Flipping an Introductory C Programming Course

Nancy Warter-Perez[1,2] and Jianyu Dong[2]

[1]Department of Mechanical Engineering, [2]Department of Electrical and Computer Engineering

California State University, Los Angeles

In flipping an introductory C programming course, EE 2450 – Embedded System Programming I, an interactive textbook (Zybook) was used to introduce students to course content through pre-class readings and textbook activities. The Zybook covers the fundamental concepts and students test their understanding through a variety of interactive exercises ranging from simple multiple choice to more complex programming assignments. Class time was mostly dedicated to a variety of active learning activities designed to help the students apply their knowledge to develop algorithms, to code increasingly more complex programs, and ultimately to complete a term project. Course assessments included programming assignments, in-class quizzes, and a final exam.

During the flipped course offerings in Fall 2016 and Fall 2017, in-class activities included 1) incorporating relatable problems and analogies, 2) paired-programming, and 3) group quizzes and peer-graded pop-quizzes. By using relatable problems and analogies, such as algorithms for making guacamole and embedded programs to control vending machines, students can more easily wrap their heads around challenging programming concepts. With paired programming one student was the driver at the keyboard (or notepad during algorithm development) who would describe what she/he was doing as she/he typed, and the other student was the navigator who would point out possible missteps (typos or question why something was being done a particular way). Partners would switch off every 5 minutes. This kept them engaged as the new driver would have to pick up where the previous driver left off. To help students review readings, group and peer-graded pop quizzes were used.

The goal of flipping a programming course is to allow all students to be able to master programming with help from their classmates and instructor. Overall, students performance improved, especially the stronger students who were able to tackle more challenging problems as compared to a traditional course. However, there were still a fair number of students who were not able to master fundamental programming concepts and successfully pass the course.

The design of the flipped course, the implementation and refinement of the in-class activities, and observations and preliminary results will be presented from the Fall 2016 and Fall 2017 offerings, as well as preliminary observations from the Spring 2019 offering. During Spring 2019 a number of improvements will be implemented based on student feedback and discussions with other faculty flipping courses, including: 1) clicker/poll-everywhere questions (instead of group/pop-quizzes) to help students quickly review important concepts from the readings; 2) more straight-forward, simple programs that allow students to master syntax and semantics before attempting more complex programs that require more critical thinking skills; 3) modeling problem solving before asking students to program (rather than solely relying on the Zybook examples), where the instructor will be the driver and a group of students in the class will be the navigator; 4) small white-boards for algorithm development to allow students to easily modify and refine their algorithms; 5) instructor formed teams; and 6) one-minute papers to identify which concepts students are struggling with.