

Article 1

The Uncommon Approaches of Teaching the Programming Courses: The Perspective of Experienced Lecturers

Jamal Othman, Naemah Abdul Wahab
Department of Computer & Mathematical Sciences
Universiti Teknologi MARA Pulau Pinang Branch

Abstract

Programming courses have become one of the important courses for all sciences programs which are offered at the tertiary level (university). Currently, the students are compulsory to enroll in at least one basic programming course in order to fulfill the requirement of the university. The university needs more experienced lecturers to teach the programming courses. Most of the students are not interested in programming subject and they have considered the programming course as a high failure rate course. As a lecturer, we should be able to change their negative perception and motivate the students to use and apply the programming skills in their life.

Keywords: *teaching, experiences, approaches, programming, mentor & mentee, courseware*

Introduction

Teaching programming courses is not an easy task. In a paper by Abid (2011), he mentioned that Rist (1996) has acknowledged the lacking in professional and expert skills in programming language by novice learners as one of the main reasons of the program comprehension difficulties. It is a challenging task to make the students understand the topic that sometimes, the lecturers are required to use creative and innovative teaching methods in their class lessons. Only lecturers who taught the programming subjects will understand the complications and challenges faced in the process of teaching and learning of programming courses.

Most of the students who registered for these courses are considered as beginners with zero knowledge, and it is the responsibility of the course instructor to produce a valuable and knowledgeable human capital with an outstanding programming techniques. According to Gomes and Areiad (2008) the programming courses are generally labeled as difficult and the dropout rate is quite worried. Most of the students have set in their mind that the programming subject is a high failure rate course. They are overshadowed by this negative perception that they do not realize their hidden potential in programming skills.

In general, for those students who are specialized in computer science or any programming fields, they must enroll in at least 3 programming courses throughout their studies. The students should be able to grasp the concepts of programming techniques before they applied for industrial training or employment. Riley (1981) has identified the major problems of our students are the lack of skills in problem solving and analytical skills. The students should be exposed on the real problem and reduce the theoretical orientation of teaching approach. Most of the ICT industries have set the basic requirement for the graduates to have a good skill in programming methods.

This article will be discussing about the lecturers' perspectives and their teaching experiences on programming courses to two different categories of students, the beginner's groups and the intermediate as well as advanced students groups.

Teaching the beginners is a very exhausting process as the course instructors have to reiterate a simple topic repeatedly. Normally, you will need a longer period of time to cover a simple topic and in some worst cases; you have to recap the previous chapter after teaching the same topics for many weeks. Riley (1981) has identified the major problems of our students are the lack of skills in problem solving and analytical skills.

The students should be exposed on the real problem and reduce the theoretical orientation of teaching approach. This is one of the challenging parts and parcel of becoming a lecturer. Experiencing of failure rate more than 50% in a programming course is a common situation for all the lecturers. It is a very tough obligation when you are appointed by the Dean of Faculty to teach the programming course for the beginners. Moreover, if the basic academic qualification of the students' enrollment is very low, then it becomes a very excruciating experience to all programming lecturers throughout the entire semester.

In realizing the seriousness of these issues, several workshops and seminars have been conducted by the faculty and campuses to come out with the best solution to overcome this problem. Some of the solutions outlined are considered irrational because it has never been implemented but it is foreseen to be capable of producing a realistic outcome. Some solutions are impossible to be implemented because it will give additional tasks or workload to the lecturers. The best solution now is to come up with an easy, realistic, effective and practical way that does not require additional workload with acceptable approach to tackle the existing problem.

Methodology

There are many methods of effective teaching for programming courses. According to Zhou and Jin (2010), it is important to have good programming courses teaching quality as it will influence the learner's programming skills and computer application when they are working in the industry.

To begin with, lecturers **encourage the students to do a lot of exercises** by drilling through the pass year's examination questions and discussing the answers with their lecturer. By doing this, the learners not only concentrate on answering the questions but it also exposed the student to the precise method of writing the answer according to the answer scheme or lecturer's expectation. However, not all the lecturers including us, agreed to these common practices. This customary method in reality guided the students to become exam-oriented and narrowed up the scope of developing their programming abilities. It will only assist them in achieving good result in their final examination. Students know that some questions from the pass year's examination will be asked again in the upcoming examination, thus the scenarios of memorizing and targeting for certain questions or some particular topics also happened in our education system. Another reason, which contributed to the student's weakness in programming skill is due to the current education system that still depends on examination systems, nevertheless is very much lacking in hands-on programming practices exposures.

Maheshwari (1997) has expressed programming lessons should employ systematically designed direct instruction activities, rich in feedback and practice opportunities. Programming activities should be exposed with problem solving strategies such as planning, simplification and modeling. Teaching strategy should give high impact on students' knowledge and their ability to apply the knowledge.

