Working with Norman Ramsey
A Guide for Research Students

Norman Ramsey
Tufts University

Fall 2008

Goals

We want to write the best software and publish the best papers in the best conferences and journals. We want our stuff to be the stuff people will read five or ten years from now to learn how to build their own systems. And whether we do compiler construction, language design, productivity tools, run-time systems, or type theory, we want always to be focused on one thing: making the programmer's life better.

Introduction

Herein I attempt to write down the oft-unspoken covenant between junior researchers (undergraduate students, graduate students, and postdoctoral associates) and their faculty supervisor (me). If you are reading it, perhaps you have been invited to join my research group or are already a member. Much of what is here may be common knowledge, but writing down common expectations helps prevent misunderstandings. I ask all prospective and new researchers to read the entire document carefully. If you have questions or concerns, please talk about them; the document, like my research style, is a work in progress. I review this document periodically to make sure that my expectations and obligations are clear and that nothing important is forgotten.

In the spirit of full disclosure, I have tried to identify what's distinctive about working with me—both the good and the bad:

- Working with students and contributing to their professional development is the most rewarding part of my job. My ideal is to help *students* develop into *colleagues*.

- I am ambitious and have high standards, and I expect the same from students.

- I can be disorganized, and students sometimes have to compensate. Students must stay aware of what they are doing and what the plan is. If they don't know, they have to ask.

- Students may take on as much responsibility as they want. I hope that even students who start out doing "only" programming[1] will eventually become skilled researchers, who can identify important problems, solve them, write about them, and present them to a technical audience. To that end, I seek opportunities for students to travel, publish, and present.

- I can be obsessive about precision.

- I use a sophisticated mix of software tools and programming languages, and I expect my students to do so as well.

What's it all about?

This section gives some of my perspectives on research and research students. Graduate students undertaking doctoral study are here to learn to become independent researchers—as soon as possible, you, not a manager or a supervisor, will decide what is the best use of your abilities. Undergraduate students are more likely to be sampling research as one of a number of activities. In both cases, doing research under the supervision of a faculty member offers an opportunity to develop advanced skills while contributing to improvements in engineering and technology. Here are some of the elements:

- Although a doctoral student may begin with an "immigration project" in which he or she works for me, my goal is that each doctoral student should, as quickly as possible, be working for himself or herself, to advance his or her own professional goals.

- You must be aware of the research literature related to your project. Beginning students can expect substantial help from an advisor, but if you are undertaking a PhD, you, not your advisor, will be the expert on the literature.

- Research must be documented to be successful, and in the excitement of the moment it is easy to forget or omit vital details. *Contemperaneous* note-taking helps you remember, organize, and communicate your results. You must keep a notebook.

- Graduate research requires full-time dedication and year-round effort.[2]

[1] As if that were easy.

[2] Part-time PhD study is possible but can be ten times more difficult than full-time study, especially during the crucial final year of writing the dissertation. I am willing to work with part-time PhD students, but only those who demonstrate unusual maturity and ability to master new skills.

- The field doesn't progress unless the good work is published. Publications are also a universally acknowledged sign of personal success. Writing for publication is an essential part of the graduate research experience; a typical dissertation will be connected to several submitted papers. In computer science, publication often includes the opportunity to give talks at conferences.

- Undergraduate students are not required to write for publication, but if you want to try your hand, I offer every encouragement, including paid travel to conferences at which you can present a paper.

- The best researchers, although independent, don't hesitate to get advice from others, especially in areas in which they are not the world's experts. Students, even undergraduate students, should take initiative in discussing ideas and solving problems with other students and faculty, not only within Computer Science or within Tufts, but also outside. In my field, programming languages, we are lucky to have world-class colleagues at several local colleges and universities.

Postdoctoral study

If you've completed your doctorate, the world assumes that you have the basic skills of a researcher: to identify and solve significant problems. Why do a postdoc?

