2019 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
SCIENCE, TECHNOLOGY & ENGINEERING, ARTS, MATHEMATICS & EDUCATION JUNE 5 - 7, 2019
HAWAII PRINCE HOTEL WAIKIKI, HONOLULU, HAWAII

BLENDING TEAM, PAIRED, AND INDIVIDUAL WORK IN A COMPUTING COURSE: USING BEST PRACTICES

VANDEGRIFT, TAMMY
DEPARTMENT OF COMPUTER SCIENCE
SHILEY SCHOOL OF ENGINEERING
UNIVERSITY OF PORTLAND
PORTLAND, OREGON

Dr. Tammy VanDeGrift
Department of Computer Science
Shiley School of Engineering
University of Portland
Portland, Oregon

Blending Team, Paired, and Individual Work in a Computing Course: Using Best Practices

Synopsis:

The presentation will describe a data structures course that combines in-class teamwork, in-class paired lab work, and out-of-class individual work. Students apply skills through in-class activities, paired-labs, individual programming projects and exams. Research questions include RQ1: How did student attitudes toward group and individual learning change over the term? RQ2: How did teamwork and lab activities contribute to learning? Students felt value in the team activities in terms of solidifying individual understanding of concepts through multiple perspectives.

Blending Team, Paired, and Individual Work in a Computing Course: Using Best Practices

Tammy VanDeGrift
Donald P. Shiley School of Engineering
University of Portland
Portland, OR 97203

ABSTRACT

The paper describes the organization and materials for a data structures course that combines in-class teamwork, in-class paired lab work, and out-of-class individual work. The course is organized around weekly topics with two class sessions per week utilizing teamwork for conceptual understanding and one session per week utilizing pair programming for implementation practice. Students apply skills through individual programming projects and exams. Research questions include RQ1: How did student attitudes toward group and individual learning change over the term? RQ2: How did teamwork and lab activities contribute to learning? Survey data, team activities, homework assignments, and exams were collected from students who consented to participate in the study in Spring 2018. Overall, students' attitudes shifted more positively to enjoy learning things with other people, seeing the value of teamwork, and seeing the importance of working in groups in the profession, although none of these changes were statistically significant. Students felt value in the team activities in terms of solidifying individual understanding of concepts and students noted that participation by all members was high. When asked about process skills, students were mixed about improving their skillset in team roles. Students' perceived benefits of group work was gaining multiple perspectives from peers and understanding content. Labs helped students practice with coding and understanding concepts. Results from the final exam show that questions most closely linked to the teamwork and labs had similar grades as questions most closely linked to homework, suggesting that all three forms of practice are beneficial to student learning.

1 Introduction

With the Associate of Computing Machinery (ACM) Special Interest Group on Computer Science Education celebrating its 50th year of conference meetings in 2019, this paper describes the application of best education practices developed over 50 years from the CS education community. As the history of computing curricula demonstrate, the focus from content to professional practice and skills is apparent. The Computing Curriculum (CC) recommendations from 1968 provide detailed course titles, topics, and curriculum prerequisite structures [1]. The report prescribes required and elective course content; the computing discipline was young and establishing a shared progression of learning was important. The curriculum outcomes focused on knowledge areas, such as algorithmic processes, programming, computer organization, discrete structures, and numerical mathematics. There was no mention of professional practice skills, such as communication and teamwork. Moving to the 1978 set of recommendations, the report provides eight core computing courses along with prerequisite structures [2]. The objectives for CS graduates are listed, including an aspect of teamwork: "be able to assess the implications of work performed either as an individual, or as a member of a team" [2]. The authors encourage consideration of using teams and states "This technique is essential in advanced level courses and should be attempted as early as possible in the curriculum" [2]. In fact, CS 14: Software Design and Development has listed teamwork as an integral part of the course.

Computing curriculum 1991 provides knowledge areas for the core computing curriculum, but we see more emphasis on the professional practices of teamwork and communication skills [16]. The knowledge areas are tied to lecture hours instead of individual courses to allow for more curricular flexibility. In fact, the report includes curriculum design considerations for combining knowledge units into courses. We also see the first mention of laboratories as an instructional technique with integrated lectures and lab. Learning occurs from the interaction among students, instructors and the subject matter. The terms open labs and closed labs appear: Open labs are for unsupervised assignments for students to complete at their convenience and does not require direct supervision. Closed labs are scheduled, structured, and supervised using hardware, software, or instrumentation [16]. Beyond the computing knowledge areas, the report emphasizes additional experiences for students in working on teams, written and oral communication, and familiarizing with the profession [16].

