CHEMICAL ENGINEERING TRANSACTIONS

VOL. 56, 2017

A publication of AIDIC

The Italian Association of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Jiří Jaromír Klemeš, Peng Yen Liew, Wai Shin Ho, Jeng Shiun Lim

ISBN 978-88-95608-47-1; **ISSN** 2283-9216

DOI: 10.3303/CET1756218

How to Motivate Chemical Engineering Undergraduates to Learn Programming?

Nor Azlinda Azmi, Khairiyah Mohd Yusof*, Fatin Aliah Phang

Centre for Engineering Education, Universiti Teknologi Malaysia, 81310, Johor Bahru, Johor, Malaysia
khairiyah@utm.my

Claims such as learning programming are difficult, irrelevant and boring are always associated with the students' motivation to learn. Students were not motivated to fully engage with their learning in programming courses. This has become an unpleasant scenario in many engineering programmes including the chemical engineering programmes. So, how can chemical engineering students be motivated to learn programming? A learner-centred environment is believed to help students regulate both their learning and motivation. Learner-centred learning environment focuses on aspects of knowledge, skills, attitudes, and beliefs that learners bring to the learning process including their motivation to learn. Hence, this paper aims to present a study on the influence of learner-centred environment on chemical engineering undergraduates' motivation towards learning programming. Analysis on observations and interviews was conducted and the findings were discussed. The discovery shows that, instilling the learner-centred elements in programming course for engineering students is essential to create an effective and motivating learning environment.

1. Introduction

Modern engineering practice brings rapid developments of large-scale and complex engineering system designs. The developments have changed a common scenario, where computing tasks were usually assigned to computer engineers which now are also assigned to non-computer engineers including chemical engineers. This is mainly because of computing tasks these days have as well become closely integrated into the daily activities of non-computer engineers (Beers, 2006). In addition, due to the developments in engineering theories and practice, engineering system designs are frequently analysed and modified (Lee and Arora, 1991). Analysing and modification tasks would require computational applications in engineering. To chemical engineers, computational applications are important to augment computing elements to the fundamentals such as transport phenomena, thermodynamics, and chemical reaction engineering (Beers, 2006).

A cognitive aspect of computational applications in engineering is computational thinking. Computational thinking, as defined by computer scientists, is an approach to solve problems, designing systems, and understanding human behaviour based on the fundamentals of computing (Wing, 2008). Computational thinking is shared with engineering thinking in general where engineers can design and modify complex engineering systems. By having deeper computational thinking, engineers are able to design, analyse and modify their systems rapidly.

One way to promote computational thinking among engineering students is to introduce programming as part of the curriculum (ABET, 2014). One of the criteria for accrediting engineering programs emphasised by ABET (ABET, 2014) is to provide the students with the ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. It is important to equip future engineers with computer literacy and programming (Engineering Accreditation Commission, 2014) which considered as compulsory in any engineering curriculum. In the first-year of undergraduate studies in chemical engineering, the students will have to go through an introductory programming course. The content of the course includes the basics of programming such as the use of computer and programs, mathematical concepts, and algorithm.

Learning to program however, is difficult. The challenges to learn programming initiate negative perceptions on it among students. Programming concepts such as algorithm and flowchart, data structures and loops are crucial in programming. These require a high level of abstraction, logical thinking and problem-solving abilities

Please cite this article as: Azmi N.A., Yusof K.M., Phang F.A., 2017, How to motivate chemical engineering undergraduates to learn programming?, Chemical Engineering Transactions, 56, 1303-1308 DOI:10.3303/CET1756218

(Gomes and Mendes, 2007). The students also always claim that the teaching approaches in programming courses are boring and dull (Bowen, 2004). These negative perceptions are observed can affect the students' motivation to learn to program.

Students' motivation is an aspect that should not be overlooked in designing a course. This study holds to a viewpoint that, an effective learning environment is essential to motivate students to learn and engage them in their learning. One approach to motivate the students is to create an effective learning environment. An effective learning environment as defined by Bransford et al. (Bransford et al., 2000) in their established How People Learn (HPL) framework, is a learning environment that is conducive to learning consists of knowledge, learner, assessment and community-centred elements. Knowledge-cantered is defined to help students understand and learn knowledge that is organised with interconnections to the fundamentals of the discipline, in the context of the application and practice in the field. Learner-centred means students' background, pre-conceptions, existing knowledge and skills are considered to create meaning and motivate learners. Assessment-centred means that learners are able to receive feedback through formative assessment and from peers to help them determine their current performance and improve it. Community-centred means that the learning environment promotes a community among students and faculty members, developing a sense of belonging and bond among them, thus creates an environment that is safe to participate, ask questions, make mistakes and explore new things.