- If your ultimate career goal is to become a professor, a postdoc offers you an opportunity to *spend time doing only research.* You can spend one to three years solving interesting problems without worrying about teaching, advising, or committee work. If you want, you can even be insulated from writing proposals.

  I encourage you to view your postdoctoral years not necessarily as a time to improve your publication record, but as a time to *develop intellectual capital* that you will draw on for the first several years of your next position.

- No matter what your ultimate career goal, a postdoc offers you an opportunity to *learn new ways of doing things.* You can use a postdoc to learn new experimental techniques, new tools, new ways of thinking, new ways of teaching, new ways of running a research group or a department, even a new country and culture. A good postdoc should be an enriching experience, and you should leave with lots of new tools in your intellectual toolbox.

I have two primary expectations of postdocs:

- You must contribute to the research mission of the group. This means doing good work and publishing it. There should be a plausible relationship between your work and your source of funding.

- You must take responsibility for the success of your postdoc. Only you know what you want to take away, so only you can set goals and priorities. My job is to assist you every way I can.

Basic work rules

I expect good work habits, organization, and time management.

- Even for full-time research assistants, postdocs, and professional staff, working at odd hours and in odd places is more common than a regular schedule. But to create opportunities to interact with colleagues, I expect everyone to spend *some* regular time in the lab—at least several hours each weekday. Many people in our group prefer to work by themselves in the morning (e.g., on writing) and to work interactively in the afternoons. Some remote work (telecommuting) is acceptable, but you must also spend time in the lab so you can interact with others.

  This short blurb may give you some insight into why I expect everyone to spend time in the lab:

  > I wonder how an English professor would feel spending a week in a physics lab. Not about the scientific work, but about the frequent, ongoing interaction between students and peers, postdocs and faculty. Scientists see each other in the lab, if not daily, then at least weekly. They have frequent lab meetings, colloquia and interaction with scholars at other universities around joint research. During my graduate training in psychology at McGill University, especially in the research lab at the Montreal Neurological Institute, I spent hours hanging around the postdocs. I learned at least as much from them as I did from my interactions with my professors. The expectation was that I would be at the lab 9 to 5 or more, every day. I saw my adviser every day.
  >
  > — Gina Hiatt

  I am hoping in the future to be more successful at seeing students every day. In particular, I would love to have a short daily briefing (about 10–15 minutes) that *all* researchers attend.

- I discourage “death from overwork,” but at certain times (when conference or grant deadlines approach, or when a project’s goals are not being met), I expect extra time (evenings, weekends). After a crunch, compensatory time off is not only acceptable but recommended.

- Undergraduate students working for me part-time should try to figure out how much time they can commit; we will review these commitments every semester. If undergraduate students find they are overcommitted, they should squeeze out a few hours so their work can be handed off to someone else, not lost. Remember, it is always possible to leave a project with honor, but it is never honorable to disappear without a word.

- Graduate research assistants should plan some involvement with their projects between semesters and during long academic breaks.
- If you will need time off, please discuss it with me in advance.
- Make sure I have your current phone number and email address.
- If special circumstances require you to miss work, let me know.

The practice of research

The key principles are to know *what you have done*, to know *what you plan to do*, and to know *what is going on around you.*

Research meetings There are many kinds of research meetings.

- In an individual meeting, you meet with a supervisor or supervisors. Such a meeting is a good time to get some opinions about your new ideas, to get help with problems, and to communicate with your supervisor about progress.
- Project meetings may involve several people working on the same schedule. They usually involve planning and status reporting. This may be a good time to let other people on the same project know what you are doing.
- Group meetings may involve one or more research groups with researchers at all levels of seniority. They are a good way to develop broad ideas about what people are doing and where the field is going. They may also present opportunities to develop important skills, such as presenting work in front of groups. Finally, group meetings help build a sense of community.

Here are some suggestions for “best practices” for research meetings.

- Know when the next meeting is.
- For an individual meeting with your supervisor, it is a fine idea to bring a *written agenda* the meeting. Have ink on paper.
- *Ask questions* during meetings. Don’t take something on faith just because the person who says it is more experienced.
- *Take notes* during meetings. If you need time for notes, ask participants to slow down.

