2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #32318

A New Approach to Teaching Programming at Freshman Level in Mechanical Engineering

Dr. Pavan Karra, Minnesota State University, Mankato

Pavan Karra is Assistant Professor at Minnesota State University where he teaches in the field of Dynamics and Controls. He can be reached at pavan.karra@mnsu.edu.

A new approach to teaching programming at freshman level in mechanical engineering

Abstract:

Introduction to programming (MATLAB) course at Minnesota State University(MSU) has traditionally been a difficult class with high DFW rates. Traditional textbooks on MATLAB for engineers have examples which deal with problems the students have yet to encounter. Programming is a create/design level activity on Bloom's taxonomy. Students' perception of the course being difficult stems from having to achieve the create level, while missing all/many prior levels on example problems. This paper assumes that one of the underlying causes for this is that students are introduced to new mathematics and/or physics in examples, which makes it hard for students to focus on the programming aspect. New course material has been prepared which includes mathematical concepts from elementary/middle school to address the issue. Additional examples also deal with problems where students know the mathematics/physics of the problem until the evaluate-level of Blooms taxonomy. The effect of such material on students learning is addressed in this paper. Comparison is made on student success with the new material vs the historic student performance.

Introduction

Teaching programming at freshman level is a challenging task. Typical textbooks used for teaching programming in Mechanical Engineering (ME) contain some applications of the field to emphasize the importance of the field. These textbooks also use equations and algorithms used in the field to achieve a specific programming objective.

Programming is a creative activity. It occupies the highest level (six) in Bloom's taxonomy [1,2]. In typical classes, learning activities from levels 1-5 take place before students are asked to tackle level 6 learning activity. This forms a natural progression in learning. When it comes to programming, this natural progression is broken for the sake of accelerated effort to get the students familiar with the field as well as programming, simultaneously. This approach, while noble in intention, can be compared to trying to run without the ability to walk.

In this paper the conventional material to teaching programming to freshman MEs is replaced with newly developed material which has been designed so that students know and understand every problem assigned before they tackle the task of writing code to solve the problem. The problems are chosen in such a way that the students can solve each problem by hand before they undertake the task of using a programming language to solve it. This study seeks to address the question whether this approach will make a difference in student success, as measured by grade distribution. Success rate is defined by lower DWF rates. Other grades are also discussed as for the impact of this approach. The grade distribution is compared to historic performance and conclusions are made.

Historically programming has been taught using mathematical models(equations) with which students are unfamiliar. The problems provided in such environment were related to the field of ME at which freshman level students had little experience. [3]. A quick look at many example problems [3] shows that the examples are either without an application or with an application that the students have yet to encounter. This characteristic is repeated across many such books for introductory programming. Six levels of cognitive development are shown in Figure 1. Since programming is at the highest level of Bloom's taxonomy, students have difficulty learning programming using mathematical models they do not understand and using physics they do not understand.

Attempts to improve freshman ME education have been made for decades. A widely understood problem in this effort is teaching programming. Programming and design have stood at the forefront of the efforts to improve freshman education in ME. Many ME programs have either started or reinforced the importance of design and programming in the freshman year. One can note that both these tasks form the pinnacle of Bloom's taxonomy of learning [1,2]. It is the steepness of learning, going from level one to level six within a semester on two disparate ideas for a freshman/woman that has proved to be challenging for instructors.

Even a couple of decades ago in 2001, when robotics/experimentation for students was still not robustly available, efforts were made to integrate design, creativity, and programming with Legos and LabVIEW [4]. Enrollment increased significantly because of introducing such courses in the freshman year. Similar efforts were underway in developing experimentation or design-based programming courses to improve student interest and satisfaction [5]. This study introduces students to experimentation and design using programming and found that students' interest and satisfaction increased significantly due to the introduction of such course.

Another study [6] designed their freshman course using MATLAB and Arduino to teach programming. Another study [7] used Visual Basic as the environment to teach programming to freshman students. In this study, students were asked to develop games using the graphical user interface available with Visual Basic.

Most of these courses developed use visual methods so that students can see the functionality of the device that they program. These methods work well for visualization but fail to instill abstract programming skills needed for success in ME programs. ME programs offer courses such as Heat Transfer, Automatic Controls, and design electives require abstract programming to model systems and to provide predictions of the models thus developed.

