Learning two programming languages in one semester does not adversely affect undergraduate biomedical engineering student performance

Trevor R Ham, The University of Akron

I am a 4th year PhD student studying Biomedical Engineering at The University of Akron, as well as a member of the College of Engineering Teaching Program. I teach a programming course (using MATLAB) to Biomedical Engineering undergraduates. My research interests concern protein-functionalized hydrogels to guide neural stem cell differentiation, with a focus on regenerating damaged spinal cord tissue.

Dr. Rouzbeh Amini, The University of Akron

Dr. Amini completed a Ph.D. in Biomedical Engineering at the University of Minnesota in the field of ocular biomechanics and biotransport in 2010. He then continued his research work on the mechanics of soft tissue as a postdoctoral trainee at the University of Pittsburgh's Department of Bioengineering, where he held the Ruth L. Kirschstein National Research Service Award (NIH F32). He conducted his postdoctoral research on the biomechanics of cardiac valves. Dr. Amini has served as an assistant professor in the Department of Biomedical Engineering at The University of Akron since August 2013. The overall goal of his research laboratory is to improve human health by studying the multi-scale biomechanics and biotransport in biological systems. Dr. Amini's research has been funded by the National Science Foundation, Akron Children's Hospital, Firestone Foundation, Conquer Chiari Foundation, and American Heart Association.

Learning two programming languages in one semester does not adversely affect undergraduate biomedical engineering student performance

Introduction

As the challenges that biomedical engineers are required to solve grow more complex and multidisciplinary,[1] we should adjust their undergraduate education to match. Recently, computer programming has become integral to the duties of all engineers, and with current trends in engineering technology and manufacturing (e.g., the "maker movement," the proliferation of data-gathering sensors on consumer products, etc.), this trend is likely to accelerate even more in the coming future. In addition, computer programing has become an essential part of data analysis[2-4] and simulation[5,6] in biomedical research. As interacting with computing-enable devices becomes easier and more prevalent, it is possible that engineers will graduate with the ability to quickly navigate user interfaces but without a detailed, theoretical knowledge of computing fundamentals. Especially in the rapidly-evolving, multidisciplinary field of biomedical engineering (BME), there is constant pressure to add new subjects to the curriculum and shorten the amount of time given to each topic. When teaching the principles of programming to non-computer science students, it is important to focus on broad, widely applicable concepts (i.e., computational thinking), rather than become mired in the application and syntax of one specific language.[7]

Active learning shows demonstrable and widespread improvements in student achievement over passive learning in science, technology, engineering, and mathematics (STEM) fields, allowing better retention of new concepts, more efficiently.[8,9] In BME, active learning can consist of problem/project-based learning,[10] which requires students to undergo the engineering design process on their own. These projects can be multidisciplinary, which improves student outcomes when compared to single-discipline projects.[11]

In our course (a 200-level course on biomedical computing), we sought to rationally design, introduce, and test a project which would require students to solve a problem using a variety of engineering skills. Going further, we were also interested in introducing a second programming language to the students halfway through the semester. In traditional computer science disciplines, students are often exposed to multiple languages (e.g., Java, C++) over multiple semesters. Taking that into consideration, and given the fact that learning multiple natural languages improves overall linguistic proficiency,[12] we hypothesized that requiring students to complete a project in a second programming language within a single semester would reinforce basic programming knowledge and have a statistically significant effect on final exam performance.

Methods

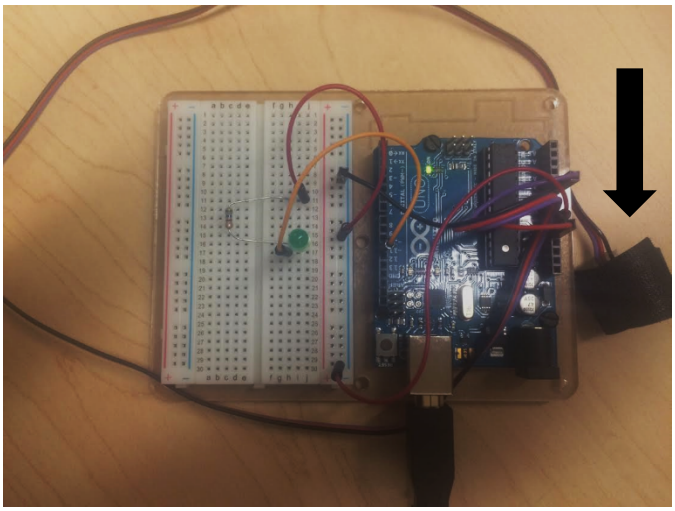

Figure 1: The heart rate monitor consisted of an Arduino UNO (blue) with a PPG (arrow) and green LED. For many students, this was their first opportunity to work with microcontrollers.

