CLPS 0950: Introduction to Programming

Course Information

- canvas <link>
- when/where: **lecture** monday/wednesday/friday @ 2pm, **office hours** tbd

Instructor Information

- *Graduate Instructors*:
  - Haley Keglovits <email>
    Office Hours: TBD
  - Ivan Felipe Rodriguez <email>
    Office Hours: TBD
- *Undergraduate TAs*:
  - Abby Perelman <email> (head uTA)
  - Alaina Cherry <email>
  - Isabella Longoria-Valenzuela <email>
  - Alejandro Torres <email>
- *Faculty Supervisor*: Prof. Thomas Serre <email>
  - Associate Professor of Cognitive Linguistic & Psychological Sciences
  - Office hours and meeting link: By appointment. Please <sign up link>.

Course Information

General orientation to the course

This course provides an introduction to programming and computational thinking. We will use the Matlab programming language as a starting point because it is concise and easy to read. It is also one of the most popular scientific programming languages in psychology and the life sciences more broadly. We will then transition to Python which has now become the main programming language used in data science. Python will be needed for those of you wanting to take additional courses in computational modeling, data science, and machine learning.

Who should take this course?

The course is designed for students in psychology, cognitive science, neuroscience, and other non-computer science majors interested in learning to program and, more generally, in developing computational thinking skills.

Beyond teaching specific coding skills, this course will support students’ development as computational thinkers. Computational thinking provides a way of solving problems, designing systems, and understanding human behavior. It has become fundamental to understanding the way people think and interact with the world and has become a critical skill to flourish in today's world. Mastering these skills will enable students to more richly understand the cognitive, linguistic, and psychological sciences — and impact society.

Course Goals

By the end of this course, students will be able to:

Think computationally and to express solutions to problems so that they can be run on a computer write Matlab and Python programs to solve many common computing challenges associated with the study of the mind, brain, and behavior — from conducting sophisticated data analyses to parsing complex data files to implementing psychophysics experiments understand the deep connection between computational sciences and the brain and cognitive sciences.

Learning activities, assessments, and allocation

This is an hybrid online and in-person course organized in weekly modules. Every week, students will be responsible for:

1. completing a sequence of brief (online) tutorials. At the end of each tutorial, students will be responsible for completing a tutorial assessment involving short programming questions to offer a chance for the student to test their own understanding of the material. These tutorials will also have a Coding in Context section with a mini-assignment/survey to help you understand how coding fits into the research world. The complete tutorial will be due every Saturday at 11:59 pm.
2. completing weekly programming assignments due every Wednesday at 11:59 pm (one per student). Collaborations are encouraged but students are expected to produce their own solution and to turn in their own assignment. If applicable, the name of all the group members who collaborated needs to be acknowledged with each student submission (we are using software to detect plagiarism, and any failure to disclose collaborations will be treated as plagiarism).
3. completing 2 group projects. These group projects offer an opportunity for students to apply their newly learned programming skills to a relevant problem in the study of the mind, brain, and behavior.

Expectations

Students are required to complete all assigned coursework including an initial pre-course survey and all weekly tutorials and to turn in all tutorial and programming assignments and group projects by their deadlines. (but see **Lateness Policy** and late-day credits for tutorials and assignments).
There will be multiple meeting times available for students to meet with instructors weekly. Attendance to these online meetings/office hours is optional but the instructor may request your increased participation if your course performance is judged inadequate.

Course Time Allotment

This class functions as a flipped classroom. Course time will be designated to answer questions you have and do demos related to the weekly topic. Over the first 6 weeks, you should expect to complete 6 weekly Matlab programming modules. Each of these modules will consist of approximately 2-5 hours of online programming tutorials and associated assessments and 8-12 hours of programming assignments (90 hours total). In weeks 6 and 7, you will be completing a first group project (25-30 hours total). We will then transition to Python in weeks 8 and 9 consisting of 2 weekly programming modules – again consisting weekly of approximately 2-4 hours of online programming tutorials and associated assessments and 8-12 hours of programming assignments (25-30 hours total). The last 3 weeks of the course will consist of a "specialization track" where you will be able to choose a programming specialty to explore in more depth. Options will include computational linguistics, computational neuroscience, computational vision, machine learning, data science, and psychophysics. You will complete an online tutorial and programming assignment which will be extended as a group project (30-50 hours total).

Academic Support

Accessibility and accommodations

Brown University is committed to the full inclusion of all students. Please inform me early in the term if you may require accommodations or modification of any of the course procedures. You may speak with me by appointment. If you need accommodations around online learning or in classroom accommodations, please be sure to reach out to Student Accessibility Services (SAS) for their assistance (seas@brown.edu, 401-863-9588). Undergraduates in need of short-term academic advice or support can contact an academic dean in the College by emailing college@brown.edu. Graduate students may contact one of the deans in the Graduate School by emailing graduate_school@brown.edu.

