A PROGRAMMING COURSE INCLUDING C# AND MATLAB FOR MECHANICAL ENGINEERING STUDENTS

Yucheng Liu
Department of Mechanical Engineering
University of Louisiana at Lafayette

Abstract

This paper introduces a programming course which provides second year mechanical engineering students the opportunity to develop engineering-oriented software and Graphical User Interface (GUI) using the principles of software engineering. Two object-oriented languages, C# and MATLAB) were taught in this course to make the students familiar with different types of programming languages. Methods used in teaching this course such as topics, class/laboratory schedule, and evaluation instruments are explained. The course objectives were well met according to the student learning outcomes and several Mechanical Engineering B.Sc. program objectives were also supported by this course. This course was taught by the author at University of Louisville and will be offered in University of Louisiana at Lafayette.

Introduction

Computer software and programming techniques have been widely applied for solving engineering problems and therefore become more critical in engineering education. For that reason, courses in programming with a high-level language (such as FORTRAN, Pascal, C, or C++) have long been included in many first and second year engineering curricula. Such courses have traditionally been justified as teaching logical thinking and problem solving, while providing the students with tools that they will use as practicing engineers. However, these goals can be achieved more readily through the use of object-oriented programming languages such as C# and modern mathematical software tools such as MATLAB. Such languages and tools are more powerful and easier to master than a traditional programming language, allowing students to solve engineering problems more easily and efficiently. This paper introduces a programming course, Structured Programming, which teaches students using C# and MATLAB to solve engineering problems. Such course is designed for the second year mechanical engineering students.

As powerful programming tools, C# and MATLAB have been taught and used in engineering courses at different universities [1]. Bernhart et al [2] and Christensen [3] discussed the explicit demand and benefits of adding a course in the Microsoft.NET platform and the C# in mechanical engineering department. Nolan et al [4] developed a program in C# to measure the surface texture using fractal dimension. This program was then revised as a case study and used for a civil engineering course. Conventionally, C# was considered too professional and only taught in computer science and computer engineering department. Recently, other engineering departments have added C# as an educational programming language in their curricula because of its simplicity, safety, and high performances in engineering applications.

Compared to C#, MATLAB is a much more popular software package that has long been used in engineering courses. Kim et al [5] showed advantages of using MATLAB in teaching linear algebra to engineering students. The authors developed web-based engineering numerical software using MATLAB and its web server tool box in order to provide basic solution tools in linear algebra, including eigenvalue problems. It was proved that the developed software is very helpful for applied mathematics education. Wirth and Kovesi [6] explored the use of MATLAB for teaching the fundamental constructs of programming languages to engineering and science students. It was found by the authors that the time taken for students to learn programming can be reduced by up to 50% because of MATLAB's simpler syntax. Vondrich and Thondel [7] illustrated the use of MATLAB in engineering education through practical example and concluded that MATLAB appeared to be the

most successful for teaching compared to other possible software products. Yaz and Azemi [8], and Cao and Wu [9] described the benefits of using MATLAB in electrical engineering courses in enhancing students' understanding of materials and reducing their amount of time spent in performing computational homework assignments. Also, the methodology of integrating MATLAB into mechanical engineering curricula to improve the quality of mechanical engineering education is presented and discussed by Habib [10].

Because of their strengths, a programming course was designed, which aims to teach the second year mechanical engineering students how to use C# and MATLAB in one semester. Meanwhile, the students can obtain critical programming skills and general computational knowledge through this course, which is not well covered in traditional curricula. Compared to similar courses taught in other universities, this course is more efficient and focused: it removes all the materials which are too profound for non-computer science students and only focuses on using both languages to solve real world mechanical engineering problems.

C# and MATLAB

Both C# and MATLAB are object-oriented language. Since 1990's, object-oriented programming (OOP) has become the prevalent technique used for developing software tools and interfaces. OOP is based on the notion that software packages can be viewed as consisting of distinct objects with properties that can be manipulated by methods [11]. Compared to traditional programming techniques, OOP offers unique virtues such as encapsulation and inheritance, which make the software easy to maintenance and make the programming language easy to learn. Because of these advantages, many modern programming languages now support OOP. A paradigm of OOP is shown in Figure 1.

