# Teaching Computer Science

Aubrey Anderson Jeremy Colebrook-Soucie
Alexis Galantino Christopher Hinstorff
Aman Javeri Kadri Jack Kerns Mary-Paule Monks
Emma Plankey Kelly Pogue

# Contents


**1 TA responsibilities** **7**

What to expect when you're expecting to be a TA . . . . . 7

What we (and now you!) stand for . . . . . 7

Getting Started . . . . . 8

Registering with payroll . . . . . 8

Billing/timecard . . . . . 9

Office Hours . . . . . 10

How to prepare for the first week: . . . . . 11

Labs/recitations . . . . . 11

Labs . . . . . 11

Recitations . . . . . 12

How to prepare for the first week: . . . . . 12

Grading . . . . . 13

How to prepare for the first week: . . . . . 13

Piazza . . . . . 14

Denouement . . . . . 14

**2 Resources Available to a UTA** **15**

Introduction . . . . . 15

Learning Objectives . . . . . 15

What Resources to Find Where . . . . . 16

Subject matter . . . . . 16

Moral Dilemmas . . . . . 16

Guidelines . . . . . 17

Teaching Methods . . . . . 18

Misc . . . . . 19

Case . . . . . 19

List of Resources . . . . . 20

Individuals . . . . . 20

Documents . . . . . 21

Links . . . . . 21

**3 You, TAing, and Ethics** **23**

Introduction . . . . . 23

Learning Objectives: . . . 23
So what do you mean by ethics? . . . 23
So I can skip this section? . . . 23
For First Time TAs: . . . 24
On fraternization with students: . . . 24
Skills and strategies . . . 25
Divide and Conquer: How to break down an ethical dilemma . . . 25
The AIR Model of Ethical Inquiry . . . 25
Syracuse Model of Ethical Decision-Making . . . 25
There's Always Another Option . . . 26
I'm stuck and overwhelmed and I don't know what to do . . . 26
Balancing Empathy and Professionalism . . . 26
Applications: Three Case Studies . . . 26
SITUATION 1: Friends in Office Hours . . . 27
SITUATION 2: Academic Honesty . . . 27
More Info . . . 29

**4 Inclusivity** **31**
Introduction . . . 31
A First Approach . . . 31
Skills and Strategies . . . 32
Mindset . . . 32
Active Listening . . . 33
Questions . . . 33
Conversational Awareness . . . 33
Cultural Understanding . . . 34
Computer Science Specific Ideas . . . 34
Examples . . . 35
Resources . . . 35

**5 Mental Models** **37**
Introduction . . . 37
Learning Objectives. . . . 38
Background Knowledge . . . 38
Skills/Strategies . . . 38
Assessing a Student's Mental Model . . . 38
Building New Mental Model . . . 39
Adjusting Incorrect Models . . . 40
Application . . . 42
Case Study 1: Variable Names vs Variable Contents . . . 42
Case Study 2: Variable Assignment . . . 42
Case Study 3: Control Flow . . . 43
Case Study 4: Order of Events . . . 43
Resources . . . 43

**6 Office Hours** **45**

Introduction & Motive . . . 45
Learning objectives . . . 45
Logistics . . . 45
How do I schedule office hours? . . . 45
Where do I hold office hours? . . . 46
What is Halligan Helper? . . . 46
Teaching a large number of students . . . 46
Types of questions you will receive . . . 47
Case studies . . . 47

**7 Pedagogy** **49**
Introduction . . . 49
Learning Objectives . . . 49
Skills/Strategies . . . 49
Diagramming . . . 50
Asking effective questions to discover misconceptions . . . 50
Rubber-duck debugging . . . 50
Creating examples . . . 50
Assessment . . . 51
Application . . . 51
Asking Questions . . . 51
Fixed vs. Growth Mindsets . . . 52
Case Study . . . 52
Challenges: . . . 53
Resources: . . . 54

**8 Professionalism** **55**
Learning Objectives for this Module: . . . 55
Professionalism Outside the Classroom: Preparation and Excellence . . . 55
Motivation . . . 56
Professionalism Inside the Classroom . . . 57
On student-teacher relationships . . . 57
The Caveat . . . 58
Additional Resources . . . 58

**9 Recitation & Lab Skills** **59**
Introduction . . . 59
Learning Objectives: . . . 59
Select skills and strategies . . . 59
Preparation . . . 59
Your first class . . . 60
Motivation . . . 60
Teaching to the class . . . 61
Interactive problem solving . . . 61
Case studies: example situations . . . 62
When students stare blankly . . . 62

When students "just don't get it" . . . . . 62
When you don't know the answer . . . . . 62
Conclusion . . . . . 63
Further reading (and/or watching): . . . . . 63

**10 When you don't know what to do** **65**
Introduction & Motive . . . . . 65
Learning objectives . . . . . 65
Select skills and strategies . . . . . 65
Fall back on your previous TA experience . . . . . 66
"What Would An Authority Figure Do?" . . . . . 66
Make use of prior art . . . . . 66
Phone a friend . . . . . 67
Case studies: example situations . . . . . 67
Further reading . . . . . 69

# Chapter 1

# TA responsibilities

## What to expect when you're expecting to be a TA

Congratulations! You have been accepted into the clandestine cabal that is the Computer Science Teaching Assistants' Guild of Tufts! Phone your parents, update your Tinder profile, and get all of the other bragging rights out of the way. Once you've got that out of your system, you might begin to ask yourself, what will I actually be doing?

Clearly the answer is having one of the most formative and intellectually stimulating experiences you will have in college, if not your entire existence. But specifically, what will you be doing with your time? How do you prepare for your first week? Here, dear reader, is where all of your questions will be answered, provided all of your questions are restricted strictly to the domain of cursory summaries of your duties in the department.

### What we (and now you!) stand for

We'll be getting to your actual duties down the road, but it's important to lay out your capital-D Duties, to the school, to the students, to the professor, and to yourself. My colleagues have written excellent chapters on decision-making, professionalism, and other important intangibles, but keeping these goals (presented in no particular order) in mind will provide a good rule of thumb in any situation while you work:

1. Motivation and support. Often the largest barrier keeping students from succeeding is their own self-esteem. Professor Ramsey considers this to be the main goal of being a teaching assistant. You'll find that frequently the

student knows the answer but needs small pushes and kind words to get them in the right headspace. Additionally, the CS classes here are tough, and personal motivation and encouragement are some of the few things that are difficult for a professor to provide in the standard lecture setting.

2. Evoking understanding in your students. That is of course one of the primary goals of any university, and as you are now an employee, so too is it of yours. When deciding whether answering a question would give too much away, or how much detail to go into explaining specific edge cases of a new concept, remember to do what would best help the students understand for themselves. (For more on these questions, see Chapter XX: Teaching Abstract Concepts or Chapter XX: Pedagogy.)
3. Professionalism and representing Tufts. With great power comes great responsibility. You are now a role model, and a bridge between the student body and the university. (For more on this, see Chapter XX: Professionalism.)
4. Guidance. You're a guide and a helping hand now, not a search engine. You are not expected to know everything, but you should be able to point to where to look if a student asks you something you do not know. On the other hand, you do have a certain responsibility to be somewhat competent in the subjects covered in class. (For more on what to do if you don't know what to do, see Chapter XX: Interpersonal Skills.)
5. Compassion. Above all, we want Tufts to be a safe, comfortable, and exciting learning environment. Oftentimes, students will spend more time with you than with the professor. It's important to always, always keep in mind the feelings of the student. One bad experience with a TA might result in a student resenting a class, which would be a loss for both them and for the school. (For more on compassion, see Chapter XX: Ethics and Chapter XX: Inclusivity. Watch Life is Beautiful. Observe elderly couples in the park. Write a letter to that one elementary school teacher who really made an impact on you.)

## Getting Started

If you've never been a TA before, there are a few administrative issues to get out of the way. Below we have shamelessly copied TA Max Bernstein's section on getting started from his Comp 15 manual.

### Registering with payroll

Payroll can be a bit of a hassle to get it set up the first time around — but after you hand in your paperwork for the first time, you do not have to do it for subsequent semesters.

Also note that this must be done in order for you to get paid! Complete it within a week of the start of the semester.

The process is as follows:

1. Fill out Bruce's initial payroll form at https://www.cs.tufts.edu/~molay/timecard/
2. Fill out a US Government I-9 (Employment Eligibility Verification) form

- Note that this requires one "super strong" form of ID (like a US Passport) or two "weaker" forms of ID (like state driver's license and Social Security Number)

3. Submit the I-9 to one of the following locations:

- Donna in the computer science main office
- A student worker at the Dowling service desk

4. Wait...

Verification normally takes a couple of days and at least a week — the first time. In later semesters, it may only take a couple days.

If you are not a US citizen, there is some additional paperwork required to assign you a Social Security Number (SSN) or other Taxpayer Identification Number (TIN). Please contact Tufts HR at 617-627-7000 as soon as possible. The process to be assigned an SSN or TIN can take a long time. Privacy

If you haven't heard of FERPA at this point in your career, please go straight to Google.com and search "FERPA". Even if you have heard of it, a refresher is probably in order. Once you have read at least the first couple paragraphs of the Wikipedia article and several from the ed.gov website, resume reading here where you left off.

{end Max's section–thanks Max!}

You're now on the payroll! Don't forget to bill any and all hours you spend on your class, including meetings, Piazza answers, or regrading. Fair is fair, and you deserve it. Speaking of which.

## Billing/timecard

To bill hours, go to http://www.cs.tufts.edu/~molay/timecard/, login with your CS username, and click on Record TA Hours. It should be fairly self-explanatory what to do–select the date, time, and thing you did. We really hope you do this at least weekly–I log my hours as soon as I'm done with them, or even while they're still going on, if I'm sure I'll be there the whole time. While it's very tempting to leave logging your hours until the last week of the semester, this creates a nightmare for the people paying you, whom you should be treating as

kindly as possible[1].

And now, on to the good stuff. For each of these duties, I'll outline the basic idea, some course-specific details, and what to do for your first week or month.

## Office Hours

We have a whole section on office hours, which is full of insightful tips on how to make your office hours the best it can be for all involved. But here's the CliffsNotes. Or are you kids using Sparknotes these days? Not that we here in academia endorse either. Anyway.

Almost every TA will hold office hours during the week. Specifics vary by class, but the idea is you make yourself available to students for questions, generally regarding homework or conceptual understandings. If you're given a choice as to when to hold office hours, consider the due dates of the assignments (which are usually consistent here in the CS department) and how stressed you like to work; closer to the deadline usually means more students showing up. Some TAs thrive on the thrill of helping thirty-odd panicking students, while some prefer to take the slower hours, which can get boring but sometimes result in meaningful conversations with the few students who start the homework the day it's assigned. No shame in either.

Some classes use Halligan Helper (https://www.halliganhelper.com/). If you don't have an account, get on that. It allows students to put themselves on a queue with their location and their question so that TAs know where and how to help them. Halligan Helper can be controversial–several classes are trying to phase it out. At the time of writing, Comp 105, 40, 11, and 15 use it, whereas Comp 170 and 160 do not. (Although this also varies by professor; I know Professor Sheldon pushes against it when he teaches 40.)

Additionally, most classes have specific areas they have adopted for their office hours. This might be officially prescribed by the professor (and possibly on the website), but oftentimes not. I'll run down the list for the core classes–if you're doing an elective you'll have to do your own research.

