Contents

1 Introduction 1

2 Diversity and Professionalism 1
   2.1 Be an adult ................................................................. 1
   2.2 About Course Staff ..................................................... 2
   2.3 Open Door ................................................................. 3

3 Course Goals 3

4 Course Staff 3

5 Course Materials 4

6 Course Structure 5
   6.1 Lectures ................................................................. 5
   6.2 Assignments ............................................................. 6

7 Grading 7

8 Rubrics 8

9 Course Policies 8
   9.1 Collaboration Policy .................................................. 9
   9.2 Hours Policy ............................................................. 9
   9.3 Handin Policy ........................................................... 9
   9.4 Late Policy ............................................................... 9
   9.5 Extensions ............................................................... 10
   9.6 Contesting a Grade .................................................... 10
   9.7 Diversity and Inclusion ............................................... 11
   9.8 Accommodations ....................................................... 11
   9.9 Harassment .............................................................. 11
1 Introduction

Welcome to CS17, the second half of “Computer Science: An Integrated Introduction.” This document is our course missive. It outlines the course structure, requirements, and policies. It also introduces you to the course staff and describes their role in fostering your learning.

CS 17/18 constitute a year-long sequence. Here are a few things to note about this sequence:

- You need not obtain written permission to take this course sequence, and there is no enrollment limit.
- You need not take both courses to receive credit for the first. Indeed, many Brown students with only a cursory interest in computer science take CS 17 or CS 19 to learn the basics of computation.
- Only in very rare circumstances\(^1\) can a student take CS 18 without first taking CS 17 or CS 19.
- CS 19 fulfills the same requirements as CS 17/18, but many CS 19 students choose to take CS 18 anyway, and benefit from doing so!

2 Diversity and Professionalism

Students:
This missive is long, and many students may just skim it, but this part is so important that I’m putting it right up front. It is copied almost verbatim from Shriram Krishnamurthi’s discussion, because he put it so well, and has given permission to reuse it.

– Prof. Hughes

The lack of sufficient diversity is an important problem in computer science. In this course, we want to help improve the situation, not make it worse. Some of the responsibility for that lies with us, the course staff, but a lot of it ultimately rests with you, the students.

\(^1\)Professor Greenwald can only remember one case in her decade of involvement with CS 17/18.
2.1 Be an adult

College is a great time, and for many of you might offer a sense of new-found liberation. Its an opportunity for exploration and experimentation of various kinds. It also, however, provides opportunities to cross various lines, and unfortunately some people do so in awful ways.

Every now and then I hear disturbing statements from students about how they have been made to feel uncomfortable in class or in the department. I dont mean intellectual discomfort—the kind you might get from having a heated debate about a technical subject with a fellow student—but the personal kind. These statements report problems that range from inappropriate comments to invitations to even touching and other physical contact. The subjects are almost overwhelmingly (but not exclusively) female students or from races underrepresented in computer science.

Theres a term for some of the behaviors I hear about. It is harassment. And let there be absolutely no doubt about this: harassment is against the law and it is completely against the norms by which we want to run this course and this department. (See Browns Title IX Web site at http://www.brown.edu/about/administration/title-ix/policy.) We—the university, the department, and this courses staff—have absolutely zero tolerance for it.

Your reaction might be to laugh it off, or to make (or think) snide remarks about political correctness or jokes about consent or other things. You might think people just need to grow a thicker skin or learn to take a joke.

However, the subject of your harassment (and thats what your remarks and actions are, harassment, even if you decide you would classify them as jokes) is forced, by the nature of classes and campus life, to be around you. That can make them uncomfortable to the point of wanting to stay away, or focusing more on you than on what they are here to learn. That hurts their education. That is not okay at all: you have no right to steal their hard-won education away from them. And often the harm goes much deeper: it hurts them psychologically in subtle and long-standing ways. And thats why these are not laughing matters.

In light of recent reports about such issues on campus, Brown is taking additional steps to reduce this form of harm. Therefore, if I cannot appeal to your decency, intelligence, and collegiality, let me at least appeal to your self-interest. Do not mess around on this matter. It will not go well for you.

