Ten Quick Tips for Delivering a Programming Lesson

Greg Wilson[1,*]
1 RStudio, Inc., Toronto, Ontario M4L 2T9

* greg.wilson@rstudio.com

Abstract

Designing a great lesson is the first 90% of effective teaching; delivering it well is the other 90%. The 10 simple rules outlined in this paper describe classroom practices that instructors can adopt immediately and at low cost.

Author Summary

Teaching well is a craft like any other, and success often comes from an accumulation of small improvements rather than from any single large change. This paper describes ten practices that can be applied to a wide variety of subjects, that are easy and inexpensive to adopt, and that have proven their value in both regular classrooms and free-range workshops.

Introduction

Teaching is a craft, and like any craft it can be improved by studying, adopting, and adapting the practices of others. This paper describes ten practices that can be applied to a wide variety of subjects and have proven their value in both regular classrooms and free-range workshops. They are easier to adopt than those described in [1, 2], some of which are briefly discussed in the conclusion; some have been inspired by [3–5], while others are drawn from the author's experience teaching programming to researchers [6].

1 Use formative assessment every 5–15 minutes.

As instructors, we always want to get through material than time allows, so we often teach at the speed at which we can talk rather than the speed at which people can learn. Having learners do something every 5–15 slows us down, keeps them engaged, and gives us and them feedback on whether they have actually understood what has just been taught.

In-class checks like this are one kind are called *formative assessments*. Good ones take only a minute or two to complete so that they don't derail the flow of the lesson, and have an unambiguous correct answer so that they can be checked in large classes. Popular kinds of formative assessment in programming classes include:

- Answer a multiple choice question.
- Write a few lines of code.
- Predict what the code on the screen will do when it runs.

- Contribute the next line of code.

Starting with a formative assessment that reviews a previous lesson is a good way to signal that class has started, and having students recall older material before tackling something new has been shown to improve learning outcomes [7]. Similarly, ending the class with such an exercise gives learners a sense of how far they (should have) progressed.

A complement to this rule is to get students out of their seats every 45–60 minutes. People's brains get tired when they are concentrating, and tired brains can't learn [8, 9]. Caffeine doesn't fix this, so have your learners get up and move around for a few minutes every hour in order to reoxygenate their gray cells. This allows those who need a bathroom break to take care of things discreetly[1], and give the instructor a few moments to answer questions and figure out what to teach next.

2 Monitor progress and learners' need for assistance.

Sticky notes are one of my favorite teaching tools, and I'm not alone in loving their versatility, portability, stickability, foldability, and subtle yet alluring aroma [10]. Give each student two sticky notes of different colors, such as orange and green. If someone has completed an exercise and wants it checked, or if they feel that they are following the lesson, they put the green sticky note somewhere the teacher can see. If they run into a problem and need help, they put up the orange one. This works much better than having people raise their hands: it's more discreet (which means they're more likely to actually do it), they can keep working while their flag is raised rather than trying to type one-handed, and the teacher can quickly see from the front of the room what state the class is in.

3 Use a visibly fair mechanism to distribute your attention evenly.

Teachers naturally focus their attention on learners who are making eye contact and asking lots of questions—in other words, on extroverts. This creates two feedback loops: the extroverts ask even more questions because they're getting attention, while other students stop trying to engage because they're not. To ensure the teacher's attention is fairly distributed, have each learner write their name on a sticky note and post it somewhere visible. Each time the teacher calls on them or answers one of their questions, they take their sticky note down. Once all the sticky notes are down, everyone puts theirs up again. This technique makes it easy for the teacher to see who they haven't spoken with recently, which helps them avoid unconscious bias and preferential interaction. It also shows learners how attention is being distributed so that when they *are* called on, they won't feel like they're being picked on.

4 Make mistakes and work through them.

Novices spend most of their time trying to figure out what's broken and how to fix it, but teachers rarely devote that proportion of lessons to analyzing and correcting errors. Watching you figure out that something is wrong, hearing you work backward through

[1] A colleague once told me that the basic unit of teaching is the bladder. When I said I'd never thought of that, she said, "You've obviously never been pregnant."

the error messages to possible causes, and seeing you make and test fixes is often one of the most valuable things you can show your learners.

For example, an effective way to teach programming is *live coding*. Instead of presenting pre-written material, teachers write code in front of their class as their learners follow along. Doing this provides lots of opportunities for making and correcting mistakes, and watching a program being written is more engaging than watching someone page through slides. It also slows the teacher down, and helps remind them just how much they are throwing at their learners. Finally, watching teachers make mistakes shows learners that it's all right to make mistakes of their own. If the teacher isn't embarrassed about making and talking about mistakes, learners will be more comfortable doing so too.