- After a project meeting or a supervisory meeting, use your notes to *write down your understanding of the decisions made at that meeting.* As well as any decisions made, be sure to cover your *plan of work* for the time until the next meeting. When you have finished, *email your summary to me* for confirmation.

  Post-meeting email is difficult for people to carry off. I keep pushing it for two reasons:

  - By writing things down, you will clarify your own thoughts about your plans. You will also get a chance *immediately* to get help tying up loose ends or solving mysteries. You might otherwise have to wait for the next meeting to get this kind of help.
  - You create a permanent record of what you've been working on. This record is an invaluable aid to writing a strong letter of recommendation: I can mine the record for lots of specifics to support my claims that you are smart, hard-working, have lots of creative ideas, etc.

- I enjoy meeting with students, and I work with my door open. Please drop in for a spontaneous meeting at any time, excepting only if my door is shut.

Records

- Get a *lab notebook* to keep a permanent record of your work, about which more below.
- Maintain an *annotated* bibliography of references useful to your project. BibTEX can be helpful here.
- If you borrow a book from me or from the lab, *leave a trail*; sign it out.

The lab notebook This is how I expect you to use your notebook:

- Every time you work on your project, write in your notebook the date and time and what you did. (If you are being paid an hourly wage, you have to keep track of your hours anyway.) Even a single sentence provides a useful record.
- If you take experimental measurements, write them in your notebook.
- Your record should include things that failed as well as your successes. Write down what it was that didn't work; you may save someone else many hours.

  You should definitely write down big things that don't work, and it may even be worth writing little things that don't work. If it's a good idea for Don Knuth, it might be a good idea for you. (See "The Errors of TEX," *Software—Practice & Experience*, 19(7):607–685, July 1989.)

- Your notebook is a good place to keep records of test inputs and outputs, transcripts, screen dumps, etc. Print them out and tape them into your notebook. In some ways, this is better than a demo, because there's a permanent record.

- You may also find it useful to use your notebook to sketch ideas, observations, measurements, proofs, code, solutions to problems, or whatever. Put these things directly into your notebook, not on scraps of paper to be transcribed later. Go wild; notebooks are cheap.

- Don't let the dog eat your notebook.

Professional interactions

- Be aware of what others are doing, both in the group and out. This knowledge and habit will serve you well throughout your career.

- Offer your experience to help others.

- Seek the experience of faculty and other students to help yourself.

- Be active at meetings and conferences. *Ask questions.* Do so respectfully, concisely, and often.

- Review manuscripts, both within the group and outside. (This activity is optional for undergraduate students.) I will provide guidance.

Good citizenship I expect everyone to work together to identify and solve problems.

- When we review code, I expect everyone in my group to participate, even if the code is not directly related to their project.

- If you see a need, find a solution. I will help you implement it.

- I may ask you to undertake service work for the group or for our department.

Group and departmental citizenship

Good citizenship is valuable not only for the research group but also for the department. Please don't hide in your office; get out and meet people at group and departmental events. Everyone should attend one group event per week, and everyone should consider attending departmental colloquia.

- *Programming group (Wednesday at 4:00).* Greg Morrisett, Mike Smith, and I jointly support a weekly meeting to discuss programming, programming languages, and related software tools. There is usually a technical talk. Everyone who works with us should be there.

- *CS colloquium (Thursday afternoon).* The Computer Science colloquium is generally held every Thursday from 2:50 to 4:00 in Halligan 111. Refreshments are served at 2:50 and the talk starts at 3:00. A good colloquium is a great opportunity to broaden your education by learning new things from people who aren't normally here. A colloquium that is not so good can at least teach us something about how not to give a talk. If a colloquium talk is too specialized for a general CS audience, it is a good idea to complain to the host!

  For undergraduate students, there is probably not much benefit in going to every colloquium: the talks tend to assume a solid undergraduate knowledge of computer science. But do keep an eye on the speakers, and if you see a talk that looks interesting, come.