However, I prefer that you think of this in positive terms. Your classmates are your colleagues. Someday you may be each others start-up partners or co-employees; one of you may even be the others interviewer or boss. So start treating one another like professionals, and I mean that in the best possible interpretation of that phrase.

In short: Be safe, be happy, and have fun without taking away anyone elses.

2.2 About Course Staff

Professionalism and respect for diversity are not just matters between students; they also apply to how the course staff treat the students. The staff of this course will treat you in a way that respects our differences. However, despite our best efforts, we might slip up, hopefully inadvertently. When we do, please feel free to talk to us about it.

Sometimes, you may not be comfortable bringing this up directly to us. If so, you are welcome to talk to Laura Dobler (https://cs.brown.edu/~ldobler/) or to the Department Chair (https://sites.google.com/a/brown.edu/ugur-cetintemel/).
As a department, we will take all complaints about unprofessional or discriminatory behavior seriously.

2.3 Open Door

You are always open to come talk to me if you are facing any such issues. I will do my best to offer whatever aid I can. I am, however, not trained in this, and also suggest that you consider one of the many resources listed at http://www.brown.edu/about/administration/title-ix/resources. If you would like to learn more about Browns policies and resources, please see the universitys Title IX site.

3 Course Goals

CS 17/18 is an introductory computer science sequence that helps students begin to develop the skills, knowledge, and confidence to solve computational problems correctly, efficiently, and elegantly. The sequence is unique in that it teaches multiple programming paradigms—functional programming through the languages Racket and OCaml in CS 17 and Pyret in CS 19, and imperative and object-oriented programming through Java and Scala in CS 18. In this way, students learn to solve the same problem in a variety of different ways; further, they learn to identify classes of problems whose solutions better lend themselves to one programming paradigm/language or another.

Part of the philosophy of CS 17/18 is that computer science transcends syntax. Indeed, students program in multiple different programming languages instead of becoming an expert in just one.

Another philosophical guideline underlying CS 17/18 specifically is that students should understand how library code works rather than use it as a black box. (There is no such thing as magic!) In CS 17, students implement higher-order functions like map, reduce, and filter before using the default implementations provided by Racket or OCaml; in CS 18, students implement data structures like linked lists and hash tables before using the default implementations provided by Java or Scala. A third driving philosophy that is emphasized in the title of the sequence is “integration,” meaning students are taught to reason about algorithmic time and space tradeoffs almost immediately.

Although CS 17/18 exposes students to some practical experience in functional programming in CS 17 and CS 19 and object-oriented and imperative programming in CS 18, the central tenet of this sequence is not to teach programming but rather to introduce you to computer science, including algorithms and data structures, using programming as a vehicle for those topics as well as a goal in its own right. In particular, we touch on introductory ideas in areas like artificial intelligence, programming languages, software engineering, analysis of algorithms, and networking. This exploration of so many different topics leaves students who complete the sequence well-prepared for any number of upper level computer science courses, each of which might delve deeper into one or two of the topics that we only have time to scratch the surface of in CS 17/18.

The CS 17/18 sequence requires no previous programming experience whatsoever. Indeed, few high school students are exposed to functional programming. Hence, even students with previous programming experience find this sequence to be an invaluable part of their education.
4 Course Staff

