

Some header

Programmers have to work with a lot of bad code. Sometimes it's their own code, but other times it's a co-worker's code. If they have to frequently work with their co-worker's bad code, they will eventually wish to approach their co-worker about their code quality, however in many cases they do not know how to do so without causing offense or resentment. So this article is about the approach I would take to such a situation.

Determine the problem

First off, verify that the problem is worth bringing up. Make sure it's not simply a case where your preferences are different from your co-workers, and that the code in question is actually harming productivity in some way. For example, simply saying "your naming convention doesn't match mine" is bad, however saying "your naming convention makes it hard to understand what is going on in the code" is a much better approach.

Gather your arguments

Once you've determined there is a valid reason for trying to change the way your co-worker does things, get your arguments prepared. Identify why the code is harmful to the development environment, and figure out the ideal way of coding it instead. Determine what your arguments are for why your ideal example is better than the existing code sample. Once again, don't use arguments such as "it's best practice", but instead explain *why* it's a best practice (readability, maintainability, reusability, etc)

Approaching your co-worker

Now that you know your arguments for why you think your code is better than your coworkers, and have a clear idea of what the ideal code should look like, arrange a time to talk with them.

This could be a casual conversation, or it could be setting up an appointment to talk with them about your concerns. How you choose to approach your co-worker is based on your level of comfort with them, the organization's environment, and both of your ranks within the organization. For example, if I wanted to talk to the programmer next to me, I might simply ask them if I could have a moment of time whenever they're available to talk with me about a piece of code, however if it's my boss I might send him/her an email asking if I could setup a brief meeting with them to discuss some code I've been working on.

Some people might argue that approaching a boss is different than approaching a co-worker,

however I feel they are both people and both deserve equal respect. The only real difference I see is that your boss's time is usually more valuable to the company, so be aware of that when talking with him/her.

What to say

When you actually talk with your co-worker, I usually find a good way to start the conversation is to ask them to explain their code. Ask them if there was a reason for their style of coding, or tell them you've never seen something coded that way and ask them why they chose that method. Really *listen* to what they have to say during their explanation. It could be that their way is right, and you are wrong, in which case take notes and learn from it.

If you still think the code is bad and worth changing, show your co-worker how you would code the same piece of code, and tell them why you would code it that way over his/her way (better performance, fewer chances of errors, easier for other programmers to read/maintain, etc).

Focus on why your method of coding is best, and not why their method is worst. Nobody likes being accused of writing bad code, although in reality we were all new once and I'm sure we've all written some horrendous code at some point in our lifetime. Be sure to keep the conversation focused on the code, and avoid turning it personal by talking about the person who wrote the code. It is always best if you can show that you understand that sometimes bad code gets written, whether it was due to time constraints, inexperience, or simply a bad day.

As a side note: don't try and get them to change their old code (unless you are their boss and want them to spend time doing so). Your goal is to educate them so they stop making the same mistake in the future, not berate them over past mistakes and punish them by extra work.

Conclusion

Afterwards, see if your co-worker still supports their coding style over yours. If they are open to improvement, they will likely change their way of coding going forwards. If they still prefer to use their coding style, you are not likely to change their opinion, so drop the subject and don't bring it up again unless their code is actually harmful.

Remember, nobody can write perfect code, so we are all always looking for improvement. The world of programming is so big that its impossible to know everything. It's likely that your co-worker doesn't know there is a problem with their code, and would appreciate the input providing you do it in such a way that you are teaching them, and not insulting them.