Paired Programming Project: Focus Groups with Teaching Assistants and Students

November, 2002

Miriam Ferzli
Eric N. Wiebe
Dept. of Mathematics, Science and Technology Education

Laurie Williams
Dept. of Computer Science

NC State University

I. Introduction

By implementing paired programming protocols in an introductory level computer programming course (CS1), a group of researchers at NC State University analyzed the effects of paired programming on students and instructors[1]. Paired programming is a practice in which two programmers work collaboratively at one computer to generate designs, algorithms, or codes. It mimics real-world scenarios for computer programmers, since a lot of computer programming is done as part of a team. Paired programming is grounded in cooperative and collaborative learning methods, during which students interact to discuss and solve problems.

This report documents research efforts, funded by a National Science Foundation grant (DUE CCLI 0088178), to evaluate the pedagogical effectiveness of the paired programming protocol in an instructional setting as well as to determine how to improve the protocol for future use. Students enrolled in the CS1 course at NC State University during the 2001-2 academic year followed the paired programming protocol during lab sessions associated with this course. Teaching assistants (TA's) ensured that students followed the paired programming protocol. At the end of the semester, researchers randomly selected students and TA's for voluntary participation in two focus groups—one for TA's and one for students.

During the focus groups, students and TA's responded to open-ended questions that were part of a semi-structured focus group interview. Responses were audio-taped for later transcription and analysis (participant confidentially was maintained at all times). Data analyses of these responses revealed numerous themes that are discussed in sections II (student responses) and III (TA responses) of this report. Overall, student and

TA responses offered an insider's view about what works and what needs improvement when implementing paired programming in instructional settings.

II. Emergent Themes: CS1 Student Focus Group

Student focus group data revealed eight major themes: Pairing and Learning; Partnerships' Negative Dynamics; Communication;, Mixed Abilities, Background & Personality; Forming a Partnership; Pairing & Real World Scenarios; Responsibility & Accountability; and Switching Roles. Students' comments show that overall, students experienced many positive effects from the paired learning setting. The few negative points or disadvantages of pairing were off set immensely by students' accounts of enhanced learning as a result of pairing. Students seem to think that the paired learning arrangement works, but they are unclear about its protocol. Issues about how to choose a partner, how long driver/navigator roles should last, and how often to switch partners seemed to dominate students' comments.

Students stressed the advantages of pairing throughout the focus group discussion. Primarily, students brought up issues about being able to ask questions immediately as problems arise rather than having to wait for an instructor. Having someone there while working on problems seemed to help students clarify ideas, pick up on minor errors, and work on understanding conceptual knowledge. Students mentioned that pairing is particularly beneficial when working on projects, because two people often complement each other by sharing strengths and filling weaknesses. Project partnerships came up in the discussion as an important aspect of learning especially for students that are lagging behind despite continued efforts.

Students also emphasized that the secret to the success of pairing is finding someone that feels comfortable with pairing. When students work well together, the partnerships seem to be successful, even if a low-ability student is paired with a high-ability student. Students also discussed the importance of taking responsibility for their own learning. They claimed that students who have no interest in learning do poorly or fail regardless of classroom arrangements. Students also found that the paired setting mimics real world settings where people are often randomly matched to work together on programming projects. Communication skills and collaboration are important components of paired learning, which the students mentioned in their comments as added benefits that will help them succeed in the future.

Themes:

1. Pairing and Learning

   I really see [pairing] as positive, because what you don't know, your partner or the person you're working with might know. And sometimes you overlook things that you normally don't see.

   …little mistakes, like I was working with my partner, and a little mistake I couldn't find, that didn't let the program work, my partner figured it out.

   I think that if you get the right partner that is used to work with partners, then pairing is a very good thing and it helps you learn.

   Like uh, for example, I had a partner that learned to program in a way that was right, and that helped me to learn the way to program.

   If one of us doesn't know something, the other one probably does. We work it out, and we've never had any problems with getting any of our programs done.

