

Tutoring in the Computer Science Department

Don Colton, CS Chair, Brigham Young University Hawaii

May 17, 2004

Tutoring is an important activity in the Computer Science department at BYU Hawaii. The first section of this document looks at tutoring from the student's point of view. The second section looks from the tutor's point of view.

1 A Student's View

As a student in a computer science course, it is likely that you will learn many new things. And it is likely that you will be expected to learn things that you **did not actually learn the first time** they were mentioned in class. There is so much to learn, and it is often so very different from things you have learned before, that it may take a while to become clear.

The CS department provides tutoring in GCB 101. There is a tutor on duty throughout the afternoon and evening. These are the current hours. (You should check to see if the hours have changed, in case you are reading an old copy of this sheet.)

Computer Science Lab (GCB 101) Tutoring Hours

Monday	3:00 PM	to	11:30 PM
Tuesday	3:00 PM	to	11:30 PM
Wednesday	3:00 PM	to	11:30 PM
Thursday	3:00 PM	to	11:30 PM
Friday	3:00 PM	to	8:00 PM
Saturday	3:00 PM	to	6:00 PM

When the lab is open for general use, and a tutor is on duty, the door to the lab must be kept open. When nobody is in the lab, the door must be kept closed.

On Monday through Thursday the lab may stay open later than 11:30 PM if students are working and the tutor is willing. The final deadline for closing the lab is 1:00 AM. The decision to stay open past 11:30 rests solely with the tutor.

The lab is generally available during the day, starting at 6:30 AM, but without any tutoring support.

You can use it whenever it is open. Sometimes classes are held in the lab. When that happens, class members will have first priority for the machines. Extra machines, if any, are available for your quiet use.

A tutoring area has been established at the front of the room. There is a large table with the "tutor computer" and several chairs. When the tutor is on duty, he (or she) will probably be at that table or helping someone in the room. This makes the tutor easy to find.

If you have a question about something on paper it may be easiest for you to sit next to the tutor and request assistance. The tutor will be happy to explain concepts to you and show you how to work problems on the sample quizzes.

If you have a question about a program you are writing, it may be easiest for you to sit down at one of the computers in the room and display your program on the screen. Then the tutor will be happy to come to your workstation, review your program, and determine what you may be doing wrong. The tutor will point out those parts of your program that seem wrong and tell you how to fix the errors.

Sometimes a tutor will be unable to figure out exactly what you are trying to do. You may need to explain your approach. Because the tutors are advanced students they can generally understand what you are trying to do and how to fix it. Because you are unique, you may have an approach that they had not considered before, so it may take a minute for them to see what you are trying to do.

Rarely the tutor will find himself unable to explain a solution to you. In such a case you may need to talk with another advanced student. There are generally a few advanced students next door in GCB 103 (the "rat lab") who are willing to spend some time helping you. If your needs are beyond them, you are stuck. You will just have to visit directly with the teacher.

Of course, you are free to go straight to the teacher anyway. Tutors are intended as a convenience. But

for easy questions, the tutors are often your best bet. And teachers, although friendly, do tend to drone on and on and on, just like this paragraph.

Things to NOT expect: There are a few things you should not expect from the tutor. They are there to help you learn and understand the material. Occasionally a tutor will report that some student seems to expect the tutor write his program. In fact, we do not allow the tutors to write any part of your program. They are not even allowed to touch your keyboard or mouse unless your computer is jammed and they cannot explain to you how to unjam it. Even then they are limited to just the minimum involvement to get you safe again. In no case can they type in a program or fix programming mistakes for you. They may tell you what to type, if it is short. They may write a similar program on the white board. They may write a program on the “tutor computer” and show you how it operates. But they are not allowed to type anything on your computer. They are limited to pointing with their finger and telling you what they think you should do. It is your job to decide whether to take their advice and how fast to let them talk. You should be in control.