<table>
<thead>
<tr>
<th>Professor</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Hughes</td>
<td><a href="mailto:jfh@cs.brown.edu">jfh@cs.brown.edu</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Head Teaching Assistants</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kathryn Scholl</td>
<td><a href="mailto:kscholl@cs.brown.edu">kscholl@cs.brown.edu</a></td>
</tr>
<tr>
<td>Harriet Small</td>
<td><a href="mailto:hsmall@cs.brown.edu">hsmall@cs.brown.edu</a></td>
</tr>
<tr>
<td>Anna Sabel</td>
<td><a href="mailto:asabel@cs.brown.edu">asabel@cs.brown.edu</a></td>
</tr>
<tr>
<td>Veda Sunkara</td>
<td><a href="mailto:vsunkara@cs.brown.edu">vsunkara@cs.brown.edu</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Teaching Assistants</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audrey Kintisch</td>
<td><a href="mailto:akintisc@cs.brown.edu">akintisc@cs.brown.edu</a></td>
</tr>
<tr>
<td>Alyssa Baum</td>
<td><a href="mailto:asbaum@cs.brown.edu">asbaum@cs.brown.edu</a></td>
</tr>
<tr>
<td>Charles Njoroge</td>
<td><a href="mailto:cnjoroge@cs.brown.edu">cnjoroge@cs.brown.edu</a></td>
</tr>
<tr>
<td>Cody West</td>
<td><a href="mailto:cwest1@cs.brown.edu">cwest1@cs.brown.edu</a></td>
</tr>
<tr>
<td>Carmen Zegura</td>
<td><a href="mailto:czegura@cs.brown.edu">czegura@cs.brown.edu</a></td>
</tr>
<tr>
<td>Eleanor Avril</td>
<td><a href="mailto:eavril@cs.brown.edu">eavril@cs.brown.edu</a></td>
</tr>
<tr>
<td>Elias Berkowitz</td>
<td><a href="mailto:eberkowi@cs.brown.edu">eberkowi@cs.brown.edu</a></td>
</tr>
<tr>
<td>Hans Halverson</td>
<td><a href="mailto:hhalvers@cs.brown.edu">hhalvers@cs.brown.edu</a></td>
</tr>
<tr>
<td>Huayu Ouyang</td>
<td><a href="mailto:houyang1@cs.brown.edu">houyang1@cs.brown.edu</a></td>
</tr>
<tr>
<td>Heila Precel</td>
<td><a href="mailto:hprecel@cs.brown.edu">hprecel@cs.brown.edu</a></td>
</tr>
<tr>
<td>Jamie Atschinow</td>
<td><a href="mailto:jatschin@cs.brown.edu">jatschin@cs.brown.edu</a></td>
</tr>
<tr>
<td>Justin Cardozo</td>
<td><a href="mailto:jcardozo@cs.brown.edu">jcardozo@cs.brown.edu</a></td>
</tr>
<tr>
<td>James Guesman</td>
<td><a href="mailto:jguessman@cs.brown.edu">jguessman@cs.brown.edu</a></td>
</tr>
<tr>
<td>Laura Case</td>
<td><a href="mailto:lcase@cs.brown.edu">lcase@cs.brown.edu</a></td>
</tr>
<tr>
<td>Mae Heitmann</td>
<td><a href="mailto:mheitman@cs.brown.edu">mheitman@cs.brown.edu</a></td>
</tr>
<tr>
<td>Neha Chaudhari</td>
<td><a href="mailto:nchaudha@cs.brown.edu">nchaudha@cs.brown.edu</a></td>
</tr>
<tr>
<td>Nikita Ramoji</td>
<td><a href="mailto:nramoji@cs.brown.edu">nramoji@cs.brown.edu</a></td>
</tr>
<tr>
<td>Peter Hahn</td>
<td><a href="mailto:phahn1@cs.brown.edu">phahn1@cs.brown.edu</a></td>
</tr>
<tr>
<td>Seth Barshay</td>
<td><a href="mailto:sbarshay@cs.brown.edu">sbarshay@cs.brown.edu</a></td>
</tr>
<tr>
<td>Shawna Huang</td>
<td><a href="mailto:shuang19@cs.brown.edu">shuang19@cs.brown.edu</a></td>
</tr>
<tr>
<td>Shoshana Simons</td>
<td><a href="mailto:ssimons@cs.brown.edu">ssimons@cs.brown.edu</a></td>
</tr>
<tr>
<td>Zhaoyong Zheng</td>
<td><a href="mailto:zzheng9@cs.brown.edu">zzheng9@cs.brown.edu</a></td>
</tr>
</tbody>
</table>

You can send mail to the entire course staff by emailing cs017tas@cs.brown.edu, or to the Head TAs by emailing cs017headtas@cs.brown.edu.