5 Follow your learners—twice.

A slide deck is like taking a journey by train: the ride may be smooth, but the route can't be changed on the fly. Live coding, on the other hand, is like going off road in a four-wheel drive: it may be bumpier and messier, but it's a lot easier to explore things that catch your attention. In particular, if a student asks a "what if" question, it's a lot easier to respond if you are writing code or proving a theorem in real time than if you are using a slide deck. Following your learners' lead also signals that you respect their time and interests; this improves engagement, which in turn improves learning outcomes [11].

But it's possible to have too much of a good thing. If someone asks a question, try it out. If they or someone else asks a follow-up, try that too, but then come back to your main point. Otherwise, you can easily find yourself pulled so far off course that you don't reach the most important points of your lesson.

6 Create a backlog.

You may not have time to answer all of your students' questions, or might not actually know the answers. To handle this, write questions on sticky notes and post them on the wall behind you, then look over this backlog during breaks and decide which questions you want to tackle. This gives you a chance to prioritize based on what's most relevant (and what you actually know). It also helps build trust: many people have learned that "I'll address that later" means "I hope you forget that you asked", so if they see you trying to tackle a few of the questions that have come up, they will forgive you for not getting to the rest.

7 Have learners work in pairs.

A sense of community is one of the primary motivators for adult learning [11], so encourage your learners to talk to teach as they work through exercises. Peer instruction puts group discussion at the center of teaching [12–14], but *pair programming* and similar practices are also effective.

Pair programming is a software development practice in which one person (the driver) does the typing while the other (the navigator) offers comments and suggestions, and the two switch roles several times per hour. It is effective in professional work [15], and benefits in teaching include increased success rate in introductory courses, better software, and higher student confidence in their solutions. There is also evidence that students from underrepresented groups benefit even more than others [16–18]. I have found it particularly helpful with mixed-ability classes, since pairs are more

homogeneous than individuals. When you use pairing, put everyone in pairs, not just learners who are struggling, so that no one feels singled out.

8 Present diagrams incrementally.

Our brains have separate channels for processing visual and linguistic information [8, 9], so people learn best when complementary material is presented simultaneously through these channels. In simple terms, that means your slides should present diagrams or other relevant images for you to talk about, and that these diagrams should be built up piece by piece rather than presented all at once. When the diagram is presented piece by piece, students' brains will correlate the arrival of new visual information with the arrival of new linguistic information [19, 20]. Presentation of either later on will then help trigger recall of the other.

9 Have learners take notes, and review them.

Taking notes forces students to organize and reflect on information as it's coming in, which in turn increases the likelihood that you will transfer it to long-term memory. Many studies have shown that note-taking improves retention [21, 22], and more recent results indicate that taking notes using pen and paper is more effective than using a tablet or computer [23].

Fifty years ago, when being able to accurately record information or the minutes of a meeting was considered an essential white-collar skill, it was common for high school teachers to require students to submit their notes for grading. This is less common today, but is still beneficial, since it gives the teacher an opportunity to identify gaps between what they thought they taught and what their students heard.

If the resources to do this are not available, have learners take a minute at the end of each class to write one thing they learned on one side of a card and one question they still have or something they're confused about on the other. Review these *minute cards* and looking for patterns only takes a few minutes, and tells you what you need to clarify at the start of the next lesson.

10 Teach together when you can.

Co-teaching describes any situation in which two teachers work together in the same classroom. [24] describes several ways to do this:

Team teaching: Both teachers deliver a single stream of content in tandem, taking turns like musicians taking solos.

Teach and assist: Teacher A teaches while Teacher B moves around the classroom to help struggling students.

Alternative teaching: Teacher A provides a small set of students with more intensive or specialized instruction while Teacher B delivers a general lesson to the main group.

Teach and observe: Teacher A teaches while Teacher B observes the students, collecting data on their understanding to help plan future lessons.

Parallel teaching: The class is divided in two and the teachers present the same material simultaneously to each.

Station teaching: The students are divided into small groups that rotate from one station or activity to the next while teachers supervise where needed.

Team teaching is particularly beneficial in day-long workshops: it give each teacher a chance to rest and think about what they are going to do next. If you and a partner are co-teaching:

- Take 2–3 minutes before the start of each class to confirm who's teaching what.
- Work out a couple of hand signals as well. "You're going too fast," "speak up," "that learner needs help," and, "It's time for a bathroom break" are all useful.
- Each person should teach for at least 10–15 minutes at a stretch, since students will be distracted by more frequent switch-ups.
- The person who isn't teaching shouldn't interrupt, offer corrections or elaborations, or do anything else to distract from what the person teaching is doing or saying, but may ask leading questions if the learners seem lethargic or unsure of themselves.

Most importantly, take a few minutes when the class is over to congratulate or commiserate with each other: in teaching as in life, shared misery is lessened and shared joy increased.

