

Pair Programming Guidelines

The guidelines for successful pair programming (adapted from UCSB CS16)

WHAT IS PAIR PROGRAMMING?

When working on homework assignments and projects, you may work alone or with one partner, using the “Pair Programming” approach. In pair programming, two programmers share one computer (or in the case of distributed or remote pair programming: one shared IDE).

- One student is the “driver”, who controls the keyboard and mouse.
- The other is the “navigator”, who observes, asks questions, suggests solutions, and thinks about slightly longer-term strategies.
- The two programmers switch roles about every 15-20 minutes; or when moving on to a new task.

Working in pairs should make you much better at programming than working alone would. The resulting work of pair programming nearly always outshines that of the solitary programmer, with pairs producing better code in less time.

“[Pair programming] makes learning programming faster and more fun. I have had previous experience working both alone and with partners. I definitely agree that working with partners is more profitable.” – CS student

To learn the “do’s and don’ts” of pair programming and to see pairs in action, view this entertaining video about pair programming from North Carolina State University: [An Introduction to Pair Programming \(Version 2\)](#). However, the most important rule of pair programming is that you must work together, co-located, on the same part of the assignment at the same time. Splitting up the work on an assignment is considered cheating.

PAIR PROGRAMMING AND ACADEMIC INTEGRITY (“THE RULES”)

- You must always be “with” your partner (physically or virtually) when developing code. You may not do any coding, including documentation, without your partner present if you choose to do an assignment together.
- You may not use or even look at any code you have previously developed without your partner. This means, if you choose to split with a partner, you must both start over from scratch with your new partner. Completely throw your old code

away. This helps to ensure that you are both starting from the same place (even though your knowledge might be different).

- You and your partner must both be active contributors to the code. This means, you must both be engaged in the development (talking, typing, suggesting, etc) and you must switch roles regularly¹.
- You may not split the code and develop it separately, even if you are sitting right next to each other. You must collaboratively develop a single solution.
- Whenever you develop code using pair programming, you must acknowledge your partner in your submission(s), and also specify the predominant interaction modality you used for the submission (e.g., “in-person”, “virtual voice-chat”, “virtual text-chat”).

CHOOSING A PARTNER

You can choose your own partner. You will need to arrange times to meet outside of class. We expect everyone to be flexible and professional in arranging those times as necessary; if your schedule is highly constrained, explore possible meeting times with your prospective partner before you commit to the partnership.

You should try to pick a partner whose experience and skill level with programming is similar to your own. This may not always be possible and it is sometimes hard to compare skill levels, but students tell us (and other data supports) that pairs are most productive when the partners start at about the same level. Still, people often pair up with another whose skills are different. This happens more often than not, as no two people have an identical skill set. The differences may be great or small, but this is exactly like most real-world working situations.

Part of accomplishing a task is to get the most out of each member and make each member stronger and more productive on subsequent tasks. Students bring different strengths to the process, regardless of how much experience they have had with programming. Both experienced and inexperienced students will need to draw on their reasoning and problem solving skills. A more experienced partner may sometimes feel frustrated or slowed down by a less experienced partner, but the experienced partner still benefits from the teamwork in many ways. *The less experienced partner's requests*

¹ While being collocated and speaking to each other in-person requires the least coordination and effort, this method of pair programming may not be preferable due to our current COVID situation. An alternative method that will take a little more coordination is to work together remotely, using virtual meeting software such as WebEx to share and voice-chat in real-time. If this too is not doable (e.g., speaking in a classroom or library), the final option is to work together remotely, using virtual meeting software such as WebEx to share and text-chat in real-time. This last method may not be ideal however, as typing chat messages and working in a shared IDE requires context shifting which may be difficult to coordinate.

for clarification often uncover flaws in an approach or solution; the exercise of providing a clear explanation solidifies and deepens the explainer's own understanding and the teamwork and communication skills they gain have great value in both the academic realm and the job market.

"My partner had never coded anything before so I was able to teach him a little bit about how it worked. The teaching bit helped me a lot with understanding the labs and passing the exams." – CS student

The less experienced partner may feel that questions hold the other partner back or that there is no benefit to participating actively, but pair programming studies show that paired work is consistently better than work the stronger partner does alone. It is each partner's job to understand the whole task; that means asking questions when necessary and answering them when possible. It may be instructive to read a selection of students' partner evaluation comments from similar courses that have used pair programming; they give a picture of what good partnerships are like (and a few disasters, too). Take a look at [Pair Programming Evaluations](#) from the University of California, Irvine Bren School of Information and Computer Sciences.

DEALING WITH DIFFERENCES

If you believe your partner is not participating appropriately in pair programming (e.g., they do not keep in touch, do not come prepared to work on the assignment, or do not seem to be engaged in the process) please first address your concerns to your partner, and try to agree on what should be done to make the pair programming experience work well for both of you. If that approach is not successful, explain the issues to your teaching assistant, who will work with you and your partner to improve the situation. If your differences cannot be resolved, or if your partner stops participating, you should mutually agree to separate for the rest of the assignment. You may complete the assignment on your own or with a new partner, but *you must start over from scratch*, as described above. You should never simply abandon your partner without coming to a mutual agreement that this is the best course of action.

HOW PAIR PROGRAMMING AFFECTS YOUR GRADE

You and your partner will receive the same score on your code.

WHEN IN DOUBT, SEEK CLARIFICATION

Pair programming is shown to help, not hinder, your successful completion of the course. It is important that you understand the processes and expectations up front so

you can gain the most benefit. If you are unsure of any of the aspects of pair programming and how it is implemented in the course, see your instructor right away.