5 Course Materials

There are no required textbooks for this course, and you do not need a computer! There are ample machines in the Sunlab (CIT 143), Moonlab (CIT 227), MSLab (CIT 167), etc. Many students have laptops and use those to do much of the course work, or have desktop machines and use those, but neither is necessary. If you feel a particular need for a laptop, but do not have one, please talk with the course staff about potentially borrowing one for the semester.
You also don’t need to purchase any software; the programs you write will all be written using publicly available and free software. The only thing you must purchase for this course is a notebook.

You should bring along your notebook to class and to labs, and you should take brief notes highlighting important points you want to remember, and write down any questions you might have so that you don’t forget them. Shortly after each lecture, we will post notes online, so you need not transcribe the lectures.

Your notebook will also be useful while working on assignments. You will use it to sketch out ideas on paper. The problems we assign require you to think. Keeping a record now of your thoughts about problems and the answers you develop will save you time later.

In addition to a notebook, you will be required to obtain an iClicker, which are available free of charge at the IT service center in the CIT lobby. You will use these iClickers in class to take short quizzes that test your understanding of the concepts taught during lecture.

Electronic Handouts and Handins  Out of deference to the trees we hope will continue to populate our picturesque Rhode Island landscape for many years to come, this course is paperless. All assignments, lectures, and course documentation will be posted on the course website at [http://www.cs.brown.edu/courses/cs017](http://www.cs.brown.edu/courses/cs017) in .pdf format. Likewise, you are expected to turn in assignments electronically. If you would like hard copies of some of the course materials, you may print them yourself; printing CS-related materials in the Sunlab is “free.”

6 Course Structure

CS 17 consists of about 36 lectures (some of which include short quizzes), about 12 hands-on labs, about 11 homeworks, 4 projects, and 1 final exam. These numbers may vary slightly as the course evolves, in order to accommodate enrollments.

Lest this sound like a daunting workload, keep in mind that CS 15 (with its four large projects) and CS17 require about the same amount of work.

6.1 Lectures

You are expected to attend all lectures.

We introduce you to new material in lectures. Lectures are interactive sessions lasting roughly a microcentury[^3] and you are strongly encouraged to participate. They are designed to give you an opportunity to digest the material being presented, and to allow the professor to adapt to your needs. You are encouraged to take advantage of this design.

Asking questions during lecture is an important part of the learning process. If you’re confused, then chances are that someone else is too, so please don’t be shy: ask for clarification when you don’t understand something that is being discussed. And if a question or idea pops into your head that draws heavily on material not covered in the class, you can write it down in your notebook and bring it up after class or during the professor’s office hours.

[^2]: It won’t cost you any money today, but it may well impose a cost on life as-we-know-it in the long-run.
[^3]: About 52 minutes, the ideal length of a lecture according to John von Neumann.
Lecture notes will be posted on the class website after each lecture. You should study the material in the previous lecture as soon as lecture notes are available. One effective way to do this is to try to reconstruct the lecture from scratch, without looking at the lecture notes. If you like, try doing this with a partner; speaking your thoughts aloud can help jog memories. This process will help you distinguish between what you really understand and what you thought you understood.

Lastly, although we do our best to ensure that the course lecture notes are comprehensive, you are responsible for all material covered in class even if, due to an oversight on our part, something is not mentioned in the notes.

6.2 Assignments

Assignments in CS 17/18 come in a variety of scales: quizzes, labs, homeworks, projects, and exams. A quiz takes about five minutes, a lab takes two hours, a homework can take several hours, and a project can take much longer (each project is spread over two weeks). Exams are take-home.

Quizzes Quizzes are designed for both students and the professor to gauge whether students understand the material being presented in lecture. If you can tell from the quiz that you do not have a good grasp on that day’s lecture material, you should go that same evening to TA hours. (If you cannot make it that same evening, just be sure to go before the next lecture to prevent yourself from falling behind.)