CC 2001 includes professional practice, with skills related to management, ethics, communication, teamwork, and life-long learning [11]. One specific recommendation is to create opportunities to work in teams relatively early in the curriculum. The 2013 version further emphasizes these professional practices, in addition to risk tolerance, time management, collegiality, patience, work ethic, social responsibility, and diversity [10]. As one can see over the past 50 years, computing curricula have evolved from specific course descriptions to outcomes that span technical skills, personal skills, and professional responsibility.

The course design and activities presented in this paper utilize best practices of closed labs and teamwork. In particular, pair programming is used in the closed labs, since it has been an effective practice for learning and in industry [5,14,15,17]. Another set of best practices is the use of active learning during lectures through team-based inquiry exercises [3,4,6,7,8,9,12,13]. Contributions in this paper include course design that includes the best practices of teamwork, pair programming lab exercises, and open lab homework assignments and assessment of the practices in terms of students' attitudes and contributions to their learning.

2 Context

The data structures (DS) course is required for both the Computer Science (CS) and Electrical Engineering (EE) programs at a comprehensive, regional university on the west coast of the USA. The university serves about 4000 undergraduates with about 100 EE majors and 180 CS majors across the four-year programs. The spring 2018 course section had 34 students. All 34 completed the pre-semester survey and 30 completed the post-semester survey and provided consent for their course materials to be included in this IRB-approved study. The participants included 13 CS majors, 16 EE majors, and 1 double major in EE and CS.

The prerequisite to DS is an Introduction to Computer Science course taught with Java, including variables, types, I/O, selection, loops, arrays, object-oriented design, inheritance, file I/O, and recursion. The DS course departs from the introductory course in terms of language; instead of using Java, the DS course uses C. Therefore, the first four weeks of the course focus on practice with C, especially memory management, pointers, and data separated from functions. The remainder of the semester covers classic data structures. A more detailed schedule can be found in Table 1.

Research questions guided the data collection and analysis:

RQ1: How did student attitudes toward group and individual learning change over the term?

RQ2: How did teamwork and lab activities contribute to learning?

3 Structure of Content and Student Work

The course met three times per week (Monday, Wednesday, Friday) for 55-minute sessions. Monday and Wednesday sessions met in a regular classroom, utilizing 19 group activities. Friday class sessions met in a computer lab to complete 11 labs. Seven individual homework assignments assessed programming data structures and operations on data structures in C.

Table 1: Schedule of topics and course activities

Week	Topics	Lab and HW Activities
1	C variables, I/O, pointers	Lab: Linux and gcc HW 0 due
2	C arrays, strings, malloc/free, structs	Lab: pointers, arrays, memory
3	C preprocessor, libraries, file I/O	Lab: structs, I/O, preprocessor
4	C style, review, exam 1	Lab: make and gdb HW 1 due
5	Binary search, recursion, algorithm analysis	Lab: recursion
6	Linked lists	Lab: linked lists HW 2 due
7	Stacks and queues	Lab: stacks, queues
8	Queues, review, exam 2	Lab: stacks, queues (cont); HW 3 due
9	Trees, BSTs	Lab: trees
10	Trees, sorting	Lab: no class for Easter recess; HW 4 due
11	Sorting	Lab: sorting
12	Graphs	Lab: graphs HW 5 due
13	Graphs, review, exam 3	Lab: graphs (cont)
14	Hashing, hash tables	Lab: hashing HW 6 due
15	Final exam	

Assessment of student learning included the following categories and percentages: 30% in-semester exams (individual), 15% final exam (individual), 25% homework assignments (individual), 20% pre-labs and labs (pre-labs individual, labs paired), 10% professionalism (individual, based on participation and teamwork during class sessions). All but the lab grades used individual assessment throughout the semester.