  For graduate students, it's not always obvious why you should go to colloquia. The benefits are primarily long-term and indirect.

  - You build up some nontrivial knowledge in a broad set of areas in computer science, including many areas in which you may never do research. This kind of breadth won't matter much while you're in graduate school, but it will become terribly important when you start looking for your first job. Especially if you are looking for an academic position, you will have to talk to many members of hiring committees who are not in your area. If you can talk with them about their own areas and can get excited about some aspect of what is going on there, they will be much more likely to want you to come.
  - When colloquia are full of people listening and asking good questions, it helps visitors learn that there is a vibrant CS community here. Building Tufts's reputation helps everyone and ultimately increases the value of your degree.
  - It probably won't happen very often, but every so often you'll get a good idea from a completely unexpected direction.

  If a colloquium is given in an area in which you have a research interest, by all means sign up to meet with the speaker. Most speakers enjoy meeting with students, and faculty from other institutions are always thinking about recruiting. If some visiting senior person comes in and asks you about your work, don't be shy! Explain what you're doing and why, *use the whiteboard*, be clear, and you'll be remembered during interview season.

Software tools

My group uses software tools and languages aggressively.

- *Literate programming* is essential for any software that more than one person will work on, or that will outlive its author's tenure with the group.

I naturally prefer the Noweb tool, but nuweb, CWEB, FunnelWeb, and other variants may be acceptable.

Noweb code should be *at most 88 columns wide* so the output can be printed. Blank lines inside a code or documentation chunk can be Good, but blank lines around a code or documentation chunk are Bad.

`http://www.eecs.harvard.edu/~nr/noweb`

- All work should appear on a server that is backed up *nightly.* The Unison file synchronizer can be used to keep a server consistent with a laptop or a personal machine at home.

  `http://www.cis.upenn.edu/~bcpierce/unison`

- Sources, documentation, test scripts, and everything else edited by human fingers should be kept under source control. As of mid-2008, Subversion seems to be the tool of choice, but tools such as darcs, git, or CVS are also acceptable. *Code committed to the repository should compile and pass regression tests.*

  Try to develop the habit of comitting your work to the repository fairly often—every few days is a good target to aim at. If you don't think about it, it's easy to let a whole semester go by without committing your work. If other people don't see the work and can't use it, there's less chance of it being carried on after you go.

- To build internal software, we use `mk`. For external distributions, we use `make`. Makefiles must be simple; GNU Make is not an acceptable alternative. GNU autoconf is evil and wrong.

- For building tools written in Standard ML, we have an uneasy truce with SML/NJ and its Compilation Manager, which we extend as needed. We prefer Moscow ML and MLton, but they aren't always practical for large programs.

- We use LaTeX and BibTeX for documents.[3] God help us.

- I encourage you to use `nbibtex` (`http://tinyurl.com/685y7r`), which helps work around the sheer ornery arbitrariness of standard bibtex keys.

- Standard ML code should use the capitalization conventions of the SML '97 initial basis:

`all_lower`	for types and type constructors
`ALL_CAPS`	for signatures and datatype constructors
`mixedLower`	for functions and values
`MixedUpper`	for structures and functors

  Never use `open`.

[3] The "Collection of Computer Science Bibliographies" is a useful source of BibTeX entries.
`http://liinwww.ira.uka.de/bibliography`

- Objective Caml code should use the Caml capitalization conventions:

`all_lower`	for types, type constructors, functions, and values
`ALL_CAPS`	for signatures (normally just `S`)
`Capital_words`	for structures, functors, and datatype constructors

  Objective Caml code avoids identifiers with `InternalCapitals`.

  As in Standard ML, never use `open`.

- C code should be strictly ANSI conforming, without `#ifdef`. Naming conventions should be those of Hanson's *C Interfaces and Implementations*. Any consistent and readable layout style is acceptable. Major builds should be tested with all available C compilers (e.g., `gcc`, vendor's `cc`, and `lcc`).

