2018

Best Approach to Teach Advanced Programming

Soly Mathew Biju
University of Wollongong, soly@uow.edu.au

Publication Details

Biju, S. M. 2018, 'Best Approach to Teach Advanced Programming', International Journal of Education and Management Engineering, vol. 8, no. 6, pp. 37-45.

I.J. Education and Management Engineering, **2018, 6, 37-45**
Published Online November 2018 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2018.06.04

Available online at http://www.mecs-press.net/ijeme

Best Approach to Teach Advanced Programming

Dr.Soly Mathew Biju

University of Wollongong in Dubai, Dubai, UAE

Received: 17 January 2018; Accepted: 06 August 2018; Published: 08 November 2018

Abstract

This paper focuses on an approach that could be effective for teaching programming languages such as advanced Java Programming, and involves a new framework implemented using the practical approach along with new framework. Assessment tools were designed to facilitate this kind of approach toward teaching programming. The new approach is implemented by conducting lectures in the lab or laptop facilitated classroom. The subject assessments and delivery methods were modified to include projects and class works. Students were encouraged to apply concepts learnt in the class in an incremental manner leading to a complete software application and also write a reflective report. It adapted the Kolb's experimental learning style theory. Effectiveness of this approach is evident through a comparison of students' results obtained after implementation of this approach, with results obtained using the traditional style of teaching.

Index Terms: Java, practical approach, project-based approach, assessment, modality, framework, long-term memory, learning style.

1. Introduction

Teaching and learning programing is considered to be a very difficult task since many years[16]. It is equally challenging for lecturers to help students acquire programming skills. We have to consider the fact that we are dealing with the Net Generation, who are exposed to high-end technology at a very early age and hence, find it difficult to attend and focus on a teaching session in a purely traditional classroom [14]. It is tiresome to listen to programming language instructions without understanding and knowing the practical application of these instructions. In spite of the importance of programming in the computer science and software engineering courses, it is often found that only half or one third of the students are able to understand the concepts correctly and put them to use efficiently [17]. Teachers usually focus on teaching concepts using one or two examples

* Corresponding author
E-mail address: dr.solymathew@gmail.com, solymathewbiju@uowdubai.ac.ae

from the text. The classroom is not equipped with PCs or laptops that would facilitate hands on experience in implementing the concept taught in the class. Learning a programming language is not only learning the syntax but it is important for students to be able to solve real life problems using an application designed using an appropriate programming language.

Student studying their first programming subject is not aware of the correct method to study and to self-train themselves to study a programming language hence rely on copying code and by hearting programs.

The teaching-learning process is very important and a delicate process that will shape the future of the students [A].

The proposed research focuses on providing students with a well-organised and well -structured method which follows the learning cycle that will help them attain programming skills.

The lectures were made interesting and interactive by following a hands-on practical incremental approach to teaching. The subject that was being considered was Java programming, covered a number of advanced topics. At many universities Java programming is taught to students pursuing their undergraduate degree in Computer Science. Since it introduces them to Object oriented programming, there are some major challenges faced by students as explained in a research conducted by Biju[6] which is out of the scope of this paper.

2. Literature Review

The learning styles theories state that every individual has a unique learning style and educational outcomes can be achieved by mapping this learning style with learning experiences. While teaching a programming language, it is imperative that students understand the theory and concepts and are able to apply it to real world problems. Teaching that focuses only on theory results in incomplete learning. Dewey[11] has stated in his experience theory that real education comes from experience. Students may not understand depth of the concept and how powerful the realistic implementation of it can be. Various methods have been adopted by teachers to make learning interesting. Interactive teaching is one such method adopted by teachers at various universities and schools. Sandra et al [24] used a practical approach to teaching coding using stubbing, where students were encouraged to write functional stubs. They have concluded that students that adopt this approach become better programmers. Practical approach for teaching software development life-cycle used by Catanio [9] enabled students to acquire the required skills. Gray et al [15] states that the traditional approach to teaching programming leads to several problems such as passive learning style, over emphasis on language form as opposed to function, premature complexity and premature abstraction. He further states that the traditional teaching approach imposes a rule-based approach on the learner and restricts their natural problem-solving skills.