Conclusion

If you are teaching an evening class after working for a full day, you and your learners will both appreciate it if you brush your teeth and put on a clean shirt. Cough drops will help you keep your voice and fend off whatever colds the learners brought with them, and your back will be grateful tomorrow that you wore comfortable shoes today.

References

1. Brown NCC, Wilson G. Ten Quick Tips for Teaching Programming. PLOS Computational Biology. 2018;14(4). doi:10.1371/journal.pcbi.1006023.
2. Devenyi GA, Emonet R, Harris RM, Hertweck KL, Irving D, Milligan I, et al. Ten Simple Rules for Collaborative Lesson Development. PLoS Computational Biology. 2018;14(3). doi:10.1371/journal.pcbi.1005963.
3. Huston T. Teaching What You Don't Know. Harvard University Press; 2012.
4. Lang JM. Small Teaching: Everyday Lessons from the Science of Learning. Jossey-Bass; 2016.
5. Lemov D. Teach Like a Champion 2.0: 62 Techniques that Put Students on the Path to College. Jossey-Bass; 2014.
6. Wilson G. Software Carpentry: lessons learned. F1000Research. 2016;doi:10.12688/f1000research.3-62.v2.
7. Weinstein Y, Sumeracki M, Caviglioli O. Understanding How We Learn: A Visual Guide. Routledge; 2018.
8. Ambrose SA, Bridges MW, DiPietro M, Lovett MC, Norman MK. How Learning Works: Seven Research-Based Principles for Smart Teaching. Jossey-Bass; 2010.

9. National Academies of Sciences, Engineering, and Medicine. How People Learn II: Learners, Contexts, and Cultures. National Academies Press; 2018.

10. Ward J. Adventures in Stationery: A Journey Through Your Pencil Case. Profile Books; 2015.

11. Wlodkowski RJ, Ginsberg MB. Enhancing Adult Motivation to Learn: A Comprehensive Guide for Teaching All Adults. Jossey-Bass; 2017.

12. Crouch CH, Mazur E. Peer Instruction: Ten Years of Experience and Results. American Journal of Physics. 2001;69(9):970–977. doi:10.1119/1.1374249.

13. Smith MK, Wood WB, Adams WK, Wieman CE, Knight JK, Guild N, et al. Why Peer Discussion Improves Student Performance on In-class Concept Questions. Science. 2009;323(5910):122–124. doi:10.1126/science.1165919.

14. Porter L, Bouvier D, Cutts Q, Grissom S, Lee CB, McCartney R, et al. A Multi-Institutional Study of Peer Instruction in Introductory Computing. In: 2016 Technical Symposium on Computer Science Education (SIGCSE'16). Association for Computing Machinery (ACM); 2016.

15. Hannay JE, Dybå T, Arisholm E, Sjøberg DIK. The Effectiveness of Pair Programming: A Meta-analysis. Information and Software Technology. 2009;51(7):1110–1122. doi:10.1016/j.infsof.2009.02.001.

16. McDowell C, Werner L, Bullock HE, Fernald J. Pair Programming Improves Student Retention, Confidence, and Program Quality. Communications of the ACM. 2006;49(8):90–95. doi:10.1145/1145287.1145293.

17. Hanks B, Fitzgerald S, McCauley R, Murphy L, Zander C. Pair Programming in Education: a Literature Review. Computer Science Education. 2011;21(2):135–173. doi:10.1080/08993408.2011.579808.

18. Celepkolu M, Boyer KE. Thematic Analysis of Students' Reflections on Pair Programming in CS1. In: 2018 Technical Symposium on Computer Science Education (SIGCSE'18). Association for Computing Machinery (ACM); 2018.

19. Mayer RE, Moreno R. Nine Ways to Reduce Cognitive Load in Multimedia Learning. Educational Psychologist. 2003;38(1):43–52. doi:10.1207/s15326985ep3801_6.

20. Mayer RE. Multimedia Learning. 2nd ed. Cambridge University Press; 2009.

21. Aiken EG, Thomas GS, Shennum WA. Memory for a Lecture: Effects of Notes, Lecture Rate, and Informational Density. Journal of Educational Psychology. 1975;67(3):439–444. doi:10.1037/h0076613.

22. Bohay M, Blakely DP, Tamplin AK, Radvansky GA. Note Taking, Review, Memory, and Comprehension. American Journal of Psychology. 2011;124(1):63. doi:10.5406/amerjpsyc.124.1.0063.

23. Mueller PA, Oppenheimer DM. The Pen is Mightier than the Keyboard. Psychological Science. 2014;25(6):1159–1168. doi:10.1177/0956797614524581.

24. Friend M, Cook L. Interactions: Collaboration Skills for School Professionals. Eighth ed. Pearson; 2016.