3.1 Exams

In-semester and the final exams included multiple-choice questions (code-tracing, choose the appropriate data structure for the task), code-reading, and code-writing questions. Due to the time constraints, code-writing questions were scaled to be at the level of code a single function or code a single data structure instead of at the level of designing interactions among multiple functions.

3.2 Team Activities

Active learning techniques have been shown to be successful for lowering D/F/W rates and increasing student retention [7]. One such technique is Process-Oriented Guided Inquiry Learning and most lecture sessions included a team activity using POGIL [6]. Students were asked to examine code, diagrams, and pseudocode and teams worked together to answer a series of questions. Teams of four (with one five-person team) were instructor-chosen to balance experience levels and majors. The same teams continued throughout the semester with the roles rotating among team members. The Manager ensured the team stayed on task, watched the clock, and ensured all members had a chance to speak. The Recorder recorded the discussion of the team and the team's answers. The Presenter spoke on behalf of the team during whole class discussion. The Reflector observed group dynamics and made suggestions to improve team function.

While student teams worked on activities, the instructor moved around the room to check pace and answer questions the teams had about content and/or the tasks. Activities often included stretch goals, so teams that worked quickly stayed engaged. After most teams completed the activity, the instructor led a whole-class discussion, often polling the presenters for the team's answer. The recorder submitted written answers, so the instructor could check for misconceptions to address the following class session.

3.3 Pre-Labs and Labs

Each closed lab had two components: a pre-lab worksheet completed individually to ensure students had practice with the data structure and/or topic and lab exercises that asked students to write functions using pair programming, an effective practice as demonstrated by many studies [14,15]. Students were instructed to switch the driver and navigator roles after each checkpoint. The pre-labs were graded according to correctness. The lab exercises were graded in real-time during the lab session; the instructor and teaching assistant examined code and output for the exercise. The lab grades corresponded to how many checkpoints the pair completed; any checkpoints not completed during the class session were due a week following the lab.

3.4 Homework Assignments

The assignments asked student to implement programs and answer post-program reflection questions. All assignments included starter code, so students could focus on the new data structures content. HW0, HW1, and HW3 asked students to complete the same program with differences in implementation. The assignment was a store receipt (basic collection of items). HW0 used Java and arrays. HW1 used C and arrays. HW2 used C and linked lists. A second grouped set was HW2 and HW4, with both solving the same problem: determining if a maze has a path from start to finish. HW2 used recursion and HW4 used stacks (depth-first search) and queues (breadth-first search). HW 5 focused on trees for the game state of tic-tac-toe. HW 6 focused on graphs to find shortest paths of flights among cities. The purpose of the post-reflection questions was to provide evidence of testing, analysis, and understanding of the code and execution.

4 Data and Analysis

To answer the research questions, survey answers and exam scores were collected from participants. An initial paper survey (PRE) was given to students to complete anonymously during the first week of the semester. A longer paper survey (POST) was given to students to complete anonymously during the last week of the semester. The PRE survey was brief to get an idea about students' preferred learning styles and if they value teamwork. The first four questions had the answer set of "strongly disagree", "disagree", "agree", and "strongly agree".

1. I enjoy learning things on my own
2. I enjoy learning things with other people
3. Learning with and from others is a valuable skill
4. Working in groups is an important part of my future profession

The final question was open-ended: Would you recommend that an instructor use group activities in their courses? (Yes/No) Why or why not?

The POST survey included all questions from the PRE survey as well as new questions that focused on the teamwork activities and labs. The Results section below contains more details from the POST survey.

Final exams from participants' were collected to determine if questions that aligned more with team activities, with labs, or with homework had the highest success rates.

5 Results

The results are presented according to the research questions.

RQ1: How did student attitudes toward group and individual learning change over the term? Answers to the common questions on the PRE and POST surveys were used to answer RQ1. Because the surveys were anonymous, only aggregate changes could be examined. Figure 1 shows the bar chart for the PRE survey responses (N=34) and Figure 2 shows the bar chart for the POST survey responses (N=30).