A large group of professionals are developing and expanding practices of the discipline in parallel with academia. One such researcher focuses on the transfer of professional practice into the classroom [13]. Use of Alice to teach basic programming concepts were also found to be beneficial in by a research conducted by Biju[7,8]. Some examples of recent contributions to the programming practices mainly offered by people outside academia are design patterns and frameworks, extreme programming, refactoring, agile development and test-driven development [3,4,10,14,21]. Many a time, computer science faculties have the misconception that theory must lead and that practical application of the theory can be carried out later. Research, however, has shown that leading with practice often provides a greater motivation for students to learn the applicable theory. Practice-driven approach allows students to develop their own theoretical models autonomously, and does not rely on models that are taught in classrooms, as stated by Linder, Nestrick, Lavelle [18] [cited in [23]].

3. Current Teaching Approach in most of the University

The conventional approach to teaching programming consists of one or two 2-hour lectures per week held in lecture rooms. The lecture room is facilitated with an overhead projector connected to the lecturer’s PC. Lectures include concepts of programming, the language syntax and snippets of code. Teaching material consists mainly of PowerPoint slides. Some fragments of codes may also be shown or explained to students in class. Concepts are covered in the class with no hands-on programming. Over the week that follows, lab sessions consisting of problems to be solved and implemented using Java programming language are held [7,8]. Students are introduced to the Interactive Development Environment in the first lab session. Lab assistants conduct lab sessions, each of which lasts for around two hours each week. The lecturer designs lab tasks comprising practical application of concepts taught during the lecture conducted in the previous week. One of the common problems that arise in a typical lab session is that of students not being able to retain concepts taught in the previous week. Students need to read through the slides, and seek help of lab assistants to solve the problem. This solution is not always feasible, as there is only one lab instructor available for every 25–30 students. Lab tasks need to be completed within a specific time, which makes it even more challenging for students. Assessment tools for the subject include two or three independent of each other.

4. Proposed Approach for Teaching Programming

Students are not able to retain concepts taught in lectures mainly due to the fact that the modality of interaction is seeing and hearing, which corresponds to only 50% of the percentage retained long term as presented by Linder [19] in Magnesen’s Table.

McLeod [22] describes Kolb's experiential learning style theory in the figure 1 below

