How to teach first-year engineering students to learn computing and programming effectively?

Hong Zhang and Charles Lemckert
Griffith School of Engineering, Griffith University
Corresponding Author Email: hong.zhang@griffith.edu.au

Structured Abstract

CONTEXT

The computer has become one of the most widely used tools in the modern world, and has had a profound impact on how engineering and science are practised today. Engineering students are required to familiarise themselves with both the hardware and software environments in a modern computing system, and how to apply the fundamentals of computing to solve various engineering problems. At Griffith University, a computing course was offered to more than 350 first-year engineering students from Civil, Mechanical, Electrical, Mechatronics and Biomedical Engineering since 2009. For the last 5 years, most students often find computing and programming difficult because they have limited mathematical and physics backgrounds to understand the situations presented to them. The most challenging task for students was to formulate engineering problems and then find an algorithm to solve them. On the other hand, the most challenging task for lecturer was to motivate the students to learn and to provide them with an effective learning environment.

PURPOSE OR GOAL

The goal of this work was to develop an effective teaching methodology to motivate and assist first-year students, from various engineering disciplines, to actively and successfully learn computing and programing. Systematic course evaluations were completed to know what works, what doesn't work and the most importantly how to improve the course. Eventually, the course aims to help the students to apply the computing techniques to solve practical engineering problems and get themselves ready for the advanced courses and future career.

APPROACH

A consistent problem solving approach was and is applied in this continuing course. Numerous engineering examples are embedded in the programming algorithm introduction. Weekly lectures and "hands-on" computer labs were designed for students to learn and practice. A problem solving assignment project was designed to cover all engineering fields. In order to engage students to learn the course more effectively, a flexible teaching assistance plan was also arranged based on students' engineering background knowledge and computing experience.

A comprehensive evaluation was carried out to assess the effectiveness of the students learning. During the semester, formal and informal "Classroom Control" feedback was collected and responded. To provide the broadest picture of the course possible, an on-line survey was completed for the course, the lecturer and all tutors at the end of the course. The course survey data was analysed systematically for all aspects by using the statistic distribution method. Finally, students' performance in the course was also used as another indicator to show the effectiveness of the teaching and learning.

ACTUAL OR ANTICIPATED OUTCOMES

While this project is still ongoing, preliminary analysis has shown the teaching methods are effective as the course survey score has been improved into the top band for student satisfcation. Student performance is also improving when compared to previous years. Therefore, the problem solving method and flexible teaching assistance were effective for this course and the approaches employed should be applicable elsewhere.

CONCLUSIONS/RECOMMENDATIONS/SUMMARY

In the study, effective teaching methods including problem solving and flexible learning assistance are introduced for the teaching of computing and programming course to first-year engineering students. The examples from different engineering disciplines aided the students understanding of programming fundamentals and how to apply them to real world engineering applications. The flexible teaching assistance provides students sufficient support in learning the challenging course.

KEYWORDS

Computing and programming, problem solving, first-year engineering, flexible teaching assistance

Introduction

The computer has become one of the most widely used tools in the modern world, and has had a profound impact on how engineering and science are practised today. Computers and programming languages provide a powerful and efficient platform for engineering design calculation, predictions, data analysis and engineering management in all engineering fields such as civil, electrical, mechanical, environmental, biomedical, and chemical engineering. MATLAB™ is an advanced and sophisticated programming language, and mathematical computation tool, which is quickly becoming a requirement for many engineering positions. Therefore, university engineering students are required to familiarise themselves with both the hardware and software environments in a modern computing system, and know how to apply the fundamentals of computing to solve various engineering problems.

Since 2009, at Griffith University, a computing course was offered to all first-year engineering students from Civil, Mechanical, Electrical, Mechatronics and Biomedical Engineering and some science degrees as well. The course assumes the student has a fundamental algebra and trigonometric concepts, and some basic physics knowledge. However, the 1st year engineering students have been admitted to the university through various pathways. Their mathematics and physics levels vary as well. For the last 5 years, most students often find computing and programming difficult because they have limited mathematical and physics backgrounds making it challenging to comprehend the problem. This course has been rated the most difficult course in their first semester in the university. The students have found the most challenging task in this course is to formulate solutions to given engineering problems and then to find an algorithm to solve them. On the other hand, the most challenging task for lecturers was/is to motivate the students to learn and provide them with an effective learning environment.