The Mann Whitney U Test was applied to the PRE survey responses per question and POST survey responses per question since the data is ordinal and not numerical. None of the p-values indicated that the distributions are not equal, so we cannot conclude that students' attitudes changed across the semester. (profession p-value .280, skill p-value .112, others p-value 0.097, own p-value 0.430). However, of note is that there are no Strongly Disagree responses in the POST survey and there are fewer Disagree responses to the question about enjoyment in working with others. If the responses are categorized into "agreement" and "disagreement", we see a higher percentage of agreement in the POST survey on questions 2 and 4:

1. I enjoy learning things on my own (PRE: 67.6%, POST: 66.7%)
2. I enjoy learning things with other people (PRE: 82.3%, POST: 96.7%)
3. Learning with and from others is a valuable skill (PRE: 97.1, POST: 96.7%)
4. Working in groups is an important part of my future profession (PRE: 91.2%, POST: 96.7%)

Students were asked if they would recommend that the instructor use group activities. Of the PRE responses, 29 of 33 said yes and 4 of 33 said no, and one was blank (87.9% yes). Of the POST responses, 28 of 30 said yes and 2 of 30 said no (93.3% yes). Again, there is no statistical significance for difference in populations, but the overall percentage for group work recommendations improved.

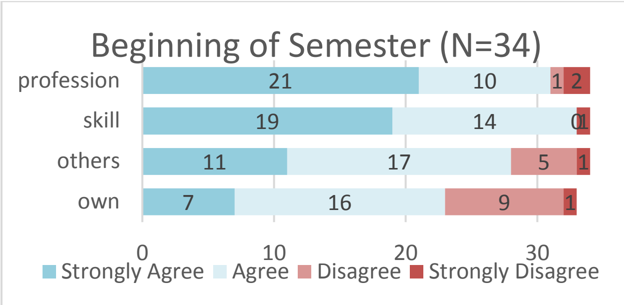

Figure 1: Responses to PRE survey by Strongly Agree to Strongly Disagree

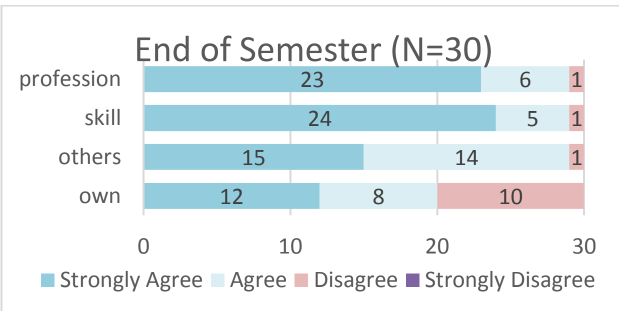

Figure 2: Responses to POST survey by Strongly Agree to Strongly Disagree

The second research question was: How did teamwork and lab activities contribute to learning? This question was answered using responses to the POST survey and scores on final exam questions. Students were asked how much each of the course activities helped them learn (answers: N/A, None, Not Much, Some, A Lot). Figure 3 shows the responses for each course activity. All but the videos and textbook received at least 85% of respondents stating the activity helped them a lot or some.

One of the POST survey questions was: To what extent did you make gains in each of the following learning outcomes? (N/A, None, Not Much, Some, A lot). Eleven of the learning outcomes listed are content-focused. Four of the learning outcomes are process-focused. A score was calculated for each learning outcome (A lot = 4, Some = 3, Not much = 2, None = 1, N/A = 0) in order to rank the outcomes by quantitative score. Table 2 shows the ranked order of outcomes.

Questions related to the in-class group activities on the POST survey were rated by the following categories: Strongly Disagree, Disagree, Agree, Strongly Agree. The survey was designed so students had to choose agreement or disagreement. For this analysis, Strongly Agree = 2, Agree = 1, Disagree = -1, and Strongly Disagree = -2. All responses had an overall positive score, with "I developed at least one skill through group work" having the score closest to zero. Table 3 shows the statements, the aggregate score, and the percentage of students who agreed or strongly agreed with the statement.

Statements related to the labs were answered on the same scale as the group activity statements and presented in the same way and shown in Table 4.

Note that "I would enjoy having group activities in future courses" was presented in the lab section of questions to verify consistency of answers. Of 30 respondents, 23 made the same mark on both. Six had different marks (SA to A or A to SA) and one student had Strongly Agree for the group work and Disagree for the labs. Only two students disagreed on both statements and also consistently responded to the open-ended question that group work should not be recommended in future courses.