- It's a character flaw, but I'm allergic to Perl. I can tolerate it if I have to, but I'm much happier with scripts that use `sh`, `ksh`, `awk`, `sed`, and even Lua.

- We use Debian Linux (testing) as our development environment. It's what allows us to use a huge range of tools without creating a systems-administration nightmare. At minimum, you will want to learn how to use the package-search tools and `aptitude`. Unless you know exactly what you're doing, avoid `apt-get`; `aptitude`'s user interface may be confusing, but it does a much better job of updating your machine without breaking it. I'm happy to give anyone a short `aptitude` tutorial on demand.

Authorship

Authorship is the most important form of credit in the academic world. All researchers employed in whatever capacity should be able to expect authorship credit for their contributions. Here, in no particular order, are some thoughts about authorship.

- For a student working on a project under my guidance, I expect that the typical case is that we write a joint paper with the student as first author. I expect such a student to put in substantial work on the manuscript as well as the project.

  I have worked with students who identified and solved problems on their own, with little or no technical support from our research group or our shared infrastructure. When this kind of work is submitted for publication, the student is the sole author—even when I advise the student, it would be inappropriate for me to be a coauthor.

- For a major group project that spans several years, students or others may make significant contributions without taking on major responsibilities for the project. Most often these contributions take the form of implementation work. It is important that these contributions be recognized with authorship credit, but such authors will typically be listed last.

- A student who joins a project already in progress might or might not wind up as a coauthor—it depends on the contribution the student makes.
- When two or more people share major responsibility for a project or a paper, it is usually obvious who should be the "senior" author, i.e., the first-named author. If it is not obvious, it falls to the junior or second-named author to make this clear, e.g., by saying "I think you should be the lead author on this paper."
- If other things are equal, when a junior person works with senior people, the junior person should be the first author. When faculty work jointly with a student, the student should be first author unless there is a good reason otherwise.
- Don't overlook past contributions of someone who has moved on to another lab by the time a paper is submitted; such people still deserve authorship credit. I have been on both ends of this mistake, and it is one people remember for a long time.

Graduate life cycle

If you're a graduate student, you'll spend some time adding depth and breadth to your knowledge of computer science, but the main thing you're here to do is become a researcher. A good researcher needs three critical skills:

- To solve a problem that has never been solved before
- To identify what new problems are interesting enough to be worth solving
- To describe a problem and its solution clearly and convincingly, both in speech and in writing

A big part of my job is to help students find opportunities to work on problems, to talk and write about them, and to watch other people doing the same.

Working on and identifying problems Most strong faculty candidates seem to have worked on about three different projects.

- When you first come in, we'll try to find you an "immigration project" where we see a clear problem and have a pretty good idea what the solution looks like. This might be a small problem we have a kicking around or a larger one on which work is already in progress. With luck, this problem will get you a quick taste of research and a publication.

- After an immigration project, different students take different paths. If you know what you want to do next, great. If not, we'll hunt for a problem that looks interesting, where we believe a solution exists, for which you can take primary responsibility.

- The final stage is for you to find your thesis problem. Finding a thesis problem is, in my opinion, the most difficult part of graduate school. Finding a good problem is always difficult, but for a thesis problem, there are special constraints. Although we don't have to know what will come out, we have to be confident that *something* will come out. You have to do something solid relatively quickly (e.g., two years), but your thesis work should also be in an area you can continue to work on for several years afterward.

To identify good problems, I rely heavily on my peers in the research community. Whenever I travel to give a talk or go to a conference, I find lots of people to talk about ideas. By hearing others' ideas and talking about my own ideas, I develop my sense of what people think is interesting. You will learn to do the same—it is an important reason to go to conferences.

Speaking and writing When you look for your first job, and later if you try for tenure, you will be known by your conference talks and your published papers. When I prepare a talk or paper, I remember that there will be people out there who will know me *only* by that one talk or paper. (This approach is one reason I publish comparatively few papers. I try hard to get them in the best places, because I've found it is almost as much work getting a paper into a small workshop as a top-flight conference.) I plan to develop significant resources at Tufts to help you prepare talks and papers.