Labs Labs are two hour, interactive programming sessions—the equivalent of wind-sprints for sports. They are intended to offer a short but intense investigation of concepts introduced in lecture, or of new concepts. It is our intent that you will learn to program quickly and effectively in lab. You will be more productive during your lab time if you have read the lab handout beforehand.

At the beginning of each lab, with the help of the TAs, you will pair up with another student so you can “pair program” the lab. You are required to work with different students over the course of the semester to gain exposure to the different ways others approach problems.

Throughout the labs, there will be “checkpoints,” which instruct you to ask a lab TA to check your work so far. The TA may ask you to improve your solution to a problem before moving on to the next section, or may let you move on immediately. Once you complete the final checkpoint at the end of the lab, you are free to go.

Homeworks Homeworks are designed to help you internalize the course material. They consist of written problems and short programming tasks. Homeworks also include practice problems, which are for your benefit but need not be turned in. However, you should feel free to discuss these practice problems in detail with your fellow students and with the course TAs. Non-practice (i.e., graded) problems may only be discussed with fellow students and TAs in a manner consistent with the CS17 Collaboration Policy. 

---

4See the Pair Programming handout for details.
5See the Collaboration Policy handout for details.
Projects  Programming projects give you practice in putting together a larger, more complicated program than you construct in labs and homeworks.

While the projects are challenging, ultimately the program you hand in need not be very large. If you get stuck, or if your program becomes awkwardly large or complicated, you are probably headed down the wrong path, and are encouraged to seek help from the TAs.

Each project has two deadlines: one for a design check and one for the final hand-in. The purpose of the design check is to make sure your design is reasonable before you begin coding. You will generally have about a week between the project release date and the date of your design check.

Completing each project will require you to manage your time responsibly. You are expected to start work on each project when it is assigned, and to work steadily until it is complete. Doing so will provide you with the time necessary to fully understand the problems posed, craft simple and elegant solutions, and seek guidance from TAs and fellow students when needed (keeping the collaboration policy in mind, of course!).

Exams  There’s only one exam in CS17, a take-home final. It is wholly non-collaborative. As such, it will help us evaluate your individual knowledge of the material in a way that collaborative work cannot. But since the exam is take-home and open-notes, it’s a lot like a homework assignment that you must do entirely on your own.

Workload  This course requires that you do a lot of work, but the work is not mindless labor; it is intellectually challenging, and alumni often tell us that it was a crucial part of their computer science education.

You must keep up! There aren’t many opportunities to take a deep breath and say, “Ahhh… a quiet stretch.” You’ll hand in a big project, and then there will be homework to do. You’ll finish a homework, and a design check will follow.

As an example of the time commitment, the projects each seem to take students an average of 20 hours. You’ve got a couple of weeks for each, so if you spread it out it won’t be overwhelming, but it’s still a good chunk of time.

To sum up: If you’re looking for an easy course, this isn’t it. But if you’re looking for a stimulating, thought-provoking introduction to the field of computer science, you’re in the right place.

7 Grading

In CS17, everyone who earns an A gets one. Your grade is independent of the grade or work of any other student in the course.

Each assignment (quiz, project, homework, exam) has work that totals up to some number of points. Often it’s 100, but sometimes it’s not. The number of points you earn is your “raw” score. After collecting all raw scores, the professor then decides what raw score marks the cutoff between an A and a B, between a B and a C, and between a C and an NC.

Your raw score and these cutoffs are then used to compute your “normalized” score for the assignment, which is a number between 0 and 100. Once normalized, the AB dividing line is 90, the BC dividing line is 80, and the BC dividing line is 70. Your final grade is then determined by the sums of your normalized scores.
line is 80, and the $C-NC$ dividing line is 70. Normalized scores are what are used to compute course grades.

An example illustrates this scoring system best. Suppose on some homework assignment, the total number of points you can earn is 80. Suppose further that the raw-scored ranges are as follows: 63 to 80 is an $A$; 40 to 63 is a $B$; and 30 to 40 is a $C$.

Then a raw score $x$ between 63 and 80 is normalized this like:

$$C_1(x) = 10 \left( \frac{x - 63}{80 - 63} \right) + 90.$$  

So, for $x = 63$, we get $C_1(x) = 90$, and for $x = 80$, we get $C_1(x) = 100$.

