Helpful Attributes of TAs

- patience
- charismatic
- listening skills
- being articulate
- relatable
- confidence
- reassuring
- calm
- positive
- enthusiastic
- prepared

Unhelpful Attributes of TAs

- hand-holding
- dismissive
- force others to think your way
- distant
- aloof
- not following student hunch
- not knowing student state of mind
- not listening to your own frustration

Do these things

Introduce yourself

Ask their issues, have them explain, be specific

Break down their problem into smaller questions,
- then address individually

Back up what you suggest - give them a "why"

Send them in the right direction, and make sure
they can somewhat do it themselves
- give them examples

Try to limit interaction to a certain amount of time
- switch with another TA
- give processing time, then come back

Draw things out -- diagrams
- encourage them to visualize their solution

When encounter compile/valgrind errors
- look at first one first, possibly replot to ones lower down

Clear up misconceptions FIRST
- identify them

If they repeat specific questions on Piazza, try to
get them to also use Piazza more often.
- otherwise, clarify Piazza posts if needed

Latch onto the terms/things they mention and
teach how to use those things
- If they are comfortable with those, THEN
introduce things

Take a few moments to think
- otherwise your response comes out as a
brain dump

Ask questions to prompt thought instead of just
giving answers

Ask what they have tried, debugging-wise

Have them walk you through the code

Bring a whiteboard to OH so that students can
conveniently draw diagrams (Bruce note: bet-
ter to just use paper, so they can keep that)

Know the specs by heart

Use the white board
- make pictures
- step through code 100% before you try to fix
anything

Explain to both partners (40)

Make them explain to you what they are doing

Ask questions more than you answer questions

Point them in the right direction
- give them the tools to solve problems (mem
leaks/seg faults)

Have students explain things back to you once
you have explained something to them

Provide general tools (terminal commands)

Start by having the student fully explain their
problem

Have the student use pencil and paper

Ask the student clarifying questions if you don’t
understand them

Ask other TAs if you don’t know

Get other resources for students (this item needs
clarification/examples)

Acknowledge to students that some concepts are
beyond the scope of the course
Emphasize that it is experience, not intelligence
Walk through fine details
Ask questions instead of just telling
- be Socratic
Make a connection between high and low level
- Make sure they have a plan/can code
Know the assignment before you get there
Encourage testing
- especially unit testing
Understand the depth and nature of the problem
(listening)
Pay attention to students
Ask people’s names
Draw pictures or have THEM draw pictures
Ask them what they have tried
Review the materials before office hours (not just the assignment but also lecture slides)
Listen to them more instead of just talking about their ideas
Guide them to find the answers themselves
When it comes to debugging, do not hand hold them to figure out the bug; instead, teach them to use print statements/valgrind/gdb
Stress the need for individuals to test on their own, before using provide
Suggest students draw their problems (particularly with memory) out
Ask students to precisely explain their problems
Know the fine line between helping a student and giving them the algorithm
Give students some time after talking to them (e.g. "try what we talked about and I’ll be back in 5 minutes")
Teach the students how to use the resources available to them to solve problems (e.g. man pages and online resources)
Don’t just encourage them to draw it out, "act" out the data structures using hands as pointers
Ask them to rephrase what you discussed while helping them
If you aren’t sure they fully understand the conversation when about to leave, ask them to write out in comments their plan, so they will have something to refer to when you leave
Use phrases like "Explain to me this design choice or data structure", rather than "Do you understand ...
Focus on teaching them debugging skills, rather than debugging for them. This includes explaining what causes valgrind errors etc.
Make sure students take a step back/a break then they are visibly stressed or have been in the lab for a while
Have students re-explain a concept after you are done explaining is useful to gauge their understanding
Do encourage students to ask questions when they don’t understand something.
Do encourage students to draw a picture of their data structure at various points throughout the program if they are having trouble debugging.
Do be patient: listen to the student explain their code, and help them talk through it.
Be patient with a student. They may not always know how to articulate their question, but that’s just another opportunity for a TA to help them.
Even if the queue is long, the student appreciates your full attention, even though that can sometimes be time consuming.
Use Halligan Helper, it makes all of our lives easier!
talk to other TAs if you have grading questions when grading to make sure that there is consistency in how certain points are rewarded
encourage students in office hours and lab to write out pseudo code and explain their algorithms in English to you instead of just fixing their problems.
Do have the student explain what they think their code is doing
Teach students how to debug their code -- not debug it for them
Do make yourself as visible as possible: circle the labs occasionally, introduce yourself to people
Do follow-up with students you help: check in with them after a few minutes to see if your suggestions were helpful, if they have follow-up questions, or if they are still stuck
Do Make students draw data structures on paper
Do give hints so that the student finds bugs on their own
Do point out the things they did get right first, to make them feel better about their understanding of the material
Do encourage students to pinpoint their problems before presenting it to you

Do NOT do these

Don’t blow them off or be condescending
Don’t talk too much
Don’t brain dump on them OR say nothing
Don’t assume what an instructor’s expectations are
- ask or go with what’s on the specs
Don’t just point to document, then leave
Don’t talk to only one partner/team member; make sure both are engaged
Assume their level of understanding
Spend too much time on a single student; give them ideas and let them work
Use jargon where not needed
TA when hungry
Offer solutions to problems they have not come to
Write code for them
Tell them what data structure to use
Get angry/impatient
Be dismissive
Spend too much time on one problem
Show them your code
Grade on the spot
Write their code
Give the full solution
Lie to the student
Leave until the student at least has a place to go
Spend too long with a student on a minor issue when there is a long queue
Act like you know everything
Introduce excessively complicated concepts when not necessary

Take the keyboard (at least without asking)
Write code for them
Leave without their explaining what they are doing or just learned
Fall back on your own code/understanding
Talk on the phone while helping a student
Ignore a partner -- talk to both
Do not treat them like little kids/ do not talk to them in a condescending way
Do not just give away the answers (unless the questions are specific to homework expectations or standards, in which case you do want to tell them the accurate answers to clear the confusion
Do not do the heavy lifting for the student--particularly when up on the homework submission deadline
Do not become impatient with the student
Do not be afraid to not have the answer to a question. Other TA's and piazza are resources for both the students and the TA's.
Do not be afraid to direct people to the specifications if the answer is clearly written
Do not allow students to be rude to you! Emotions and stress can run amok during crunch time, and if necessary ask another TA to switch students with you
Do not focus too much on one student in a partnership.
Do not spend too long trying to explain something
If your approach is not working, swap out with another TA
Don’t tell students the answer without explaining why.
Don’t talk more than you listen.
Try not to have headphones on during office hours. It makes you seem unavailable
Don’t take over typing duties from the student. It’s more helpful for them to type it themselves.
Do not skip office hours without scheduling a replacement
Do no overwhelm the student with too much information at once - focus on their problems one at a time and explain things thoroughly
instead of rushing through stuff

Do not give the correct answer without any guidance

Do not be afraid to ask for help when they are unsure of a problem

Do not give away the whole answer: ask leading questions to get the student to make the connection independently; don’t ever type on their computer and give them the answer

Do not give incorrect information: it’s better to admit your own confusion or lack of information than to send a student on the incorrect path. If you don’t know the answer, admit that and help the student find someone who can help

Do not assume the student can understand your CS vocabulary

Do not spend too long with one student if you cannot help them and the queue is too long. The best thing is to have another TA swap places with you

Do not ever make a student feel dumb or insecure for not knowing the right answer

Do not give a student an answer without first (1) figure out how much they might understand and then (2) trying to show them they maybe do know, they just were not thinking about it the right way or for long enough