We also understand that special assistance may be required under extraordinary circumstances including medical emergencies, personal or other family crises. We encourage students facing difficult situations to reach out to the graduate TAs as early as possible.

Student Support Services: Deans can also be a helpful resource to discuss personal, family, or health-related concerns, as well as a potential academic and personal plan. They are available for same-day consults and/or scheduled appointments. A list of other resources can be found here.

Inclusion

Brown welcomes students from around the country and the world, and their unique perspectives enrich our learning community. To support students whose primary language is not English, an array of English support services are available on campus including language and culture workshops and individual appointments. For more information, contact english-support@brown.edu or (401) 863-5672.

Use of technology to support your learning in this course

This course will use the following technological platforms: Canvas, Piazza, Grader, and Google Colab, Jupyter Notebook. We are committed to ensuring access to online course resources by students. If you have any concerns or questions about access or the privacy of any of these platforms, please reach out to me. The IT Service Center provides many IT Services including remote assistance, phones, tickets, and chat. Please also see the Online and Hybrid Learning Student Guide -- this course is "in person" but much of our learning will be completed online.

Course Structure and Policies

Assessment

>90% A, 80–90% B, 70–80% C, <70% NC.
Final grades will be calculated as follows:

1. Weekly tutorial assessments (10): 20%
2. Weekly programming assignments (8 mandatories with additional optional ones; best 8 scores will be taken out of all programming assignments completed): 50%
3. Group projects (2): 15% each
4. Bonuses: Up to an extra 5%
5. 1% penalty for every late day past submission deadlines for tutorials and weekly programming assignments

Books, Supplies, and Materials

The course does not use any textbook. Students will need a standard laptop. If your Brown undergraduate financial aid package includes the Book/Course Material Support Pilot Program (BCMS), concerns or questions about the cost of books and course materials for this or any other Brown course (including RISD courses via cross-registration) can be addressed to bcms@brown.edu. For all other concerns related to non-tuition course-related expenses, whether or not your Brown undergraduate financial aid package includes BCMS, please visit the Academic Emergency Fund in E-GAP (within the umbrella of "E-Gap Funds" in UFunds) to determine options for financing these costs, while ensuring your privacy.

Learning Management System, Submissions, Communication

Canvas will be used to manage the course material including tutorials, labs, homeworks, projects as well as grades and due dates. All submissions including labs, homeworks, and reactions will be submitted through Canvas. We will use Canvas for weekly announcements and discussions and Piazza for all other communications including chats.

Lateness

To provide additional flexibility to students, we give you 5 late-day credits which can be used flexibly for tutorials and programming assignments (but not group projects) throughout the semester without incurring any penalty. Beyond these 5 days, the equivalent of 1% of your final grade will be taken off per late day for each and every assignment.

Collaborations and Academic Honesty

Discussion, collaboration, and research on the web are encouraged. However, the work being turned in for credit must be the student's own. Sources and collaborators need to be acknowledged.
Brown's academic codes for undergraduate and graduate students can be found here. Plagiarism is considered a very serious offense and we take it seriously. We use software to detect plagiarism. Any violation of the code will be reported to the Dean of the College and may lead to academic sanctions.

Grade Complaints

Please address these by emailing all instructors at <email>, and include information about the assignment, the section that was penalized (if known), and your reasoning for the complaint. If you wish to meet with the primary instructor or a TA (either during office hours or separately) to discuss the grading, please specify this as well. Someone will follow up with you individually to schedule an appointment (if requested), or to communicate either an explanation for the grade or a confirmation that the grade has been appropriately adjusted.

Tips for Success

This intro to programming course is intensive. You are expected to dedicate 10-15 hours of work every week. There is no secret! Learning to program is hard. It is not unusual for students taking an intro to CS course to spend 40 hours weekly. While this course is not meant to be as intensive as a CS course you should expect to spend significantly more time than your typical psychology or neuroscience course!

Learning to program is like learning a foreign language. It is hard and it is slow and the best way to learn is by being totally immersed. I know it is hard to speak Matlab at the dinner table so the analogy breaks down quickly but if you want to learn to code and to learn well -- there is no secret. You need to practice and practice and practice -- ideally every day! The key is to start early so that you have sufficient time to meet an instructor before the deadline. We also encourage you to create your own teams for accountability and for potentially collaborating on assignments. Below is a suggested schedule for you to stay on top of your tasks and weekly assignments.