- Comp 11: Halligan collaboration room (upstairs, down the hall)
- Comp 15:
- Comp 40: Generally in 116 and 118
- Comp 61:
- Comp 105: Halligan upstairs kitchen
- Comp 160: Halligan collaboration room
- Comp 170: Halligan extension

[1] Not that being nice will get you paid any more or faster. Being nice is just the right thing to do.

### How to prepare for the first week:

Usually, the first week or so has fewer students than usual, so hopefully you'll be able to become accustomed to the routine before you hit the real brunt of the work. This depends on when the first homework is due, but in my experience, students who come in the first week can have more administrative questions, or questions about setting up their text editor. If you don't know the answer, don't sweat it! That's not going to be representative of the kind of question you'll be asked throughout the semester. Usually an "I'm sorry, but I don't know" will do, but if you can help them, go for it.

Make sure to thoroughly read the assignment, as well as the course website and the syllabus. You'll find that many questions you get, in the first week and beyond, are just trying to understand the question, or can be found on the syllabus. You don't need to know everything, but it helps to be knowlegeable on syllabus-related things.

When you get to office hours, just relax, put yourself on Halligan Helper if applicable, and if there's anyone working, say aloud that you're on duty. It's not your responsibility to know everything, so don't sweat it if you don't know the answer to a question. Remember, you're just a second, slightly more experienced set of eyes. On the other hand, you might be surprised at how much comes back to you.

## Labs/recitations

Once again, we have a beautiful chapter on recitation skills on page XX. The idea with labs and recitations is that you will be overseeing what is most likely the first time the students put into practice the skills and concepts they learned that week in class. In labs, this is usually a coding challenge, whereas recitations tend to be more conceptual problems done on paper. There is some overlap between the two, but they're distinct enough to warrant differentiation here.

### Labs

Labs are done in Comp 11, 15, and 40[2], and a few electives. You might be the lab leader or a lab assistant. For the majority of the time, both roles will roam the classroom, answer questions and making sure nobody's dead lost. The primary difference comes from the beginning of the class: Usually, and this will depend on the class and on how confident the students are on the material, the lab leader will give a brief overview of the relevant topics as well as the problem.

[2]In 40, labs are generally run by the professor or the head TA, so you won't have to worry about this section as much. Unless you're the professor. Hi, Noah!

If you're a lab leader, you'll know it, and you'll be going over the content of the labs each week in a meeting. You may need to go over the topics from class one more time, but it's up to you to gauge how confident the students are on the subject matter. Most likely if you're a lab leader then you've done this before or at least have been a TA for the class for a semester or two (although if you haven't, don't worry–that you were chosen for the position means you're already more than qualified). The biggest tip I can give here is try to get your opening spiel done with enough time for the students to get their work done. If something's unclear, you can answer questions individually.

If you're a lab assistant, you will probably be in charge of collecting and grading the pre-labs. While the leader is explaining the problem, listen along so you understand the purpose of the lab. After the lab leader's done, both of you will wander the class in search of anyone who needs help. If you see someone who looks lost, mosey on over to them and ask them if they have any questions. You'll be surprised at how often students will have questions that they for whatever reason didn't want to ask.

## Recitations

Recitations, as I've said, are generally done on paper, and go over concepts rather than the implementation of coding ideas. Make sure to somehow keep track of attendance, and make an effort to learn names. Each week the professor or a graduate TA will go over the problem so you won't be in over your head, but make sure to run through it a few times in your head to anticipate any questions you'll receive. Once again, chapter XX has a bunch of tips on how to make your recitations run smoothly and successfully.

## How to prepare for the first week:

If there's a prelab, do it yourself! It'll get you familiar with the focus of the lab if you don't have a copy of the lab itself beforehand. (Also you are often not given answer keys for the prelabs, so it's nice to know what the answers should be.) If you're a lab or recitation leader, try working through the problems on your own time, or if not, work through the solutions. It's imperative that you fully understand the content of the recitation, so don't be afraid to ask clarifying questions or even to go over whole concepts with the professor or another TA.

Don't forget to bring at least a piece of paper and a pencil to take attendance with, and arrive 15 minutes early to the classroom (if possible) so earlier students know they're in the right place, and can introduce themselves if they want. That you're even here means you're more than qualified, so don't stress it. Relax, and the rest will come.

# Grading

Everyone has to grade homework, tests, or both. (At the time of writing, undergraduate TAs do not grade exams for 40 or 170.) Classes vary widely on how grading works, but the idea is the same. Some classes will work on a specific website, some on a shared Google document, and some will print out the provided work and grade on paper. Some classes grade together in a single sitting, and others are completed individually. Rest assured the details will be given to you by the professor.

Generally rubrics are provided. If you feel a rubric does not capture points that a student deserves, never feel afraid to ask a more experienced TA. Depending on the class, extra credit may be given out frequently. But make sure to confirm with the group so that you can have consistency.

Do leave notes on why you are deducting points–ideally they will learn in the future! I make it a personal rule never to deduct a point without leaving a note about it, even if the rubric makes it clear anyway.

Depending on whether the class allows selection of students, try to stick to the same group, or students you recognize from having spent time working with them (NOT friends!). That way, you can understand their individual struggles and progress. Best not to mention who grades a student outside of class, though. (There have been talks of increasing the anonymity of assignments, so this may not be possible in the future. It's not a huge biggie either way, though.)

It's a balance of thoroughness and time. With hundreds of students, it's not always possible to spend half an hour pouring over every line of code. However, always try to give students a fair readthrough.

The first few students will always be slowest to grade. If you're grading on a computer, keep common responses in a notepad for copy-pasting, or if you're technologically savvy you can add macros. Comp 40 and 105 do this automatically.

And remember when it comes to grading: fairness über alles, and when in doubt, always ask another TA. We aim for consistency. Nobody should lose points just because they had a stricter TA.

## How to prepare for the first week:

If you did this assignment, look at it and read your feedback. Familiarity with the homework itself will also help, although at this point you should already know the gist of it from office hours. As you grade the first few submissions, look for common mistakes and indicators that the student did or did not understand the concepts. Always give the whole submission a fair scan-through, but having some concrete points to look for can make grading easier and more objective.

## Piazza

Almost every CS class at Tufts has a Piazza page[3] (though at the time of writing, 61 and 170 do not). While it's not necessary that you answer questions, you should absolutely read every question and response. They will often be the same ones that come up in office hours, and can better your understanding of the homework.

If you do answer questions, try to avoid giving away solutions. Merely clarify what the question is asking or perhaps give an explanation of the topic, not the answer. If a student presses further, encourage them to come into office hours.

If you see a question that has code from the student's solution or that gives away too much of the answer, mark it as private. You can leave a friendly note saying "I'm going to mark this private in case this gives too much away." An instructor or more senior TA may change it back, but no harm has been done.

On the flip side, if someone asks a good question that would be helpful to the class but makes it private, write something along the lines of "This sounds like a good question to share with the class. Would you mind making it public?" Leave it up to them to decide, though.

The sociology of these Piazza pages is fascinating; every class seems to have its own atmosphere and set of expectations. If you're not super familiar with the style of answers in your class, maybe wait a bit and read how much of a response from TAs is expected. In some classes students are much more likely to answer each other's questions, and in that case only step in if there's a question nobody can answer, or an administrative question. In some classes, giving anything other than the broadest hints is discouraged, so watch out, and follow the instructor's lead.

## Denouement

In your hands lies a holy grail of information. The words on these pages represent dozens of hours you will be spending over the next year. If it seems like a lot, don't worry about it: these are all merely observations my colleagues and I have collected in our time as teaching assistants, and I'm certain you'll get the hang of things as well. This section merely serves as a guide so you aren't as in the dark about what on Earth you will actually be doing. Bon chance, mon ami!

[3]Note for TAs of Comp 40: Tread carefully. Your professor will probably give you a talk about this as well, but it merits mentioning: a large part of Comp 40 is careful reading of the spec. If a student poses a question that could be answered upon a careful reading of the assignment, don't give it away. Comp 40 is fairly unique in that regard.

# Chapter 2

# Resources Available to a UTA

## Introduction

A UTA is expected to be reasonably knowledgeable of the subject matter of the course which they assist; that being said, no UTA can be expected to know it all from the get-go. Especially as a new TA, and often as an experienced one, we need to refer to resources beyond ourselves to learn about subject matter, help resolve a conflict, or simply confirm a hunch. This section will highlight the resources you have available to you, and some that you can share with students, to aid you in being a super UTA.

## Learning Objectives

By the end of this section (and occasional reference back to it) you will know:

- Why external resources can be useful for a teacher or educator, and when it's worth it to seek guidance from said resources.
- Which people/groups of people you can approach for help with questions related to course material, conflict resolution, course information, grading help, or moral dilemmas.
- What written resources you can refer to when unsure about course material, rules or general TA guidelines.
- Where you can find the aforementioned resources.

## What Resources to Find Where

You know why resources are useful, and know when to refer to them. This section tells you where to find them. Resources are broken down by the type of problem you're having.

### Subject matter

No matter how experienced a UTA, there are always times we come across a weird thing with pointers we are unsure about, or simply don't recall how many bits are in a short. The first often most effective option here is a quick Google (Bing is unacceptable). This tends to resolve most little slips in memory, or syntax related questions. There are a few specific websites to look out for (which you are likely aware of, having taken CS courses) like Stack Overflow and Tutorials Point. It is a good idea to point students toward such online resources as well, when appropriate. Like through a Piazza post. It helps enable students to do similar searches themselves in the future. This is particularly important in introductory classes.

If the issue is conceptual or a google search wasn't helpful enough, there are multiple options on how to proceed. If there's another TA from the course around, ask whether they are comfortable with what you're struggling with, they may be able to do a quick explanation (or a long one). Else the textbook for the course is a good place to look (assuming you have access to one).

If the problem is really a toughie and you're still not clear, decide whether you need a quick answer or whether this can wait. If it isn't urgent, post the question to other instructors on Piazza; if you need a quick response it makes sense to approach either one of the graduate TA's or the professor of the course to help clear this up.

### Moral Dilemmas

Often the guidance we need isn't as straightforward as course material issues. Moral dilemmas aren't uncommon for any TA and we can't hope to not get stuck in a situation where need moral guidance at some point. What's most important here is that you know that there are people to reach out to, and that you do reach out whenever you feel the need to.

Although this is a persona; decision, and will vary depending on whom you feel most comfortable discussing such things with, there are people within the department who you can work through such a in issue with you.

The ethics and inclusivity sections of this handbook could be particularly useful as written resources. They would be a good first option. Of course one often

feel the need to talk through a moral dilemma rather than just read, given how specific to the situation such issues can be. A fellow TA who you are close to or comfortable chatting with could be the easiest to reach and feel at ease with.

Another alternative may be to come in to the TA office hours/discussion groups that will hopefully start soon. This would allow you to talk over your situation in person with at least one other experienced TA.

Depending on your relationship with the professor of the course, you can choose to approach them, or another professor in the department whom you have a relationship with.

Of course this is only to highlight that you do have options within the department itself. If there's a friend or family member that you usually discuss such things with, that could still be a great idea. As long as you do reach out when you feel the need to.