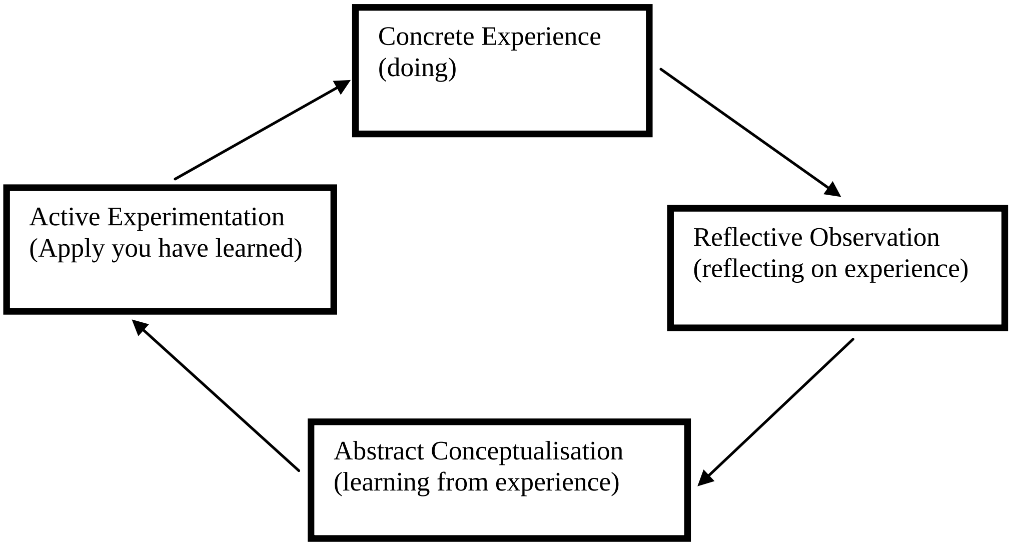

Fig.1. Kolb's Experiential Learning Style Theory

McLeod[19] mentions that Kolb's experiential learning style states that effective learning is seen when a person progresses through a cycle of four stages: of (1) having a concrete experience followed by (2) observation a report and reflection on that experience which leads to (3) the formation of abstract concepts (analysis) and generalizations (conclusions) which are then (4) used to test hypothesis in future situations, resulting in new experiences.

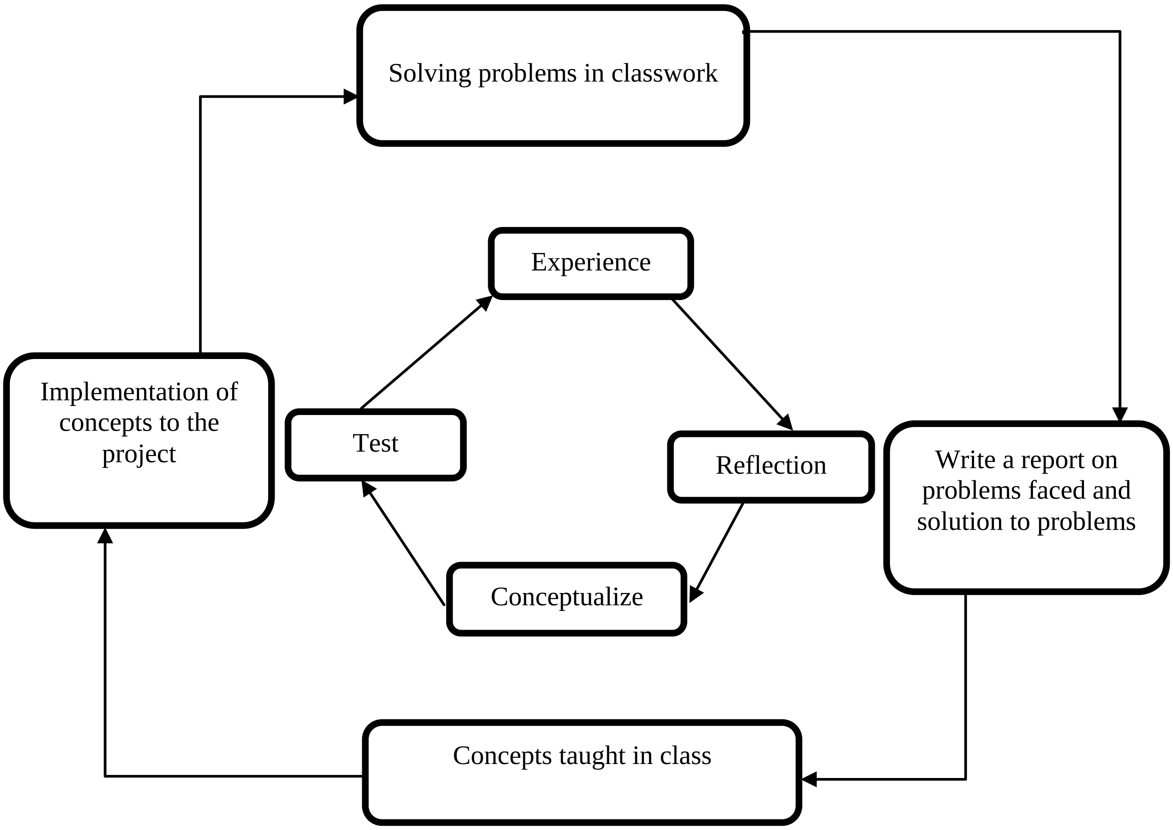
Solving problems in classwork
Experience
Implementation of concepts to the project
Test
Reflection
Write a report on problems faced and solution to problems
Conceptualize
Concepts taught in class

Fig.2. New Teaching Framework

Kolb [18] views learning as an integrated process with each stage being mutually supportive of and feeding into the next. It is possible to enter the cycle at any stage and follow it through its logical sequence.