   …other people bring their own ideas, like how they want to write a program. If it's different than yours, you're going to have to say, "OK," and compromise with them. Say, "I like how you wrote this, but not how you wrote that." So, in the end, you'll end up with a better program.

   I really like working in partners, because it helps a lot, because you get thinking in your own one track and you have a hard time getting off that track, but if you bring in another person, they may have different ideas. You may say, “Oh, that’s a good idea, let me think about that.” So it gives you just more options and flexibility, and I think it helps a lot.

Yeah, I think that it’s a good approach, but at the same time, I think that if someone is struggling, and is making below 70 or a bad grade, sometimes having a partner would help them out, if they were willing to try and learn…

…[pairing on projects is] actually really good, because if there’s something you don’t understand, your partner can work on that specifically and you can put the time into what you do well, and you can complement each other.

I personally learned. Like when I first got there, I didn’t know a whole lot of programming. I got paired with different partners that some of them knew what they were doing and some didn’t, so I had to learn theory. And now I pretty much understand the material, why I’m doing it. And I see myself getting better on projects and I could work by myself now. I couldn’t do it before, but now I could…

It’s definitely improved my skills. You can’t say it’s bad for you at all.

…when you’re learning and you don’t have any idea about what’s going on, if you have somebody to ask questions and stuff, then it would be easier to learn that way.

For me the whole experience has been rewarding.

Having a partner helped me most, because it meant that I don’t have to start from scratch when I’m working on an assignment. I have someone there to bounce ideas and questions. It’s important to have that kind of help, especially on the projects.

…Working by myself may have been better, because I would have had to think really, really hard…but with a partner, they’re going to know some parts that you don’t and it makes it a whole lot easier…I mean working by yourself, you have to stretch it… It would’ve gotten done, but it would have been much more of a struggle…It’s just more efficient I think.

The two-person group I think is good. I don’t find anything wrong it. The smaller the group the better, because with the whole class effort, you’re not going to think as much. You may just sit there and learn nothing.

I just find that having a second opinion really helped me to avoid common mistakes, and helped me to save time, and not get stuck as often.

…the partner, who you work with, they could probably explain it better than the professor can. Sometimes the professor is talking over you head, and you don’t know what they’re talking about.

I tell her that I think I know what the problem is that you didn't really have someone there, because, you know one-to-one is crucial to learn. If you have one guy for 30 people in a lab, he can only do so much. But if you have someone else there who might know more, you can definitely pick it up.

…let's say you have one TA for 15, 20 people, then it takes time for the TA to know your code, like how you coded and find out where the problem is… so it takes more time, you know, for people to wait, if you have no other friends or partners.

2. Partnerships' Negative Dynamics

…if you get a partner who is not used to working with partners or doesn't like to work with partners, he may want to control everything. Once I had a partner, and he had a bad temper, so he took over everything and I didn't find that I was learning much.

…when I'm working with a partner, I get lost, because my partners goes off into something and I don't always follow what he's doing. In the experiences that I've had, he's doesn't really tell me what he's doing.

Even though we're supposed to work on the same computer, a lot of times he works on his computer and I work on mine. Then understanding what he's done is really hard and figuring out which part of the program is mine, that's really confusing to me.

…when you work with someone who's not used to working with a partner. They want to do everything themselves and do not cooperate.

Some people like to work individually and you want them pairing and they're not cooperating with the process.

His partner is just doing what he wants, and didn't care about him.

The thing is that students are required to pair and some can't catch up with another, and they have to struggle.

Some people just don't want to work with anyone.

3. Communication

I guess when you're with a partner, you talk everything out and it takes a little bit longer. But putting it into words really helps understand.

Well, in order to know what your partner thinks, you have to communicate, right? I mean if you partner and you are not communicating, then it's not going to work.

With a partner, we could discuss what we were doing, and then figure out the coding together.

I just couldn't write the code in JAVA…So my lab partner for the first couple of labs, he knew C++ well and had mastered it well, and I asked him a lot of questions and like, learned a lot of stuff like writing JAVA stuff…

…JAVA is not complicated, but if you don't get it you won't be able to understand what's going on, except like if you ask questions to someone who knows JAVA a little or maybe more.