In Person Meeting / Office Hours Policies

In person meeting/office hours for CLPS 0950 are intended to be collaborative learning spaces for students of the course to seek help from your instructor and teaching assistants (TAs) and work through the assignments with peers. In order to ensure that these hours are consistently helpful and can reach as many students as possible, we ask that you read and abide by the following policy.

1. Please do not email your instructor or TAs individually: Emailing a single TA reduces the speed of response and deprives the rest of the class of the benefit of

the answer. Instead, please post all academic questions publicly in Piazza. In the case that the question is personal, very specific, and/or involves a lot of code posting, you may email it to <email>.

2. Office Hours: Office hours are intended to be collaborative. Therefore, we want to ensure that students understand that it is acceptable – and encouraged – to attend office hours to work on programming assignments with peers. However, if you want direct assistance from a TA please follow these guidelines:
    a. Have a specific question in mind. This can be on anything from the instructions to specific bugs in your code, but cannot be along the lines of "How do you do the homework?"
    b. Be prepared to show your previous attempts. When asking a question we expect that you have already attempted to solve the problem. Be prepared to explain the steps you have already taken and the different methods you have tried to debug your code or otherwise address the challenge you are facing.
    c. Use your "fair share" of the TA hours. We want to make sure we can get to as many students as possible, but it is often challenging to keep track of the students to whom we have already talked. So if you notice that someone has been waiting for longer than you or that someone has not been helped while you have already been helped multiple times, please let the TA know!
    d. Email or message all instructors ahead for questions outside of course scope. Each student will work on a final project that applies the skills from the course to a new topic or computational challenge. The projects are always fun and exciting but may cover software or concepts with which the TAs are unfamiliar. Email the questions, including background information on the project or topic, before office hours (at the latest the night before) to ensure that the TA can help.
    e. Work collaboratively and respectfully. Take opportunities to work with your peers, and to help those struggling if/when the TAs are otherwise occupied. Please be respectful and considerate of everyone's level of background in computation and the ways in which this impacts their experience in the course.

Course Calendar and Weekly Schedule

The course will be organized in roughly 3 sections:
First, we will introduce the fundamentals of programming using Matlab. We will then transition to Python. Finally, students will be able to choose their own "specialization tracks" including computational vision, psychophysics and data analysis, and computational linguistics. Tutorials will be offered for the different tracks and students' training will culminate in the final project which they can choose to complete in Matlab or Python.

Daily Schedule

	Suggested activities	Deadlines	Weekly meetups* (1-2 daily, with instructors)	
Saturday	Take one day to rest and one day to either finalize your tutorial or get an early start on your programming assignment.	Tutorial due at 11:59 pm	TBD	TBD
Sunday			TBD	TBD
Monday	Work on your programming assignment (every day for 3-5 hours)		10AM	2PM
Tuesday			TBD	TBD
Wednesday		Assignment due at 11:59 pm	10AM	2PM
Thursday	Work on your tutorial (every day for 1-2 hours)		TBD	TBD
Friday			10AM	2PM

Calendar: <link>

Fundamentals of programming Matlab	Week 1	*Introduction to computational thinking with Scratch & Variables* Tutorial due on Jan 29 at 11:59 pm. Programming assignment due on Feb 2 at 11:59 pm.
	Week 2	*Functions, Operators & Conditionals* Tutorial due on Feb 5 at 11:59 pm. Programming assignment due on Feb 9 at 11:59 pm.
	Week 3	*Loops* Tutorial due on Feb 12 at 11:59 pm. Programming assignment due on Feb 16 at 11:59 pm.
	Week 4	*Vectors, Figures, and Plotting* Tutorial due on Feb 26 at 11:59 pm. Programming assignment due on Mar 2. at 11:59 pm
	Week 5	*Matrices and Images* Tutorial due on Mar 5 at 11:59 pm. Programming assignment due on Mar 9 at 11:59 pm.
	Week 6	*Basic IO and Debugging, Matlab Desktop, Github.* Tutorial due on Mar 12 at 11:59 pm. No programming assignment-- work on group projects.
	Week 7	Group project Group project due on Mar 23 at 11:59 pm.
Python	Week 8	*Transitioning to Python* Tutorial due on Apr 9 at 11:59 pm. Programming assignment due on Apr 13 at 11:59 pm.
	Week 9	*Python continued* Tutorial due on Apr 16 at 11:59 pm. Programming assignment due on Apr 20 at 11:59 pm.
	Week 10	*Python jupyter notebooks* Tutorial due on Apr 23 at 11:59 pm. Programming assignment due on APR 27 at 11:59 pm.
Specialization	Weeks 11–13	*Final group project (Choice of language)* Tutorial due on Apr 30 at 11:59 pm. Programming assignment due on May 4 at 11:59 pm. Final project due on May 11 at 12:00 pm.