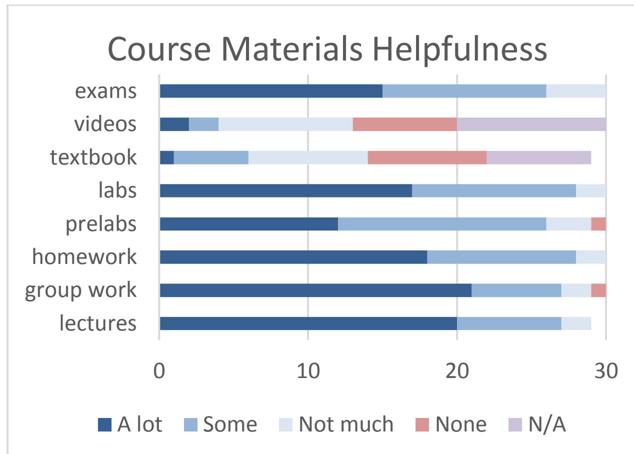

Figure 3: Responses to POST survey regarding course materials

Table 2: Outcomes rated by students' learning gains. *shows process outcomes

Outcome	*Score*
Understanding code written in C	3.90
Using Stacks	3.87
Using Queues	3.80
Using Trees	3.80
Writing programs/functions in C	3.77
Using recursion	3.63
Using linked lists	3.53
Working with others to solve problems *	3.53
Understanding sorting routines	3.50
Finding and using resources to solve problems *	3.50
Using graphs	3.47
Confidence in ability to do computer science *	3.40
Explaining my ideas to others *	3.37
Analyzing running time of programs	3.27
Using hash tables and hash functions	2.90

Table 3: Statements ranked by students' in terms of group work. Score = Average of Strongly Agree (2), Agree (1), Disagree (-1), Strongly Disagree (-2).

Statement	*Score*	*Percent SA/A*
Students in my group came prepared and willing to participate	1.63	100.00
Students in my group listened to one another	1.63	96.67
Students in my group participated freely in discussions	1.57	100.00
Everyone contributed to the success of the activity	1.53	96.67
Students in my group asked questions when in doubt	1.50	96.67
I solidified my own understanding of concepts when other group members explained their understanding of concepts	1.50	93.33
I would enjoy having group activities in future courses	1.47	93.33
Clear explanations were given by group members	1.37	90.00
I solidified my own understanding of concepts when explaining them to my group members	1.37	90.00
No one dominated on my team	1.33	90.00
The group activities encouraged me to apply the code or algorithm	1.17	90.00
The group activities encouraged me to invest in reading and understanding the code or algorithm	1.17	86.67
I developed at least one skill (manager, spokesperson, reflector, notetaker) through the group work	0.10	50.00

Table 4: Statements ranked by students' in terms of lab work in partners. Score = Average of Strongly Agree (2), Agree (1), Disagree (-1), Strongly Disagree (-2).

Statement	*Score*	*Percent SA/A*
My partners came to lab prepared	1.60	100.00
I solidified my own understanding of concepts when explaining them to lab partners	1.53	100.00
I solidified my own understanding of concepts when other lab partners explained their understanding of concepts	1.50	100.00
Both partners contributed to the success of the labs	1.43	96.67
I would enjoy having group activities in future courses	1.37	90.00
No one dominated in my lab partnerships	1.27	90.00

Most final exam questions corresponded with activities from the course:

1. Series of multiple choice questions about variety of topics; some code tracing; some big-O analysis; some choose the best data structure
2. Debugging: find four errors in code snippet
3. Big-O analysis of functions
4. Linked lists: code to delete first node with positive value
5. Linked lists: code to determine if two linked lists have equal length
6. Queues: tracing enqueue and dequeue operations
7. Trees: code to recursively determine how many nodes in the tree had a key between min and max
8. Binary search tree: draw tree after inserting keys, tree traversal for search and delete
9. Graphs: depth-first search execution, find minimum spanning tree using Kruskal
10. Hash tables: insert items into a hash table (open address linear probing and open address double hashing)

Problems 1 and 2 did not have explicitly linked course activities, since multiple choice covered a range of topics and the second problem asked students to debug faulty code. The multiple-choice questions had an average score of 70.1%. The debugging question had an average score of 50.8%. Table 5 shows the question content areas, the corresponding activities, and the average exam scores. The overall exam average was 79.3%. (Note that the ten questions had unequal weights toward the overall score.)