The tutor’s job is not to get your work completed, or to “do it for you.” The tutor’s job is to **help you understand** so you can complete your own work. (They get in big trouble if they go beyond that. It is considered an honor code violation.)

Please Come! We are delighted to provide a tutoring opportunity for students in CS classes. We try hard to provide coverage at convenient hours. We hope you will find this service useful. If you have comments or suggestions, please share them with one of the teachers or send email to me, don@colton.byuh.edu. Thanks!

2 A Tutor’s View

This section looks at tutoring from the tutor’s perspective. You should also read the first section so you know what we want the students to experience. Then read this for some of the nuts and bolts that go on behind the scenes.

If you are an ordinary student **not** planning to be a tutor any time soon, reading this will be like sitting through a boring orientation meeting. You have been warned.

2.1 Our Mission

Your mission as a tutor is to use your best efforts to support the CS department in educating students. Your primary responsibility is to help students understand how to do their work, without doing their work for them.

2.2 Your Preparation

To be a tutor, you must be able to help the students who come in. Typically this means you must have completed CS 101, CS 201, CS 202, CS 210, Math 201, and Math 202. If you completed alternate versions of these courses, or skipped some, you will probably need to go through the current course outline and work the problems that the students are working. This should give you the background to be helpful.

2.3 The Pay

Tutoring is an on-campus job. It pays typical on-campus job rates, which is to say far less than we think you are worth. This is balanced, of course, by the fact that tuition is much lower than it would be elsewhere, and the whole campus economy is designed to help us keep our minds off of the money. There are some imperfections in the system, but generally it works pretty well.

The **real pay** is, of course, the experience that you receive. It is a very enlarging experience to help another student write a program. First, you have the experience of helping another person meet their goals. This is a good activity in itself, if the goal is worthy. Second, you have the opportunity to see another approach to a problem. Often you will be forced to look at a problem in a different but correct way. For most programming problems, there are many correct approaches and solutions. This will expand your outlook. Third, you have the opportunity to see and fix mistakes that you might never make yourself. This will improve your own programming skill. The teacher (or tutor) always learns beyond the student. Tutoring is a way to become really good at programming. It is a great job.

2.4 Hours per Week

Tutoring is not a 19-hour-a-week job like some you might find on campus. It is an as-needed job with a 19-hour cap. Currently as-needed means that primarily you are paid for just being there to help the

students. That pay is for about 16 hours per week, and starts the first day of class and ends the last day of class. It does not exist during finals week. It does not exist during the breaks between semesters. It does not exist during summer term.

The tutors are responsible as a group for setting their own hours. The overall requirement is to provide coverage according to the time schedule shown in the first section: 3 PM onward, Monday through Saturday. The department has arranged funding to cover one tutor at a time over all of these hours. Each semester the tutors meet to decide who will cover which hours, and the resulting schedule is returned to the CS chairman or his designee and posted on the GCB 101 door for students to see.

There are some opportunities for additional hours. An energetic tutor who wants a 19-hour job can probably find activities and start approved projects that will provide those additional hours. It requires the tutor to take on additional projects, and this usually happens when the tutor himself has a suggestion and gets it approved as a project. If you have an idea, do not be timid.

2.5 Typical Tutoring Activities

Tutoring itself is fairly simple. As mentioned in the mission section above, your mission is to use your best efforts to support the CS department in educating students. Your primary responsibility is to help students understand how to do their work, but not do their work for them.

Tutoring can be proactive or reactive. In the simplest reactive case, you just sit at your station (the “tutor computer”) and wait for students to approach you. When they approach, you direct your attention to the problem they bring, and you give them instruction and advise on solving that problem. In the proactive case, you greet students as they enter, occasionally get up and wander through the room checking to see if anyone needs or wants help, and you may even invent group opportunities, such as organizing a “Malama session.”