Another resources available to you is Tufts Counseling and Mental Health Services (CMHS). If you're particularly stressed about a moral conflict, or feeling particularly worried/upset about something you can always make an appointment to meet with a counselor through CMHS. You do NOT need to be diagnosed with a mental health illness in order to talk to a counselor. Sometimes you just need somebody to talk to, and nobody you usually chat with fits the bill. It is in times like these that talking to someone may be most helpful.

## Guidelines

There aren't all that many guidelines to follow as a UTA so far. Most resources for guidelines can be found in the course specific sections toward the end of this chapter. The university policy for academic integrity may be helpful too. (https://students.tufts.edu/student-affairs/student-life-policies/academic-integrity-policy) Referring to this link may be particularly helpful when unsure about what counts as academic dishonesty or plaigarism. There is even an entire section dedicated to Computer Ethics. This is, however, a page filled with a lot of content which may be hard to parse when looking for answers to very specific questions.

Academic honesty is a case where approaching the professor of the course directly may be the best option depending on the severity of the offense. Since Tufts and the department take such issues quite seriously it may be worth sharing whatever you have found or believe with the professor who can then take the matter forward. You should not be directly concerned with any disciplinary action (apart from something the professor may ask you to do).

Of course, there are cases where whatever you have found may not seem serious enough to bring to the attention of the professor, like if you see a harmless piece of code placed in a Piazza post. In such a case where the offense is clearly

quite trivial, if you are unsure on how to proceed it may be worth chatting with a fellow TA or a grad TA. Sometimes it is fine to just remind the student of the rules and carry on. Do make sure that you know what is excusable in this manner and what isn't though. If you are unsure, it is worth clearing up with the professor of the course.

Here again, for the most part, fellow TA's, more experienced TAs within the department, teaching fellows, grad TAs, and the professor(s) would be appropriate people to approach.

## Teaching Methods

No matter how super we may be as teachers, there's always room for improvement. Working toward developing and growing as a teacher is a large part of the UTA experience that deserves adequate attention. So if you have that drive to develop your teaching skills, where can you go?

This handbook: You're in the right place right now, because the rest of this handbook has loads of helpful little tips and explanations of teaching methods.

TA Office Hours/Discussion group: Come chat with other UTAs about what's been effective, what works what doesn't. Experience is a great teacher, shared experiences save you the trouble of going through them yourself.

Google: 28 million results for how to become a great teacher, here are plenty of resources and more. Here's a couple that were interesting from the first page itself:

- 25 pointers for all kinds of teachers: https://www.teachthought.com/pedagogy/25-things-successful-teachers-do-differently
- A great TED talk on the matter: https://www.huffingtonpost.com/dr-mariappan-jawaharlal/be-an-original-be-yoursel_b_10708882.html
- An entertaining/motivating poem about teaching: https://www.youtube.com/watch?v=8I_JK6tT

Of course, narrowing the vast swaths of content down to CS-related pedagogy is important. Going through the computer science related material on Khan Academy is one good place to start: https://www.khanacademy.org/ (Most of the material is quite basic, but the teaching methods and format of explanations are the things to watch out for.)

If you are curious about how to teach specific material it is worth doing a google search for a tutorial on the specific content you want to be able to teach better. For example, toward the end of last semester I needed to explain friend classes and wasn't sure how. A quick search for "friend class c++ tutorial" lead me to this tutorial which was great refresher for me as well as an example on how to explain the concept with examples included.

The chapter specific resources at the end of the Pedagogy, Recitation Skills, Mental Models, and Interpersonal Skills chapters of this handbook have some

super links and material that are worth checking out if you're interested in reading beyond what is already mentioned in the chapters.

### Misc

Sometimes you need a resources but the problem you're facing doesn't fit into any of the categories above. In such cases it is a good idea to go through the list of resources at the bottom of the section to see whether any of the options seems just right for you right now. Below are a few people that you may consider approaching or referring students to.

College Transition Advisors: Particularly for freshman that are confused about their college career. You do not have to play the role of college counselor, although you may feel free to do so if you like. There are resources provided by the university, however, to help students who feel lost in this manner. CTAs are a good start.

Professors: If you are unsure about details of grading, coursework or something to do with the department as whole, approaching either your course professor or another professor in the department with whom you have a relationship can help you gain some clarity. Most professors are quite open to meeting and having discussions with students from the department, they are often even more open to hearing from TAs.

Advisers: CS major advisers are particularly helpful when a student has questions about the department as a whole. You can suggest they discuss their plans for a major either with their current adviser or a professor who they take a class with who may be willing to discuss what the CS department is like and give some insight about the courses. Of course, you as a TA may be able to give a fair bit of insight yourself, but it's always useful for students to get advice from the professors in the department.

Student Services Center: The Student Services Center can be greatly helpful with a wide variety of academic and general student life related questions. They would be a good resources to consider referring students to if they have questions about study abroad and potential to study CS at other schools or more broad questions about campus life.

## Case

If you are ever doubtful of any information you are sharing with a student, then it is likely a good idea to refer to a resource first. As UTA's we have the responsibility of not only being knowledgeable about course matter, but to correctly guide a student toward the correct answers even when we are unsure ourselves. An example will probably narrow this rather broad idea:

Last semester, I was working with two students on one of the homework assignments for COMP11 when they had a question about whether they could return an integer value to represent the grade of a student in a subsection of one of the exercises. I was unsure of the answer here, but felt like it would be fine to just do integer division and not return the exact answer. There were other students waiting and I wasn't going to go running around for an answer. Seemed pretty trivial.

As you may imagine, this wasn't a great decision on my part. I ended up being incorrect and costing a few students (the two I worked with and a couple others they spoke to) 2-3 points off their homework.

The alternative here would have been to first read through the spec again and see if that answered anything, then ask any of the other TAs that may have been on duty whether they knew, if not, then quickly check Piazza, and let the students know that I would get back to them about the answer if it wasn't already answered on Piazza. I could later ask one of the teaching fellows or grad TAs or the professor of the course since at least one of them would be likely to be around Halligan. This wouldn't have taken too much effort to resolve, and although seemingly a small matter I caused students to lose points and possibly caused them to be more weary of listening to what TAs have to say.

Of course there are an innumerable amount of other examples where choosing to access a resource is the better option for everyone involved, which I'm sure you know of. The point of this case is just to emphasize when to choose to access a resource, and the answer is whenever you are at all doubtful of information you are about to share with a student that may affect their understanding of concepts in the course or their grade.

## List of Resources

There are many, many resources to refer to, only a subsection of which have been discussed in this chapter. Below is a list of what has been discussed in this chapter plus a few more. Do go through it if you just need a reminder of what's available to you, or if you want to know your options so you can choose your next steps.

### Individuals

- Fellow TAs
- Teaching fellows
- Graduate TAs
- Course professor(s)
- Major Advisor
- Friends/family

- Other professors
- Counselor
- TA discussion groups
- Student Services Center
- College Transition Advisers
- Academic Deans

## Documents

- This handbook
- Tufts Academic Integrity Policy

## Links

- Khan Academy website
- Tufts Counseling and Mental Health Services - CMHS
- Stack Overflow: Great for any material related unclarities.
- Tutorials Point: Super explanations and info on functions and libraries specially for C/C++
- Course website for the course you are TAing

# Chapter 3

# You, TAing, and Ethics

## Introduction

### Learning Objectives:

By the end of this chapter, you should be able to:

1. Identify an ethical situation.
2. Use an inquiry model to analyze your options and how they affect others.
3. Have had practice working through difficult ethical situations and applying models to said situations.
4. Make meaningful choices that balance professionalism and empathy.

### So what do you mean by ethics?

This section will not be dealing with ethics as an academic field– rather, we're going to talk about ethics and the practical ways in which they relate to your day-to-day as a TA. A good rule of thumb for determining whether you're in an ethical dilemma is whether or not you feel conflicted about what to do, or if you feel like the "right thing" is up to interpretation. There are, of course, professional guidelines to these sorts of situations, but you also have a duty to be empathetic toward your students. That being said, it's okay to feel uncertain, as there isn't one right answer to every ethical situation.

### So I can skip this section?

No. I understand that this section will be, by its nature, less tangibly relevant than the others when you are first starting out. Trust me, it will be of dire

relevance. The purpose of this chapter is to give a framework in which you can evaluate complex issues that you may face as an undergraduate student working with other undergraduate students. These are issues that you may not feel comfortable discussing with your coworkers or your professors, so it's up to you to be prepared to know how to respond.

## For First Time TAs:

One issue common to new TAs is balancing friendship with professionalism. It's entirely possible that some of your friends are also your students this semester, or that you will grow to become friends with some of your students. That's great. What's not great is when they message you at 1AM the night before an assignment is due going on about how they're panicking and could-you-just and you also have a midterm the following day so you really can't and it's not your job but they're your friend and AHHHHHH. It's important to set clear boundaries with your friends from the beginning of the semester. Make sure they know that, while you're glad to help them like you would any other student, you have to keep that part of your relationship professional. You can help them during office hours, but outside of office hours, their understanding of the material isn't your responsibility.

On the topic of friends, it is also relevant to disclose any close personal relationships you may have with students to the professor, so that the professor can decide how to handle minutiae such as whether you should be grading their assignments or leading

## On fraternization[1] with students:

Don't. Being someone's TA creates an inherent imbalance of power between the two of you, and that makes consent tricky. If you are in the circumstance that someone with whom you are already involved becomes your student, you should probably disclose that information to your professor (of course keeping it anonymous to respect your partner's privacy). Your professor can decide how best to proceed.

[1] Does this mean only romantically? Because I'm pretty sure "fraternization" also covers just being friends, which I hope is allowed.

# Skills and strategies

## Divide and Conquer: How to break down an ethical dilemma

The mark of an ethical situation is that it leads to internal conflict. This conflict can be overwhelming to think about, let alone make rational decisions about. Because of this, we use thought models to organize and understand what all is at play.

## The AIR Model of Ethical Inquiry

1. (A): Awareness

- Create a safe context for ethical inquiry and discussion. Make sure you are in a context where you can speak honestly and freely without judgement.
- Classify the type of situation you are in. What are the associated values? Honesty? Professionalism?
- Be aware. Ask yourself questions about the situation. What assumptions are you making? Who will be affected by your decision?

1. (I): Investigation

- Consider one possible course of action. How would this prioritize the relevant values? How would it affect the stakeholders in the situation?

1. (R): Practical Action

- Having thought through several courses of action, you're ready to choose one and act on it. Go in with a plan and try to anticipate how it will play out.
- If things go differently than expected, be willing to adjust your course of action. However, make sure you give adequate thought to your new course of action.

## Syracuse Model of Ethical Decision-Making

1. Identify the problem.
2. Identify the potential issues involved.
3. Review relevant ethical guidelines.
4. Know relevant laws and regulations.
5. Obtain Consultation.
6. Consider possible and probable courses of action.
7. List the consequences of the probable courses of action.
8. Decide on what appears to be the best course of action.

The Syracuse Model is in many respects similar to the AIR model, but it also raises the important point that one should know relevant laws and regulations.

### There's Always Another Option

If, after examining a situation in the above two frameworks, you have only come up with one possible course of action, pause, and think through it again. Though each situation may have one "best" course of action from your perspective, it's important to thoroughly think through several, weigh them against each other, and then decide which one is really the best. It will oftentimes not be the first that comes to mind.