Therefore, the goal of this study was to develop an effective teaching methodology to motivate and assist first-year students, from various engineering disciplines, to actively and successfully learn computing and programing. Systematic course comparisons and evaluation were completed to understand what works, what doesn't work, and most importantly; how to improve. Eventually, it aims to help the students to apply the computing techniques to solve practical engineering problems and get themselves ready for their advanced courses and future career.

Course development

The curriculum development for 1st year computing and programming course is extremely challenging (Haubold, 2007). Firstly, the students lack an understanding of fundamental concepts in physics and mathematics, due to the national and international diversity of standards. Secondly, the students' interests in the engineering disciplines are different. It includes civil, mechatronics, mechanical, environmental, electronics, microelectronics and energy engineering. The curriculum design principles have been considered to address the challenges faced by educators in developing graduate attributes within the engineering.

Figure 1 shows a typical learning cycle which links graduate outcomes to activities and assessment tasks (Cameron and Lewin, 2009). It is an important educational design guideline. This approach implements important aspects of Biggs' constructive alignment concepts that helps formalize the learning and teaching designs (Biggs, 2007).

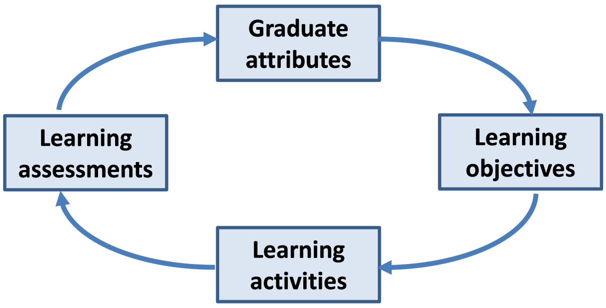

Figure 1: The learning cycle around building graduate attributes (Cameron and Lewin, 2009).

For this computing course, the learning cycle will start from the "Graduate attributes" and the "Learning objectives". For the Bachelor of Engineering degree, the computing foundation course is required by other advanced courses in the degree and also for the students' future engineering career development, where programming and computing become an integral part of the analysis, simulation and design methodology. A course development team from different engineering disciplines and the university's Institute for High Education was formed. It was/is clear that the course aims to develop students' fundamental skills in problem conceptualization, formulation, and solution in one of the most powerful and versatile programming environments - MATLAB. After successfully completing this course, the students should be able to: (i) understand the fundamentals and theory of computer programming; (ii) apply the programming theory to the MATLAB software environment; (iii) Design the algorithms, and use MATLAB to solve engineering problems.

Within the course, the students learn the programming fundamentals and computing theory in weekly lectures. Practical reinforcement of concepts presented in lectures were undertaken in 3 hour, weekly, laboratory sessions. Systematic engineering examples have been used and delivered to students through traditional lectures, small group labs and individual research assignment. It helps students to explore various engineering disciplines, which suit the 1st year engineering students and improve their computing skills. In order to create and maintain a stimulating, engineering disciplinary environment, a consistent problem solving method has been adopted in all lecture examples, laboratory activities and assignment projects based on first-year engineering students' mathematics and engineering knowledge levels; which lets engineering students understand the importance of computing skills in their future endeavours. Table 1 summarises the engineering disciplines' relevance to computing and programming.

The next step was to develop effective "Learning assessments". In this course, four different assessments were used. (i) Students were/will be assessed on the effort at each laboratory session. This was designed to encourage students to attend laboratory sessions, where students can develop and fortify their understanding and application of programming concepts. (ii) In-class quizzes were used to permit students to gain continuous feedback on their learning progress and assess their theoretical understanding of the conceptual material delivered in lectures and developed through personal study and laboratory experience. (iii) An assignment project consisted of real engineering/science problems, for example the popularly showed *Top Gear* Beetle car project, the Dam operation project, the traffic safety project, the manufacturing project, the resistor selection project, that students are required to solve them using the engineering knowledge and programming skills learnt from the course. A written scientific report was required to submit with the programming codes to assess the students technical and communications skills. (iv) A final examination was used to examine students' overall knowledge and skills acquired during the course in an examination environment under pressure and time limited.

Table 1: Engineering disciplines and relevance in the computing course.