2.5.1 Be On Duty

The minimum acceptable standard of performance is that you are present in the CS lab during your assigned hours, and available to students. It is **not** appropriate to run errands during your on-duty time, even if there is nobody around that needs your attention. Like a guard at a military checkpoint, you

should not leave your post. You must be there. If for some reason you must leave for a few minutes, for example to go to the restroom, you must make every effort to get a temporary tutor to sit in your place while you are gone. Usually a temporary tutor can be found in the “rat lab.” Someone must be on duty.

Occasionally there will be a real emergency. Maybe you get a phone call that your daughter has broken her leg or something, and you feel you must go to take care of this more important responsibility. In such a case you should either secure a replacement or immediately communicate with your supervisor (probably Bro Colton) by telephone or email telling him your situation and that you are leaving. At that point you can go. You should not go without leaving word.

In a similar vein, you may be responding to a real emergency that prevents you from arriving at work on time, or causes you to miss your agreed-upon shift. You should make every reasonable effort to get a replacement and/or to inform your supervisor at the earliest possible time. After the emergency is over, you must make a full report to your supervisor indicating the nature of the emergency, your efforts to secure a replacement for yourself, and your related plans for the future.

When I say that “be on duty” is the “minimum” acceptable standard of performance, I mean it. To be absent without leave is a sufficient reason to be terminated from this employment.

2.5.2 Try to be Helpful

After simply being there, your next task is to try to be helpful. Students appreciate your efforts to be helpful. Students complain when you ignore them. Being helpful requires that you direct your attention to the problems and questions that the students bring. If you cannot solve their problem, you should at least help them find someone who is better able to help them solve their problem.

2.5.3 Helping with Sample Tests

In the case of sample tests that are provided for many of the classes, you are permitted to work through the problems in any level of detail that seems appropriate to you. Students do not get any credit for doing the sample tests, so by giving them answers you are not committing any sort of honor code violation. The students know that their task

is to prepare for the real test, the one at the testing center, so their practice-test goal will be understanding.

2.5.4 Helping with “I Don’t Understand”

Another scenario is the student who brings his book, or just brings a question, and says something like this: “I don’t understand subroutines. Can you explain it to me?” This is an open-ended challenge that requires you to explain a concept which you yourself understand to someone that has desire but does not yet understand. There are no easy rules for how to do this. You are being a teacher. You can follow the examples of teachers you have known. You can draw diagrams on the board. You can ask the student questions to help them stay focused as you lead them to the answer. Helping them understand is the most difficult and most rewarding activity you will do.

2.5.5 Helping with Programming

Many times a student will need help with a program. Their program can range from a total mess to something that almost works. If the program almost works, your task may be to explain the error messages and show where and how to fix the errors. This is an acceptable activity, even if you point to the end of a line and say, “you need a semi-colon there.” The key thing is that the student did their own work. You are just helping them fix it.

2.5.6 Whose Algorithm to Use

Sometimes the program does not work, but the student has done substantial work. There are no syntax errors, but the program does not work right. You may need to understand what the student is doing to give them advice on how to fix it. Read their code. If you cannot understand it, challenge the student to insert some documentation (comments) that provide signposts for their intentions. Have them explain their algorithm to you. **DO NOT EXPLAIN YOUR ALGORITHM TO THEM.** At least not yet. The goal is **not** for them to do it **your** way. The goal is for you to help them do it their way. If the code is difficult, you may ask them to print a copy of their program. Then you can sit with them at the front of the room and write notes on the paper, and draw lines and circles and arrows that help you understand and discuss the operation of the program.

2.5.7 Trace-based Debugging

Often you can help debug a program by having the student add a large number of **print** statements that show the values of variables they are using. When the program runs, these **print** statements produce an execution trace. The trace shows what is really happening. The trace should be examined to see whether the variables are behaving in the way that is expected. This can often identify the formula or statement with the bug.

Some GradeBot labs allow students to include debugging print statements in their output. You should know how to do this.

More advanced students will learn how to use a programming debugger, such as **gdb**, that allows them to run their program and get a trace without the extra effort of adding **print** statements.