### I'm stuck and overwhelmed and I don't know what to do

Consider who else you could ask for input on this situation. You're welcome to discuss tough decisions with your fellow TAs, just be mindful of keeping the situation adequately anonymized. Other people to whom you can reach out for help include the course Teaching Fellow (if it has one), graduate TAs, and the professor(s).

### Balancing Empathy and Professionalism

Your job is to assist students in the manner that is specified by the professor. You are required to follow the guidelines set forth by the professor. That being said, this does not bar you from valuing empathy and understanding when making decisions. For example, if a professor requires you report a certain type of behavior, you should do so. This doesn't bar you from keeping said report anonymized and explaining the context for the behavior when reporting it.

## Applications: Three Case Studies

In this section we will explore three situations using the AIR Model of Ethical Inquiry as a reference for what the process looks like. These are not meant to be read as course policy, but rather as a framework of decision making that you can apply in your TAing.

We have separated the situations from our analysis of them, as to give you the opportunity to use these as an exercise and compare your results.

## SITUATION 1: Friends in Office Hours

You host office hours once a week for two hours on Wednesday night, the night before homework is due. You share this time block with another TA, Balthazor, who has a reputation for using a lot of technical vocabulary and confusing students. Usually this block is pretty busy, so you and Balthazor try to help as many students as possible, but now and then you won't be able to get to someone before it's time to leave.

A friend of yours, Zagdar, is in your class. For the first time this semester, she comes to your office hours, panicked, because she doesn't think she'll finish the assignment. As you start to help her, you see that she has only just started working on it, and it's a particularly long assignment. You spend half an hour with Zagdar, helping her get started, but there's a line of other students forming, waiting for help. What do you do?

## SITUATION 2: Academic Honesty

It's a month into fall semester. You're off-duty and working on a project in Halligan, you notice two first-year students from your class who are looking over each other's code. You know that this is against course policy to look at other students' code, and by this point in the semester, they should as well. Listening to them, you can tell they're not copying each other's answers, but "checking if this looks right". What do you do?

### CASE 1: Friends in Office Hours

Let's break it down:

- Your friend needs your help, but as a TA, it's your job to treat all of your students equally.
- Values at play include loyalty, professionalism, fairness, etc.
- Relevant laws and regulations? None really
- Who can you talk to? Your fellow TA on duty, but aside from that, you have to make a snap decision
- There are many potential courses of action, but here are two:
- Continue to help your friend for the rest of your office hours
  - She may make significant progress, but may or may not finish
  - Other students who need your help won't get it
  - Other students will see you giving preferential treatment and may feel frustrated
- Move on to the next student, let your friend wait in the queue:
  - She may or may not finish the assignment
  - Other students who need your help will get it

  - It's possible that she will blame you for not helping her when she needed it
- Choose some action/combination of actions to take. Here's an example:
- You choose to help other students, and you make your friend wait on the queue just like any other student. She does not finish the assignment, and she believes you would had you spent more time helping her. You explain your professional obligations to her, and hope that she'll be understanding.

**CASE 2: Academic Honesty**

Let's break it down:

- Students are engaging in behavior that is technically considered cheating, but don't necessarily understand the implications of their behavior.
- Values at play include honesty, academic honesty, professionalism, compassion, etc.
- Relevant laws and regulations?

Institutions of higher education are required to report academic honesty violations

- Who can you talk to? Consult with your peer TAs, and with your professor (keeping the behavior anonymized)
- There are many potential courses of action, but here are five:
- Do nothing
  - They get away with it, and continue to make this sort of violation over the course of the rest of the semester
  - Other students hear that you're lax about enforcing this, and start to behave the same way.
- Report the students for their violation
  - They could get severely punished/expelled for what may have been an honest misunderstanding of the rules. However, it is students' responsibility to understand the rules
- Explain to the students why their behavior isn't acceptable, and let it slide this time
  - They could learn from this and avoid the dishonest behavior in the future
  - They could also learn how to better disguise their dishonest behavior, and continue doing it
- Make an example of the students in front of the class, publically shaming them for their mistake
  - This would embarrass the students. They may even drop the class, and their grade will no longer be a reflection of dishonesty if they aren't getting a grade at all
  - This will likely form a permanent stain on your rapport with students at large
- Give the students a zero on this assignment

  - Students are likely to take a lesson to heart when it affects their GPA
- Choose some action/combination of actions to take. Here's an example of a combination of actions you could take:
  - After speaking with the professor, she decides to give the students a zero on this assignment. She asks you to privately explain to them that this behavior is unacceptable, and then she publically reiterates course policy to the rest of the class to prevent the behavior from repeating.

## More Info

Laws and Regulations:

- FERPA
- https://en.wikipedia.org/wiki/Family_Educational_Rights_and_Privacy_Act
- https://www2.ed.gov/policy/gen/guid/fpco/ferpa/students.html

# Chapter 4

# Inclusivity

## Introduction

Inclusivity is the intentional focus on the needs of individuals in an effort to create an atmosphere that is supportive of diversity. In education, it is crucial to creating a learning environment that enables all students to learn and perform well. As TAs, we have the direct ability to shape a course culture, and overall the culture of CS at Tufts. Given our role, we have an imperative to create a space where all students at Tufts have equal opportunity to learn and engage with computer science.

This chapter covers what it means to be an inclusive TA, as well as some skills and strategies. It certainly won't be able to cover everything, so at the end there are many resources to help guide you when you're not sure of what to do.

## A First Approach

This chapter isn't going to do the work for you on becoming an inclusive TA- it will take time and effort and a conscious decision on your behalf. But I'm hopeful that because you understand that an inclusive environment is paramount to student learning you will want to grow and change, so this framework is to help you get started.

1. Examine your own background and your current mindset. Take a bit of time and put away technology and think about your experience. Where this goes will be different for everyone, but you can start with things like what your school was like growing up, whether or not you think your teachers like you, and what it's been like working on group projects. Then, think about what things might be different if you attended a different

school. What would it be like to go to school where you feel safe, or where you do not? What if you attended a university without your favorite/most influential professor?
2. Be aware and mindful of when in the classroom, both as a TA and as a student. You don't need to pass judgement on every thought you have- that's not only energy consuming and impractical. Rather, try and notice the thoughts you have and let them come and go freely (when you're not paying attention to lecturing, that is).
3. After being aware of yourself for a while, focus your attention on those around you. Notice the people who come from a different background as yourself as well as those with similar experiences. How much do people participate in discussions? How confident do students seem about their answers? What seems different than how you would react to a particular question or situation?

After doing this, repeat whatever you think will be helpful. Additionally, during this process it is also important to hear the perspectives of others. If someone is giving you insight into their experience as a student or TA, listen. Do your own research into inclusivity, feel free to start with the resources below.

## Skills and Strategies

This chapter cannot be an exhaustive resource on becoming an inclusive TA- nor does it strive to be. Becoming more inclusive is not a task with a discrete goal, but is more like a state of awareness. No one can be perfect, we just have to change our behavior and thoughts as we can. These skills and strategies are only meant as basic tools to get you started and to handle situations commonly seen within computer science at Tufts.

### Mindset

Being inclusive starts with the right mindset — in order to be inclusive as a TA, you must approach your students and colleagues with empathy.

- To show empathy, you must put aside your own viewpoint and validate the perspectives of those you interact with.
- It can take a lot of effort, but examining your own attitude is a key first step in being empathetic.
- If you're struggling to find your own biases through reflection, the Harvard Implicit Biases Tests found in the resource section can be particularly helpful and insightful.

## Active Listening

It's impossible to empathize with a person if you do not hear what they say.

- Wait a few seconds before responding to make sure a student has finished talking. Often they will add more interesting thoughts.
- Reflect back what a student has said to make sure that your understanding matches with what they are trying to express.
- Often as TAs it is our job to give advice and, well, to teach. It's what we're hired for. But sometimes a students just want to be heard. If a student mentions how hard they've been working all day, don't automatically follow up with five new suggestions. Hear what they are saying and validate their emotions in this. If you are unsure if a student wants help, you can then ask if they have questions or want suggestions, but do not give unwarranted advice. It can come off as unempathetic and callous.

## Questions

In order to better understand where your students are coming from, ask them questions.

- Some questions are necessary to start a relationship with a student, such as asking their name, preferred pronouns, and what they enjoy outside of the classroom. If students seem shy or hesitant, offering up your name and other information first might be a way to "break the ice". These are crucial at the start of the semester to setting up a positive dynamic.
- Make sure that the questions you ask are based on getting to know that specific student better and do not reduce them to one facet of their identity. Asking students about their experience as a `_____` could make them feel as if you only see them for that label, if you have not already established a deep relationship.
- Stick to topics that students will feel comfortable talking about. If you are unsure, give open questions that allow for students to answer in different ways. This flexibility can let the student steer the conversation if you've touched upon something that might be sensitive.

## Conversational Awareness

Any interaction you have with a student should be a conversation. Even if you're lecturing, students will be contributing to the conversation with body language and hopefully questions.

- If you and a student are having a dialogue, make sure that you are not dominating the discussion and are giving the student space to think and respond. Keep in mind that some students are often more-talked over than

others (students who are women, people of color, and/or disabled), so as a TA it is your job to make sure they feel like an equal participant in the conversation. A good strategy to ensure this is to wait 2 or 3 seconds after a student finishes speaking to respond, and to respond thoughtfully.
- Even if your words are all kind, useful, and inclusive, you could still create a hostile learning environment with your body language. Be aware of the space around you, and make sure that you are not "getting in someone's bubble". Whenever possible, meet students on their physical level, rather than talking down to them.

### Cultural Understanding

Each person has a cultural background, whether you are aware of it or not.

- If your personal cultural background is not something that you have had to think about before, think if your language and social circles matched your neighbors, or what's in mainstream culture. Keep in mind that for many of your students, this is not the case. Use universal analogies, or tailor it to a student's particular interest, if applicable.
- As a follow up, when giving examples that involve example people, make sure to not only describe people who happen to look like yourself. This is an innate tendency for teachers to have, but if a student cannot even see themselves within an example situation in CS it will be more difficult for them to envision their place within the classroom.
- Natural language ability does not reflect a student's ability to perform well in the course- give synonyms for particularly obscure and new words. Be redundant in your speaking if possible and phrase the same concept in a variety of ways.

## Computer Science Specific Ideas

- Studying CS is difficult, but what students find challenging is subjective. Do not invalidate students by describing problems as easy or trivial.
- You will see a variety of emotional reactions to a course. Be ready to recognize students who are frustrated, discouraged, and overwhelmed, and as a TA part of your job is to empathize. It is only from a position of mutual trust that you will be able to motivate a student and guide them back onto a healthy direction. That said, you should not be a student's only support system. In ambiguous or extreme circumstances, consult the resources below.
- Be open and welcoming to student questions about computer science and your experience, classes, and being a TA

## Examples

1. A student switches into your lab, and you strike up a conversation after class to get to know them, as you've never met before. This student says they have switched labs because they felt uncomfortable in their last section, but does not describe further. How would you make the student feel more comfortable and able to learn in your lab, and how would you address the potentially problematic environment?
2. You notice a student who comes to office hours when you work, along with a few other TAs. This student only asks questions to TAs who are male, and was rude one time to a TA who is female by interrupting her often when she tried to help the student. Other than that, things go smoothly when the student can work with whatever TA he feels comfortable with. Do you address this situation? If so, how?
3. You have a student in your lab that prefers to go by a different name than is on the class roster. You and your fellow lab TA have been using this roster to help learn student names, and you notice that your fellow lab TA often mistakenly calls out the roster name for attendance, or uses it when calling upon the student. How do you talk about this subject with the other TA?