- At some point my students and I will start meeting with one or two other professors and their students for a weekly talk. These talks may range from very informal "blue-sky" talks to relatively polished practice talks. It can also be quite helpful for you to give a talk about someone else's paper you have been reading. The meeting should provide a friendly place where you can get some practice giving talks in a low-key setting.

- When a student gives an external talk (conference, workshop, visit, or whatever), we start by working together on the presentation. The next step is a focused practice talk with detailed feedback. We also invite selected people from other groups at Tufts. These practice sessions can be grueling but invaluable.

- On my web page, I've collected some suggestions about preparing and giving talks.

- To discuss writing and help improve skills, I have from time to time organized a writing group, which has met once a week for 90 minutes, usually to discuss a short text written by one of the participants. (For details, see my manual for "Teach Technical Writing in Two Hours a Week" at `http://tinyurl.com/6nry8l`.) Sometimes this discussion leads to research ideas. It is possible that I may offer a course in technical writing; if you are interested in such a course, please let me know.

- For questions of style and usage, I turn to the *Chicago Manual of Style* and to the fine books by Fowler (*Modern English Usage*, in the *second* edition, not the third) and Garner (*Modern American Usage*). Every writer has pet peeves; my primary peeves are that I insist on the correct use of the hyphen and on author-date citations. I also frown on the use of the word 'this' as the subject of a sentence.

Sponsored research

Funding for research students (as well as my summer funding) normally comes from "sponsored research." This category typically includes grants from government agencies like the National Science Foundation, contracts with industry or with agencies like DARPA, and the occasional outright gift from industry. Different agencies want different things in return for their money. DARPA, for example, often wants software, documents, and other "deliverables" as well as command performances at conferences and other meetings. I try to get most of our funding from NSF, because what NSF wants is topnotch research results, pure and simple.

Part of your job as a research student is to help make sponsored research successful. Ronald T. Azuma has written at length on this topic, in his essay "So long, and thanks for the PhD!," and I quote him here.

> Academia is a business, and "graduate student" is a job title. This is especially true at private universities. Academia is very peculiar type of business. It is certainly not the Real World and does not work in the same way that the ordinary corporate world does. However, it is a business nonetheless and as a graduate student, you must treat it that way. Graduate school made a lot more sense and became much easier for me after I realized this. If you think of graduate school as an "Ivory Tower" free of politics, money problems, and real-world concerns, you are going to be severely disappointed. If you don't believe me, read *The Idea Factory* by Pepper White for one account of graduate life at MIT.
>
> A few graduate students are independently wealthy or have fellowship and scholarship money that cover all their expenses for their total stay in graduate school. Such students are rare, however. Most of us needed financial support, in the form of Teaching Assistantships or Research Assistantships (RA's). In general, RA's are more desirable to students since those can directly fund the research you need to finish.

Where does the money come from to fund RA's? Your professors have to raise funds from external organizations. These include government agencies such as the National Science Foundation (NSF), Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR), and others. Private companies also fund some university research, although this tends to be less common, in smaller amounts, and in the form of equipment donations. These organizations don't just give money away as charity. They expect their money to accomplish something. Increasingly these days, this takes the form of a contract for a working demonstration that must be shown at the end. That means once the money is delivered, your professors must come through with the working demonstration. It is rare that they do this by themselves. Instead, they find some very capable, young, self-motivated people who are willing to work long hours for small amounts of pay. In other words, they fund RA's.

The RA job is crucial to the academic business. If the RA's cannot successfully conduct the research, then the demonstration will not work in the end and the funding agencies may not be happy. They may choose not to fund your professor in the future, which will bring his or her research program to a halt. And there are many professors and other researchers chasing too few research dollars these days; it is a competitive market. Thus, each professor wants the best students available. These students are the most capable ones who can get the research done required to fulfill the funding contracts.