Other advanced students will add a global variable into their program, maybe **debug**, and set it to 1 when a trace is desired and 0 for normal operation. Each **print** statement then becomes **if(debug)print** or **if(debug>3)print**. Many production programs in the real world contain such lines to aid in debugging when the program is modified.

2.5.8 Helping with the Total Mess

If the program is a total mess, you may need to sit with them and help them organize their attack on the problem. Go to the white board and plan that attack. **DO NOT TYPE CODE INTO THEIR COMPUTER.** Typing code into the student’s computer is a termination offense and possibly an honor code violation. Students must do their own work. Your task is to help them achieve understanding but they must do their own work.

2.5.9 Helping with Lots of Students

Giving help properly can take a lot of time. You may end up with other students waiting for your attention. If you are spending a lot of time with one student, it is perfectly acceptable to handle short questions from other students on an “interrupt” basis. You should avoid making students wait too long for attention.

2.5.10 Helping with a Difficult Student

If a student is taking a lot of your time, and you are getting behind on helping the other students, you must make a decision. Your assignment is to

satisfy as many students as possible. You may decide that the problem is so difficult you cannot spend any more time on it at that moment. You are permitted to tell the student that “this is a difficult problem, and I think you need to talk to your instructor or another CS faculty member about it.” By saying this, you are saying that you are too busy to help. Make sure it is true. Do not send a student away simply because you do not understand what they are doing, so you are giving up. Send them away because you have other students who need your attention. Perhaps you can send them back to their computer to add comments to their program and try it again while you deal with some of the other students who have problems. See how cleverly you can use your time wisely. See how helpful you can be.

2.5.11 Using the Rat Lab

GCB 103 houses the “rat lab.” This is a special computing experience for advanced students (and sometimes newer students) to give them a place to be together and develop into a group of colleagues. They also have an explicit mission to act as backup support staff to the tutor on duty. Being a rat-lab student carries with it an obligation for service.

This means that when you are occasionally overwhelmed with students, you may step into the rat lab and ask one of your friends there if he or she can help a student for a few minutes. Then send in the student. Do not worry that you are imposing on the rat lab student. Helping out occasionally is part of the rent they pay for the privilege of being there. But also be careful to not overdo it. Occasional helping is expected and required. Frequent helping is not expected, required, or fair, as the rat lab students also have their own homework to do. But if someone in the rat lab is just playing a computer game

2.5.12 I’m Always Too Busy

If you discover there are too many needs for assistance, you should notify your supervisor and suggest a solution. It may be that we should schedule two tutors during certain hours. Let us know.

2.5.13 Helping the Wrong Way

Yes, it is possible to do tutoring in the wrong way. In the past we have had more than a few cases where students went to the tutors day after day asking for guidance in doing their homework or labs. The

tutors in these cases were guilty of simply giving the students pieces of programming that would contribute toward the finished assignment. A brick here and a brick there and pretty soon the house is finished. The students in these cases were actually having the tutors totally write their programming labs while the students only gained the “people skills” of getting others to do their work for them. When the final exams came, these students failed, although their lab work indicated that they could do everything that was asked in the class. This is not a happy outcome. The worst part is that the instructor never knew that the student was cheating in this way. Therefore, when you provide actual code to a student that can be used substantially as-is in completing the program, you have committed an honor code violation and can be terminated from your tutoring job.

2.6 Beyond the Minimum

As stated above, your mission as a tutor is to use your best efforts to support the CS department in educating students. You can be proactive. There are many activities you can and should do during your spare time.

2.6.1 Doing Your Own Homework

If you have nothing else to do, you are allowed to do your own homework while you are on duty. If you do this, take special care to greet all students who enter the lab and make sure they are aware that you are there to help them. Students have sometimes complained to me that the tutors appeared too busy with their own work so they did not ask any questions. Make sure this is not true about you.