## Resources

For general questions and concerns

- Graduate TAs/other TAs that you are comfortable with
- The course professor(s)
- Other members of faculty that you have a relationship with (advisors, former professors)
- how to learn from other TAs by watching

For self-education on inclusivity

- Strongly recommended (Harvard's implicit Bias Tests)
- More on creating an inclusive learning environment (University of Michigan)
- Building Inclusive Classrooms (Cornell Center for Teaching Innovation)
- Specific ways to be inclusive within CS (Stanford)

Institutional Resources

- Tufts Counseling and Mental Health
- Tufts Office of Equal Opportunity
- for getting more information on policies at Tufts, and for escalating situations beyond your control

# Chapter 5

# Mental Models

## Introduction

One of the important parts of a TA's is to help students build and improve their mental models. What is a mental model? Concisely stated, it is an explanation for why a person thinks a certain way. Examples specific to computer science include a student's internal explanation for variable assignment, pointer dereferencing, or recursion.

A model is either viable or nonviable. A viable mental model is one that proves sufficient when tested. In other words, while not necessarily completely correct, it is correct enough to get the job done. A nonviable model is one that may be internally consistent, but leads a student to incorrect conclusions. An example of a nonviable model for recursion is the idea that a recursive function behaves similarly to a loop. While this is not entirely unreasonable, it will not lead a student to success.

Computer science has a couple unique challenges regarding mental models. First, it is one of the few disciplines that collegiate students typically encounter with a blank slate. In other words, most college students in introductory computer science courses have never programmed before. Most other disciplines taught in college are scaffolded on top of years of primary education; for instance, the mental model for calculus is built a top the model for precalculus and algebra. Computer science is deeply different from this. Second, the functional correctness of a program provides immediate verification of a mental model. A compiler error informs a student that their mental model is not entirely correct. While valuable, this can be deeply frustrating and can be barrier to learning.

### Learning Objectives.

After completing this module, you will:

1. Have a working knowledge about what a mental model is and how it relates to computer science.
2. Be prepared to incorporate this into your teaching.

In order to accomplish this, we will supply you with several concrete techniques.

1. You will learn how to walk through a student's thinking in order to understand where they went astray.
2. You will learn best practices for constructing mental models from scratch.
3. You will learn strategies for adjusting a student's malformed mental model.

Finally, we will tie these together in an several case studies where you will be able to test you knowledge.

## Background Knowledge

Constructivism is a modern education paradigm. It is rooted in the idea that student's learning is scaffolded on top of past knowledge. In other words, all new concepts must be connected to past concepts in order for a student to effectively learn them. Another core idea is active learning. In the eyes of constructivism, a student needs to work through difficult concepts and "construct" new knowledge themselves. Only a student can do the hard work of connecting new knowledge to old.

Passive Learning is another education paradigm that stands in opposition to constructivism. It states that students are like empty sponges or blank slates upon which instructors can inject information. This is occasionally effective for surface level learning, the type of learning that is sufficient to do well on an exam. However, it rarely successfully contributes to deep learning, which is required for true mastery.

This handbook is written unashamedly from a constructivist perspective.

## Skills/Strategies

### Assessing a Student's Mental Model

When you start working with a student, you should first assess the student's current mental model. This will inform the guidance that you give the student.

Here are a few techniques.

### Step Through a Program

Generally, a student will come to you with something concrete that they do not understand. Perhaps this is an example from recitation, a piece of code that the student wrote, or a proof. If you can, try to locate the part of the code where the student went ary. In order to save time, it makes sense to focus on this part.

At this point, you should ask the student to mechanistically step through their logic on a whiteboard. Have the student draw a picture representing the current state and walk through the program, step by step adjusting the state. At every nontrivial step, you should have the student explain why they think that makes sense. For proofs and other abstract concept, it is harder to draw a picture. In this case, focus on justifications that a student gives to get from one argument to another.

### Figure Out What a Student Already Knows

Generally a student will have absorbed something from reading, lecture, or recitation; hopefully talking to you is not their first time encountering material. Even if it is, there is certainly material that they have learned earlier in the course that provides a relevant scaffolding for the new material that the student is struggling with.

You actively determine what the student already knows. Ask questions about relevant past assignments or what the professor has been talking about in class recently. Nail down the line between what the student is comfortable with and what they are starting to learn. This will provide both of you with a good place to start instruction.

## Building New Mental Model

Sometimes a student will approach you and ask you to teach them something for the first time. Maybe a homework has gotten ahead of lecture or a student missed an important concept in lecture. Teaching new information can be quite challenging to do. Regardless, it is your job to rise to the challenge and help this student form an accurate mental model.

Here are some strategies to help with this.

### Solidify Language

When introducing a new concept, its very important to also formalize the language used to talk about that concept. As an expert, it is easy to assume that the names that you use to talk about a concept are self explanatory or already known to the student. In general, this is simply not the case. Formalizing

language allows a student to better ask questions and collaborate with their peers. It gives them a basis to effectively ask other TAs for help. Moreover, knowing what to call things is frequently half the battle to understanding them.

Make sure that the language that you use is shared by the professor and other course staff. It is incredibly frustrating to learn the same concept by three different names.

#### Connect to Preexisting Knowledge

Generally, a new concept will not exist in a complete vacuum. Even in intro courses, the student has generally been to lecture a few times and has seem some material there. By connecting new material to old, you make things easier for a student to understand. Moreover, you will also reinforce old material and boost a student's confidence.

Its advisable to start a session by figuring out the most recent material that a student is comfortable with. This gives both of a you a good jumping off point to start teaching.

#### Concrete Didactic Examples

A didactic example is an example intended for teaching. Generally these isolate the key concepts and are designed to allow a student to have a moment of realization. It's a good idea to have a few didactic examples up your sleeve for current course material. This is important because thinking these up on the spot can be hard.

Didactic examples have a couple advantages over helping a student directly on the assignment. First, because they are simple it's possible for a student to completely grasp them over the course of a single session. Second, because they are not assigned homework, you can help a student reach understanding without worrying about giving the student the answer or violating academic integrity.

When actually working through one of these examples, it's important to let the student think for themselves. Do not narrate them through the example. In many ways, this is the natural response. However, it is one of the least helpful things to the student. Allow them to fail and focus on asking questions to keep them on track.

### Adjusting Incorrect Models

One situation that arises is a student will be convinced that their incorrect mental model is in fact correct. Perhaps they've misinterpreted some lecture notes or even had some success with the incorrect model. However, you have

identified their model as nonviable. How do you then convince the student that they are wrong?

Here are some techniques.

### Tell the Student They're Wrong

The easiest thing to do is to just tell the student that they are misunderstanding something crucial. The most direct approach is the simple, however it can also be very risky. First, if you are too harsh, you risk damaging the student's confidence. Confidence in one's ability to understand is crucial for building more understanding. A TA brutally putting you down is very hurtful and damaging to this. Second, you run the risk of seeding more misunderstandings. If you state something without really supporting, you risk the student believing you but doing so for the wrong reason.

If you decide that this is the best approach, it's important follow up your explanation with many questions in order to ensure that they believe you for the correct reasons. In order to maintain confidence, it's also important to state what parts of the students understanding were correct.

### Getting Student to Recognize Inconsistency

In an ideal world, you will get the student to realize that they are wrong without you ever explicitly telling them so. A major advantage of this is that you set the student up to debug their own thought process in the future without being reliant on TAs.

A good approach is to focus on a very targeted didactic (teaching) example. Ideally, it will make the students misunderstanding into a blatant, logical inconsistency that they can't help but realize. You then use this example and careful questioning to guide the student to reassess their assumptions.

### Addressing Trial and Error

The ability to run a program is a powerful tool. It can immediately validate if a student has a working solution. Unfortunately, this can lead to a deeply negative pattern: trial and error.

We've all been there. Make some changes, see if it works. Make some more changes, see if it works. Get frustrated and repeat. Eventually you stop thinking about your changes and just keep making them until something sticks. This is not a viable way of building a mental model and it is certainly not active learning. A student is not reasoning about the things that they have learned and trying to apply it. Instead, they are just trying stuff and seeing if it works.

Unfortunately, early on in intro courses, this is often a good enough solution. The problems are basic enough that trial and error can get a student to a passable solution. This builds bad habits that are not sustainable in harder courses, where trial and error simply leads to frustration and wasted time.

When you're helping students, try to identify if they are using trial and error to brute force their way to a solution. Ask them questions about why they made a change or get to them to justify a certain decision. Do not give them the answer. Make them reason their way to it. Try to demonstrate correct thought processes for reasoning about problems so that they can adopt these strategies themselves.

# Application

## Case Study 1: Variable Names vs Variable Contents

A student is confused about the following example from lecture.

```
int sum(int &a, int &b) {
  int nine = a;
  int ten  = b;
  return nine + ten;
}
```

They claim that the function will always return 19. It seems pretty clear that the student is conflating how variables are named and what they contain. How would you go about correcting this student's mental model?

## Case Study 2: Variable Assignment

A student is assigned to write a simple swap function that exchanges the values of two references.

```
/* Correct code */
void swap(int &a, int &b) {
  int temp = a;
  a = b;
  b = temp;
}

/* Student's Code *.
void swap(int &a, int &b) {
  a = b;
  b = a;
}
```

The student thinks that their code isn't working because they don't know what to return from the function. They seem unclear what a reference is and have no idea what the ampersand means, and admits that they have missed some class. How would you go about teaching references?

### Case Study 3: Control Flow

A student comes to you having written the following code snippet.

```
void foo(int size) {
  if (size == 10)
    cout << "print" << " ";
  for (size = 0; size++; size < 12)
    cout << size << " ";
}
```

They expect the foo(4) to print `0123456789print1011`. However, the correct output is `01234567891011`. You're not really sure what is causing this incorrect analysis. How would you go about finding that out?

### Case Study 4: Order of Events

A student comes to you having written the following code snippet.

```
int main() {
  int width, height;
  int area = width * height;
  cin >> width;
  cin >> height;
  cout << area;
}
```

The student expects the program to read in two numbers and return the product of those two numbers. However, the program does not compile. The student has no idea why the program isn't working, and seems pretty frustrated. They've been trying bunch of things and nothing seems to work. How would you help this student?

## Resources

There is a pretty large body of academic work about constructivism within computer science. These articles formed the basis for some of the ideas in this sections and are pretty interesting reads.

1. Constructivism in Computer Science

- This paper is a good survey of preexisting work and gives a good set of definitions for some of the more abstract concepts.
- https://dl.acm.org/citation.cfm?id=274308

2. Bricolage Forever

- Bricolage is a fancy word for trial and error. This paper gives interesting insight into the ways the students use trial and error when writing code.
- http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.4965

3. Conceptual Bugs

- These papers covers several common language-agnostic conceptual bugs in novices understanding regarding computer science and working with the terminal. Both are a bit old, but many of the issues that students faced in the 1980s and 1990s are just the same as today.
- http://journals.sagepub.com/doi/abs/10.2190/689T-1R2A-X4W4-29J2
- http://journals.sagepub.com/doi/abs/10.2190/3LFX-9RRF-67T8-UVK9

# Chapter 6

# Office Hours

## Introduction & Motive

Office hours are a set time when a TA is available to students for fielding questions and offering guidance.