A learner-centred environment is summarised as a learning environment that considers the students' background, pre-conceptions, existing knowledge and skills. It also includes the teaching practices that match what the students' beliefs, understandings and cultural practices (Bransford et al., 2000). Among the teaching practices that can be employed by the instructors are such as discover what students think about the problems given, discuss the students' misconception rigorously and give the students' situations which can provide opportunities to realign their ideas that lead to the solutions. The idea of discovering what the students think fits the concept of "diagnostic teaching" which requires the instructors to conduct a research on the process of teaching and learning. The information about the process can be acquired through observation, conversations, or activities' reflections. It is to get the students to explain and develop their knowledge through their own predictions and the reasons of the predictions. About the misconception, after rigorously the discussing the students' misconceptions, the instructors should select tasks that can help the students to overcome the misconceptions by giving ideas on the reasons and methods to change the misconceptions. A learner-centred instructor is also sensitive to the cultural and language practices of the students. This is to provide the students a basis for further learning and coordinate every day and scientific discourse to assist their understanding. It is about how the instructor connects everyday practices and talk to what the students are learning. This matches with the teaching approaches that provide opportunities for students to realign their ideas that lead to solutions. Learner-centred environments have instructors who are aware that the students are constructing their own meanings while learning, starting with the students' beliefs, understandings, and the cultural practices that the students bring to the learning.

Focusing on the learner-centred element, this paper aims to present how a learner-centred learning environment plays a significant influence on chemical engineering undergraduates' motivation towards learning to program. The research questions that are sought to be answered are:

1. To what extent the programming course matches the criteria of learner-centred learning environment to motivate engineering undergraduates to learn to program?
2. How do engineering undergraduates develop their motivation to learn to program within a learner-centred learning environment?
3. What are the recommendations based on the learner-centred element to improve the learning environment and motivate their students to learn to program?

2. Methodology

To answer the research questions, this study inclines towards an exploratory study and thus qualitative methods which are observation and interview were employed. The study was conducted in an introductory programming course in a chemical engineering programme in a university in Malaysia. The course matches the sampling criteria: a programming course in an engineering faculty. The course outline also shows the need of the course to be included in the curriculum of a chemical engineering undergraduate study. The course was conducted by one one-hour and one two-hour lecture per week, in the computer laboratory. 86 students enrolled for the course, consisting of first-year undergraduates. The students were randomly divided into three different sections by the course's coordinator where each section was assigned to three different instructors. Table 1 describes the three different sections and their number of students.

Table 1: Description of sections and number of students

Section	Number of students
1	27
2	31
3	28

The data collection methods consist of classroom observations and interviews. The observations were conducted for a whole semester of 14-weeks. For each week, the observations were done using observational field notes. The observations were conceptually driven, based on the learner-centred element of HPL framework as explained in section 1.

At the end of the semester, ten students were selected as the interview respondents to get their perception of the learning environment and information on their motivation to learn to program. The reason to conduct the interview at the end of the semester is to ensure that the interviewees are familiar with all the activities (i.e. lectures, tests, assignments, quizzes and mini project) of the whole semester. Purposive sampling method was used, to enable the full scope of exploration where the respondents were selected based on their scores in sampling survey conducted a week before the interview starts. The sampling survey adopted Motivated Learning Strategy Questionnaire (MSLQ) instrument by Pintrich et al. (1991). MSLQ was designed to assess college students' motivational orientation and learning strategy across all learning contexts. This study chose to use only the motivation component due to match objective of this study which is to investigate the students' motivation. Based on their scores, three to four students consist of the lowest, average and highest score were selected from each section. An interview guide was used as a guideline and the researcher was allowed to probe into new questions during the interview. The interviews were conducted for maximum one hour for each student and were later transcribed into interview transcripts.

The data analysis employed thematic analysis technique to all observation field notes and interview transcripts. This method suits the nature of exploratory research of this study where the findings are in the forms the emerging themes. There are six phases in the thematic analysis based on the techniques by Braun and Clarke (2008) which are data familiarisation, generation of initial codes, theming, defining the themes and reporting. To familiarise with the data, all of the interview transcripts were read at least three times before the initial coding. The codes were sorted into themes and later discussed based on the element of a learner-centred environment.