Next, in the present education systems, **we should expose the students to a variety of assessments** such as assignments, quizzes, tests, project and final examination. In a study by Ramirez and Munoz (2014), they pointed out that assessment such as programming project enhances learner's programming comprehension as the students get the opportunity to apply their programming knowledge and abilities during the project development and implementation. From our past experiences, students tend to copy the assignment from their peers.

Furthermore, the assignment will not be an effective method of evaluation if you are giving the same question to each student. Therefore, we suggested the course instructors to give a new question to each of their students to be solved within 15 minutes during the practical programming laboratory session. This technique will determine whether the assignment presented to the lecturers is originally prepared by the student's or copied from another person. We have practiced this method for several semesters and we do not have to worry about their final examination because the students are capable of solving any types of question given to them. Additionally, they are also competent to present different kinds of unique answers or solutions and outstanding methods of problem-solving in answering the programming question.

Another method of teaching that we have implemented to our students are by **enquiring them to create questions, which are related to the current topic learned during the class lesson**. The lecturers are not required to answer the question; however, the questions will be answered by the remaining group of students. This method is normally applied in suggesting or proposing the topic for student's final year project. We should also apply this technique in programming courses to encourage students in becoming a creative thinker. By asking the students to create the questions, we are able to identify the level of students' knowledge in the programming topic, inspiring creativity in producing the questions, encourage ability to develop the question according to the degree of difficulties and finally, the course instructors are able to create a database of questions bank.

Nevertheless, you have to make sure that the students really understand the overall concepts of a topic before this practice can be implemented. We have tested this method for a couple of semesters and we found that the outcome is excellent. This teaching approach will really benefit both students and lecturers because it stimulates the elements of knowledge sharing if we are able to implement it successfully.

As a lecturer, we always **encourage the students to ask questions**. Nevertheless, do you think that this is really effective? From our observation, it only assists a small group of students. Flowerdew and Miller (1995) found that the Asian students are actually shy of asking questions to the lecturer. They are concerned if they asked the wrong or silly questions, asking for a personal reason, asked to be commended or the answer of the question can be obviously found in any

textbooks. In contrast to the western students, they ask questions because they believe they have the rights to inquire.

Mentor and mentee systems, which is practiced for three semesters, seems effective in motivating the students to improve the programming skills and logical thinking, boost up confidence level and creating the programming interests. The mentor and mentee systems are handled by students themselves. The lecturers must ensure that the group of mentor and mentee system should include a person who is very proficient in programming language and skills. This person will act as a facilitator and mentor who are capable to create stimulating environment to all the mentees. In a study conducted at Monash University, Freeman, Jaeger and Brougham (2004) concluded that group work brought advantages for the students since working in group “can make computer coding more accessible to those with minimal or no background in this field”. In turn, this may contribute to a higher level of programming aptitude and programmer confidence.

Another preferable method by **students is learning through examples**. The lecturers must show and explain many examples to students before they can understand the concepts of programming logics. Silva et al. (2003) had mentioned that the examples should be practical and interesting. Moreover, Zhou and Jin (2010) added that some universities provided programming examples resources on their websites for student’s easy acquirement to improve the learner’s interest, motivation, skills and decreasing their learning time. The examples should be from different perspectives, situations, conditions, cases or problems. The students are required to do a project will prefer the course instructors to show samples of their senior’s project based on our experiences.

Thus, from there, they will get some ideas to produce the project proposal. From the lecturer’s point of view, this practice is not considered as a plagiarism, but it is a way to provide the students with a basic overview and rough idea of a project proposal as well as the lecturer’s expectation of the project requirements from their students. Moreover, as a lecturer, we will be very amazed and totally pleased to see that most of our students will be presenting an excellent completed task beyond our expectation that fulfill every requirement. Furthermore, we believe that through the implementation process of this project, it enhances their programming skills as well as improving their interpersonal and intrapersonal communication.

Other than that, the **lecturers should encourage the students to explore more examples from the websites**. According to Chapman (2002) the Internet is a relatively the most important channel to retrieve valuable resources and contains huge quantities of information. Nowadays, you will realize that the answers of the project given to the students are available in the websites. In fact, hundreds of project samples can be found on the Internet and are free to be downloaded. Therefore, the students will have a tendency to download the answer and modify the source code of a programming project according to the requirement of the questions. This norm amongst the students is difficult to be controlled. However, since the lecturer must make sure the originality of the project, their students should be allowed to reproduce the project but with a few conditions. They must understand the technicality of the source code, modify the source code or sample of the application according to the project or user’s requirements and finally customize it according to lecturer’s specifications or the company’s operation.

The same situation happens in a working environment when the lecturers are required to do a research. Some studies particularly involving the research on project development, has been conducted by other academicians, thus, most of the source code can be downloaded from the open source websites. We are welcomed to download the source code without any violation to the copyright issues. Another channel to get the source code is by joining the open source community. This community will assist you in solving any technical problems or sharing specific source code that fulfills most of your requirements. Interestingly, we will receive a prompt reply or the sample source code within 3 days of our inquiry.