The exact roles and responsibilities of an undergraduate TA during office hours vary between courses (e.g. answering questions about exam grades). General guidance is provided in this chapter.

## Learning objectives

By the end of this section, you will:

- Be clear about what kinds of questions you are able to answer
- Understand how to balance your time between any number of students who seek your help
- Be aware of what resources are available to you during Office Hours

## Logistics

### How do I schedule office hours?

Office hours are scheduled with by senior course staff. Before the end of the first week of classes, you can expect to be reached out to for your availability. Most courses request you host office hours at the same time every week. Some courses assign office hours on a week-by-week basis.

You may be asked to hold office hours at the same time as one or more other TAs from your same course.

## Where do I hold office hours?

Office hours are traditionally held in Halligan. Courses may have requirements or suggestions for you to sit in a particular room or area so students can find you. Some courses require you to walk around to wherever students say they are on Halligan Helper.

## What is Halligan Helper?

Halligan Helper is an online resource at halliganhelper.com that allows students to enter a virtual queue to request TA assistance. It is run and managed by a nebulous consortium of current and former students. TAs are given access to the TA features by their professors, grad TAs, teaching fellows, or other senior staff. If you do not have access, these should be your first points of contact.

Some courses use Halligan Helper as a sort of "TA delivery service", where the TA walks around and physically goes to the location that students provide. Some courses simply use it as an easy place for TAs to broadcast where their office hours are taking place, but don't actually use the queue functionality. Teaching a small number of students

When there's a small number of students during your office hours, you're able to give lots of specific guidance to each person. Be careful not to sit and work with a student. The point of homework assignments and problem sets is for the student to learn, not you. If you're spending too much time with a student, they won't always tell you to leave, so be prepared to make a swift exit.

## Teaching a large number of students

When you have more students in your office hours than you can feasibly help one at a time, your strategy should shift from that of an instructor to that of a facilitator. You should connect students who have similar problems to talk things through with each other. Since this situation tends to only arise close to assignment due dates, courses with Piazza pages generally have a fair number of questions answered on the Piazza pages. Students can be encouraged to use the resource of Piazza during these times as well.

### Types of questions you will receive

#### "Am I thinking about this right?"

For intro-level courses, TAs generally should be able to fully understand the explanation and give feedback for where the student is going wrong or if they're correct. For some higher-level courses where TAs cannot be expected to know the entire material and every possible way to implement it, they can still often find places where logic breaks or things go wrong.

Frequently, encouraging a student to draw out their thoughts or program is enough for them to figure things out by themselves.

Frequently, merely explaining their ideas to you makes a student realize where they went wrong, even without any actual feedback from you. This is called "rubber duck debugging".

#### "Debug this for me."

If you debug a student's code, they tend not to actually learn anything. The student's actual problem is either "I don't have a clear and correct understanding in my head of how to do this". As such, you should provide general guidance of how to approach and think about the problem.

Often times, it is sufficient to explain the meaning of an error message. This is especially helpful for things like syntax errors.

#### "Why was my exam or homework graded wrong?"

Courses vary greatly on how they want regrade requests processed. Most courses require the requests happen via email and regrading happens without the student present. Make sure you're aware of how your course handles regrading and be prepared to outline the course policy to the student. If the course requires you to do regrading live in front of the student, you should expect to be trained appropriately by the specific course's senior staff.

## Case studies

While hearing about strategies in the abstract can be nice, it is often helpful to look at some concrete examples of situations that have happened in the past and will happen in the future. This section presents four sample situations.

1. A student is lost and wants you to stay and help debug code

- How do you give guidance without doing the debugging yourself?

- How do you tell if the non-working code indicates a lack of understanding?
- What specific debugging tips can you suggest to them?

2. A student is lost and wants you to stay and help plan out code

- How do you give guidance without doing the planning yourself?
- What resources already exist to help the student with planning?
- What do you do if the student hasn't even started planning yet?

3. There are 30 students needing help and you are the only TA on duty

- Is it possible to help each student individually?
- What do you do differently from when you have only 3 students?

# Chapter 7

# Pedagogy

## Introduction

Pedagogy is the method of teaching, especially teaching a particular subject or in general. It is important to understand the pedagogy of computer science in order to be an effective teacher during office hours, labs, or recitations. Generally, when you are working with students you are covering subject material that they have (or at least should have) learned already in lecture. Therefore, your goal is to find their point of misunderstanding and correct it. Useful strategies include to help students include crafting examples, asking questions, and assessing learning. This section explains some effective methods of teaching computer science with a focus on identifying and correcting misconceptions.

## Learning Objectives

By the end of this section, UTAs will be able to:

- Ask effective questions to gauge a student's level of understanding
- Convert students from a fixed mindset to a growth mindset
- Teach students to use rubber duck debugging and diagramming
- Choose the best method to use for a given situation
- Understand what level of help is and isn't appropriate

## Skills/Strategies

This section outlines appropriate strategies and methods for teaching computer science

### Diagramming

This is an activity that students can use to help understand how their program is represented within the computer. Students draw out the data structures that their program uses and modify the picture as they walk through the code. The key to this activity is asking the student to explain their process as they modify the drawing. This forces the student to make their thought process clear to both you and themselves. If they make an incorrect assumption or make too large of a logical leap, it can indicate a misunderstanding that you can identify and fix. As such, this can be a useful tool for finding a student's misconceptions.

### Asking effective questions to discover misconceptions

This is a tool that you can use when interacting directly with students. Figuring out what questions are appropriate to ask can be difficult. However, your goal is to get a student "to find out what they think by hearing what they say" (Christenbury). By making them speak their thought process out loud , they might realize where their misconception is. Sometimes a student is so lost that they can't figure out where they went wrong even after vocalizing their thoughts. They might be confused by an important concept or just unable to figure out a subtle difference between their solution and the correct solution. At this point it is reasonable to point out their misconception, step away from the current line of questioning, and start discussing the concepts behind their bug.

### Rubber-duck debugging

This is an activity that students can use to actively think about their code. The idea behind it is that the programmer walks through their code and their debugging process out loud. This process can be supplemented with diagramming to improve the effectiveness.The goal of this process is the same as the above point of asking questions. In this case though, the student does not need someone to ask them questions. Because it can be a solo activity, rubber-duck debugging is an effective tool to explain to students and therefore enable them to debug their code by themselves.

### Creating examples

When asking a student to diagram or walk through their code, it can be a laborious process if the input they are working with is large. You can reduce the time and energy spent on the process by identifying (or asking them to identify) a smaller test case with the same issue. The same process can be used when doing examples as part of the explanation of a subject. It isn't worth the time

to go through a long example when a shorter example would get the point across just as well.

### Assessment

This can conjure thoughts of pop quizzes and tests. However, assessment covers a much broader range than graded work. Assessments are anything that help give you insight into how your students are learning. Asking questions is an assessment of parts of a student's mental model. Prelabs are an assessment of the level of understanding that the students have on the lab topic as well as an indicator of how well the subject was covered in lecture. Even simply asking your students how comfortable they are with a subject is a form of assessment. The point of assessments is the gather information about how well students are learning so that you can reinforce concepts with which they are struggling (in addition to grading them).

## Application

When first interacting with a student about a question they have, it is important to gauge their level of understanding. For questions on subject matter, you can ask the student to give a summary of their understanding of that part of the material. This allows you to see where they are coming from and correct any major issues they have in their understanding of the material.

Different situations require different teaching styles. For instance in labs and recitations you need to address the whole class at times. If you teach a topic they're already very familiar with, you'll lose their attention. On the other hand, if you assume that they know more than they actually do they'll be confused. This is where prelabs and taking a poll of the class can be useful tools to assess the knowledge of the class. By determining what they do or don't know, you can determine an appropriate level of complexity at which to teach the lab or recitation.

### Asking Questions

There are a number of different forms that your questions can take to effectively serve your students. Depending on the situation, some forms are more beneficial than others. For instance, when covering subjects in lab or recitation it can be useful to ask questions that are open-ended and characterize the material. This allows a wide range of responses and allows different students to incorporate the pieces of the material that they understand into the activity. When in office hours, other approaches must be used. Generally, in office hours, the students that are there have an issue with some point of their code and want help solving

it. Diagramming, rubber-duck debugging, and asking questions can all work well together. Although they are presented as separate concepts, they are inextricable when actually working with students. Part of your line of questioning may be to ask the student to draw a picture of what is going on in their program. When a student comes to you with an issue, your job is not to solve it for them. Rather, it is to provide them with the tools with which they can obtain the solution for themselves.

## Fixed vs. Growth Mindsets

When working with a student, sometimes it becomes clear that they believe that they just "don't get" CS. When a student says something like that, it means that they believe that some people have an innate ability for CS that they don't have. This is called a Fixed Mindset. A fixed mindset is caused by the belief that a person's intelligence can't change. If the student is struggling with a problem then a fixed mindset says that the student is not intelligent enough to solve the problem. This is bad. We want the students we teach to have a growth mindset. A Growth Mindset is the idea that intelligence can be developed. That by struggling with an idea and overcoming challenge you can build intuition and intelligence for a subject. This is the mindset we want students to have. When students have a growth mindset they are more willing to engage in learning practices such as rubber-duck debugging and diagramming. They are less likely to get frustrated and fed up in office hours. And finally, they are more likely to believe that they can succeed in a course no matter how much they are currently struggling.

When you identify that a student has a fixed mindset, it is important to help them to change their mindset to a growth mindset. It will make it easier for you to teach and for them to learn. Changing their mindset can be tricky. However there are a few methods you can use to help urge them in that direction. One method is to try to relate to the student. Identify a time when you had similar feelings of inability and mention it. Talk about how you struggled for a long time, but in the end you figured it out and now you understand the subject much better. The student might see themselves in your shoes and see the light. Another method is to reframe the issue that they are having as an opportunity for learning. There is a link (https://www.opencolleges.edu.au/informed/features/develop-a-growth-mindset/) in the resources that has a big list of other methods for changing a fixed mindset into a growth mindset. Each student is unique and will require their own motivation to shift their mindset.

## Case Study

A student comes up to you during office hours and says the they have a segfault in their program. They push the computer over to you and ask you to fix it.

(This is an incredibly vague, but also incredibly common situation. Hence why we it is offered as a case study to consider here.)

There are a number of possible ways to respond to a situation like this. How you respond will depend on your personal style of teaching and relating to people. Although this situation is not applicable to every CS class (not all CS classes have segfaults) it is a useful exercise to discuss how you would approach this situation with your fellow TAs.

- Consider the mindset of the student: is it a fixed mindset or a growth mindset?
- Have you worked with the student before? How would that change your interaction?
- What types of questions would you ask the student?
- How would you deal with the student if they are unwilling to go through the process but just continue to insist that you fix their code?
- What activities can you have the student do?
- How long should you spend with the student?

There is no single correct way to handle this situation. They are many reasonable interactions you can have. However, there are also many wrong ways to handle the situations. Think of some bad ways to handle the situation. Take note so that you can stop yourself or other TAs who are acting that way. Talk to your fellow TAs the next time you are holding office hours that no one is showing up to about this case study. Have a discussion about it. Think of other common situations that cause difficulty when teaching. Discuss those. You will be a better TA and better at helping students the more you think and reflect on your experiences and your interactions.