That means you must treat an RA like a job. You must prove to your professors that you are capable of getting the work done, being a team player, communicating your results, and most of the other characteristics needed to do well in regular jobs. That's why many of the upcoming sections in this guide sound like ones written for the regular workplace.

What do you get out of this? At the start, you may have to do tasks specifically related to the funding contracts. But eventually your professor must be flexible enough to fund your own specific research program that leads to the completion of your dissertation. Your stipend and tuition waiver should be enough to live on frugally without going into debt. You will learn the state of the art in your chosen speciality and conduct cutting-edge research on a subject that you find interesting and enjoyable. If you don't find this compensation sufficient, then you shouldn't be in graduate school in the first place.

The bottom line: realize that academia is a peculiar kind of business and the role you play in this enterprise. If you do your job well (and have good negotiation and interpersonal skills, as discussed in future sections), both your needs and your professors' needs will be met. But don't enter an RA position thinking that the computers, research equipment, staff members and other resources that you are provided with are your birthright. Don't take them for granted! Most of those exist only because your professors have been able to raise the money to provide those to you. In turn, you must fulfill your end of the deal by doing great research with those resources. If you don't do your job well, don't be surprised if your professors choose not to fund you in the future. They do not have to provide you with an RA job or let you use the computing equipment they acquired. And the student who has no funding, no tuition reimbursement and no access to required computing resources is the student who leaves the university that semester.

My responsibilities as advisor

Funding I cover wages, stipend, or salary, and also materials, travel, publication fees, etc.

Project definition I help with

- Goals and relevance
- Time span
- Approach
- Initial set of references for bibliography

Work environment I provide space, equipment, computer accounts, software, etc. I will also provide pizza to anybody who is working late in the lab and gets hungry. Just send me the bill.

Guidance I will

- Be available for meetings (scheduled and unscheduled) and to help with software problems in the lab.
- When new skills are needed, teach them, find other instruction, or develop a plan for self-teaching. Follow up to check that skills were learned.
- Help plan experiments, coding, testing, and path of project.
- Discuss results and analysis.
- Help over roadblocks.
- Constructively criticize writing and presentation skills as part of professional development.
- Evaluate your work thoroughly and carefully.
- Encourage when things go badly; praise when things go well.
- Criticize only actions, not people.

- Require challenging goals to be set and met.
- Help frame questions, and give students opportunities to find answers.
- Give students experience in interacting with sponsors.

Publication I will

- Suggest when and where results should be published.
- Provide opportunities to present results at national meetings.
- Outline papers with students.
- Correct or make suggestions about drafts — critical review.
- For undergraduates, write draft of paper (if desired).
- Help students respond to reviews of submitted papers.

Proposal, thesis, or dissertation I will

- Provide time and schedule for writing
- Critique student outline
- Critically review drafts
- Help form committee

Preparation for permanent employment I will

- Give you a view of the financial and administrative sides of research
- Help with job-hunting
- Write letters of recommendation[4]
- Nominate you for appropriate awards and fellowships

Proposals

Whether it is required or not, a thesis proposal is a find idea. A thesis proposal can head off potential problems and avoid having a student invest lots of time in avenues of exploration that have low probability of paying off—or that have already been explored by someone else. A review of the proposal should also keep students from biting off more than they can chew. When a proposal is approved, it typically binds the faculty supervisor and committee, *not* the student. That is, if you do everything in your proposal, we promise to grant you your degree, *but* it is often not necessary for you to do everything in your proposal in order to earn your degree. As you undertake the research you may well identify more

[4] If you are a graduate student, I will not write a letter for you until I have seen the first draft of your thesis. This policy protects you from trying to finish a thesis on an impossible deadline, and more importantly, it helps ensure that when you go onto the job market, you are properly prepared.

interesting problems or solutions that result in an even better thesis than the one you proposed—or you may find a problem you proposed intractable, while a related problem yields interesting results. These are acceptable outcomes; it's all part of doing research.