One time the whole class was struggling with the same problem, and the TA encouraged everyone to work together. And this was more than just a partner…and I learned a lot from just listening to the discussions taking place.

4. Mixed Abilities, Background & Personality

Even if they're not at your level, if they are lower or not as advanced as you are, sometimes they can learn. Because if you provoke them to explain why that is, explain to me such and such, they'll be able to learn the material also. And that's very positive.

I had a partner once, and like the way she thought about it was so different from the way I thought about it. And she made it so difficult that she would confuse me, and we ended up being confused.

I think it just depends on how you think with your partner. If the thinking process your partner and yours is similar, then the pairing works perfectly. But if you think differently, like totally differently, then I don't think that it's a good idea to work as partners.

…if you want students to work in pairs or groups, maybe give them a psychological test or something to figure out how they think, or some simple test. That way they can match people out, and that works out fine.

…the way that people think psychologically that's the best thing to match people up.

…figuring out their backgrounds and how does it measure to others' backgrounds and stuff. Like some students from here, they're going to think totally differently than students from other parts of the world.

…if you think the way that your partner thinks, then you'll work at the same pace. Like your partner will not work faster or slower if you think the same and stuff.

…a lot of people work hard and study the book, but suppose they don't get at least 70%? They would be able to do a lot better on their projects and everything if they paired with someone that did do over 70…if they're working with someone who's already grasped it, then they can learn it and continue to build on that.

I didn' t know anything about programming, and I was having a really hard time and the project was due so soon into the course… The partner I had knew a little more about what we were doing, and that helped me to figure things out.

I think that some people are more social than others. For those people, pairing is definitely a good help, but others like to work individually, and they work like that best.

5. Forming a Partnership

Well, random pairing is like, sometimes it works out and sometimes it's not going to work. So I don't think there's anything you could do about it. It's like getting a random roommate, sometimes it can work out but sometimes it doesn't. I think maybe trying to match up personalities or match up intelligence, or whatever.

…in pairing programmers, we should let the students choose to participate in pairing.

I would prefer to be able to pick who I want, because there are situations when you get stuck with someone that you just don't get what's going on.

…you get to know each other eventually, especially in lab, because it's so small. In lecture, the chances of getting to know someone are pretty small.

… I don't really know that there is a real effective way to pick a partner than just to randomly pick a partner. You don't know the people, you don't know their personalities, or whether they're going to take advantage of you or not. So it's kind of like trial and error, I guess.

…[projects are] good because you get to choose your partner, someone who works well with you.

Yeah, you should definitely be able to choose your partner on projects, because they're so crucial, a big chunk of your grade. And you don't want to get in a bad situation, like if both people don't understand the concept, then there's nothing you can do about it.

…on projects, you're on your own. So if you're able to pick someone that you know, from watching them work in lab, you can pick someone who knows their stuff and be sure that you can complete this project. If you're assigned someone, you might be in a situation where you've never met this person. You don't know how the two of you are going to work…

You try to pick people you know, people you've studied with. Like this girl I worked with, I met her in class, and we started talking and studying together.

Yeah, I mean, after a while of sitting in class, and you're both working, you figure out who's sharing what you think about trying to figure out the course.

You also ask about their grades, and that way you know who to pair with and who not to pair with.

I chose my partner based on the partnering in the lab section. The guy I got partnered with in lab, we worked well together, and that's why I chose him for the project, because he did such a good job as my lab partner.

Well, I want to say that I chose my partner, also because we worked in lab together and he knew the material well.

I think that if we switched more often, like for every lab, we would meet more people.

It would help you to get to know more people and find someone that you work well with and that you can pair up with for projects.

The problem is that, you know, if you work with them two or three times…the thinking process takes time to see.

I think people should be given the choice to pair or not, and then all [the] problems would go away.

I think that people should be able to choose the lab type they want (paired vs. non-paired).