The proposed material attempts to address both, the problem of lacking understanding of the mathematics/physics of the problem before solving it and to impart abstract programming skills needed for completing most challenging courses in ME.

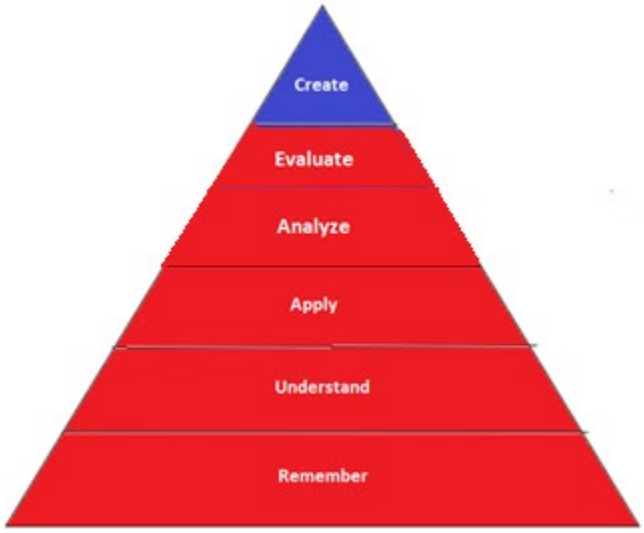

Figure 1: Six levels of cognitive development in Bloom's taxonomy

Material Developed

Material developed for the programming portion of this introductory programming course attempts to address the above-mentioned drawbacks of existing content. The content developed includes a brief introduction, example problems, and exercise problems. The content is split into 12 lab manuals with each lab expected to span two hours. On average, one hour is spent lecturing and for group activities. The second hour is spent quizzing students, and students working on exercises individually.

Example problems and exercise problems developed for typical programming concepts have the following general characteristics:

1. Students can solve them by hand within a few minutes without further explanation from instructor.
2. Problems have real-world applications, so that students understand the relevance.

The following table shows sample content and problems for many concepts covered in the programming class.

Table 1: Sample of problems developed for the new material.

Topic	Sample Example Problem	Sample Exercise Problem
Sequential execution and expressions	Ask user to input enter mass and height, and then calculate BMI (formula provided)	Ask user to enter a length in cm and convert it to inches
If conditions/Boolean operations	Write software to help a movies theatre kiosk. Ask for the person's age and then determine the ticket cost based on age (child/adult/senior citizen)	User enters exam score in percentage, write code to display their letter grade
Compound if conditions	Given IRS flowchart, determine if the user needs to file Income Tax Return based on user inputs	Write software for elementary school kids. User enters shape, and dimensions asked, you calculate and display area of the figure.

While loops	This software you write is for elementary school kids. The kids enter two numbers. Your task is to see if the first number entered is a factor of the second number. When a kid is successful, you need to display a congratulatory message and then tell the kid how many attempts it took to enter a successful pair of numbers (first number factor of second number). If the kid does NOT enter a successful pair, then you display a message saying so and you need to repeat the process.	This software you write is for elementary school kids. The kids enter a number. Your task is to list all factors of the number.
For Loops	Help elementary school kids by writing a program to find least common multiple of two numbers.	Ask an elementary school kid to enter a number. Tell the kid of that number is a prime number.
Arrays and Plotting	Using provided data, plot employment opportunities for MEs as a function of year for the past 20 years	Using the excel sheet provided, plot the global CO_2 levels and temperatures as a function of year for the past 100 years.
Loops and arrays	Given the grades of a class, find the maximum grade	Given the grades of a class, find the average grade
Functions	Given the grades of a class, find the maximum grade	Given the grades of a class, find the average grade

Results

The programming component of the course was setup for a weekly quiz, two mid-term exams, and a final exam during the Spring 2020 semester. Programming assignments using robotics kits were incorporated into one week's schedule but was not implemented due to Covid-19. Students' performance for this semester is compared to historic performance in the figure below. As can be seen from the figure, the performance of students in this course far exceeded historic performance. In 2019-2020 year, due to Covid-19, some grades were listed as P as per MSU policy. These grades are counted towards C, the least grade for the course needed to progress in the program. Other grades during this year included 4 NC (no credit) and 1 incomplete. The NCs and incomplete(I) are added to the list of failed. The total class size in 2019-2020 is 90.

