

Getting to know your students in CS1 on the first day and beyond

KATE LOCKWOOD, St. Paul Academy

This implementation paper describes how to use the resource “Day One Ice Breakers”. The goals of these ice breakers are (1) to start to build community in the classroom, (2) to introduce students to strategies for success in computer science, (3) to “bust some myths” that have been associated with the intro courses in our major, namely that the course is intentionally hard or a “weeder” course and that many of the other students have previous experience. I use these activities to set the tone for the rest of the semester and refer back to them at multiple points.

1. THE FIRST DAY OF CLASS

When students show up on the first day of CS1, they often come with some misconceptions and even misgivings about learning computer science. This is particularly true if CS1 is a required course for students who don't plan to major in computer science. For example, students may think that CS is a hard or “weeder” class, or that CS is a hard topic. Students also often think that other students are more prepared for the course. Usually the first day of class is spent reviewing the syllabus whose grading policies and plagiarism warnings do little to make the classroom seem like a collaborative, welcoming place and sometimes can end up reinforcing the misconceptions students came in with.

The slides in the first day of CS1 resource take a new approach to the first day. The students are assigned the task of reading the syllabus and taking an automatically scored multiple choice quiz on its contents via the LMS. Class time that would normally be spent reviewing course policies, is used instead to start to build community among students and to help set the ground work for students to understand how to be successful in computer science courses. There are two main ice breakers used. The first asks students to stand up whenever a statement applies to them. The second asks students to engage in a creative problem solving task (creating a meme). The slides conclude with material that is aimed at letting students know how to be success in my class.

1.1 Activity 1: Stand up/Sit down

For this activity, I project one statement at a time using PowerPoint and ask students to stand up any time a statement applies to them. I've found it's important to start with basic information that doesn't have the potential to be at all uncomfortable to answer (for example, “I am a freshman”), this eases students into the activity and gets them participating. In order for the activity to work, it's important for the instructor to also be engaged with the activity and to be interested in the student responses. I'll often ask follow up questions (for example, if a student doesn't stand up for any of the majors I have listed I'll ask that student what their major is).

Once students are warmed up and engaged, the statements ask more “difficult” questions about previous experience in computer science and preconceptions about the CS1 course. These questions have two main goals from my perspective (1) let students see that very few of their

classmates have any previous programming experience (2) set up the conversations that we will have later in the class about how to be success at learning to program and problem solve.

1.2 Activity 2: Creating a Meme

For this activity I ask each student to create a meme using a picture either that they took, or of themselves. The activity purposely has minimal instructions because I want the students to practice figuring out how to break a larger problem down into its component steps (e.g. step 1 is find a picture). I also want students to think about the ways they might find information to help them complete an unfamiliar task. I make sure students know they are welcome to ask questions or to collaborate with their classmates but each student needs to turn in their own meme. It is important to circulate through the class while the students are working. There are usually 2 or 3 students who just sit in their seats unsure of where to start. I try to identify these students and give them a gentle prompt – “What do you think the first step might be?” or “Where might you have a picture?”

After all the students have completed the assignment, we go through the uploaded memes as a class. We also have a conversation about the activity itself. I use the conversation as an opportunity to talk about problem solving strategies and appropriate ways to get help – particularly how to responsibly use Google to find information in class. Some topics that usually come up that I try to make sure I follow up on are:

- Different students did the assignment in different ways – but many of the different approaches are correct – for example, some students may have edited a picture in MS Paint while others used an online meme creator –*there is no ONE right way to do this assignment*
- Most students had not previously made a meme
- If a student didn’t know how to make a meme, they could break the problem into component steps and figure out how to do each step independently
- Students likely got help from (1) each other, (2) the instructor (3) the Internet

1.3 Success in Programming Discussion

The last part of my first day activities involves helping students understand how my class is structured to help them succeed. I teach introductory programming using the inverted classroom model to attempt to give students more guided practice with problem solving and programming. I have an additional teaching paper that outlines the details of how I implement this, but basically students are assigned workbooks to complete before class. These workbooks introduce conceptual information and then we spend class time practicing programming.

The success strategies that I introduce in this lecture and then refer back to throughout the semester are (1) PRACTICE! Many of my students are used to classes where studying consists of memorizing. I purposely make my assessments open notes to help drive home the fact that memorizing is not the most effective way to become a better programmer. I stress to students that programming is a skill like driving or playing a sport or instrument – you can’t learn it just by watching someone else do it. (2) Mistakes are GOOD! Many of my students want to know that they are doing something correctly before they start. Throughout the semester I try to transition them from asking “is this right?” to being able to determine independently when their programs are ready to go. We spend a lot of time talking about how mistakes are expected and that the important thing is to notice a mistake and then trouble shoot how to fix it. We talk about how a quarterback doesn’t just pick up a football and throw it correctly 100% of the time, etc.

2. CONTINUING TO BUILD COMMUNITY

Much of my first day is spent starting to build community in my classroom and to help students start to feel comfortable coming to me for help. I try to continue to show them I am interested in them throughout the semester. One way I do this is through sign-in sheets. I don't count attendance in grades, but every day I pass around a sign in sheet with an "ice breaker" question on it (instead of students signing their name, they answer the question). I will occasionally follow up on a student answer or even show the class a chart of the (anonymized) responses – for example "80% of you have dogs at home!" Students have a lot of fun with these and they've been surprised when I follow up with them when they come to office hours for help.