Similarly, a raw score between 40 and 63 is normalized like this:

$$C_2(x) = 10 \left( \frac{x - 40}{63 - 40} \right) + 80.$$  

Again, for $x = 40$, we get $C_2(x) = 80$, and for $x = 63$, we get $C_2(x) = 90$.

Finally, a score between 30 and 40 is normalized as:

$$C_3(x) = 10 \left( \frac{x - 30}{40 - 30} \right) + 70,$$

and a score less than 30 is normalized as:

$$C_4(x) = 10 \left( \frac{x - 0}{30 - 0} \right) + 0.$$  

Note, for example, that $C_1(63) = C_2(63)$, so it doesn’t matter which formula is used to convert scores on a boundary between two grade ranges.

Late penalties (see Section 9.4) are applied to the normalized scores. Again, let’s look at an example: if, in the scenario above, your raw score is 67, then your normalized score is about 92.4, a low $A$. But if you hand your assignment in 30 minutes late, the score used in computing your final grade will be 82.4, a low B.

The following table shows how much each type of work counts in computing your final grade:

### 8 Rubrics

Your work in CS17 will be graded **anonymously** by undergraduate TAs, with professor participation at times, using rubrics developed by the professor and course staff. If you have any concerns about this structure, please contact the professor.

In CS17, your work is graded based on functionality, design, testing, style, and efficiency. We describe here the rubrics for labs, homeworks, and projects. The rubric for exams usually resembles that of homeworks, but we reserve the right to deviate from the standard homework rubric where appropriate.
9 Course Policies

This section describes our course policies. If you have any questions about any of our policies, ask for clarification now. Do not wait until you have violated one of our policies!

9.1 Collaboration Policy

The CS17 collaboration policy is described in a separate document.

9.2 Hours Policy

The CS17 hours policy is described in a separate document.

Be sure to read this document carefully. It describes how you can help TAs use their limited time to help you efficiently and effectively.

9.3 Handin Policy

The work you do in lab is evaluated in person by the lab TAs. Every other assignment handout contains a section entitled “How to Hand In,” which you must follow in order to hand in your work electronically. We will not accept hard copies of your assignments, or handins via email.

When handing in assignments, make sure you save your files in the required format (e.g., .txt or .pdf), and that you name your files using the file names provided. The handin tools will not accept wrongly named files.

N.B. All code you hand in must compile and run, as-is.

9.4 Late Policy

Quizzes You cannot make up quizzes. Come to class!

Labs Labs are designed to take the full two hours. If you are late to lab by 20 or more minutes you will receive no credit for that lab; Regardless, we recommend completing the lab, as you are responsible for mastering all information taught during lab.

If you discover that you cannot make your lab section during a particular week, you can email the TAs to temporarily switch into another lab section. You should send this email at least 24 hours in advance of your scheduled lab. That way, the TAs will have sufficient time to read your email, help you find an open slot in another lab, and confirm the change. If you email the TAs with less notice, say only 2 hours in advance of your scheduled lab, and no TA is able to help you before your usual lab time, you are expected to go to your assigned lab. If you miss a lab without having a confirmed lab switch, you will receive no credit for the lab.

Homeworks Homeworks that are turned in on their due date, but beyond the published “due time” (e.g., 5:00 p.m.), will be considered late. You will receive a 10 point penalty if your homework
is late by no more than 1 hour, and a 20 point penalty if it is late by no more than 24 hours. You will receive no credit on any homework that is more than 24 hours late.\footnote{24 hours is 24 hours (unless otherwise noted, of course!). In particular, school breaks, such as Presidents’ weekend or Spring Break, do not alter the definition of 24 hours.}

\textit{Deadlines are strictly enforced; there is no grace period.}\footnote{FastX seems to have a nasty habit of running out of space precisely when a homework deadline is approaching. Please plan \textit{in advance} for this possibility; extensions are not given for handin difficulties caused by FastX.}

In the event that you do not hand in an assignment, we suggest that you at least become familiar with the material, as the course material builds on itself over the course of the semester.