## Challenges:

Pedagogy has a number of challenges relating to a student's learning. Sometimes after going through the above processes a student will still struggle to grasp the subject material. At this point it is necessary to ask targeted questions with the goal of determining what part of the material they don't understand. Once that is determined, you can try to reframe that explanation in a different way. Try drawing a picture or looking for animations online. Ask another TA to try to explain the subject. It could be that your method of conveying information is not conducive to the student's learning. If none of this helps, it can be useful to suggest that the student go talk a walk and clear their head. Thinking about a problem for too long can cause mental fatigue that will make it difficult to solve.

Another challenge that you will face when teaching students is how directly to involve yourself in their code. There are a few different philosophies on the matter. Some TAs like to avoid looking at the student's code as much as is reasonable and instead have the student explain their problem in their own words. On the other hand, it can be useful to look at a student's code to see

if you can identify what is wrong. With that knowledge, you can then ask the student questions or have them do an activity that is more directed specifically at their problem. However, you must avoid telling a student directly what to type to fix their problem. Telling them prevents them from learning the material and creates the idea in their head of TAs as fixers rather than helpers.

## Resources:

This section lists resources that TAs can use for more detail on teaching theory.

- Asking Questions
- Eric Christenbury - Questioning: A Path to Critical Thinking
- https://www.cte.cornell.edu/teaching-ideas/engaging-students/using-effective-questions.html
- Fixed vs. Growth Mindset
- https://www.brainpickings.org/2014/01/29/carol-dweck-mindset/
- https://www.opencolleges.edu.au/informed/features/develop-a-growth-mindset/

# Chapter 8

# Professionalism

## Learning Objectives for this Module:

- Defining professionalism
- Providing motivation for being professional
- Outlining how one might go about becoming more of a professional

Teaching assistants need not wear a pant suit to office hours, nor will they lose their job for showing a smile or cracking a joke. There are many misconceptions surrounding professionalism. One might argue that professionalism defines an impersonal, even cold demeanor, but I would assert that a strong sense of empathy and respect for others is necessary for teaching in a professional manor. As an undergraduate teaching assistant with relatively limited qualifications and experience, one might erroneously believe professionalism is not a necessary or useful attribute. In this module we will explore how best to approach becoming a professional TA, and why it is important to do so.

## Professionalism Outside the Classroom: Preparation and Excellence

Many believe that the only requirement of a teacher is to understand what they are teaching at a level that is higher than the understanding of their students. Those who believe this would be incorrect. Teaching requires preparation, and understanding the material from multiple perspectives: through the viewpoint of a teacher, student, or otherwise. To promote and motivate learning, one must be familiar with the mindset of a learner. A teacher's ability to consider the material from the perspective of one unfamiliar is essential to preparing quality teaching strategies and examples. Remember, each TA was once a student themselves

of the course they now teach, and is in fact still a student of other courses and disciplines; recalling the mindset of a student is critical to a TAs preparation and evaluation of material. In short, preparation is not an action so much as it is an attitude.

Here is a sequence by which one might approach preparation:

- Assessment: By examining the material through the eyes of a student, one increases their chances of identifying what might be difficult for a student to understand, what classroom materials may or may not be helpful, what questions students may have, etc.
- Implementation: The results of this examination should naturally prompt questions such as how to present the material, what examples might best convey the essence of a lesson, what sorts of practice problems might be useful to attempt... these are all important questions that must be addressed before stepping into a lecture/lab/recitation, or even office hours.
- Broaden Scope: Consider how different students might answer those questions; be aware of inclusivity principles.

While the goal of preparation is to create a plan, it is also about establishing contingencies, or being able to formulate appropriate responses to unexpected situations, which becomes much easier when those situations have already been considered/prepared for (and if not, at least the thought already put into a lesson would surely be of use). This is the heart of preparation for teaching.

It goes without saying that a TA should be knowledgeable and very proficient in the course material. It is a responsibility of all course staff to be honest with themselves and their level of competence. If it is the case that a TA is for any reason incapable of attaining this threshold of adequacy, it is their obligation to correct this. Fellow TAs and professors are both excellent resources should one seek to review material or bolster one's understanding. It follows that one should also make an effort to keep up with changing course materials and expectations.

## Motivation

In general, computer science courses (at Tufts especially) are not easy, and require students to work a considerable amount of hours outside of lecture in order to understand the material and complete assignments. Students work hard, and as a teacher it follows that you should be expected to match their efforts. Professionalism is built on the idea of mutual respect, and part of that respect is recognizing that students expect a certain level of preparation, and fluency from an instructor, as well as being able to match those expectations. Not only is preparation essential for teaching well, it is essential for teaching in a professional manner (are those really so different?). Simply "winging it" is disrespectful, and thus unprofessional.

There are more concrete aspects to professionalism outside the classroom that

are important to mention:

- Maintaining professional qualities outside of scope of being a TA, be it academically, socially, digitally, etc. provides a piece of the foundation that makes for respectful student teacher interaction. As a teacher, particularly as an undergraduate TA and therefor peer of your students, setting a professional example is important, and reputation is an essential component of this.
- One should do their best to adhere to TA responsibilities: specifically being on time, not cancelling office hours and such (and if those must be cancelled, with advanced notice), respecting course pedagogy, being respectful in online correspondence, responding to emails in a timely manner, etc. While fostering a professional mindset, these particulars should be located in the realm of common sense.

# Professionalism Inside the Classroom

Returning to the idea of respect, the classroom is where the professional attitude, and how well it is maintained, is truly tested. For a teacher to be considered a professional, their relationship with each of their students ought to be formal and work focused. While it has surely been covered extensively in prior modules, respecting each student is of the utmost importance.

## On student-teacher relationships

- When developing a relationship with a student, a teacher should hold a detached and objective position. As an undergraduate TA, this can prove rather difficult, as often one might be responsible for teaching/grading their peers, acquaintances, and friends. While it is unrealistic to suggest complete separation between the student-TA relationship inside of the learning environment and outside the learning environment, this is what should be pursued.
- In these scenarios, a TA is partially responsible for the learning and evaluation of their peers or friends, and because of this the TA occupies a position of authority.
- Befriending students is a natural compulsion, but when this desire for friendship and acceptance compromises and interferes with the established power dynamic, things might get messy. Prioritizing a relationship with a student over that student's learning is a serious problem and compromises one's ability to teach effectively.
- It is entirely possible to be friends with students, but keeping that relationship as professional as possible within the classroom setting is essential to being a good TA.

A TA represents the course staff, professor, and department of the course they are involved in, so the job should come before the TA.

Each of the following is an action a professional would certainly take:

- Learning students names/preferred pronouns.
- Using appropriate language when possible (the intersection of technical and accessible is the target).
- Adhering to one's ethical responsibilities as a TA.
- Dress in appropriate attire and conform to basic standards of hygiene.

More relevant to labs and recitations, it is nevertheless worth mentioning that a lesson should begin with an outline of expectations, or learning objectives. Students should be aware of the lesson's purpose and what they will be expected to learn from it. These objectives should be reiterated throughout the duration of the teaching encounter.

## The Caveat

While this general guide to professionalism is certainly useful, it is by no means an authority on the subject. What one individual may see as professional, another might not. Moreover, determining what is and is not professional is informed by a set of assumptions that should certainly be questioned. Assumptions referencing class, race, ability, gender, etc. all influence one's standards of professionalism. It is extremely important to be mindful of how these facets of identity influence the accepted or working definition of a professional. Identity should not be a barrier to being a professional, and if indeed such a barrier exists, a professional TA would work to demolish it.

## Additional Resources

If you ever find yourself in a situation that has the potential to compromise your sense of professionalism, there are many resources within the Tufts CS department or elsewhere that could be useful. Resources such as more experienced TAs, teaching fellows, graduate student TAs, professors, etc should all be willing to offer some advice or assistance. Naturally many ideas of professionalism discussed in this module apply in contexts other than computer science; professionals in any field could be a great resource as well.

# Chapter 9

# Recitation & Lab Skills

Interactive learning in a structured classroom environment

## Introduction

Recitations and labs are a time for students in large lecture courses to work together with a subset of their class and a dedicated TA to interactively solve problems. As the leader of these sessions, it is your job to guide the discussion, supply the expertise, and facilitate a good, inclusive, and productive learning environment for everyone in the room.

### Learning Objectives:

In this section, you will learn:

- How to prepare to teach a recitation (and why you need to)
- How your first class differs from the rest (and how to make it count!)
- Active learning strategies and methods

## Select skills and strategies

### Preparation

Even though you've likely taken the class and been through the material before, it's a whole new world to teach it! For that matter, even if you have TA'd the class before, it has probably been at least five months since you last taught the

upcoming section, so it's still good to review the material sometime in the week before your recitation or lab section.

Work through the problems on your own, just as you expect your students to do. Note any inconsistencies and anything that confuses you, because those things are likely going to be much more confusing to students. In particular, if you see any mistakes in the handout or answer key, try to get them addressed with your course staff before going to teach it to students; there is not much more anxiety-provoking for students than finding out the course material itself is leading them astray. If you don't have time to get the mistake fixed, at least note it for yourself, so that you can help guide students around it when it comes up in class. As soon as possible, let the course staff know about the mistake so that it can be fixed for the future.

## Your first class

For the very first class of the semester, you'll want to start by introducing yourself to the class. Write your name on the board, then say it out loud; students are typically more comfortable asking questions during group work time if they can get your attention using your name, rather than "hey excuse me um." Then, go around and have each student introduce themselves, indicating their name and pronouns. If possible, try to check them off your roster and write their pronouns next to their name as they do this; this will help you connect the pronunciation with the spelling, and will also serve as your attendance taking.

After everyone has been introduced, begin talking about content. Take some time here to talk a bit about the course, how recitations are going to work, and what your expectations of the students are for that time. Explain to students that recitation is a good environment to try things out, because you are there to guide them if they make mistakes.

At the end of your first recitation, allow time for any and all questions students may have about the material or the course in general. Explain to the students that you are a resource to them for this course, and perhaps finish by introducing yourself by name one more time.

## Motivation

At the beginning of each recitation, introduce the topic and goals for the day, and be sure to talk briefly about the motivation for each part. In other words, if the main exercise for the day has to do with queues, remind them that queues will be on their upcoming (or current) homework. Connect topics that you introduce back to the lecture material, and show the students how this particular recitation fits into the overall roadmap of the course. Better yet, connect the

material to real-world applications, so that students can see how this material will fit into their careers in technology.

## Teaching to the class

When it's time to do a bit of teaching, be it to introduce topics or explain answers, there are a few main skills that will serve you particularly well:

- When explaining concepts, use examples! This helps students internalize concepts that may otherwise be unclear. See the chapters on Pedagogy and Teaching Abstract Concepts for more information.
- When you ask a question to the class, pause for at least 20 seconds to give students time to answer. This may feel like a very long time, but it allows students space to really think, instead of deciding quickly that they either do or do not know the answer. If necessary, remain silent until somebody proposes an answer; eventually, people will become uncomfortable enough to propose thoughts, at least. If after a brutally long time, no one has answered, try rephrasing the question or asking a leading one instead.
- When you teach recitation you're not a lecturer, you're a coach. Just as if you were coaching someone in an activity like sports or music, the first priority is to correct very large mistakes. As the students fix those big mistakes, move on to correcting the smaller ones.
- Be ready to adapt. You might be able to get through the material for the day easily, or you might find that the students have absolutely no idea what you're talking about, and that you have to essentially teach them the topic from the start. Both of these scenarios are ok; you just have to go with the flow.
- Every so often, pause to let students catch up with you. A very good phrase, shared by a professor here, is "what questions do you have?" This slight rephrasing of "are there any questions?" invites the group a bit more a safe space for questions; it says, I know you have questions, let's talk about them.