The experiments conducted in this study were performed under a formal exemption by the University of Akron Institutional Review Board (IRB) as research performed under normal academic settings. The study described herein took place within a single section of a one semester, 200-level course entitled Biomedical Computing. The general concept was to guide students to complete a project in one of two different programming languages. First, we allocated students into two evenly distributed categories based on exam 1 performance: after listing the students in descending order by exam 1 grade (so that "student 1" had the highest grade and "student 28" had the lowest), we assigned odd-numbered students into one group and even-numbered students into another. We then sorted each group into 5 teams (4 teams of 3 students, 1 team of 2 students) for each group. Second, both groups were instructed to build a heart rate monitor (HRM) from a commercially available photoplethysmograph (PPG, World Famous Electronics, New York City, NY, USA) and an Arduino UNO (Adafruit Industries, New York City, NY, USA). The device can be seen in Fig. 1. The first group was asked to write a program to collect and display the pulse data using Arduino Integrated Development Environment (AIDE, a scripting language that is freely available for use with Arduinos and which closely resembles C syntax), while the second group was asked to write the same program using MATLAB (the main focus of the course and the language used for all of the exams). To complete the project, the students needed to handle I/O (receiving data from the Arduino UNO), use loops to repeatedly calculate and update heart rate interval, use selection statements to filter the raw PPG data, and graph the results (both PPG and calculated heart rate). Last, we required both groups to complete a survey regarding their perceptions of the project and compared the final exam scores of both groups.

To complete the project, both groups were given one day of introductory exercises, with separate sessions for the MATLAB and AIDE groups. The MATLAB group was shown how to interface Arduinos with MATLAB, while the AIDE group was also shown the similarities and differences between MATLAB and AIDE. We briefly explained the differences between variable typing in MATLAB (weakly typed) and AIDE (strongly typed) without giving the students any formal framework. The students were then given 4 weeks to complete the project on their own. The final deliverable was a written report (5 pages maximum); students were graded based on the quality of the report itself and the structure/completeness of their code.

To analyze the data, we compiled anonymous survey results. Wherever numerical measures were not inherent, we categorized the different types of responses (e.g., in response to "Did this project help prepare you for the final exam?" we scored each response as Yes or No) and then counted the number of responses in each category. We also compared the final exam grades

between the two groups. The final exam was an accumulative exam designed to assess students' ability to master the course outcomes including,

- Understanding the use of logical operators, logical functions, and control structures in MA TLAB
- Utilizing special features of MATLAB including matrix calculations, plotting functions, symbolic toolbox, numerical integrations and differentiations, and ordinary differential equation solvers

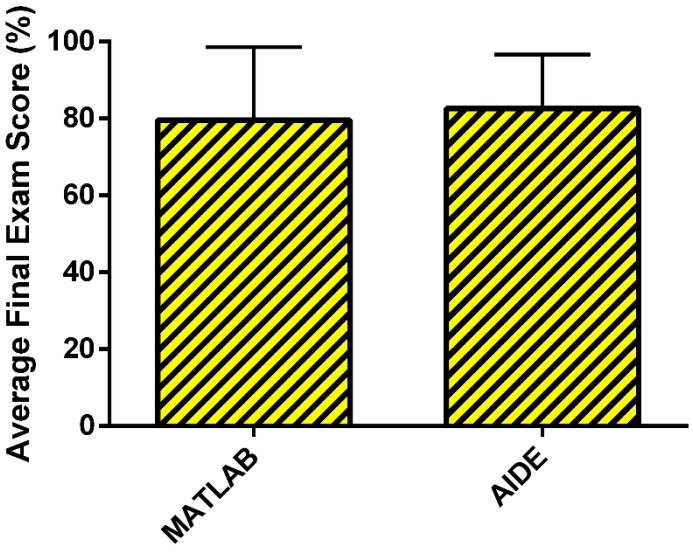

Figure 2: Requiring students to complete the project in a second programming language had no statistical benefit as determined by Student's t-test. Data reported as mean ± SD, n = 14 for each group.

Although no questions pertaining to Arduino UNO were asked in the final exam, we expected that the hands-on project would further improve students' knowledge and skills listed in the above course outcomes. For all statistical tests, an α of 0.05 was considered significant.