C# is a simple, general-purpose, OOP language developed by Microsoft and supported by the .NET Framework's Common Language Runtime (CLR) [12]. As stated by Microsoft, .NET will replace COM as a software

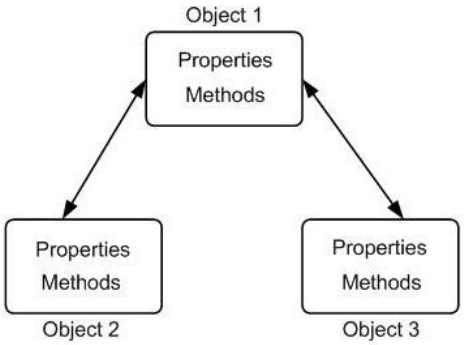

Figure 1. Object-oriented program structure.

component architecture and it provides an easy and efficient way to create window applications. Therefore, it is important for mechanical engineering faculties to teach C# and .NET to the students to prepare them well for this challenging era and allow them to become successful mechanical engineers.

MATLAB is a special-purpose computer program optimized to perform engineering and scientific calculations. As another object-oriented programming language, MATLAB is different from C# because it is an interpreted language. There are two major ways to implement a programming language: interpretation and compilation. C# is a compiled language, whose implementation is typically compliers (translators which generate machine code from source code). However, the implementation of an interpreted language often takes the form of an interpreter, which is a step-by-step executor of source code, where no translation takes place. Because of that reason, MATLAB execute more slowly than C# and is more expensive. Despite its defects, MATLAB offers a lot of advantages: it is easy to use and is supported on many different computer systems; it has an extensive library of more than 1000 predefined functions plus many special-purpose toolboxes; the plots and images generated from MATLAB can be displayed on any graphical output device supported by computer. These advantages make MATLAB very suitable for mechanical engineering applications. We teach both C# and MATLAB in Structured Programming course so that our students can have experiences in using and dealing with two typical languages that represent different implementation schemes.

Approach and Methodology

It is important to understand that this course was designed for students who want to learn computational knowledge and programming techniques, and pursue a career in mechanical engineering. This course was designed to be taken by students at the sophomore level or above and provides them with (1) an introduction to engineering software development using an object-oriented programming language (C#) within a modern develop environment (Microsoft.NET), and (2) instruction in the use of high-level numerical computing environment and programming language (MATLAB) for engineering applications.

This course has a well defined schedule and syllabus so that the instructor would cover the various sections of the class, which was required by the breadth of material and subject matter. Visual and hands on learning techniques were used wherever possible by the instructor of this course. In many regards, the mechanical engineering students who took this course have different background in computational and software knowledge. To help in this regards, instructor was available for one-on-one office time with students. Students were able to use class time and available office hours to discuss on special topics with the instructor. Descriptions of a course syllabus are presented.

Course Syllabus

Structured Programming

Course Learning Objectives

The Mechanical Engineering Department has a set of five primary program objectives and 11 educational outcomes associated with the B.Sc./M.Eng. curriculum. Structured programming supports the program objectives by developing:

- An ability to apply knowledge of mathematics, science, and engineering in the field of mechanical engineering.
- An ability to identify, formulate and solve problems in the field of mechanical engineering.
- A recognition of the need for, and an ability to engage in, life-long learning in the field of mechanical engineering.
- An ability to use the techniques, skills, and modern tools necessary for the practice of mechanical engineering.

Text Book

- J. Liberty, B. MacDonald, Learning C# 2005: Get Started with C# 2.0 and .NET programming, 2nd edition, O'Reilly.
- S. Chapman, MATLAB Programming for Engineers, 3rd edition, Thomson.

Organization

Three 50 minute sessions per week devoted to lecture/discussion and problem solving. The difficulty of this course is such that a minimum weekly commitment of 8-10 outside study hours will be required.