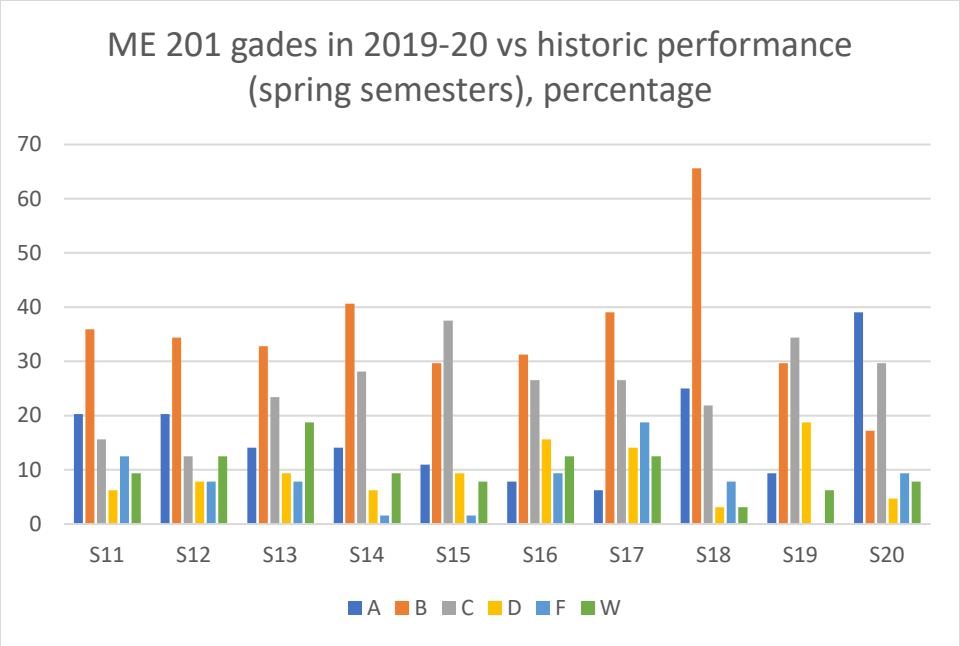

Figure 2: Student performance in ME 201, S20 vs historic

As Figure 2 shows, the grades have shifted up for the year 2019-2020 for which the new material is implemented. There is a decrease in the DWF rates (22%) compared to previous years' average of 27%. This difference is not statistically significant due to small sample size. The number of students achieving A has significantly increased (39% vs 25%), higher than any other year for the past 10 years.

Some of the study is complicated by the fact that some students were unable to access education resources due to Covid-19. Further study is needed to determine the trends due to the implementation of the new material.

Conclusions:

New material has been developed and implemented in the historically difficult freshman programming course (ME 201) at MSU. The new material is designed such that, for all example and exercise problems, the students the mathematics and physics behind the problems at least up to level 5 of Bloom's taxonomy. The new material is implemented in Spring 2020. Results show that student performance at the top grade has improved, beating any other semester in the past 10 years. While DWF rate for S20 is lower than ten-year average, it is not statistically significant. Future study includes continuous monitoring of student performance using the new material.

Future Work

These assignments will be used for the next few years and student performance will be measured and documented to determine if the improvements can be sustained.

References:

[1]. B.S. Bloom, “Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners”, New York: Longmans, 1956., pp 10-24.

[2]. L. W. Anderson, D.R. Krathwohl, “A Taxonomy for learning, teaching, and assessing”, Addison Wesley Longman, 2001, pp 27-37.

[3]. S. Chapman, MATLAB Programming for Engineers, 5th edition, Cengage Learning, 2015.

[4]. E. Wang, "Teaching freshmen design, creativity and programming with LEGOs and Labview," 31st Annual Frontiers in Education Conference. Impact on Engineering and Science Education. Conference Proceedings (Cat. No.01CH37193), Reno, NV, USA, 2001, pp. F3G-11, doi: 10.1109/FIE.2001.963943..

[5]. E.A. DeBartolo, R. Robinson, A Freshman Engineering Curriculum Integrating Design and Experimentation, Vol-35, International Journal of Mechanical Engineering Education, 2007.

[6]. M. Utayna, An Introductory Engineering Course for Freshman Students- From Programming to Implementation, Proceedings of the 2014 ASEE North Central Section Conference, 2014

[7]. M. Naraghi, An Effective Approach for Teaching Computer Programming to Freshman Engineering Students, 2001 Annual ASEE Conference, 2001.