When we talk about people not wanting to give up the keyboard and stuff, having it as an option would solve those problems.

I think that TA's should do both—have both paired and unpaired.

6. Pairing & Real World Scenarios
   I think that pairing should be forced on us, because in the real world, it's not a choice when your boss says, you have to work with him…You just have to work things out.

   …in the real world people are all mixed together and they're being paid. They sit down and say, OK this is the process that we're going to go by—they communicate.

   I think it's more realistic to work in this environment, because you have to learn to work with other people. I mean, you can't your whole life work alone. You're going to be put in situations, especially if this is your major and you plan to have this as your work field, you'll have to be able to work with other people.

Well, in the real world, we are forced to work with people that we don't agree with or get along with, so we should practice it. I think that we should have some labs for pairing and some for not pairing in order to make it work.

7. Responsibility & Accountability

   In school, people don't care. They're not being paid.
   …in school your GPA is going to determine what kind of job you're going to get. Like if you have a low GPA, you may get kicked out and not get a good job.

   …if you can't catch up, I mean, you have to take responsibility to learn the material yourself. And you have to learn to work with your partner and whenever your partner is finished, you take the work and learn what he or she did or put it in your own words.

   …sometimes you might get an advanced partner, and if he or she is leading you and it's too advanced when working on a program in lab, you have to take the initiative to make them slow down.

   It takes personal initiative. People are going to want to take on these roles, otherwise it doesn't matter who's at the keyboard.

   It's all about what a person wants to learn.

   Well, like I said before, people should be responsible for their own learning, so they should take the initiative to switch on their own.

   Well, it's not like this is Kindergarten…if you really want to learn, you're just going to make certain demands to be on the keyboard…

   …switching should be up to you. It's a partnership, and in some labs, I may be better as the navigator or the driver, but it should be left up to people to figure out what works best for them.

8. Switching Roles

   It's like when you switch off every twenty minutes…and I would ask my partner, "you know what I'm doing?" and he'll say yeah, yeah, but I just kept having to wait every twenty minutes until I can get back on it and continue doing what I was doing. That's frustrating.

   I think [changing roles is a] complete waste of time. It's not like one person gets on the keyboard and the other person is completely silent. The other person at the keyboard, if they don't know it, their still feeding on the other person…

   …[not changing roles] means that only one person is getting stronger while the other person is just sitting there.

> I think the time for switching is all wrong. I mean, 20 minutes is not going to work for everyone.
>
> Well, someone is always going to take the leadership.
>
> I think that we should switch, but not every lab. Maybe only half the labs.
>
> I think that if we're going to have switching, then the TA needs to really enforce it. If some people do it and some don't, then it doesn't work.
>
> Well, [switching roles] every twenty minutes is just too fast. Sometimes, that's how long it takes to even think about what you're going to do.
>
> Well he (the TA) switches the partners every two or three labs, but as far as switching roles, he doesn't enforce that at all.
>
> Me personally, I always get to use the keyboard. We switch, but not during the twenty minutes. We just take our time.
>
> I need thirty minutes or so [between switching]. For people who get done early, then twenty minutes may be a good time.
>
> Whenever people feel like switching, they should. That's what we've been doing.
>
> I think you should be told when to switch… Yeah, because that way everyone is ensured that they will get on the keyboard.
>
> I think that the TA should tell people to switch but let people take they're own time to switch.
>
> The good thing is that we switch partners, so you may not have to deal with that (a negative partnership) for a long time.

III. Emergent Themes: CSC 116 TA Focus Group

The comments that TA's gave during the course of the focus group revealed thirteen major themes: Pairing Protocol: Time & Efficiency, Student Interaction, Higher Order Thinking & Active Learning, Unpaired Labs, Reluctance to Pair/Switch Roles, Motivation, Student Compatibility, Pairing & Learning, Pairing Protocol Logistics, TA Responsibility & Evaluation, Project Pairing, Confidence, Future Success. TA's all agreed that implementing the paired protocol gave them flexibility and time that they could use to give students equal opportunities for questions, discussions, and other support. TA's also found their jobs fun and easy when teaching in paired labs. They observed that students interacted positively for the most part, and that partnerships helped students reduce major problems associated with programming. As compared to traditional labs, students displayed more active participation in their learning. Their questions were on a higher level than questions from traditional lab students, and their efforts and willingness to learn surpassed their "traditional" counterparts.