	Basics				Algorithm & programming				Applications			
	Graphic user interface	Vectors, matrices	Elementary functions	Plotting, figures	Sequential structure	Selection structure	Repetition structure	User-defined functions	Statistics	Data management	Numerical methods	Image
Civil Engineering	x	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	
Electrical Engineering	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	x
Electronics Engineering	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	x
Mechanical Engineering	x	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	
Biomedical Engineering	xx	xx	xx	xx	x	x	x	xx	xx	xx	x	xx
Environmental Engineering	x	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	

x - less relevant xx - more relevant

Finally, the learning cycle was returned to “Graduation attributes” and the course’s “Learning objectives”. Through the students’ performance, the course survey, teaching survey and peer review, the course was/is updated and revised according to the above evaluation outcomes.

Course delivery

Engineering Problem solving methods

A combined approach was employed in this course. The traditional lecture in a large lecture theatre was/is used to introduce all the computing and programing, as the class size is around 380 students. For this size of class, it is essential to know who the students are, their age range, the gender mix, and most importantly, their physics and mathematics knowledge standards and experience. In all lectures, a consistent problem solving approach was/is important throughout all engineering and science disciplines. Numerous engineering examples are embedded in the programming algorithm introduction in weekly lectures. It is evident that problem formulation is always the most challenging step for most of the students. A diagram or picture helps students to understand the problem, and if the students have a clear understanding of the physics, they are more likely to be able to solve it. For effective learning, the quality of understanding of theory and concepts by the students is more important than the quantity of information presented to them. Therefore, the appropriate level and discipline of the examples have been selected.

Students learn to do well only if they practice doing what they have learnt. Therefore, weekly “hands-on” computer labs were designed for students to practice and apply the theory to new problems. These labs give them the opportunities to think critically, analyse information, communicate scientific ideas, make logical arguments, and acquire other desirable skills. They are encouraged to practice over and over in many contexts.

Effective learning also requires that students make connections of new problems to lecture examples and restructure their thinking radically. A problem solving assignment project was designed to cover all engineering fields. The students have also been assessed through the engineering problems they have solved and presented in a report, which is designed based on the relevant engineering graduate attribute competencies and course objectives.

Teaching assistance team

In order to engage students to learn the course more effectively, a flexible teaching assistance plan was also arranged based on students' engineering background knowledge and computing experience. A good teaching team creates space for learning. With over 380 students approximately 20 lab sessions were/are needed as each lab has a capacity of 20 students. A lab tutor team consisting of 6 members are formed under the supervision of the course convenor. All tutors have extensive computing skill experience. Most of them are engineering PhD students who apply computing skills in their research projects and a couple are engineering students in their senior years, who learned the course and achieved an outstanding performance in their first year study, and who are able to share their learning experience with these freshmen. However, for a consistent and quality delivery of the lab activities to the students, the lecturer also provided weekly training to the tutors to strengthen their understanding of the theory, the concepts, and the problem formulation techniques.

At the university, another voluntary learning assistance, Peer Assisted Study Sessions (PASS), were available to help students in the traditionally more challenging courses, such as the first year computing and programming course. Students who have previously achieved excellent results in the course facilitate the sessions. The lecturer guided them to prepare the tutoring material and exercise problems. This learning group gives the students a more in-depth understanding of the course content; develop closer relationships with other students, and to learn as a team – thus strengthening their overall learning experience and opportunities/styles.

Outcomes

Currently, there is no standard measurement for evaluating learning effectiveness. In this study, both the performance based assessments of learning and the perception based assessments of learning (Moody and Sindre, 2003) were/are applied to show the effectiveness of the learning.

Performance Based Assessments of Learning

It is difficult to make sensible comparison between exam scores year to year, because there are many potential confounding variables. In the study, the students' performance in the computing course was compared with their performance of another 2 concurrent courses, *Mathematics 1A* and *Engineering Materials*. They are both 1st year 1st semester courses and are technically relevant to the computing course. In the course development, the physics and mathematics background have been taken into careful consideration. The students' performance in the courses is shown in Figure 2. The students performed the best in *Mathematics 1A*, with over 45 students whose marks are greater than 90%. *Engineering Materials* had no student who received a mark of over 90%. However, the students' performance was inter-compared between the courses, which was shown in Figure 3. It is clear that the performance in the computing course is highly correlated with both the mathematics and the engineering material courses. The correlation coefficient is 0.78 for both. This supported that the computing course is highly related to the mathematics fundamentals and engineering knowledge.