\textbf{Projects}  You will sign up for design checks by running a signup script for each project. You will receive instructions via email about how to do this. \textit{The email will contain a deadline by which you must sign up for a design check; if you miss this deadline, you will not get any credit for the design check!} Further, if you do not show up for your design check at your scheduled time, you will lose 10 points. (If one partner does show up, only the one that doesn’t show up will lose points.)

During your design check, you should schedule a time for your final grading. Like design checks, if you do not show up for your final grading at your scheduled time, you will lose 10 points. (If one partner does show, and one doesn’t, only the one that doesn’t show will lose 10 points.)

One project may be turned in late, by no more than 24 hours, without penalty. For the projects that are programmed in pairs, \textbf{both} partners must not have used their late day for late handin to still be available. Please take note of this rule, particularly when choosing partners for later projects.

For all other projects, you will receive a 20 point penalty (on the “normalized” or “rescaled” grade) if it is late by no more than 24 hours. You will receive no credit for any project that is more than 24 hours late (once your or your partner’s free late day has been used up).

\textbf{Exams}  \textit{Late exams will not be accepted.}\footnote{Except in the most extreme of circumstances; see Section \ref{sec:extensions}.}

\section{Extensions}  
Extensions are only granted for good reasons. Illness \textit{with a note from health services} is always a good reason. So is a death in your family.

A request for an extension must be made directly to the professor at least 48 hours before the assignment’s due date (except in cases where an emergency arises at the last moment). Extensions may be requested on homeworks and projects, but extensions will not be granted on labs\footnote{Labs may be rescheduled; see Section \ref{labresched} re: Labs.} except in the event of documented illness.\footnote{Forgive us for sometimes creating policies in our own self-interest!}

The exam is take-home. Only in the most extreme of circumstances is an extension ever granted on the exam.

Only the professor can grant extensions, so please do not ask any of the TAs for an extension.
9.6 Contesting a Grade

You are encouraged to look back over your assignments after they have been graded. If you find a possible error or believe that you lost too many points, please email or come to the hours of the TA who graded that problem. The rubric for each problem will include the login of the TA who graded it.

If there was a simple arithmetic mistake or if the TA misgraded the problem so that the given grade is inconsistent with the established rubric, you will receive points back immediately. If the dispute cannot be resolved, the TA may refer you to an HTA or consult with an HTA themselves and get back to you by email.

9.7 Diversity and Inclusion

Brown’s computer science department is committed to diversity and inclusion, and strives to create a climate conducive to the success of women, students of color, students of any sexual orientation, and any other students who feel marginalized for any reason.

If you feel you have been mistreated by another student, or by any of the course staff, please feel free to reach out to one of the CS department’s Diversity and Inclusion Student Advocates, or to Professor Greenwald, Professor Nelson, or Professor Doepplner (the CS departments director of undergraduate studies) or Professor Cetintemel (the CS department chair). We, the CS department, take all complaints seriously.

9.8 Accommodations

If you feel you have any disabilities that could affect your performance in the course, please ask SEAS to inform the course staff. We will support accommodations recommended by SEAS.

9.9 Harassment

Please review Brown’s Title IX and Gender Equity Policy. If you feel you might be the victim of harassment (in this course or any other), you may seek help from any of the resources listed here.

10 Where to Get Help When You Need It

Our goal in CS 17/18 is to guide you as you develop the skills, knowledge, and confidence to solve computational problems correctly, elegantly, efficiently, and with ease. That’s why we assign a significant amount of challenging and thought-provoking problems. To help you cope with the workload, Brown CS in general, and CS 17/18 in particular, provide many resources to support you: approachable professors, a strong and enthusiastic TA staff, and online documentation. Take advantage of these resources!

Part of your learning experience in CS17, and throughout your time at Brown, will involve a productive struggle with problem solving. If/when you encounter feelings of frustration, don’t be afraid to ask for help. Rather than beat your head against a wall, seek out your friendly staff.
We are here to help!