3. Results

The results are presented in two parts. The first part is the observation data followed by the second part which consists of the interview data.

3.1 Observation Data

To answer the first research question "To what extent the programming course matches the criteria of learner-centred learning environment to motivate engineering undergraduates to learn to program?" class observations were conducted throughout the semester.

An overview observation was done in the first week. Each student was given their own computer and space. Traditional lectures and tutorials were identified as the teaching methods used in the course. No significant changes were observed in the teaching methods throughout the semester. While the classes were conducted, students will use the computer to program using MATLAB. The instructors would also use the whiteboard to show mathematical calculations or flowcharts and the computer to show the coding steps.

The observation also shows that the instructors for section 2 and 3 were concerned with the students' prior knowledge on programming. A short survey was conducted by them, asking whether their students have previous exposure to programming and the programming language that they had learned in the previous institution if they had. This has helped the instructors to assign the students' with experience as indirect tutors to the students with no experience in programming.

In section 1, the instructor chose to use the local language as the instructional medium instead of English. This could be done since section 1 consists of only the local students. This also shows that the instructor was trying to connect the students' language culture with programming concepts. The students showed that they were happy when they can communicate in their own mother-tongue during lessons. It did not eliminate the fact that they face difficulties to understand the concepts.

Students in section 2 and 3 were observed to be more active and comfortable with the instructors. This is because the instructors were observed to be more caring and friendly with the students. However, in section 1, even though the instructor was observed to be caring too, the students had the difficulties in understanding what were taught. This happened because the instructor did not show actions that he will look into the problems again when the students faced difficulties. Instructors in section 2 and 3 made efforts to revise and look at the problems in different methods. Such situation also depends on the instructors' ability to provide opportunities for rehearsal, both in the form of repeated exposure to models and in the form of time to reflect on the material or skills. They guided the students individually and provided opportunities for students to discuss and seek help from others. For this reason also, towards the end of the semester, students in section 2 and 3 some students were observed to be more active. Students in section 1 were found to be more passive and silent during learning throughout the semester due to the limited opportunities to discuss with each other.

3.2 Interview Data

To answer the second research question *"How do engineering undergraduates develop their motivation to learn to program within a learner-centred learning environment?"* interviews with the students were conducted. Table 2 shows the sample data with codes under the element of learner-centred which are prior knowledge, the learning culture, and clarity. The students needed consideration from the instructor where there are students who are new to programming (Students A) and they have their own learning styles that might not match the instructors' styles of teaching (Student D). Through this element also, the study finds that the students come from a various background of computing education, have different prior knowledge on computer programming and have different learning styles.

Table 2: Codes and examples of quotation

Code	Student	Example of quotation
Learning culture	Student D	For me, my learning styles is more on looking at the questions, try it and imagine the solution, then look at the given solution and later compare my own solution with the given solution.
Prior knowledge	Student A	It is because, in the class, not all of them have experience taking programming courses. He should take all of the students as zero level students. Because everything was not explained from the beginning, they were too sudden. Their motivation became low, because they didn't understand.
Clarity	Student C	Yes, when the lecturer explains well, students will understand, and then the students' motivation will be high. Because during my matriculation, I like the programming class, because the lecturer's teaching was very clear. The explanation was more detailed. It started with the input, not straight away asked us to create an algorithm.

4. Analysis and Discussion

This discussion sought to answer the third research question which is to provide the recommendations based on the learner-centred element to improve the learning environment and motivate their students to learn to program.

In the perspective of the learner-centred learning environment, the students' background and prior knowledge are important information for instructors to plan their teaching approaches. Some students have received earlier exposure to computing and programming concepts. The students who are familiar with programming valued the learning more in contrast to the students with no prior knowledge. Students with prior knowledge knew that programming requires deep-thinking learning skills and were able to regulate their learning. These students also probed into questions and tried to explore the programs regularly.

Students with no prior knowledge were very dependent on the lecture, learning materials and peers. This affected their level of expectancy in learning. They were easily frustrated, gave up during the learning process and believed their success rate is low. Factors that could motivate them to learn are the grades and people's expectation and they believed that they cannot succeed the course without relying on exercise and examples. The differences in characteristics between these two groups of students support the observation findings where some students depend on each other during learning where the students managed to create discussions rather than studying individually.

