with this course, some specific to the course and others which hold more generally in the university.

4.1 Diversity Statement

Our aim is to present material in a way that respects the diversity of the student body. If we fail to do this, please make us aware. Any suggestions are welcome and appreciated. We also expect students to appreciate and respect the unique opportunity they have to participate in a diverse student body like ours.

4.2 Disability Statement

If you require disability accommodations, please contact us as soon as possible. You should provide us with the appropriate documentation, available through Disability and Access Services. In order to receive accommodations, you *must* be in contact with us.

If there's a policy that we're failing to comply with, please reach out. And if you'd like accommodations that aren't covered by existing services or policies, feel free to contact us and we can see what we can do. We want everyone to feel able to fully participate in the course.

4.3 Sexual Misconduct

Please read the Sexual Misconduct Policy and review the entire page for information on talking to someone confidentially about experiences of sexual misconduct, filing a report, and any other relevant information. Above all, you should feel safe, and able to be productive. If this is not the case, please reach out to us or someone else immediately.

The members of the course staff are considered "mandated reporters" and are required to report cases of sexual misconduct. Therefore, **we cannot guarantee the confidentiality of a report**. We must provide our Title IX coordinator with relevant details such as the names of those involved in the incident. The university will consider a request for confidentiality and respect it to the extent possible.

With that in mind, if you come to any of us with questions or concerns, we will handle the situation to the best of our ability and connect you with available resources.

4.4 Academic Integrity

Please read the Academic Conduct Code and review the entire page for information about what constitutes academic dishonesty and what penalties arise as a result of violations of this code. This is taken very

seriously at BU and we take it seriously in this courses. There are a couple policies about which we'll be strict:

- You must submit your own work for all assignments and mini-projects. Submitting the same file as another student, something notably similar (e.g., identical wording or code in large parts of the solution), or something suspected to be wholly generated by AI, is considered academic misconduct and will be handled accordingly.
- Copying or information-sharing during quizzes and exams is considered academic misconduct and will be handled accordingly.

If you work with others, consult materials found on the Internet, or use an AI assistant, you should cite your sources. This is a useful skill in any setting, and so we recommend being as conservative as possible regarding citations. In an assignment, these citations should appear next to every corresponding problem (in comments if the submission is code). Some examples:

- I discussed problem 1 and 2 with Leah Smith. She helped me understand X and Y aspects of the problem.
- I saw the stack overflow post stackoverflow.com/questions/6681284/python-numpy-arrays which informed my solution.
- I helped Zihan Guo with problem 4. I told them to try using X.
- I asked ChatGPT "what's the largest eigenvalue of this matrix?"

When in doubt, err on the side of longer, more descriptive citations. We do not consider missing or poor citations is a direct act of academic misconduct, but we will consider this grounds for further investigation in suspicious cases. Above all, use your best judgment and remember:

- We care about your success in this course. We provide a number of avenues to ask for help, please use them.
- You will have to answer questions on quizzes and exams without external aids (and in interviews when you apply for a job).
- If you don't know how to start thinking about a problem, it's okay to ask for pointers in office hours and on Piazza.
- We have safeguards (like dropped homework assignments) in the case you are unable to complete an assignment. In other words, don't submit someone else's work when you can drop an assignment.

4.5 Generative AI

The appropriate role AI in learning is an open question with many possible answers. But these tools exists, and the university has made them more accessible with the introduction of TerrierGPT, to which all students of the university have access. Keep in mind that this is all still an experiment; we don't know if our policy makes sense in the long term. But it's our attempt to come to terms with the appearance of these tools in our courses.

This semester, we've re-weighted evaluation sites in order to maintain the policy that *AI is allowed for use on assignments and labs*. Of course, part of the reason we've introduced this policy is that these tools are becoming ubiquitous. It's not unlikely that you'll be *expected* to use AI assistants in your future job.

Likewise, *nearly all questions that we can ask you on assignments can be easily solved by existing models*. This is why we have quizzes: it's not enough to produce the work for assignments, you have to demonstrate it in a closed-book setting.

An obligatory concluding remark: that existing models can solve our assignments does not negate the value in knowing how to do them without the help of external tools. To draw an imperfect analogy, we don't learn a new language in order to have memorized a vast vocabulary and grammar rules, but in order to *internalize* the language, and learn how to *interact* with it and in it. This is our goal in this course and beyond. AI tools can be incredibly useful in the process of learning, but the internalizing is what we really want to achieve; we want to be able to look at a problem and *sense* the underlying structure to which we can apply our knowledge from this course.

4.6 Additional Attendance Policies

As we've noted, we won't take attendance in our course. Instead, we remind you that, according to the Attendance policy at BU, you're required to attend the courses in which you're registered.

Absence Due to Religious Observance

According to the BU policy on Absence Due to Religious Observance: you "shall be excused from any such examination or study or work requirement, and shall be provided with an opportunity to make up such examination, study, or work requirement that may have been missed because of such absence on any particular day; provided, however, that such makeup examination or work shall not create an unreasonable burden upon such school."

Bereavement

According to the BU policy on Student Bereavement: you "should be granted up to five weekdays of bereavement leave for the death of an immediate family member." Your advisor should help you coordinate your leave.

4.7 Additional Grading Policies

Regrade Requests

Regrade requests may be submitted on Gradescope for up to one week after receiving the grade for an evaluation site. Regrade requests will only be considered in the case that the grader has made a mistake in grading. Any regrade requests which solely appeal for a higher grade will not be considered.

Grading Grievances

According to the BU policy on Grade Grievances: you may "contest a final course grade received in a unit-bearing Boston University course when that grade is alleged by the student to be arbitrary." Read the policy for more information. We recommend contacting us before submitting a formal appeal.

Incomplete Grades

According to the BU policy on Incomplete Coursework: "An incomplete grade (I) is used only when the student has conferred with the instructor prior to the submission of grades and offered acceptable reasons

for the incomplete work. An incomplete grade may be appropriate when the student has participated in and completed requirements representing a majority of the course, and circumstances prevent the student from completing remaining requirements by the conclusion of the course.” In particular, **you must contact us before the last day of the semester in order to receive an incomplete grade.**

Closing Remarks

Quite a bit goes into organizing (and taking) a course. In light of my comments on citations, I'll note that much of what's in this document is based on similar documents (often taken without permission) written by Mark Crovella, Mark Bun, Jonathan Appavoo, Preethi Narayanan, Ravi Chugh, Andrew McNutt, and others we may be missing. All told, we hope that most of this logistical information will be overshadowed in your memory by the concepts of the course, and that we can focus on having a good time doing math and programming.

5.1 Course Agreement

We consider this document a contract. The following is what you agree to in remaining a part of the course.

I agree to the policies outlined in this document, and I will uphold them to the best of my ability. I will also, generally speaking, try to be a reasonable person and be nice and good and respectful to the people around me taking—and running—the course. In return, I expect a high-quality learning experience and respect from those around me taking—and running—the course.

5.2 University Resources

There are quite a few BU resources, it can sometimes be overwhelming. Here's a small list of the ones we think are important. If you're struggling in this course due to personal/health conditions, we can't guarantee we can help, but if you're comfortable reaching out, feel free to send us an email and we can see if we can point you towards the correct resources. If you're not comfortable reaching out to us, that's okay too, hopefully this list can help you find what you need. Also, keep in mind you can post anonymously on Piazza if you want to ask for help without including your name.

- Disability and Access Services
- Student Health Services
- Outreach and Prevention
- Behavioral Medicine
- Survivor Support (SARP)
- Educational Resources Center
- International Students & Scholars Office