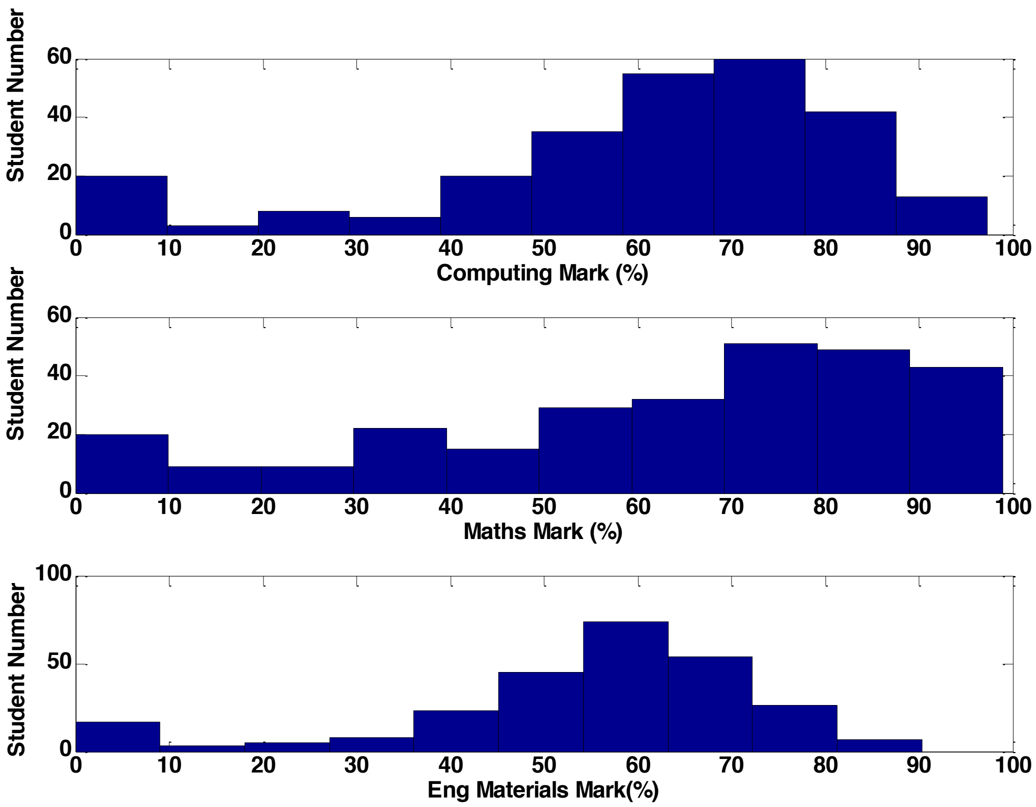

Figure 2: The mark distributions for 3 courses.

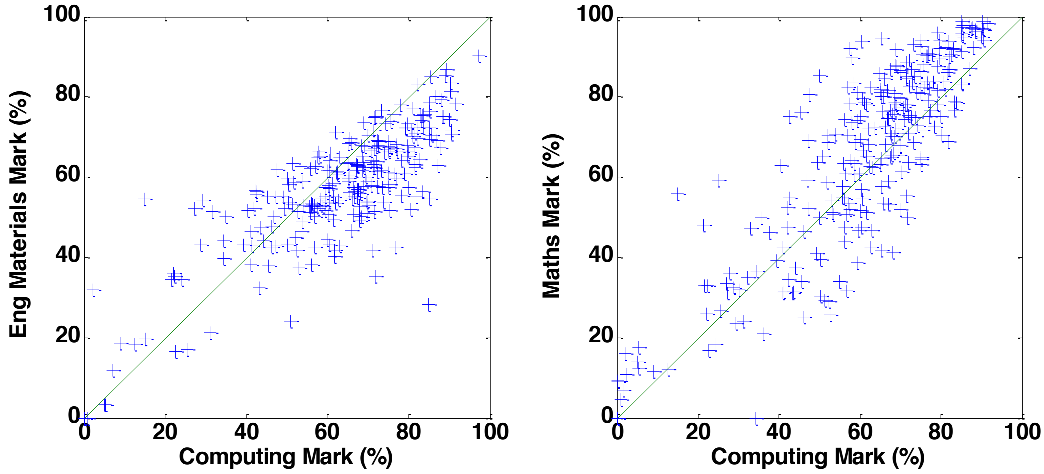

Figure 3: Comparison between the computing course and (a) Mathematics 1A. (b) Engineering Materials

Perception Based Assessments of Learning

A comprehensive course evaluation, through an on-line questionnaire, was carried out to assess the perceived effectiveness of the students' learning and their overall experiences. The outcomes of the survey for 2014 are presented in Figure 4.