Grading

Homework	20%
Quiz	10%
Midterm Exam	10%
Final Exam	10%
C# Project	20%
MATLAB Project	20%
Professionalism	10%
Total	100%

Final grades will be calculated according to an absolute scale:

≤ 59 F		
60-62 D-	63-67 = D	68-69 D+
70-72 C-	73-77 = C	78-79 C+
80-82 B-	83-87 = B	88-89 B+
90-92 A-	93-97 = A	98-100 A+

Topic/Activity

Week 1: Introduction (computer programming languages, C# and .NET software development platform); C# fundamentals (types, constants).

Week 2: C# fundamentals (variables, expressions); variable types and arrays.
Week 3: Flow control (Boolean logic, branching, looping).
Week 4: Functions, OOP fundamentals (classes, objects).
Week 5: OOP fundamentals (properties, fields, methods, inheritance).
Week 6: OOP fundamentals (polymorphism, events); Window application (window forms collection).
Week 7: Window applications (input and output).
Week 8: Window applications (graphics with GDI+).
Week 9: Window applications (C# .NET project).
Week 10: Introduction to MATLAB.
Week 11: Arrays.
Week 12: Modular program (functions and script files).
Week 13: Input and output; Control structures (decisions, loops, and vectorization).
Week 14: Data types (cell arrays and structures); MATLAB project.
Week 15: MATLAB graphical user interface (GUI).

Design Projects

Two exams were given in this course, one for C# and one for MATLAB. However, both exams only included problem solving and answering, computer operations and programming were not covered in the exams. In compensation, two course projects were assigned to each student, which were major components in this class. The two projects mirrored typical design projects found in industry. It is believed that students will not be able to develop their engineering software development techniques and a complete understanding of computational application in mechanical engineering without two projects that focus on C# and MATLAB, respectively. Two examples of the course design projects are presented below, where the C# project focuses on engineering software development and the MATLAB project focuses on mechanical engineering problem formulation and solution.

Project 1 – C#

Assignment:

Using C# to design an online checking account. After log into this account, user can perform following operations: add deposit, withdraw cash, transfer balance, view current balance, and view the history statement. The final program should be able to handle all kinds of exceptions; an account number and password have to be set for this account.

Deliverables:

1. Project report – briefly describe the project problem and explain how the program is designed; list all the classes and forms used in the C# program. (15 pts).
2. Help documents – show users how to use the online checking account; use screen shots of different interfaces to explain the functions of different buttons and controls. (15 pts).
3. An oral presentation needs to be given to show that the final program meets all the design requirements and is safe enough to handle wrong operations. (70 pts).

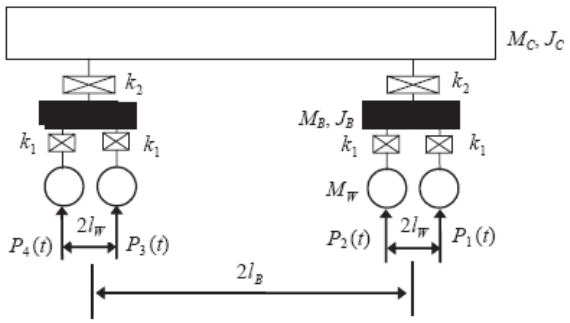

Figure 2. A vehicle-suspension-wheel system.

Project 2 – MATLAB

Assignments:

Fig. 2 shows a vehicle system that will be used for rigid body analysis. Neglecting the damping effects, the governing equation of such system is

$$[M_V]\ddot{X} + [K_V]X = 0 \qquad (1)$$

In Eqn. (1), X is the displacement vector of the vehicle system; M_V is the mass matrix, which is defined by:

$$[\mathbf{M}_V] = \text{diag}(M_C, J_C, M_B, J_B, M_B, J_B, M_W, M_W, M_W, M_W)$$

where K_V is the stiffness matrix defined as:

$$[\mathbf{K}_V] = \begin{bmatrix} 2k_2 & 0 & -k_2 & 0 & -k_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2k_2 l_B^2 & -k_2 l_B & 0 & k_2 l_B & 0 & 0 & 0 & 0 & 0 \\ -k_2 & -k_2 l_B & k_2 + 2k_1 & 0 & 0 & 0 & -k_1 & -k_1 & 0 & 0 \\ 0 & 0 & 0 & 2k_1 l_W^2 & 0 & 0 & -k_1 l_W & k_1 l_W & 0 & 0 \\ -k_2 & k_2 l_B & 0 & 0 & k_2 + 2k_1 & 0 & 0 & 0 & -k_1 & -k_1 \\ 0 & 0 & 0 & 0 & 0 & 2k_1 l_W^2 & 0 & 0 & -k_1 l_W & k_1 l_W \\ 0 & 0 & -k_1 & -k_1 l_W & 0 & 0 & k_1 & 0 & 0 & 0 \\ 0 & 0 & -k_1 & k_1 l_W & 0 & 0 & 0 & k_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_1 & -k_1 l_W & 0 & 0 & k_1 & 0 \\ 0 & 0 & 0 & 0 & -k_1 & k_1 l_W & 0 & 0 & 0 & k_1 \end{bmatrix}$$

Select proper values (with units) for all the parameters and construct matrices [Mv] and [Kv]; based on the matrices, find natural frequencies for this vehicle system (the number of natural frequencies should equal to the rank of the matrices); list natural frequencies in a table from low to high; plot these frequencies in a 2-D plot (vs. mode index); each natural frequency is accompanied with an eigenvector, which is an animation mode of the vehicle system, resort these eigenvectors according to the resorted natural frequencies; create a function to normalize an eigenvector based on a input mode index number and plot at least one normalized eigenvector with a specified mode; save all functions into M-files.

Deliverables:

1. Project report (briefly describe the project problem, include the figure of vehicle assembly, list all selected values and write out the matrices, list all the M-files and functions generated for this project, attach all the natural frequencies and eigenvectors found using MATLAB). (40 pts).
2. A table to list the sorted natural frequencies. (15 pts).
3. A plot for the natural frequencies. (15 pts).
4. Solve the eigen problem to find all normalized eigenvectors. (15 pts).
5. A plot for a normalized eigenvector with a specified mode. (15 pts).

Student Assessment

This course was taught by the author to the mechanical engineering sophomore at University of Louisville in Fall of 2007. As an entry-level programming course, Structured Programming effectively developed the students the programming skills and capabilities of solving complex engineering problems using popular engineering and mathematics software package. 24 students enrolled in this class and by the end of that semester, favorable outcomes were received from the student feedback as well as from the course evaluation. From the evaluation results

(Table 1), it was reflected that this course was well designed that the students had learned fundamental knowledge of computer language and programming, developed basic skills to develop engineering-oriented software and graphic user interfaces using the principles of software engineering, and matured in the use of a suitable programming language such as MATLAB.

Student feedback was also exciting, from which it showed that the students had benefited from this class. Selected comments from the students include:

"Amount of knowledge gained was tremendous. This class will be very helpful in my career."

"This is one of the toughest classes, but Dr. Liu is extremely effective at communicating difficult concepts in the classroom."

"The class was well organized as always. Dr. Liu presents a caring, considerate and open attitude toward the students."

"I have learned a lot from this class and the instructor has earned my respect!"

With the success of offering Structured Programming in University of Louisville, the same class is planned be implemented into Mechanical Engineering curriculum at University of Louisiana at Lafayette in the near future.

Results and Conclusions

Because of the increasing significance of software knowledge and programming techniques in mechanical engineering, a Structured Programming course was designed for the second year mechanical engineering students. The course syllabus which includes the learning objectives required course textbooks, grading breakdown of the course, and covered topics are presented in this paper.

This course includes hands on and visual lessons to better help students to learn the two different programming languages, C# and MATLAB. Different evaluation tools such as homework, quiz, exams, and design projects are used to help students to grasp the concepts being taught. It is believed that the best way to solidify the main ideas of this course is the design projects which develop students' ability to use C# to develop engineering software and use MATLAB to formulate and solve a mechanical engineering problem. Oral and written communication skills are also required in both exams. A project overview and description can also be found in text.