1. Class work (Step A) - In case of teaching programming classwork is considered as Concrete Experience state.
2. Reflective writing (Step B) constitute Reflective Observation state.
3. Concepts taught (Step C) in class through classwork constitute Abstract Conceptualization state.
4. Implementing the concept learnt in class to the project (Step D)is seen as Active Experimentation state.

To follow the new approach, some changes were made to the elements of assessment. Assessment for the new approach consists of project, mid-term, classwork and final examination. This proposed teaching approach the lectures are conducted in a lab equipped with an overhead projector connected to the lecturer's computer, a white board and 30 PCs for the students. The new approach consists of the following steps.

Step A: The lecturer is seen as a facilitator. The lecturer designs classwork sheet with the main objective to cover a particular concept. The students are expected to complete the class work and execute the program within the time given. Since the students are not only seeing and hearing, but also doing and discussing, the percentage retained long term is 90%, as per Magnesen's Table [19] this is implemented by having students solve a problem in class thus studying the concepts intended to be taught through the session.

Step B: The students now write a reflective report based on the concept they have learned through completing the classwork. The report will have the concept learnt, solution to the problem, details of any technical problems faced and solution to the same.

Step C: The lecturer use a power point presentation to explain the concept taught through the classwork. The lecturer also discusses the possible solutions. This helps the lecturer design more challenging tasks for the lab session in the following week.

Step D: The students are now required to implement the concept learnt on their project before the next lecture session. They also do a reflective writing in the form of a report where they had to state what they learnt from their experienceof developing the project, the problems they faced and solution they applied to solve these problems. This completed the learning cycle.

Assessment tools designed for this approach are very different from that used in the traditional approach. A project with 30% weightage replaced the existing two assignments. This project was divided into phases. The first phase consisted of the class diagram design showing details such as attributes and methods of each class and the relationship between various classes. Phase 2 consisted of GUI design and event handling and connection to databases. This helped students implement various concepts taught in the class to their project, in an incremental manner which maps onto the next state experience gained by implementing the concepts to the project.

It proposed approach exposes students to hands-on programming immediately after learning a concept. Every topic taught has a respective practical classwork question associated with it. This helps the lecturer to assess whether the students have understood the topic taught in the lecture.

One of the sample questions given to work on after an introduction to classes, objects, interfaces, and exception classes is as follows:

4.1. Classwork 3

The code given below is an incomplete code. The complete code is written to perform the following tasks:

When the button “OK” is clicked by the user, the statement “Hello there” will be displayed on the console Fill in the missing statements so that the code produces the required result. Import

```
java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ButtonTest extends JFrame {
//…(1)mention appropriate statements to declare the button


// set up GUI
public ButtonTest ()
{
   super( " JButton Test" );
   // get content pane and set its layout
   Panel panel1 = new Panel();
   panel1.setLayout( new FlowLayout() );
```

```
  // (2)…mention appropriate statement here to create the button and add it to
panel 1 this.add(panel1);
  //(3) …mention appropriate statement(s) here to register listeners for Button
     setSize( 275, 100 );
     setVisible( true );
  }
  public static void main( String args[] )
  {
     ButtonTest application = new ButtonTest();
     application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
  }
  // private inner class for ActionListener event handling
  private class ButtonHandler implements ActionListener {
  // (4)... appropriate statements will appear here to
  //define method of the interface ActionListener
  }
  }// end class
```

This classwork consists of topics taught in the previous session along with new topics taught in the current session, in an incremental manner. Quizzes conducted indicated that students had fewer syntax errors and were well-versed with the development tools. This indicates that the students have reached the coding level of learning and are ready to progress to the understanding level, according to Bruce model[8].

