

Pair Programming

For each assignment, you may select a different person to work with, or work alone. Once a homework assignment is started as a pair, it must be completed as a pair. When working in pairs, no significant work is done individually—all work is done completely as a pair, with a single, joint, assignment turned in by one member of the pair.

Pair Programming Technique

In the pair-programming model, you must spend the bulk of your time working together as a team, with one person at the keyboard and the other at their side watching and making suggestions. All work should be a joint effort. You will be violating the **Honesty Code** if you divide the work such that you work separately with one person doing one half of the work and the other person doing the other half.

Rules

1. All code is written with two programmers sharing one machine (at any given moment, one person is at the keyboard and is known as the driver; the other person is observing and is known as the navigator)
2. The driver and navigator collaborate on all aspects of the software development: design, coding, debugging, etc.
3. The driver does the actual typing of the code.
4. The navigator is thinking about broader issues, such as How does the current code fit into the larger picture? How will we test the code to see if it works? Are the types correct? Is there a helpful function in the library we can use to help out? Could we use simpler/clearer code here? Do we need a comment here? What should we do next? Are we making any discoveries that require changes to code already written? The navigator also observes the code being written and checks it for defects, but the primary role of the navigator is not to say "Oh, I think you've indented that *where* loop body wrong." This role is called the navigator because the job is to steer the software development in the right direction.
5. **Periodically, the driver and navigator change roles. This is a crucial part of the methodology — you are required to spend roughly equal amounts of time driving and navigating.**

Honesty Code

Specifically, pair programmers must contribute equally, including

- Spending equal times driving and navigating
- Reporting any violations of the rules, either pair programming, or otherwise, promptly.

What You Can Do Alone

Before you meet together, you can and should read over the assignment. Print it out. Make notes on it. Highlight important parts.

The assignments can be involved, so marking them up your self can be very helpful.

You can think about your algorithms and how they could work better, and if you have a bug, you can think about it. Alone, you may also reread the code the two of you have written, but you usually cannot change it without your partner present unless the change is minimal and trivial.

Finally, you may sometimes fix some bugs; specifically, if you discover a “Duh!”-style bug (e.g., forgetting to initialize something), you may fix it, if and only if:

- The fix is “trivial”
- You check in the code with your fix immediately and email your partner

You may not make substantial or non-trivial changes to the code by yourself. This prohibition includes "cleaning up" the code.

What Pair Programming Is Not

Be sure you understand that pair programming does not mean dividing the assignment in half and each doing one part. That is one kind of software development team, but it is not what you will do in this class and it does not provide the benefits outlined above.

To reiterate: in pair programming, all the work is done with both team members at the same machine at the same time.

How to Succeed

There are just a few of requirements for a successful pair programming project.

- In the first place, the partners must be compatible. Working with a friend is a really good idea; working with someone you don't like or normally disagree with is a bad one.
- Another key ingredient is compatible schedules. Since you have to work at the same time, you have to be available at the same time. When forming a team, be sure to discuss your schedules and be certain that you have adequate opportunities for working together.
- A final consideration has to do with ability levels. In any class, there is bound to be a range of programming abilities. Pair programming will work for you no matter where you are in that range, but it will work best if both partners are at about the same ability level.

Tips for Good Pair Programming

- Respect your partner. If you say you'll meet at a particular time, be there. Know each other's mobile phone numbers so that you can call if something's up.
- Agree clearly on what you'll have done before you meet. Time together is precious, so use it well, and meet the commitments you make to each other.
- Stay fairly close to 50-50 on driving. Let the less experienced partner start and maybe drive a little more.
- Make sure your partner stays focused and on-task.
- Keep your feedback positive. For both, this is an opportunity to improve.
- Don't take things too seriously. If your partner picks out a bunch of errors as you type, be glad. But do not always agree. Have healthy disagreement/debate. Finding the fine balance takes adjustment.
- Sit side-by-side and program, simultaneously viewing the computer screen and sharing the keyboard and mouse. Slide the keyboard — don't move the chairs.

Benefits of Pair Programming

- When one person gets stuck, the second can commonly get them "unstuck".
- When both people get stuck, they spend less time beating their heads and go seek help sooner.
- Your software will be cleaner, better organized, and have fewer bugs.
- You will learn more of the basic concepts that the assignment is designed to teach.
- Your exam performance will be about the same as if you had worked on your own.

Why It Works

It is easy to see the benefits of the paired approach. As you have no doubt experienced, just "bouncing your idea off someone" is often very enlightening and very helpful. Explaining your solution helps crystallize it in your mind and helps you see any pitfalls. Add to that the fact that your partner will chime in with ideas of her/his own, and you can see that the power of two is much greater than the power of one.

Another great benefit often surfaces when you are working on some new aspect of programming. A common occurrence is simply getting stuck; that is, you arrive at some point where you don't know how to proceed or where your code is not working for reasons that are unclear. Sticking points like these can consume a few hours or even a few days.

Now imagine that you always have a partner to help you. In many cases you will find that where you are stuck, your partner will know the answer, and vice versa. Almost everyone who tries pair programming reports that there is far less "down time" due to programming difficulties.

Pair programming will also help you learn more. When your partner explains something to you, you learn something new. When you explain something to your partner, you learn it better. When you figure something out together, you both learn it, in less time than if you were working on your own. Mutual tutoring and problem solving is truly a win/win situation.

General Programming Tips

When coding

- Start simple, develop incrementally
- Edit the code to simplify, add flexibility, etc.

To find out about pair programming have a look at:

http://en.wikipedia.org/wiki/Pair_programming
<http://www.extremeprogramming.org/rules/pair.html>
<http://www.wikihow.com/Pair-Program>