Concerning on the existence of these two groups, the instructors' efforts to obtain the students' background information is considered as a diagnostic teaching. Diagnostic teaching is a way for instructors to gain information from the students to improve their teaching such as observations, conversations, surveys and reflections (Bransford et al., 2000). The information allowed the instructor to think on the level of difficulties of the tasks, provide scaffolding to the students with no prior knowledge and provide opportunities for discussions.

The students are also used to a teacher-centred learning environment which is part of their learning culture. The students required more examples and exercise for the subject to satisfy their learning styles thus to develop their motivation to learn. The students' dependency on examples and exercises is also related the students' cultural practices and the practices can affect their learning beliefs. For such case, a growing mindset paradigm is necessary. This can happen by starting the course with informal ideas that students have about their daily life and help them to transform the ideas to understand programming. It can improve their understanding rather than focus on examples and exercises which are memorisation oriented. Another benefit of shifting from giving examples and exercise to this type of learning is the student will get high satisfaction when the task is completed and thus motivates them to pursue more self-exploration on programming.

Furthermore, the learning materials, programming terms and visualisation of the concept should match their pre-existing knowledge and misconceptions. If the concepts to be learned and the way they are organised match neatly with a learner's pre-existing knowledge base, then the learning is likely to be smooth and rapid (Gomes et al., 2012). This is also related to the clarity of the instruction, students find that the instruction was unclear and leads to misconceptions. This also frustrates and demotivates them. Unclear instruction means there could be unaligned pre-conceptions in programming discourse. One way to overcome this is by looking at a term called 'conceptual change'. Conceptual change can be defined as a process of the students to re-conceptualise their deeply rooted misconceptions that interfere their learning (Bransford et al., 2000). The instructors might not realise when this process happens but proper strategy can help to ease the process. Such situation also depends on the instructors' ability to provide opportunities for rehearsal, both in the form of repeated exposure to models and in the form of time to reflect on the material or skills (Schunk et al., 2014).

5. Conclusion

Motivation has been a common addressed issue in engineering education. It is almost impossible for instructors to change or easily influence factors which are outside the classroom, such as the institutional and community norms, the attitudes and beliefs of the students' surrounding people, or the students' family background and beliefs, but it is possible to change and control what they implement in their own classrooms. This study indicates that a change can be made to the traditional learning environment to learner-centred learning environment for programming courses in engineering curriculum. It also provides some insights on issues that should be considered in the course planning based on the learner-centred environment criterion.

Acknowledgments

The authors would like to thank Ministry of Higher Education and Universiti Teknologi Malaysia for supporting this research under flagship grant R.J130000.2409.02G57.

Reference

ABET (Accreditation Board for Engineering and Technology), 2014, Criteria for Accrediting Engineering Programs–Effective for Evaluations during the 2015-2016 Accreditation Cycle, Engineering Accreditation Commission, ABET, Baltimore, United States.

Beers K.J., 2006, Numerical Methods for Chemical Engineering - Applications in Matlab, Cambridge University Press, Cambridge, United Kingdom.

Bowen J.D., 2004, Motivating civil engineering students to learn computer programming with a structural design project, Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition, 20-23 June 2004, Salt Lake City, Utah, United States, Paper 13064.

Bransford J.D., Brown A.L., Cocking R.R., 2000, How People Learn: Brain, Mind, Experience, and School, National Academy Press, Washington D.C., United States.

Braun V., Clarke V., 2008, Using thematic analysis in psychology. Qualitative Research in Psychology 3 (2), 77-101.

Engineering Accreditation Commission, 2014, Criteria for Accrediting Engineering, Engineering Accreditation Commission, Kuala Lumpur, Malaysia.

Gomes A., Mendes A.J., 2007, Learning to program-difficulties and solutions, International Conference on Engineering Education – ICEE 2007, 3–7 September 2007, Coimbra, Portugal, Paper 411.

Gomes A.J., Santos A.N., Mendes A.J., 2012, A study on students' behaviours and attitudes towards learning to program, In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education, July 2012, Haifa, Israel, 132-137.

Lee H.H., Arora J.S., 1991, Object-oriented programming for engineering applications, Engineering with Computers 7 (4), 225-235.

Pintrich P.R., 1991, A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ). National Center for Research to Improve Postsecondary Teaching and Learning, Ann Arbor, United States.

Wing J.M., 2008, Computational thinking and thinking about computing, Philosophical Transactions of the Royal Society of London A: mathematical, Physical and Engineering Sciences 366 (1881), 3717-3725.