The course survey data was analysed systematically for all aspects by using the statistic distribution method. The students' responses for all questions including course organisation, assessment, engagement, teaching style and overall quality, were rated in the best quartile band. The rate distribution of each questions were also better than the quartile band comparison with similar courses within the university. Finally, students' evaluation in the course indicated they have felt they had learned effectively.

Quantitative Summary:

	Question	#		Score	Quartile Band Comparison	%+ve %-ve	Median	Std dev	Mean	Comparative mean 25%	Comparative mean 50%	Comparative mean 75%	Quartile Band Rank
Q1	This course was well-organised.	90	SA	45.2%	33.5%	+90.9	4	0.66	**4.4**	3.8	4.1	4.3	4
		91	A	45.7%	45.9%	-0.5							
		17	N	8.5%	13.6%								
		1	D	0.5%	4.9%								
		0	SD	0%	2%								
Q2	The assessment was clear and fair.	87	SA	43.7%	32.3%	+85.9	4	0.79	**4.3**	3.7	4.0	4.2	4
		84	A	42.2%	44.7%	-2							
		24	N	12.1%	14.4%								
		2	D	1%	6.1%								
		2	SD	1%	2.4%								
Q3	I received helpful feedback on my assessment work.	69	SA	34.7%	23.1%	+70.4	4	0.97	**4.0**	3.4	3.7	3.9	4
		71	A	35.7%	35.4%	-7.5							
		44	N	22.1%	26.7%								
		12	D	6%	10.7%								
		3	SD	1.5%	4%								
Q4	This course engaged me in learning.	62	SA	31.3%	26.4%	+72.7	4	0.86	**4.0**	3.4	3.7	4.0	4
		82	A	41.4%	38.9%	-5.1							
		44	N	22.2%	21.7%								
		10	D	5.1%	9%								
		0	SD	0%	4%								
Q5	The teaching (lecturers, tutors, online etc) on this course was effective in helping me to learn.	79	SA	39.9%	32%	+81.3	4	0.86	**4.2**	3.6	3.9	4.2	4
		82	A	41.4%	38.9%	-5							
		27	N	13.6%	18%								
		9	D	4.5%	7.5%								
		1	SD	0.5%	3.5%								
Q6	Overall I am satisfied with the quality of this course.	70	SA	35.4%	27.2%	+87.4	4	0.72	**4.2**	3.6	3.9	4.2	4
		103	A	52%	44.5%	-2.5							
		20	N	10.1%	18.2%								
		5	D	2.5%	6.9%								
		0	SD	0%	3.2%								

Figure 4: Summary report of student experience of course.

Conclusions

In the study, effective teaching methods, including problem solving and flexible learning assistance, were introduced for the teaching of a first-year engineering computing and programming course . Examples from different engineering disciplines aided the students understanding of programming fundamentals and how to apply them to real world engineering applications. The team teaching approach provided students sufficient support in learning the challenging course. Both the performance based assessments of learning and the perception-based assessments of learning show the effectiveness of the learning of the computing course.

References

Biggs, J. (2007). Teaching for quality learning at university, Society for Research in Higher Education (3rd edition). UK: Open University Press. ISBN 0335221262.

Cameron, I.T. and Lewin, D.R. (2009). Curricular and pedagogical challenges for enhanced graduate attributes in CAPE, Computers & Chemical Engineering, Volume 33, Issue 10, 14, Pages 1781-1792.

Haubold, A. (2007). Matlab for First-year College Engineers, The 37th ASEE/IEEE Frontiers in Education Conference, October 10 – 13, 2007, Milwaukee, WI, USA, T1A-1 – T1A-6.

Teodoro, V.D. and Neves, R.G. (2010). Mathematical modelling in science and mathematics education, Computer Physics Communications, In Press, Corrected Proof, Available online 9 June 2010.

Tiernan, P. (2010). Enhancing the learning experience of undergraduate technology students with LabVIEW™ software, Computers & Education, Volume 55, Issue 4, Pages 1579-1588.Moody, D. and Sindre, G. (2003) Evaluating the effectiveness of learning interventions: an information systems case study. In Proceedings of the Eleventh European Conference on Information Systems, 1311-1326, Naples, Italy.

Copyright statement