TA's observed that some students were reluctant to pair, but most students eventually experienced the advantages that pairing has to offer. The biggest issue with pairing students seems to be compatibility. TA's emphasized that partnerships work best when students working together are compatible. In these cases, students seemed able to problem solve with their partners with relatively little help from their TA's. Students who fell behind, were those that lacked the motivation to be active learners. These students relied heavily on their partners without making much progress in their learning. For this reason, some TA's argued that although pairing is beneficial to learning, there should be a balance between paired and individual work. This argument was especially focused on student projects. TA's seemed divided on whether pairing is beneficial to students working on projects. Some TA's noted that students would benefit immensely from pairing during projects and were puzzled by students who opted not to pair. Other TA's felt that projects should give students an opportunity to work alone so that they can feel confident when programming on their own.

Despite agreeing with the paired protocol and suggesting very little need for change, TA's suggested that frequency of switching roles between driver and navigator needs to be tailored to specific labs. Some labs are too short for students to switch very often. TA's also stressed the importance of their roles in their students' learning. TA's feel that no matter how much potential a teaching strategy or protocol may hold, its success is ultimately determined by the TA's teaching investment. Overall, TA's found the paired protocol to be very successful and worth considering as an alternative to traditional ways of teaching CSC 116 laboratory sections. TA's wondered about students' performance in subsequent courses—the effects of paired learning in students' future programming skills, computer science knowledge, and overall successes.

Themes:

1. Pairing Protocol: Time & Efficiency:
   …And it's really like night and day, because I had to work so hard to get the students to understand things in the unpaired sections, and there were a handful of students who took up the majority of my time with questions.

   I've found it easier to get done what I have to get done in lab. There are fewer questions. Sometimes, they are great labs. Less of them don't compile, most of them know what they are doing. If they have a mistake, it's usually very, very minor, it's never anything major like they're missing a method, or they say, this isn't really right.

   Very few questions get asked in the paired lab as compared to the unpaired lab…

   It's a lot easier to get my job done, because they ask each other questions more often than asking me. And in the unpaired labs, I usually had sometimes up to 16 students who don't know what's going on, and I had to walk through the lab, and this is a real pain.

   …usually, it turns out, that I'm not needed.

   … students have more resources…

   …she was able to learn, and I didn't have to spend so much time having to explain everything to her.

There are very few times when I have to go to a pair and go through the entire problem, which usually happens in a traditional lab. I don't have to spend the entire lab re-teaching it to a pair and getting them through the task. Usually, I just start them off, and then I just gauge how they're doing.

Usually, it's more about troubleshooting rather than having to teach them.

I've seen [pairing] in action, and more oftentimes it works. If it works, don't try to fix it. If there's anything that I'd like to change, I haven't found it yet.

It's a good idea, and it works in most cases…it's a good thing to pair them all together.

I think it's been very successful, especially compared to my experience of having a traditional unpaired lab.

It's been a step up from the non-paired sections. It makes it easier on the TA—easier to get their job done, and easier to address questions, and reach more of the class at one time instead of just a handful of students…All in all, it's been good and fun to teach. I think students are really learning.

We have to find a happy medium somewhere. Perhaps, doing it differently every semester until they hit the right balance.

2. Student Interaction

…when they are working with each other and feeding off each other, they can ask the other person a question, and the majority of the time that person will know.

I've seen students try to rush through these problems, and the other students will say, no, no, I think we should really go through this…

Apparently, students working together, one of them has the book and pulls it out to find the answers.

…one student was doing all the work while the other student was just sitting there relaxing.

…one student was new to programming and the other student had some experience and he was able to teach her as they were going along.