5. Research Population

To evaluate the effectiveness the new framework used in delivering the programming language, data will be collected from two semesters for the course 'Java Programming and Object Oriented Concepts'; the same instructor will teach both the semesters.

Data will be collected from performances of students in quizzes. The quizess for both the research groups will be similar and will be marked by the same lecturer using the same marking criteria.

One of the major changes that will be made to the course will be the replacement of two assignments with one project that was divided into phase 1 and phase 2. The teaching style will be changed to include class work to be completed on the laptop or PC during the lecture, followed by writing a reflection report. The students had to implement the concept in the project. All these changes will require time and a lot of effort by the lecturer. The lecturer has to be very creative and capable to developing interactive activities aimed towards teaching a particular concept. These activities are not only focused on finding the correct solution but help student explore new experiences. Diverse activates not only help students acquire programming skills but also makes learning enjoyable. The results of these changes were very significant and hence, are worth it. Performance in quizzes held in classes following the traditional approached will be compared to the result obtained based on the proposed approach which will provide students with proper guidance through the tasks given in class, reflective writing and completion of the project on incremental basis.

6. Challenges Faced

The implementation of the proposed framework requires a significant amount of time invested by the lecturer. This will increase the workload of the academics and made it challenging due to limited of preparation time between lectures. The lecturer may have some issues with time management and may not be able to correctly judge the amount of time students will require to complete the class work. In case the student is unable to complete the tasks given in class, one solution proposed is that the lecturer should encourage the students to complete the work at home.

7. Conclusion

Various methods have been implemented in the past by lecturers teaching programming, many of which followed a traditional teaching style. Lectures were conducted in the lecture hall using PowerPoint presentations and students did not have the opportunity to execute or modify the code explained. This method tends to leave a gap between learning a concept and application of the concept, which makes it difficult for students to retain the concepts taught. This research shows that the most effective way to enable students to easily retain what is taught in class using the new teaching framework.

Assigning projects to students could highly motivated them. This will also help them work on the concepts taught in the class throughout the semester by applying it to the project on an incremental basis and helped in completing the learning cycle.

Lecturers are suggested to adopt the proposed framework to teaching programming languages. This approach will provide students a new and effective learning experience and will require the lecturers to design class works/tasks covering the concepts taught during the session. This will help student acquire programming skills that Afra and Rizwan[1] argues that most student lack.

The researcher recommends that the proposed would prove to be beneficial in teaching advanced programming subjects in the Computer Science at all universities in order to improve the ability of students to follow programming concepts.

Acknowledgements

I would like to thank all the anonymous reviewer for their valuable suggestions that helped me improve my paper.

References

[1] Afra A. Alabbadi, Rizwan J. Qureshi,"The Proposed Methods to Improve Teaching of Software Engineering", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.8, No.7, pp.13-21, 2016.DOI: 10.5815/ijmecs.2016.07.02

[2] Bassey Isong (2014), "A Methodology for Teaching Computer Programming: first year students' perspective", *I.J. Modern Education and Computer Science*, 2014, 9, 15-21 DOI:10.5815/ijmecs.2014.09.03

[3] Beck K (1999) Extreme Programming Explained: Embrace Change. Addison-Wesley Professional.

[4] Beck K (2003) Test-Driven Development: By Example, Addison-Wesley, pp. 240.

[5] Butler M, Morgan M (2007), Learning challenges faced by novice programming students studying high level and low feedback concepts, Proceedings ascilite Singapore 2007. Available at: http://ascilite.org.au/conferences/singapore07/procs/butler.pdf. (Accessed Retrieved 25 May 2014).

[6] Biju SM (2018), Benefits of Working in Pairs in Problem Solving and Algorithms - Action Research, Athens Journal of education, forth coming papers Available at: https://www.athensjournals.gr/education/2018-1-X-Y-Biju.pdf (accessed 16 March 2018).