10.1 Course Website

The course website is [http://www.cs.brown.edu/courses/cs017](http://www.cs.brown.edu/courses/cs017).

All documentation, lecture notes, and assignments will be posted on the website. On it, you will also find links to the syllabus and the schedules for TA hours (see Section 10.4), workshops (see Section ??), and labs.

10.2 Piazza

In CS 17/ 18, we use an online academic forum called Piazza to handle questions and comments that arise outside class and office hours.

In CS 17/ 18, we encourage you to post questions and comments privately (i.e., to the course instructors only). That way, you don’t risk conveying any information about an assignment’s solutions to your fellow students, in possible violation of the course collaboration policy.

Soon after you post, a friendly TA will respond with the “Instructor’s Answer” to your post, and if they think that the thread might benefit the entire class, they will anonymize it, and make it visible to the class.

Before you post, please read through the posts relevant to your topic to check whether your problem has already been addressed. If you post something redundant, you will soon realize it, because you will not receive a response from a friendly TA in a timely manner. That is our subtle way of telling you to read the forum before posting.

Furthermore, we try to give timely responses, but from midnight to 7 a.m, you should not really expect prompt answers, or any answers at all.

Lastly, feel free to comment on any questions/answers/comments you read on Piazza. Piazza is a forum for discussion and we welcome your contributions!

10.3 Email

Questions or comments not directly related to the course material (for example, “I can’t log in to Piazza!”) should be emailed to the entire course staff at cs0170tas@lists.brown.edu. The only exception to this general rule is in the case of a grade complaint; then, you should start by contacting your specific grader. If you and your grader cannot resolve the issue to your satisfaction, you can then contact the head TAs, at cs0170headtas@lists.brown.edu.

You can also email the head TAs about other more personal concerns, such as a personality conflict with your partner on a project. If the head TAs cannot resolve your issue or answer your question, feel free to contact the professor.

Again, we try to give timely responses, but from midnight to 7 a.m, you should not really expect prompt answers, or any answers at all.
10.4 TA Hours

Each of the TAs holds at least two hours of walk-in office hours, per week. Each week’s schedule and location is posted on the course website.

Contrary to what you might have experienced in high school, attending TA hours in CS 17/ 18 does not send a message to your peers that you are falling behind. Au contraire, the most successful students are usually those who come to hours early and often—whenever they have questions about topics covered in lecture or would like some guidance as they work through an assignment.

10.5 Course Mailing List

All students belong to the course mailing list. Messages sent to this list are sent to your CS email account. For your convenience, the setup script you run during the first lab of CS 17 forwards all mail received by your CS account to your Brown account. For more information about your CS email account, please visit [http://cs.brown.edu/facilities/system/email](http://cs.brown.edu/facilities/system/email).

10.6 Announcements

Announcements, information about upcoming talks, interesting links, assignment clarifications (if necessary), and more will be sent to the course mailing list, posted on Piazza and/or the course site, and/or announced during class. You are responsible for keeping up to date with all messages, regardless of where they are posted.

11 Ergonomics

Much of the following is from [http://cs.brown.edu/about/system/ergo.html](http://cs.brown.edu/about/system/ergo.html), which contains links to other places with still more information.

Working at a computer may seem harmless, compared to (say) working with nuclear energy, but there are actually some substantial risks associated with this line of work. One source of risks is being immobile (except for the hands) for long periods of time. This can lead to back and neck and wrist injury, each of which can take a very long time to correct. Worse still, the harm done can accumulate little by little over years, only showing up much later. Now is the time to start good habits.

- Make sure you are sitting properly: Is your lower back supported? Is the top third of your screen level with your eyes? Are your feet on the floor? Are your wrists as un-bent as possible?

- Take frequent breaks. Move the rest of your body and give your hands/wrists frequent rests. Never push it. Listen to your body: If it says you need a break, take it.
Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS17 document by filling out the anonymous feedback form: [http://cs.brown.edu/courses/cs017/feedback](http://cs.brown.edu/courses/cs017/feedback).