2.6.2 Web Reviewer

A better use of your time is to look for ways you can improve the CS department. Maybe you can surf the CS web pages and find links that are broken or things that are out of date and need correction. You can send email to webmaster@cs.byuh.edu with these suggestions, corrections, or notifications that there seems to be a problem even though you may not have a solution for it.

2.6.3 Webmaster

You may have enough interest to become an assistant webmaster, or even the primary webmaster for

the CS department, in addition to your tutoring responsibilities. This would give you something to do during those slow hours when no students are seeking your assistance. It could also work into hours on the clock between semesters when there is no tutoring to be done.

2.6.4 Malama

When you find you are answering the same question over and over again, or there is something about an assignment or a chapter with which students are having trouble, you may want to run a Malama help session to reach many students at the same time. The “tutor computer” is connected to an overhead projector that is sometimes used during class. You can use it. You can take control of the GCB 101 lab at pre-arranged times to run a study group on a particular topic. By pre-arranged, I mean that you have cleared it with Bro Colton and advertised it to students, maybe by making a posting on the lab door. You may even arrange a standard weekly session for students in the same class.

2.6.5 Other Things

The best way to get these jobs and others is to volunteer. Do not wait to be assigned. Take responsibility for using your time wisely and suggest activities or tasks that you want to do and that would be good for the CS department. It is not good to be commanded in all things. I mention this again at the end, under Initiative.

2.7 Staying Open Late

The first rule is: whenever the lab is open, the door must be open. This applies to all times when a tutor is on duty.

Tutors have reported that closing the lab at midnight (our old policy) was often difficult because students were diligently working on their labs and the tutors were diligently involved in helping them. To resolve this problem we revised our policy.

The tutor on duty must keep the lab open until 11:30 PM, Monday through Thursday, even if there are no students present. Between 11:30 PM and 1:00 AM, the tutor may close the lab, lock the door, turn out the lights, clock out, and go home when there are no students working in the lab. Even if there **are** students working in the lab, the tutor has the discretion to decide whether to close the lab or keep it open starting at midnight. The door should be

closed and lights out by 1:00 AM under any circumstances.

One of the worries on campus about late hours is the “making out” problem. The honor code office has seen plenty of cases of students spending time with their significant other in dark and deserted classrooms late at night. I myself have walked in on students vigorously making out on the floor of a campus room. Because these issues have existed and will continue to exist, the campus administration and myself are very interested to make sure we take due precautions against any improper use of the privacy afforded by the CS labs. That is one reason the **lab doors are to be kept open when people are inside**, and why many people know the combinations to the doors, thus decreasing the expectation of privacy that some might feel. Avoid the appearance of evil. Avoid temptation. Doors must be kept open at night.

2.8 Your Supervisor

The tutors are supervised by a designated member of the Computer Science department faculty. Currently that supervisor is the CS Chairman, Brother Colton. The supervision style will probably vary according to who is supervising.

2.8.1 Communication: Reporting In

In all of your employment activities, communication is important. If you were digging a ditch or building a road it would be easy for your supervisor to see your value and achievements. In a job like tutoring, your product is often invisible. If you do not report your work, you are also invisible.

I (Bro Colton) expect reports weekly or more often. You can accomplish this by personally visiting with me or by sending me an email mentioning your accomplishments. Let me know what is going on. What is going well? What problems or needs or opportunities are there? What suggestions do you have? What proposals do you have?

2.8.2 Initiative: Beyond the Minimum

As suggested above, you can propose activities beyond what is required. Web assistance. Malama. Update the bulletin board. Research job openings for graduates. Research OPT opportunities. Work as a grader for one of the instructors. It is your responsibility and opportunity to make yourself more valuable by finding things that you think need to be

done. Then volunteer to do them as part of your job.

Some tasks require more than a worker. Some tasks require a “hero” to go forth into battle and lead the charge. When such tasks are merely assigned, the results are often disappointing. But with a volunteer that has vision, the results can be amazing.

The most valuable employees are those that can see beyond the “minimum acceptable standard of performance” and develop new services to improve life for everyone around. Be that person.