Results

A comparison of final exam grades between the two groups yielded no significant differences ($p = 0.622$, Fig. 2). Therefore, as applied by us, there was no statistical benefit to learning the second language within one semester. Importantly, however, such outcome also indicates that the converse is true: requiring students to learn a second programming language did not negatively impact their final exam performance. The student surveys showed that the overwhelming majority of students enjoyed the project (24/28). In fact, more students from the AIDE group reported enjoying the project (13/14) than did students from the MATLAB group (11/14). Interestingly, the majority of students in the MATLAB group indicated that they thought the project helped them to prepare for the final (10/14) while none of the students in the AIDE group thought so (0/14). A full summary of the survey results can be seen in

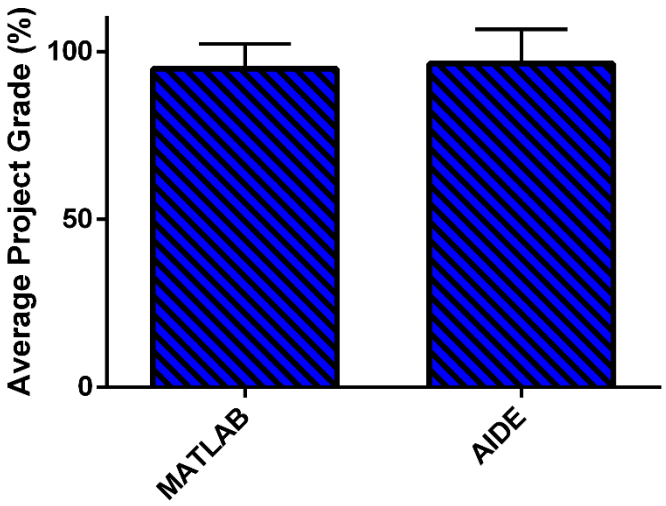

Figure 3: Average project grades for both groups. The overall averages are very close, while the standard deviation of the AIDE group is higher (10.1 vs. 7.4). Data reported as mean ± SD, n = 5 for each category.

Table 1. Due to our intentional grouping of students by grade, the quality of the reports was varied, especially within the AIDE group (Fig. 3), where grades ranged from 78% - 110%. Some groups failed to turn in working HRMs while other groups added bonus features, such as heart rate variability monitoring.

Table 1: Summary of survey results.

MATLAB Questions	**Response Summary**	**AIDE Questions**	**Response Summary**
Was MATLAB your first experience learning a programming language?	11/14: Yes	Was MATLAB your first experience learning a programming language?	11/14: Yes
Did you feel that completing a project in the MATLAB environment helped to prepare you for the final exam?	10/14: Yes	Did you feel that completing a project in a different programming language helped to reinforce concepts from the rest of the course?	9/14: Yes
Did you encounter any major bugs in your program?	10/14: Yes	How much time did you spend learning the syntax of AIDE?	Responses varied from 3 hours to multiple weeks
Was the amount of instruction provided inadequate, adequate, or more than adequate?	3/14: Less than adequate, 9/14: Adequate, 2/14: More than adequate	Did you encounter any bugs that were a direct result of having to use AIDE?	8/14: Yes
Did you enjoy the project?	11/14: Yes	Which language did you prefer, MATLAB or AIDE?	12/14: MATLAB, 2/14: Both
How much time did you spend completing the project?	Responses varied from 10 hours to 3 weeks.	Did completing the project in AIDE help to prepare you for the final exam or did it distract from the focus of the course?	0/14: Yes, 8/14 distraction
		Was the amount of instruction provided inadequate, adequate, or more than adequate?	5/14: Less than adequate, 9/14 Adequate
		Did you enjoy the project?	13/14: Yes
		How much time did you spend completing the project?	Responses varied from 10 hours to 3 weeks

Discussion

Our results did not confirm our hypothesis. However, they do point to a new hypothesis: that BME students can successfully incorporate a second programming language into a single semester class without losing comprehension of the material. This opens up new doors for their intellectual pursuits and may even lead to improved job opportunities. Importantly, the students were able to experience the difference between strongly (AIDE) and weakly (MATLAB) typed languages firsthand. Working in a strongly typed environment was something that many of the AIDE students reported as a major challenge that they had to overcome when completing the project. Due to the anonymous nature of the surveys, we cannot correlate between reported difficulty with the strongly-typed environment and project grade. However, we did not observe a connection between the quality/success of each group and any specific struggles with strong variable typing (i.e., groups who struggled did not struggle with variable typing *per se*). Broadening their range of exposure in this way is extremely important for students, especially if they intend to learn another programming language in the future. In addition, many students may opt to use Arduino microcontrollers in their senior design class and using AIDE is preferred in more complicated systems. While MATLAB has many advantages regarding numerical calculations and advanced engineering analyses, it is, in our opinion, less ideal than AIDE for the HRM project. Since most of the students had formal exposure only to MATLAB, the project also served to demonstrate that simply being familiar and comfortable with a tool does not mean that it is the best tool for every job.