Table 5: Final exam scores by question; each question had one or more course activities that related to the question

#	**Topic**	**Group work**	**Lab**	**HW**	**Average**
3	Big-O	X			83.3%
4	Delete node	X	X	X	70.3%
5	Equal length		X		86.4%
6	Queues	X			87.2%
7	Trees		X	X	86.3%
8	BSTs	X			89.7%
9	Graphs	X	X	X	91.0%
10	Hashing	X			91.1%

6 Limitations and Reflection

Limitations: A limitation of this study is that student names were not collected for the PRE and POST surveys. This was an intentional design to encourage honest feedback from students; however, we cannot look at individual student changes in attitude. The study design favored honest data from students over the ability to link the PRE and POST answers.

In addition to the PRE and POST surveys being anonymous, a second limitation of this study is that the data was collected from a single cohort during one semester from one university. A subset of this data (just group work) was collected during a prior semester and had similar results. This version of the study was to look more comprehensively at all course activities. Activities may work at one institution and not as well at another institution due to institutional culture, size of classes, and the population of students.

Discussion: The PRE and POST survey results show that students' overall attitudes toward group work and individual work shifted in the positive direction. Although not statistically significant, the results indicate that using a blend of group work and individual work may have contributed to this slight change as students became more experienced in working in teams.

The combination of POST survey responses and final exam scores show that all forms of work assisted in the learning process. At least 90% of students agreed or strongly agreed with almost all statements regarding the group activities and lab activities, so they perceive benefits to the course components. The one skill that was not as well-developed was team roles as manager, spokesperson, reflector, and manager. The instructor introduced these roles and responsibilities at the beginning of the semester and rotated roles among teammates for each activity, but there was little continued emphasis on roles throughout the semester. The instructor observed groups working during class, but the instructor focused more on the content and process of learning than the team roles. It is not surprising that this statement had a 50/50 split on agreement and disagreement about gaining skills in the roles.

While no official observational data was collected, the instructor witnessed high student engagement in class and lab activities. There were no team issues reported by students, probably because all teamwork was conducted during class sessions. There was no need to coordinate schedules outside class time and no disputes about some team members spending more time than others on a project. The average attendance rate in the class was 98.1% with the minimum attendance rate being 90.0%. Attendance levels indicate that students placed value on the activities conducted during class time. There was also high student engagement during office hours to discuss course concepts and to get additional help with the homework assignments.

While not an official part of the data collection, it is noted that one goal of the course is to give students opportunities to meet a variety of people and build community. Students worked with the same group of four students during the group work, but worked with ten different students during the lab activities. As a small, private, comprehensive university, a feature of the educational environment is to foster this sense of community among students.

From the instructor's perspective, group work during lecture time is a departure from a more traditional "show the students" approach to teaching and learning. While there were a few lectures to demonstrate the material, much of the content delivery was moved to the group work activities. One advantage of this approach is that the instructor gets real-time feedback about student misconceptions by observing group answers and leading class discussions. A second advantage is that students get multiple perspectives from several people instead of the single perspective of the instructor.

Students' perceived benefits of group work and labs support the instructional goals of giving students multiple perspectives and a peer network. Students were asked to consider the in-class group activities and the lab activities prior to the agreement/disagreement to statement questions, so that students could give their honest view of the activities prior to the closed-formed statements. For each of the group work and lab work, students were asked what was the best outcome for you? The responses were coded into emergent themes. For the benefits of group work, themes and number of responses to themes were:

- Perspectives of peers (12)
- Understanding content (7)
- Teamwork (3)
- Practice with material (3)
- Communication (2)
- Creating a community (1)
- No benefit (1)

Three students did not respond to the question. Here are some sample responses from students: "They really helped me understand the concept when we were first learning it as I got the input of various different people and was able to work through problems with them. [Participant 5]", "Getting the chance to be wrong before I had to apply skills. [Participant 20]", "Being able to speak my thoughts was a great way to error check. Also, going into this course, I had no idea how to converse professionally about CS but now I am better. [Participant 21]", and "First it created a little study group option if we didn't know that many people second it got second and their ideas to bounce around, if one explaination (sic) didn't cover it, someone else would [Participant 24]".

