Mellon Evidence-Based Learning Proposal:
Problem-Busting Jam (PBJ) Pilot in CS111 Lab
Sohie Lee, Computer Science Department

High-level summary

This proposal aims to pilot group work in one CS111 lab during the Fall 2018 semester with collective ownership of problem solutions and an emphasis on verbal strategy articulation.

Background

CS111 is an introduction to problem solving through computer programming. Using the Python programming language, students learn how to read, design, debug and test algorithms that solve problems. It is the first required course for CS majors.

In Fall 2018, we have four lectures and seven labs, for a total of 120 students (CS111 serves ~200 students each academic year). Students attend two 70-minute lectures, one 110-minute lab and office-hours/drop-in hours each week. Weekly problem sets take an average of 6.92 hours to complete (using data from Spring 2018). The four lectures use the same lecture notes, quizzes, problem sets and exams. The labs are not tethered to any particular lecture; so any student in any of the four lectures can enroll in any of the seven labs. CS111 has one SI leader and nine tutors. The course generates more tutor-tutee contact hours than any other course at the college (Spring 2017, 84% of CS111 used drop-in hours; tutors were available ~20 drop-in hrs/week). This is a course where most students seek a lot of assistance outside of class.

Currently, CS111 labs consist of randomly assigned student pairs with no graded components. Lab is intentionally designed to be a place where their work is not graded, and the environment is question-friendly. This pilot project introduces peer-to-peer learning in group problem-solving with accountability[1,2]. In this document, we will refer to the proposed pilot project as Problem-Busting Jams (PBJ).

Implementation Details

Students will be assigned to work in groups of 3-5 people[3] to solve programming problems over the course of a 2-hour weekly lab. PBJs emphasize the verbalization of programming concepts and strategies as well as paper sketches (rather than directly typing code). In this way, students are validating their code with each other rather than on the computer. Once a solution is derived, each group member must give it an individual "thumbs-up". The collective ownership of the solution means the members in the group are responsible for making sure their peers understand and can defend their solution[4]. The group sharing of knowledge and strategies benefits all group members.

There are two key innovative components of PBJs: 1) problem solving commences with conversations and diagrams, rather than with code; and 2) each group member is held accountable for their group's solution. We believe that the higher degree of ownership and accountability of work, combined with the required verbalization and written depiction of strategy will lead to deeper understanding and more satisfaction for the students[5].

Our goal is that the students participating in the PBJ pilot will be able to:

- Verbally/pictorially decompose a problem into smaller parts

- Explain a solution with words and drawings to a peer who does not understand it
- Describe multiple plausible approaches in English and model a potential solution on paper
- Reason about which type of programming flow should be used (if, loops, LC, recursion) and what variables and data structures should be used (lists, strings, dictionaries, etc)
- Contribute meaningfully to effective team dynamics

Specific improvements we hope to observe are as follows:
- removing the immediate feedback from the compiling/execution of code will promote better understanding of programming constructs
- students view their peers as valuable resources and view themselves as integral classroom members
- students can identify and explain why the chosen appropriate data structure and program flow is appropriate and optimal
- a closer knit classroom with greater sense of community (a more inclusive classroom)
- a higher level of articulation of the programming process (a critical skill for CS majors)
- more overall confidence in coding and ability to self-advocate (necessary for women who enter the male-dominated field of CS)
- group dynamics and peer-to-peer learning may reduce course cultural dependency on drop-in hours

Staffing
Each group in the PBJ lab will have a “coach”. Coaches will be carefully selected and trained students who have previously succeeded in CS111 and possess the maturity and interpersonal skills to assist with group dynamics (rather than merely supplying answers). Coaches will be knowledgeable about group dynamics and willing to proactively coach their teams.

Assessment/Evaluation
The pilot is designed to be contained within one of seven CS111 labs, so we can compare the PBJ pilot lab to the six other labs. We will assess the PBJ pilot students’ performance on quizzes, problem sets and exams, as well as through surveys and interviews. With the help of Dr. Sarah Pociask, we will survey our students with respect to sense of community (both within CS111 and within the CS department), individual self-confidence and positive attitude throughout the semester.

We will construct specific questions on weekly problem set quizzes and/or problem set reflections (for all CS111 students) that will measure how well students can identify and explain their choice of data structure and program flow. We will also track students’ ability to select and explain across the semester (both within the PBJ lab and across the entire CS111 course).

As a side note, CS111 uses web-based Codder (provides automated real-time code feedback, written in-house) and tracks Codder usage. We can examine the relationship between Codder use and course performance (the range of Codder is wide: some students use it hundreds of time per problem, others never use it). A cursory glance at the data shows that some students reflexively use Codder without anticipating what will happen; we hope the PBJ lab will reduce this reflex.

We will track drop-in hours via Tutor-trac (the PLTC’s cumbersome, non-intuitive software for tutors that records tutee sign-ins) and can compare to previous semesters of the course.

The PBJ student assistant will perform the following tasks:

Before the semester begins (~3 hrs total):
-- cull training materials on effective group dynamics and collaborative learning and assemble a training packet. The training packet will have required reading in advance of training that covers, among other things, how to encourage students to articulate their coding plans and how to best enforce even group contribution.
-- poll coaches and determine a suitable day and time slot for training

On a weekly basis (~2.5-3.0 hours/week):
-- help delineate and refine quiz/reflection questions to assess articulation and defense of code choices (both data structure and program flow)
-- extract the data from the google-form reflections or hand-written quizzes and analyze skills of interest (articulation ability and defense of code choices) with respect to PBJ and non-PBJ students
-- compare/contrast PBJ data to non-PBJ data (more time-consuming as the semester progresses and there are more data available)
-- share relevant PBJ materials with coaches in advance of the PBJ lab
-- check-in with coaches after each session with feedback/ways to improve
-- collect feedback from the students' mini-reflection at the end of the PBJ session for what worked and what didn't
-- attend weekly cs111 coaches' staff meeting for brainstorming and feedback
-- document events of each PBJ lab (content, coverage, effectiveness, reactions) in shared google doc as record of the semester

References

1. Stockwell, B.R., Stockwell, M.S. & Jiang, E. Group Problem Solving in Class Improves Undergraduate Learning, ACS (American Chemical Society) 2017 Jun 28; 3(6): 614–620.
2. Michaelsen, L.K., Knight, A.B. & Fink, L.D. (Eds.) (2004). *Team-based learning: A transformative use of small groups in college teaching.* Sterling, VA: Stylus.
3. Collaborative Learning: Group Work, Cornell University Center for Teaching Innovation.
4. Abel, Z. Team work in MIT's 6.042 *Math for Computer Science,* Spring 2018.
5. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H. & Wenderoth. M.P. *Active learning increases student performance in science, engineering, and mathematics*. 2014 PNAS June 10, 111 (23) 8410-8415.

Budget (attached as separate document)