As programming language choice varies between academia and industry (and within industry is divided amongst many different languages),[13] learning a specific language, associated syntax, and quirks should not be the focus of an undergraduate-level programming course. Rather, the focus should be on learning computational problem-solving and the skills necessary to use programming to solve real-world problems. We believe that learning a second language helps to accomplish this goal, as it demonstrates to students that the connection between programming and problem-solving is not language dependent, but rather that it revolves around a core set of skills. Additionally, students are exposed to the idea that they can apply these skills to new programming languages. As the final exam was undertaken in MATLAB, the students in AIDE groups could not observe a direct link between the project and preparing for the final. This observation may reflect perceptions on behalf of the students that the course is a "MATLAB" course, rather than a more general programming course. In the future, as we continue to implement elements of programming which are not strictly tied to MATLAB, this perception will likely change.

It is possible that our analyses did not detect a statistical effect due to a lack of refinement, small sample size, or imperfect implementation. A *post-hoc* power analysis (of the t-test shown in Fig. 2) indicated a power of 0.084. Future adjustments to the project may be needed (perhaps using a different, more universal language) to improve student outcomes. The simple method that we used for assessment is limited – perhaps, a difference might be observed if we followed students' performance in successive classes (e.g., senior design) or if we used a larger sample size (e.g., by comparing two sections of the same class). Regardless of whether future studies indicate a positive effect or not, we plan to continue this approach, as students were able to successfully incorporate a second language with no apparent loss of conceptual knowledge. In addition, we

intend to introduce the project earlier in the semester so that the students have more time to take advantage of this hands-on opportunity.

Conclusions

Having undergraduate BME students complete a project in a different programming language did not significantly affect final exam score. As such, students were able to learn a second programming language while not reducing their overall comprehension of the principles of programming.

References

1 Sharp, P. A. & Langer, R. Research agenda. Promoting convergence in biomedical science. *Science* **333**, 527, doi:10.1126/science.1205008 (2011).

2 Powell, T. A. *et al.* Elasticity of the porcine lens capsule as measured by osmotic swelling. *Journal of biomechanical engineering* **132**, 091008, doi:10.1115/1.4002024 (2010).

3 Amini Khoiy, K. *et al.* Surface Strains of Porcine Tricuspid Valve Septal Leaflets Measured in Ex Vivo Beating Hearts. *Journal of biomechanical engineering* **138**, doi:10.1115/1.4034621 (2016).

4 Kloefkorn, H. E. *et al.* Automated Gait Analysis Through Hues and Areas (AGATHA): A Method to Characterize the Spatiotemporal Pattern of Rat Gait. *Ann Biomed Eng*, doi:10.1007/s10439-016-1717-0 (2016).

5 Amini, R. & Barocas, V. H. Reverse pupillary block slows iris contour recovery from corneoscleral indentation. *Journal of biomechanical engineering* **132**, 071010, doi:10.1115/1.4001256 (2010).

6 Gsellman, L. & Amini, R. Patients With Intravitreal Gas Bubbles at Risk of High Intraocular Pressure Without Exceeding Elevation of Surgery: Theoretical Analysis. *Investigative ophthalmology & visual science* **57**, 3340-3347, doi:10.1167/iovs.15-18010 (2016).

7 Muller, C. L. & Kidd, C. Debugging geographers: teaching programming to non-computer scientists. *Journal of Geography in Higher Education* **38**, 175-192, doi:10.1080/03098265.2014.908275 (2014).

8 Freeman, S. *et al.* Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences of the United States of America* **111**, 8410-8415, doi:10.1073/pnas.1319030111 (2014).

9 Handelsman, J. *et al.* Education. Scientific teaching. *Science* **304**, 521-522, doi:10.1126/science.1096022 (2004).

10 Clyne, A. M. & Billiar, K. L. Problem-Based Learning in Biomechanics: Advantages, Challenges, and Implementation Strategies. *Journal of biomechanical engineering* **138**, doi:10.1115/1.4033671 (2016).

11 Hotaling, N., Fasse, B. B., Bost, L. F., Hermann, C. D. & Forest, C. R. A Quantitative Analysis of the Effects of a Multidisciplinary Engineering Capstone Design Course. *J Eng Educ* **101**, 630-656 (2012).

12 Griessler, M. The Effects of Third Language Learning on Second Language Proficiency: An Austrian Example. *International Journal of Bilingual Education and Bilingualism* **4**, 50-60, doi:10.1080/13670050108667718 (2001).

13 Ben Arfa Rabai, L., Cohen, B. & Mili, A. Programming Language Use in US Academia and Industry. *Informatics in Education* **14**, 143-160, doi:10.15388/infedu.2015.09 (2015).