They don't know anything about programming, they' ve never taken programming, but they know that they make a lot of mistakes, so it's funny to watch them work, because they have a lot of fun beating each other up and getting the code to work.

…this pair didn't know anything about programming and they were always the last pair to leave. And they'd talk a lot and sit there and stare at the screen or read the book, but you could tell that they were both learning it at about the same speed. They would both interpret things and then check with each other.

3. Higher Order Thinking & Active Learning

   …when they do ask questions, they are very intelligent questions not about how to do it, but how they can improve what they have done, how they can do it better.

   …oftentimes in lab, I've had one or two students, that after they've finished the lab one time without switching, they'll switch and do the lab one more time again to make sure the other guy gets it.

   …it's higher order questions.

   I think what I've seen is more whole-hearted effort, and a slow but honest effort on the practice problems to try to understand what the lab is all about.

   …they go back and review what they've done, and then call me if I'm needed.

   … sometimes he goes through the lab as a pair, but then goes back and reworks it to make sure that he learned what he was supposed to learn.

   They'll say, "I think we understand it, but I just want you to look at it to make sure." It's a lot less about "what should I be doing in this program?" and more about "why am I getting errors?"

4. Unpaired Labs

   I have two friends who are teaching the unpaired labs, and I think that the students there tend to work a lot harder, because they are working alone.

   I was in a non-paired section, and I had myself to depend on, and I felt bad disturbing other people while they're working and using their books.

   Traditional labs, you have to start from square one. They don't know what they're doing.

   It's like you have to give them every step of the algorithm along the way. . .

   But in the traditional labs, they just sit there with the book, and they don't even touch the keyboard until I come around and tell them what to do.

   In unpaired labs, they don't understand what they're doing, they don't get the concepts.

5. Reluctance to Pair/Switch Roles

   And there are a couple of students who don't want to do it, because they could just type in the code and be out the door, and that's another two hours of free time that they could have.

   Sometimes students will pair but will refuse to switch so they are locked into these two roles, so one student will learn all the programming while the other one ends up having to watch.

   He really didn't want to [pair]. He said he was used to programming on his own, and I understood, but I told him this was an experiment and he either partnered or he was out. So I made him partner, and he pretended to partner and just went through the motions.

   Eventually, the reluctant students become less reluctant, because they see how much better the paired students are doing. . .They realize that they don't know what they're doing after all. A lot of times they see that [pairing] is not such a bad idea.

   Some students don't want to give up control of the keyboard.

   Yeah. If there is a problem that someone is working on, they're unwilling to switch.

6. Motivation

   If you have a partner, and that person comes in with the work already done, then the other person is likely to just say, OK, and then just sit back. But if that person was alone, he would have to do it all on his own…

   …in some lab sections…[t]hey'd just come in with completed labs or just wanted to come in and get the job done and were not really there to learn.

   …if one student is more motivated than the other, then that student can carry more of the weight than the other.

   …sometimes a student will pair with a student doing really well, and get by on his progress.

   Yeah, and people take the path of least resistance, so they're not going to want to adapt. I mean that's not a natural thing to want to adapt. It's easier to be lazy.

   [Partnerships] can work out if the guy that doesn't know the stuff really wants to learn and not just get out of lab.

   The advantage of not being paired, is that it forces you to have a willingness to learn. If you're paired, it doesn't actually force you to have a willingness to learn, unless the other guy doesn't want to learn either.

I mean there are benefits to pairing, I'll give you that. And the disadvantages are that if [students] don't really want to learn, [they] really don't have to.

Well those students (students who don't want to learn), even if they did it by themselves, they probably would not find the motivation to learn anyway. It would be just as bad.

7. Compatibility
   I think that when students are on the same level, as far as wanting to work, as far as having the work done, and I guess, and the same knowledge in general, then paired learning is optimal.

   …only disadvantage of paired programming is …if they're not on the same level, either with programming experience or motivation to work…

   If maybe some time at the beginning of the lab there is a way to assess their motivation or programming skills, that would help.