[7] Biju, S. Mathew. 2013, 'Taking advantage of Alice to teach programming concepts', E-Learning and Digital Media, vol. 10, no. 1, pp. 22-29.

[8] Biju, S. Mathew. 2013, "Difficulties in understanding object oriented programming concepts", in K. Elleithy & T. Sobh (eds), *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering*, Springers, Netherlands, pp. 319-326.

[9] Catanio JT (2006) An interdisciplinary practical approach to teaching the software development life-cycle In: SIGITE '06 Proceedings of the 7th conference on Information technology education 3-8.

[10] Cockburn A (2002) Agile software development. Boston: Addison-Wesley.

[11] Dewey, I (1963) Experience and education, New York: Collier.

[12] Diana G. Oblinger and James L. Oblinger (2005) Educating the Net Generation, Is It Age or IT: First Steps Toward Understanding the Net Generation, pp12-13.

[13] Fincher S and Petre M (2004) Computer science education research London: Routledge Falmer.

[14] Fowler M (1999) Refactoring: Improving the Design of Existing Code Addison-Wesley, pp. 464.

[15] Gray J, Boyle T. and Smith C. 1998, A constructivist learning environment implemented in Java. In proceedings of ItiCSE '98 - *Integrating Technology into Computer Science Education*, pp. 94-97. ACM Press.

[16] Guram N. Beltadze,"Game Theory - basis of Higher Education and Teaching Organization", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.8, No.6, pp.41-49, 2016.DOI: 10.5815/ijmecs.2016.06.06

[17] Huang T, Shu Y, Chen C , and Chen M(2013),The Development of an Innovative Programming eTeaching Framework for Modifying Students' Maladaptive Learning Pattern, *International Journal of Information and Education Technology*, Vol. 3, No. 6.

[18] Kolb, D. A., & Fry, R. E. (1974). Toward an applied theory of experiential learning. MIT Alfred P. Sloan School of Management.

[19] Linder PS, Nestrick EB, Lavelle LC (2001), Facilitating active learning with inexpensive mobile robots. Available at: http://www.cs.dartmouth.edu/~spl/publications/2001/FacilitatingActiveLearning.pdf. (Accessed 27 March 2014).

[20] Magnesen V (1983) A Review of the Finding from Learning and Memory Retention Studies. Innovation Abstracts, 5 (No.25).

[21] Martin, RC (2003) Agile Software Development: Principles, Patterns, and Practices, Upper Saddle River, NJ, Prentice-Hall, pp. 529.

[22] McLeod, S. A. (2013). Kolb - Learning Styles. Retrieved from http://www.simplypsychology.org/learning-kolb.html

[23] Ohlsson, LJ (1995) A practice driven approach to software engineering education IEEE Transactions on Education 38(3): 291 - 295.

[24] Sandra G, Reza K, Jacob RM, John WM (2001) Stubbing: a practical approach to teaching coding, *Journal of Computing Sciences in Colleges*, Volume 17 Issue 3, February 2002:29-34.

Authors Profile

Dr Soly Mathew Biju had a PhD in Computer Science and is currently an Associate Professor in the Faculty of Engineering and Information Sciences, University of Wollongong in Dubai. Dr. Biju has achieved the Chartered IT Professional status which is a symbol of excellence in the field of IT. She has recently completed 'Nature of Learning and Teaching' course with a distinction from Swinburne University. She is also ISTQB-certified software testing professional. She is also an active BCS Approved CITP Assessor. She was awarded UOWD excellence in Teaching in 2015. Her research interests include Machine learning, Big data analytics, software testing, cryptography, e-learning, innovations in teaching, agile software development, data science and programming techniques. She has papers published in reputed journals and books and presented and reviewed papers at national and international conferences.

How to cite this paper: Soly Mathew Biju,"Best Approach to Teach Advanced Programming", International Journal of Education and Management Engineering(IJEME), Vol.8, No.6, pp.37-45, 2018.DOI: 10.5815/ijeme.2018.06.04