This cause has been taught by the author in Mechanical Engineering Department at University of Louisville. The course objectives were well met according to the student learning outcomes and an assessment from students also ascertained their interests in this course. Such course is thereby ready to be added into current curricula of Mechanical Engineering Department at University of Louisiana at Lafayette after small modifications.

Table 1. Selected course evaluation reports.

Query	Strongly agree	Agree	Undecided	Disagree	Strongly disagree
I learned a lot in this course	20 (84%)	4 (16%)	0	0	0
This course challenged me to think	16 (67%)	8 (33%)	0	0	0
Summary	Excellent	Good	Average	Fair	Poor
Overall, I would rate this course as	20 (84%)	4 (16%)	0	0	0
Overall, I would rate the effective-ness of this instructor as	20 (84%)	4 (16%)	0	0	0

References

1. A. Pyster, R. Turner, D. Henry, K. Lasfer, L. Bernstein, K. Baldwin, "Current state of software engineering masters degree programs", *Proceedings of the 21st Conference on Software Engineering Education and Training*, Charleston, SC, USA, April 14-17, 2008, pp. 103-109.

2. M. Bernhart, T. Grechenig, J. Hetzl, W. Zuser, "Dimensions of software engineering course design", *Proceedings of the 28th International Conference on Software Engineering*, Shanghai, China, May 20-28, 2006, pp. 667-672.

3. J. E. Christensen, K. Ribu, "A method for learning software tools in engineering education", *Proceedings of the 9th International Conference on Engineering Education*, San Juan, PR, July 23-28, 2006.

4. J. M. Nolan, F. N. Kremm, D. Budny, "Advances in on demand programming solutions developed in C# by non computer science majors", *Proceedings of the 9th International Conference on Engineering Education*, San Juan, PR, July 23-28, 2006.

5. A. S. Kim, C. Park, S.-H. Park, "Development of web-based engineering numerical software (WENS) using MATLAB: applications to linear algebra", *Computer Applications in Engineering Education*, 11(2), 2003, pp. 67-74.

6. M. A. Wirth, P. Kovesi, "MATLAB as an introductory programming language", *Computer Applications in Engineering Education*, 14(1), 2006, pp. 20-30.

7. J. Vondrich, E. Thondel, "The application of MATLAB in engineering education", *Proceedings of 2003 International Conference on Simulation and Multimedia in Engineering Education*, Orlando, FL, USA, January 19-23, 2003.

8. E. E. Yaz, A. Azemi, "Utilizing MATLAB in two graduate electrical engineering courses", *Proceedings of the Frontiers in Education Conference on 1995*, 1(1-4), 1995, pp. 2c6.1-2c6.4.

9. Y.-J. Cao, Q.-H. Wu, "Teaching genetic algorithm using MATLAB", *International Journal of Electrical Engineering Education*, 36, pp. 139-153.

10. M. S. Habib, "Enhancing mechanical engineering deep learning approach by integrating MATLAB/Simulink", *International Journal of Engineering Education*, 21(5), 2005, pp. 906-914.

11. J. Mackerle, "Object-oriented programming in FEM and BEM: a bibliography (1990-2003)", *Advances in Engineering Software*, 35, 2004, pp. 325-336.

12. J. Bishop, N. Horspool, "C# concisely", Pearson Education Ltd. 2004.

Biographical Information

Dr. Yucheng Liu is an Assistant Professor in the Department of Mechanical Engineering, University of Louisiana. He received his Ph.D. in mechanical engineering from the University of Louisville in 2005 and has research and teaching experience in the areas of programming and development of interactive software, mechanical and machine design, computer modeling and simulation, structural mechanics, finite element analysis, crashworthiness analysis, vehicle system design and analysis, applied mathematics, etc. To date, he has authored more than 70 publications in his research areas and has been the PI or Co-PI on over $1.5 millions of research funds. He is an editorial board member of an international journal and also has served as reviewer for 18 refereed journals and 5 international conferences. Dr. Liu is a professional engineer registered in Ohio, and holds memberships in ASME, SAE and ASEE.