   For my computer engineering course, I had to pair up with someone for the duration of the course. We had to take an assessment that wasn't based on programming skills, it was based on personality types.

   Well, sometimes you may want to put a guy who knows his stuff with a guy who doesn't know anything, if the guy who knows his stuff has the ability to teach it.

   They were both on the same level, and had the approach that they would work through things together and eventually they were going to get it right. Since then, one of the people has paired with someone that is at about the same level, so she continues to get that.

   I think my problem is more an issue of partner incompatibility. When partners don't communicate with each other, they don't have that balance, and they're not understanding, that's whe n it becomes difficult to help one person out while trying to help the other one.

   I think the only thing that I would want to change is what we were talking about earlier, the issue of compatibility—partnering them up in that hierarchy of compatibility. . .

   . . . Not only testing them in their programming ability, but testing them on what type of personality they have and what type of motivational level.

8. Pairing & Learning
   …I think that in general, and from what I've seen with the kids in my labs, in a general case, pairing with each other does help them learn better.

The advantage I see with pairing with different people is that you have to adapt to new people all the time and to their style, and they have to adapt to you. If you just have one buddy for the whole class, chances are that you're not going to get to know anyone else in the class. Odds are, that their learning style will get locked into one particular learning style because they're with one partner the whole time.

… different people have different styles of programming. So it's good to change partners, like he's saying, because you're getting new techniques, but it's hard to adapt.

With just one pair, like he said, you end up depending on one learning style to learn. Having several pairs allows you to be more adaptable.

…since she had the first few labs with the more experienced programmer and was able to learn the basics of the programming that way, [the pair was] more productive.

I never have a situation where one student knows it and the other is lost. By the time they ask me to check, they either both have it or they're both lost.

I mean when you're working in a pair, there is a chance that someone in the pair will be able to bring in their resources and figure it out.

. . . in the pair section, they understand all the steps they have to complete in lab. They may not know exactly how to do it, but they understand what they have to do.

Usually, if the partner doesn't get what's going on, the other guy will, and will say, "OK, this is what needs to happen. Do you see how it solves the problem? OK, now you help me code this, because I don't know how to code this."

Yeah. You cannot learn from the book by yourself unless you sit in front of the computer with the book for hours and hours and try to code until you learn something.

9. Pairing Protocol Logistics

But on the most part, most everyone takes part in the switching—navigator and driver--, and I haven't had any problems with that.

But I think that one thing that may help is not to switch partners so often. And I think it's kind of a gamble to switch off every three labs.

I think you can do several labs with random pairing so that the TA gets an idea of who should be together, and then those people should be together for the rest of the semester.

I think that the major problem is that they change so often…I often have students that are not there and students have to pair with someone else for that lab, and they end up partnering with everyone by lab 7 or 8.

Yeah, I think no system is perfect either. Some students will miss lab anyway and students will have to adapt to new partners no matter what. It's inherent—the adaptation for students.

I suppose time of day is a factor, because I teach mine both in the middle of the day, 11:30-2:30, or something like that. Less of my students end up missing labs, and they end up changing partners at most three or four times, and usually the partner when they miss a lab, it's because they're sick.

…when people are absent, you could pair them up with another group that has similar ranking, or something like that.
It's a really good idea to switch them, but not as often. If they are in a long lab, then it's OK to switch at the intervals we're using, but otherwise there's no point in it.

Personally, I've never had any difficulty making them switch. I used to set the timer and tell them to switch, and eventually they would switch on their own when they heard the timer. They may stay and finish what they're working on, and then switch after that.

. . .back to the driver/navigator issue, I do think it's beneficial, because you isolate two aspects of coding, and they learn how to learn how to do those two separate jobs.

. . . [one] guy seems to know at least 75% what's going on, but the other guy is left out of the loop. And I think that's the other good thing about switching off, because this guy that knows everything is going to have to explain it to the driver what to do, and the driver is going to have to understand it.

. . . Maybe only having two partners the whole semester.

