eemokey.blog
June 3, 2025

Oski Does Well in CS Classes (with no CS background)

Having survived Berkeley CS, here's what worked well for me.

Going into my freshmen year, I decided to do CS at Berkeley with no prior programming experience.

And now, I'm doing my PhD in CS.

Read on for my thoughts on classes and study guidelines.

1 A Holistic Approach

CS courses at Berkeley are not just about knowing the material or getting good grades.

I think their true purpose is to train students in problem-solving and other skills that will carry beyond any single exam.

Key takeaways could include:

- Problem solving: The ability to dissect open-ended problems without relying on rote memorization
- Resilience: Debugging code or refining a proof while failing repeatedly before succeeding instead of giving up
- Endurance: Being able to sustain repeated, extended periods of emotional and intellectual endurance
- Self discovery: Knowing how you best learn and what strategies work best for you
- Redefining limits: Realizing that you could do things you never thought possible

2 Study Strategies

2.1 Learning the Material: See the Material at Least Twice

Since I had no prior CS experience, I found it difficult to pick up concepts after only hearing it the first time.

To address for this, I did the following things in addition to the typical course workload:

1. Before the class starts: Watch and sometimes take notes on relevant past lectures[1] during winter/summer breaks. Try to attempt publicly released notes and worksheets
2. Attend discussion sections from 2-3 different TAs each week. Typically 1 section per week is recommended. For example, if discussion section for CS70 is held on Tues/Thu, then I would attend two on Tues and two on Thu hosted by two different TAs
3. Take notes with pen and paper when attending in-person lecture and review sessions. After coming home from these, re-organize the concepts in a way that makes it easier for you to recall the information
4. Practice. Links to TBP database and HKN exam archives.

I learned that I absorb information better in-person and by interacting with other students. If this is how you also like to learn, maybe give these a try.

2.2 Test Taking: Manage Your Anxiety

Say you follow the advice related to knowing the material for test taking such as prepare well in advance and make a cheatsheet (even if the test doesn't allow you to bring with you). However, it all won't matter if you are too anxious while taking it and perform suboptimally. This happened to me during my first semester of Calculus (Math1B).

One way to reduce anxiety is to keep your self occupied e.g. listen to music on the walk there, read your notes while waiting to be let into the building, pace back and forth if it's too dark outside to read your notes, doodle while waiting for the timer to start...etc. Do anything but nothing.

Another way is to develop a pre-test routine e.g. On the day before the test, find the test taking location on campus to reduce anxiety the day of the test. This way you know where the building is and don't get lost and arrive late. On the day of the test, do a light workout, brief review, meditate/visualize yourself succeeding...etc.

These are approaches that I've used as an athlete (I did figure skating). They are well known in athletics training and they translate pretty well.

We don't want to eliminate the anxiety completely, but reduce it to a manageable level that does not influence your test taking performance.

More tips on test taking here: Doing well in your courses - a guide by Andrej Karpathy at Stanford.

2.3 Day to Day: Have Endurance

It is typically advised to form a study group. I loved spending time with mine.

Friends not only help with intellectual endurance (focusing on long problem sets and projects) but also emotional endurance (managing stress, impostor syndrome, overcoming setbacks and failure).

Both types of endurance are needed to succeed.

[1] My blog on Course Capture

- Being stuck is normal
  - Spending hours debugging late at night is normal. Your brain is like a muscle, it gets stronger the more you use it (even though it might hurt)
  - I don't know what I'm doing and everyone else is so smart. Remember that we've all been there, from your TAs to the 4.0 students. If you have the courage to keep trying, I believe you can succeed
- Failures are your best lessons
  - Failed a midterm? What can you do instead next time so the same error doesn't repeat itself?
  - Failed a course? Low GPA? It is OK, many successful people I know have done the same
- Track progress, not just deadlines
  - Note down when you understood a concept or the number of hours spent on something rather than just HW completed
  - Small wins can help you stay motivated

"We all fall. It's how we get up that matters."

National Get Up Day (February 1st) is a day to celebrate perseverance, inspired by figure skaters who literally "get up" after falling down.

3 Time Management

3.1 Planning

Use your precious time wisely.

Planning well in advance helps with this.

Create:

1. Degree plan e.g. CS sample
2. Semester plan - already provided, it is the syllabus
3. Monthly plan - What are the next set of important deadlines such as test and projects? How do other things like fun with friends, periods, or club activities fit into this? How might I shift my time around during the month to juggle things?
4. Weekly plan - What are my time commitments for this week? What do I need to get done each day?

3.2 Pick a Priority Class(es)

Pick a few classes that you really want to do well on each semester. It can be 1 or 4 classes, depending on how comfortable you are with the material and other commitments.

An example of a poor decision, knowing myself, would be to pair CS61B and CS70 together in the same semester. This relates to the CS major declaration requirements back then [2].

I did 1-2 each semester for my lower division courses:

- 1st semester - Data8, Math1B
- 2nd semester - CS61A, EE16A
- 3rd semester - CS61B, EE16B
- 4th semester - CS70
- Summer - CS61C

3.3 Estimate a Manageable Workload & More

To get a good sense of how much workload each class might be, see Berkeley Time [3], Codebase's Guide to Berkeley Computer Science , or EECS 101 Piazza/Ed.

Note. Register for your courses on time - exactly when the minute changes and always have a backup plan if you cannot get into your top picks. (Why is) Signing up for a course spot is like trying to get famous concert tickets...unless you are Regent's I suppose.

3.4 My Experience

For me, I would spend anywhere from 40-80 hours each week on courses. The 80-hour weeks would typically correspond to project deadlines rather than tests. Research (2nd year and onwards) and other activities such as teaching (2nd semester and onwards) would be in addition to this. This absurd amount of time spent was mostly because I had no prior CS experience and wanted to go to straight to grad school - no AP CS, no bootcamps, never wrote a line of code before college.

My schedule was much less structured than Prof. Sahai's famous 80-hour 4-technical course plan[4]. I often merged serentity and physical exercise into studying where I would walk and think, letting problems and ideas marinate while not staring at a screen.

I recall a Berkeley Professor (I think Prof. David Patterson) talk about the power of shower thoughts and fostering creativity during off time. Work hard, then go do things like sit in the hot tub and think, and it would be productive. But, don't just go sit in the hot tub first.

Consistent effort made things easier, but it was a very flexible learning process, like drinking water from a firehose. Courses did become easier as time went on, starting around my sophomore year.

[2] How to Declare the CS Major Fall 2022 and earlier: Students must get >= 3.30 overall GPA in CS 61A, CS 61B, & CS 70 to be admitted to the CS major

[3] My blog on Berkeley Time

[4] My Perspective on Your Time meme

4 Beyond Grades

- Build things that excite you! Seek out research, internships, and personal projects to gain real world experience
  - See my blog on how to get involved in research here
  - Class problems are different from real engineering problems. Try applying your newly-gained fundamentals to a personal project and learn how to build something interesting
- Learn how to learn! It is about training how to think, not what to think.

”You’re Learning So Many New Things and I’m Proud of You.” - Mister Rogers

5 Related Resources

- HKN course guide
- Effective Strategies for In-Person & Virtual Learning by Berkeley Student Learning Center
- 5 Tips: EECS Major at UC Berkeley

Acknowledgments

Thank you to Justin Qu and Michael Chien for reading early drafts of this post and providing helpful feedback.