Another method of effective teaching of programming courses is by **providing the appropriate courseware for the students**. The courseware should be related to the topics covered in the syllabus. It should be interesting, but importantly, enable the students to understand the concepts, and at same time will attract the students' interest on programming courses. We have applied this technique to the programming courses and the outcome is positively unexpected. Some courseware has inserted elements of humor; with user-friendly designs that inspires the students to comprehend the programming topics easily. In addition, the lecturer's workload can be reduced since the courseware supports the students to learn each topic individually. To provide stimulating problems for students to solve, the environment needs to provide a rich set of graphical capabilities such as 3D features, as mentioned by Carlisle, et al. (2005). Soloways (1998) has mentioned in his article that the creativity of lecturer of applying animation to show program execution will minimize the students' difficulties.

The final teaching method that we are going to proposed will be it terms of the number of students in a group or classroom. **The size of the group should not exceed 25 students**. A bigger number of students in a group will result in fewer students focus on the topics that are taught and you as a lecturer cannot deliver the knowledge effectively. Programming requires more examples and exercises practiced by the students. Working with a small number of students allows a higher degree of interaction and discussion opportunities in the class. Brown and Atkins (1996) proposed this technique and it is applicable in technical subjects such as the programming paper. Studies on small groups have also been developed by social psychologists, Argyle (1983), who suggested that working with twenty students is the limit for developing good interaction.

Conclusion

In conclusion, the methodologies that have been mentioned in this article are based on the lecturers' experiences. We cannot presume that these methodologies are totally effective. Nevertheless, you can apply the methodology in your teaching and establish the effectiveness. There are many articles discussed on these matters and we have perceived that most of it is the subset of methods that have been mentioned in this article.

References

- Abid, S.H., Zehra, S. and Iftikhar, H. (2011). *Using Computer Aided Language Software for Teaching and Self-learning*. In Proceeding: 14th International Conference on Interactive Collaborative Learning (ICL2011), pp. 102-106.
- Argyle, M., *The Psychology of Interpersonal Behavior*, 4th ed: Penguin, London, 1983.

- Brown, G. and Atkins, M., *Effective Teaching in Higher Education:Routledge*, 1996.
- Carlisle, M. C., Wilson, T. A., Humphries, J. W., and Hadfield, S. M. 2005. *RAPTOR: a visual programming environment for teaching algorithmic problem solving*. In proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. ACM Press, 176-180.
- Chapman, L. (2002). *Russian roulette or Pandora's box: use of Internet as a research tool*. Paper presented at VALA 2002. 11th Biennial Conference and Exhibition, 6-8 February, 2002, Melbourne. Victoria, Australia: Victorian Association for Library Automation Inc. Retrieved 24 December, 2006, from <http://www.vala.org.au/vala2002/2002pdf/18Chpmn.pdf>
- Flowerdew, J. & Miller, L. (1995). *On the notion of culture in L2 lectures*. TESOL Quarterly, 29(2)345-373.
- Freeman, S. F., Jaeger, B. K. and Brougham, J. C., *Pair Programming: More Learning and Less Anxiety in a First Programming Course*, Vol. 2004, 2002.
- Gomes, A., Areias, C. M., Henriques, J. & Mendes, A. (2008). *Aprendizagem de programação de computadores:dificuldades e ferramentas de suporte*. Revista Portuguesa De Pedagogia, 42, 2, 161-179.
- Kessler, C. & Anderson, J. (1989). *Learning flow of control: recursive and iterative procedures*. In Soloway & Spohrer: Studying the Novice Programmer, pp. 229-260.
- Maheshwari, P. (1997). *Improving the learning environment in first-year programming: Integrating lectures, tutorials, and laboratories*. Journal of Computers in Mathematics and Science Teaching, 16(1), 111-131.
- Ramirez L., A., Munoz D., F. 2015. *Increasing practical lessons and inclusion of applied examples to motivate university students during programming courses*. In : Journal of the Social and Behavioral Sciences, Vol. 176, pp. 552-564.
- Riley, D. (1981). Proceedings from Technical Symposium on Computer Science Education '81: Proceedings of the twelfth SIGCSE Technical Symposium on Computer Science Education, pp. 244-251, New York: ACM Press
- Rist, R. (1996). *Teaching Eiffel as a first language*. In: Journal of Object-Oriented Programming, Vol. 9, pp. 30-41.
- Silva, I.,H., Pacheco, O. and Tavares, J., *Effects of Curriculum Adjustments on First-Year Programming Courses: Students' Performance and achievement*, presented at 33rd ASEE/IEEE Frontiers in Education Conference, Boulder, Colorado, 2003, p. 3.
- Soloway, E. M. (1986). *Learning to program = learning to construct mechanisms and explanations*. Communications of the ACM, 29, 850-858
- Zhou Q., Jin J. 2010. *A Novel Student-Centered Teaching Reform on Programming Courses*. In proceedings of the International Conference on E-Health Networking, Digital Ecosystems and Technologies. IEEE, 221-223.