Lab benefits were slightly different than the group work benefits. Students appreciated the chance to practice coding with the instructor present for questions and confusion. The themes from responses were:

- Coding practice (12)
- Understanding content (11)
- Building confidence (2)
- Good preparation for HW (2)
- Can ask questions / get feedback (2)
- Can make mistakes (1)
- Working with peers (1)
- Active learning (1)

Three students did not respond to this prompt. Sample responses include: "It helped a lot to have to actually write code on the subjects we learn in class so I can see if I really get the concept or just the general ideas. Labs help me a lot for doing HW & studying for exams. [Participant 3]", "It gave me a chance to work with actual coding first hand instead of just reading it. Made me more confident about being able to write functions in C. [Participant 5]", "Great. I got to practice the course material with the instructor right there to help if I needed. [Participant 9]", In sum, blending teamwork, paired lab work, and individual homework is a successful methodology for student engagement and learning based on the past 50 years. Let's see what we learn about best engagement practices in the next 50 years.

REFERENCES

[1] W.F. Atchison, et al. 1968. Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. *Communications of the ACM*, 11(3), pages 151-197.

[2] R.H. Austing, et al. 1978. Curriculum 78: recommendations for the undergraduate program in computer science – a report of the ACM curriculum committee on computer science. *Communications of the ACM, 22*(3), pages 147-166.

[3] J. Bergin. 2006. Active learning and feedback patterns: version 4. In *Proceedings of the ACM Conference on Pattern Languages of Programs*, article 6.

[4] T.A. Budd. 2006. An active learning approach to teaching the data structures course. In *Proceedings of the ACM SIGCSE Symposium*, pages 143-147.

[5] M. Celpkolu and K.E. Boyer. 2018. The Importance of Producing Shared Code Through Pair Programming. In *Proceedings of the ACM SIGCSE Symposium*, pages 765-770.

[6] CS-POGIL. 2016. Process Oriented Guided Inquiry Learning in Computer Science. http://cspogil.org. Last accessed August 30, 2018.

[7] S. Freeman et al. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences of the United States of America.* https://doi.org/10.1073/pnas.1319030111

[8] H.H. Hu and T.D. Shepherd. 2013. Using POGIL to Help Students Learn to Program. *ACM Transactions on Computing Education, 13*(3), article 13.

[9] H.H. Hu and T.D. Shepherd. 2014. Teaching CS 1 with POGIL activities and roles. In *Proceedings of the ACM SIGCSE Symposium*, pages 127 – 132.

[10] The Joint Task Force on Computing Curricula (ACM, IEEE Computer Society). 2013. *Computer Science Curricula 2013, Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, New York.

[11] The Joint Task Force on Computing Curricula (IEEE Computer Society, ACM). 2001. *Computing Curricula 2001 Computer Science*. ACM, New York, New York.

[12] C. Kussmaul. 2012. Process Oriented Guided Inquiry Learning for Computer Science. In *Proceedings of the ACM SIGCSE Symposium*.

[13] C. Mayfield. 2017. Guiding students to understand CS concepts and develop process skills using POGIL. *Journal of Computing Sciences in Colleges, 32*(5), May, pages 102-103.

[14] A.D. Radermacher and G.S. Walia. 2011. Investigating the effective implementation of pair programming: an empirical investigation. In *Proceedings of the ACM SIGCSE Symposium*, pages 655-660.

[15] B. Simon and B. Hanks. 2007. First year students' impressions of pair programming in CS1. In *Proceedings of the International Computing Education Research Workshop*, pages 73-86.

[16] A.B. Tucker. 1991. Computing Curricula 1991. *Communications of the ACM, 34*(6), pages 68-84.

[17] L. Williams. 2007. Lessons learned from seven years of pair programming at North Carolina State University. *ACM SIGCSE Bulletin, 39*(4), pages 79-83.