## Interactive problem solving

After you introduce the material, it's time for the problem solving portion of the class. In labs, this is the lab exercise, and in recitations, it's usually a small problem set. Pair the students up so that they have small groups to work in, if applicable; it's best to match people up by level of interest, if possible, so that more reserved, quieter students don't get steamrolled by very excited and engaged students. Then, set them to work!

While they're working, roam among them, checking in and discussing their thinking every so often. To do this, you might ask a student, "what are your thoughts so far?" or "can you explain your solution to this problem?" Pay

attention to any struggles or questions students are having. Allow students time to struggle a bit, because recitation is one of the best environments for them to do so. However, don't let them struggle too much; if more than a few groups are confused about the same thing, it may be worthwhile to pause the group work time for a few moments to come together and talk about that thing.

## Case studies: example situations

### When students stare blankly

Sometimes, you'll stand in front of a recitation, introduce a topic, ask the class to get started on the problem set for the day, and instead the entire group will stare blankly. In this case, it's best to ask them whether they've seen the material before or know what you were talking about, because unfortunately it is possible that they have not and do not. If that's the case, then you will have to bring the recitation back together, and perform a more thorough review of concepts. Remember, be ready to adapt! As a wise man once said, "Improvise. Adapt. Overcome."[1]

### When students "just don't get it"

Sometimes, you can explain a concept over and over again to a student but they are just having the absolute darndest time trying to understand. You definitely want to to help the student learn, but keep in mind that you are also responsible for balancing the learning of the other 12 or so people there. If this comes up in a more lecture-like part of your class, then it would probably be best to ask them to pause for a bit so that you can teaching the recitation to the rest of the class. Then, at the end or while everyone else is doing group work, you can go back and spend time with this one person in particular.

### When you don't know the answer

Sometimes, a question will come up during recitation that even you do not know the answer to. The most important thing here is to be honest, and don't attempt to conjure an answer on the fly and pass it off as a well-reasoned response. It is totally ok to admit that you don't know an answer! In this case, one strategy may be to tell the student to ask the question on Piazza, so that other TAs can see it. You also may go to other TAs yourself, ask the question, and relay the answer back to the student. Whatever you do, again, it is important to be honest, and hopefully provide a mechanism by which students can actually get their answer eventually.

[1] https://i.imgur.com/rZcCVi0.jpg

## Conclusion

Keep in mind that teaching a recitation is, for many TAs, a non-trivial exercise. That said, many find it to be a rewarding experience! A recitation is one of the best times to work directly with students on key concepts in the course. It's also a fantastic place to catch misunderstandings early.

As a reminder, the steps you need to follow for each recitation are:

1. Prepare the topic
2. Teach the recitation
3. Introduce yourself
4. Introduce the material
5. Motivate the material
6. Set the students to independent work
    1. Make yourself available for questions
    2. Check in with individuals to spot-check progress and understanding
7. Facilitate discussion about answers
    1. Ask students to share their thoughts
    2. Correct mistakes gently
8. Summarize key points
9. Reflect and provide feedback to senior course staff

## Further reading (and/or watching):

1. Video from an MIT professor on how to ease students into active learning: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-06-principles-of-automatic-control-fall-2012/this-course-at-mit/acclimating-students/
2. A journal article on effectiveness of different styles of leading a recitation: https://journals.aps.org/prper/pdf/10.1103/PhysRevSTPER.3.010104
3. General tips on teaching recitations: https://www.unl.edu/gradstudies/current/teaching/recitation http://ctl.gatech.edu/sites/default/files/students/taweb/handout/recitation.pdf

# Chapter 10

# When you don't know what to do

## Introduction & Motive

This handbook cannot cover every topic under the sun. It is very likely that at some point in your career you will encounter a situation that you are not prepared to handle. This section is meant to prepare you to stop, breathe, and think on the spot.

### Learning objectives

By the end of this section, you will:

- Become familiar with common problems that a teaching staff faces
- Develop a model for dealing with challenges that you have not explicitly seen before
- Create a short-list of people you could ask for help

## Select skills and strategies

This handbook would like to present several methods for answering the unexpected questions that life tends to throw at us. Each of the tactics relies on contacting somebody or something else to help you solve problems.

## Fall back on your previous TA experience

If you are a new TA, consider skipping this section and coming back to it — in reflection — next semester.

If you have been a TA for more than one semester, you will likely see patterns in student behavior. Odds are, you have encountered a similar — if not identical — situation in the past. Let how you handled that situation, whether you handled it well or poorly, inform your decision-making now. What could you have done better? Does that apply here?

## "What Would An Authority Figure Do?"

It is reasonable to use a figure of authority as a model in unfamiliar situations. Maybe it's Mark Sheldon. Your mother. Your aunt Josie. Your high school robotics coach. Whoever it is, it should be somebody you know that has experience dealing with people and ethical situations. Sit down and think about how that person would approach things.

Perhaps this figure has vocalized his model of ethical inquiry already, and you feel comfortable adopting it. Perhaps you were an observer in this very situation in the past. Perhaps you can take a moment, step back, call him up, and ask "Hey, what would you do in this situation?"

If you have the time, a pen, and a piece of paper, take a moment and get those out. Write down three to five people you could contact if you needed help dealing with something. Why did you think of them?

## Make use of prior art

Since there is a long history of undergraduate teaching assistants at Tufts, there is a rich body of knowledge from which you can draw. Nearly all internal documents, important conversations, and decisions about matters of consequence are archived in course folders semesterly. This is something Tufts does well. Some semblance of that documentation will be available to you and your fellow course staff, so you should ask your professor and other TAs where best to look. When you find an answer, please add some of your new-found knowledge to this manual.

In addition, there may be record of a similar situation on the course Piazza page, the course website, or even by email. Make sure to check those sources as well.

## Phone a friend

There are likely many other undergraduate TAs, Teaching Fellows, graduate TAs, and professors within your arms' reach. Since TAs are generally helpful people, they will likely be more than willing to give you advice in a time of trouble.

Consider the other undergraduate TAs of the course: who can you approach? Odds are, if you are proctoring an exam, on duty in your office hours, leading a lab, or some other standard TA duty, you can reach out to another person who is also working. Ask them if they have experienced something similar, and what they might do.

It's entirely possible that either there will not be an undergraduate TA, or that TA will not know what to do. In that case, you should contact either a Teaching Fellow or a graduate TA that you know and trust. Given their tenure as TAs, they will hopefully have some wisdom to impart.

Since we're really assuming the worst here, you, dear reader, may not be able to immediately find another TA. In that case, contact the professor for the course. Your professor will either have encountered the situation you are experiencing or will have a very refined way of handling such things. In either case, take some time to listen and learn, so that you may contribute your findings to other members of the teaching staff.

## Case studies: example situations

While hearing about strategies in the abstract can be nice, it is often helpful to look at some concrete examples of situations that have happened in the past and will happen in the future. This section presents four sample situations.

### You feel uncomfortable because of student behavior

*Situation*: Students are often emotional when it comes to their work. Occasionally this comes out in bursts. For example, some classes have in-person grading for a subset of the projects. Occasionally, a student will feel stressed when they think they are not doing well or that they cannot get their point across — and may get angry, raise their voice, stand up, or potentially be violent.

*What to do*: If there is ever a time when you feel physically threatened or uncomfortable because of a student, your first priority should be to remove yourself from the situation. After leaving the situation, get help. Help could be from a professor or the police. If the threat is imminent, call 9-1-1. If getting away resolved the problem, let the professor know as soon as you can. Explain the situation, who the student was, and your professor will help navigate the situation.

*Situation*: You are helping a student when you notice that they are "checking you out". You feel uncomfortable.

*What to do*: This is one of those moments that is harder to address verbally. What if you misinterpreted the look or otherwise do not feel comfortable correcting the behavior in the moment? In this situation, it may help to firmly establish the boundaries and roles in the situation. Revert to a strict separation of student and TA if there was anything more. Use overt teaching vocabulary. Hopefully this makes it clear that what the student is doing is inappropriate.

### You come across students cheating and they ignore your demand to stop

*Situation*: Imagine you happen upon students sharing code. You tell them to stop, but they continue. The students knowingly ignore your directives to stop and continue to flout university policy. They leap take pictures of the board. You do not know who they are.

*What to do*: There are many things wrong with this picture. First, the students are sharing code. This is an academic integrity violation, and staff is required to report the students immediately. Second, the students ignore your authoritative direction to stop breaking academic integrity rules. Third, the students – despite your directive – start taking pictures of the shared material, further breaking academic integrity.

Because you are required to report the students, you should note down their names and send them to the professor. If you do not know the students, jot down their descriptions and what they are wearing. At worst, you can take pictures of them for later identification.

If you do not deal with the situation swiftly, other students' work may be at stake: if the work with integrity violations is graded in the same pool as "normal" work, it could affect everybody's grades. This is a serious issue.

This does not address the root problem: feeling the need to cheat. However, that is not your responsibility; it is that of the professor. If you feel as though there is a systemic problem in the course, raise it to the attention of the professor as soon as possible.

### You are approached off-duty for help

*Situation*: You are at a social function (off-duty). A student approaches you and asks for help with something course-related.

*What to do*: Consider your options. Should you help the student? Help them but tell them that this is inappropriate? (Is it inappropriate? (Yes.)) Tell them that you won't help them?

Whatever you decide has implications for both the rest of the course staff and the course staff of other courses. If there is an agreed-upon code, then you should follow it no matter the situation. Rules about social interaction exist to protect you, the TA, as well.

If you help the student, you either set or support the precedent that an off-duty is fair game for student questions. You are not getting paid for your work, and future TAs will get pestered for questions when they are not required to be answering them.

But you may feel uncomfortable telling the student that they should not be asking you for help. Here is a handy guide for the interaction in the worst possible case:

```
[STUDENT and YOU are both at a social function]
[YOU are having a Good Time]
[enter STUDENT]
STUDENT: Can I have some help with my problem?
YOU: Actually, I am off-duty right now.
STUDENT: But I really need help or I might fail this class. And if I fail,
  I can't get a job!
YOU: If there are TAs on duty, you should go see them in Halligan right
  now. They can help you. If there aren't, then I'm afraid there is not
  help available right now.
STUDENT: But I can't fail this class! [some very personal reason] so it
  would be really bad if I didn't get help right this second.
YOU: I suggest you post privately on Piazza / email the professor / talk to
  your dean / [other course-approved suggestion] about this situation. I
  cannot help you.
[YOU remove yourself from the situation]
```

Unfortunately, it is not a pleasant interaction to have. Fortunately, it is reasonably rare that the student does not understand that you are a human, too.

## Further reading

https://www.cmu.edu/teaching/resources/PublicationsArchives/CollectedWisdom/collectwisdom-teachingstrategies.pdf

https://www.cs.umd.edu/grad/ta-handbook