Switching them, that's also good to give them both the opportunity to code and to guide.

10. TA Responsibility/Evaluation
…it's our responsibility to make them try to learn. And I do my best to try to do that. If they don't want to learn, then you have to get it through their heads, hey you paid for me for three hours, but I'd rather earn my money than watch you walk out.

If there's a question with a concept, I usually talk to them about it, and one student will say, "yeah I get it," and the other says, "I don't see it." So what I do is have the one student explain it to the other student, and I stay with them to make sure that they're explaining it right. That's how I deal with those issues.

. . . and I think it’s the TA’s responsibility to get [the student] back into [the programming].

Well, there’s already the peer evaluation, which affects their grade. But no one wants to give their partner a bad grade. But if you leave it out of the grade, then people would be more honest. The peer evaluations work sort of good. I say sort of, because not everyone is completely honest in those evaluations.

It is not fun to do the peer evaluations. It’s long, it’s tedious, you have to press the buttons, and sometimes you have to type.

I think that we need more TA input. I mean, there’s TA input as far as how the pair is working and interacting, but maybe having that work more towards their grade than what the actual peers have to say.

I think the TA should hold everyone accountable for his or her own work. You have to say, “all right, you better be working equally, and I don’t want to see anyone hiding in the background, because if I do, that’s points off your grade.”

Ultimately it comes down to, I’ve done everything I can to get this student to work. Usually I play the encouragement card before I play taking their points away.

11. Project Pairing

. . .some people working on the projects, don’t take advantage of pairing.

More often than not, they’re working in pairs in the lab, but not on the projects, and I don’t understand.

Yeah, [it’s] so strange [when students don’t pair on projects], because most of the time one person gets a really good grade and the other gets a really bad grade, and they could have both done well if they had paired.

I think my lowest grade project grade was a 10. No one gets that in lab, so I don’t understand why they don’t say that they need help.

I think some students [don’t pair on projects], because they want to see if they can do it on their own, and this is an opportunity for them to do it on their own, because they have no choice in the lab--they have to pair--since that’s the way it’s designed.

If they’re both at the same ability level and learning about the same amount, then they should do approximately the same grade on the project. And as far as I’ve witnessed, that’s been the case at least in my labs. People have done really well, probably not as good as if they had paired, but well.

I found that working with someone on projects, when he worked something out, figured something out, I felt like, well that could have been me. I felt like I should have been the one to solve the problem. I should have done that.

I like the way things are now with pairing in lab and having the option of not pairing outside of lab. But if you take that option away and you say that they have a paired component and an unpaired component, you'll find that there's people that won't do as well. You'll find that some people need others to do well.

12. Confidence

    If you're pairing up with somebody and leaning on them, then you're not going to get that. You're not going to have the same self-confidence as someone who works through it on their on, and knows how to do it, and has that experience.

    They need some reassurance that they can do this and not just be dependent on somebody else. Being dependent on somebody else is not really going to work on a test, obviously, so they're going to run into some trouble.

    Yeah, well, you don't want to make them dependent on any one person.

    I think working with another person is to make it a more conducive learning environment and not to making it a situation where they're saying, "help me, help me."

    It will probably happen that a student will become dependent on working with someone else. But I think that will be a very rare case, and there are steps we can take to solve that—to help a student see another person as a resource and not as a crutch.

13. Future Success

    I mean pairing is a nice benefit built into the lab, but eventually when they get out into industry, they're going to have to program by themselves in some situations.

    The only thing that I would like to add is to see how some of the students do after 116, when they get into classes where they can't pair.

    But the way the program is designed is so that they don't become dependent on someone. It's just to help them learn better about programming. In fact, if that's the case, they can take that along with them to 216.

    I think it would be interesting to see the number of cases of cheating in 216 [because] …they're so used to collaborating…

IV. References

1. Williams, L., Wiebe, E., Yang, K., Ferzli, M., and Miller, C., "In Support of Pair Programming in the Introductory Computer Science Course," Computer Science Education, September, 2002.