

Teaching Beginners Android Application Development

Minjie Hu, Tony Assadi, Chalinor Baliuag
WeITec, New Zealand

Keywords

Android Application Development, Flipped Classroom, Teaching Innovation

Abstract

In recent years, Mobile Application Development (MAD) has been widely taught at most tertiary education providers, polytechnics and universities at different year levels and used in the final year degree capstone projects (Tigrek & Obadat, 2012; Mahmoud, 2011; Khmelevsky & Voytenko, 2016). Our institute offers MAD courses in the second year of the degree, as well as in the postgraduate IT programme in Level 8. In the degree programme, some students learned second-year advanced programming in Java language in the first trimester while others only have the first-year introductory programming in C++ or C# language in the previous year before enrolling in the 2nd year MAD paper. This creates a problem for the class with a group of students with different learning experiences and skills at the start of the 2nd year MAD paper.

It is challenging to teach MAD using Android Studio and Javato a group of students with a different prior learning experience and without adequate knowledge of Java and object-oriented (OO) development. Based on our experience early on in the trimester, it became clear that only using the traditional lecture PowerPoint slides and text-based lab instructions with minimum figures was not enough to meet the learning requirements by students with different computer programming background.

This research aims to investigate the use of action research and multiple teaching approaches with a flipped classroom in the degree programme for students from a diverse programming background. In particular, the research question is:

- Does the use of multiple teaching approaches and flipped classroom enhance student learning in the MAD course for students from a diverse programming background?

Facing the challenges experienced in teaching MAD to a group of students from a diverse programming background, we were inspired by Mahmoud's (2011) advocating by using traditional programming strategies. First of all, we focus on developing a smooth transition for all students from window platform development to mobile development environment. After a brief introduction of Java and OO programming, we made a comparative study of Window GUI and Android UI (User Interface) components. We then introduce variables, data types, and basic statements (input, process, and output) in the first Android project. Secondly, considering students who have difficulty in following the text-based lab document, we provided video clips for them to track the practice step-by-step. Thirdly, for advanced students who had learned Java language

before joining the MAD class, we applied the flipped classroom approach with the entire course resources (Gren, 2020) for them to focus on problem-solving. Beginners were also able to further advance their skills in the course through the flipped classroom.

Early results indicated that with the knowledge that students had from the Windows application development platform, students found it easy to implement their first mobile app using AndroidStudio. Further, students were very interested to find out how to work on OO method in Android. Taking advantage of such a motivation, we introduced how to use Android studio automatically generating class method code such as constructors, properties, and toString() method. We then used class constructor to link to Android UI for creating objects. Through this way, the beginners learned OO process while the more advanced students reviewed their knowledge and applied it into a new development.

Although the online resources for the MAD course included video clips for students to use, some students still could not fully understand and were struggling to complete their tasks. To address this, additional workshops were put in place for students as an additional learning opportunity. In the workshops, we discussed both overview and details of layout and program code. We focused on explaining the linkage between layout and Java code. We also provided intermedia level worked examples and final sample solutions for students to help in the completion of the tasks that they were expected to produce.

Due to Covid-19, in the fourth week of our trimester, our course had to change to online delivery mode, using Zoom and Moodle. The flipped classroom model met our online teaching well for both beginners and advanced students. From students' Zoom attendance and their activity of answering questions, it indicated that our strategies engaged students' learning. The first assignment is UI development, with two extra new features by student self-learning. There were 24 out of 30 enrolled students who completed this assignment in a workable project with diverse features.

From this study, we learned that traditional programming strategies can be applied to teaching Android by comparison to two different software development environments. To meet student requirements from a diverse background, we need to provide various types of course materials for online learning. Flipped classroom approach fits our current situation due to Covid-19. In further research, we would investigate how our strategies work on both beginners and advanced students in the advanced topics in this course by evaluating their results in future assignments.

References

Gren, L. (2020). A Flipped Classroom Approach to Teaching Empirical Software Engineering (pp.155-163). IEEE Transactions on Education.

Tigrek, S., & Obadat, M. (2012, June). Teaching smartphones programming using (Android Java): Pedagogy and innovation. In 2012 International Conference on Information Technology Based Higher Education and Training (ITHET) (pp. 1-7). IEEE.

Khmelevsky, Y., & Voytenko, V. (2016, May). A new paradigm for teaching mobile application development. In Proceedings of the 21st Western Canadian Conference on Computing Education (pp. 1-6).

Mahmoud, Q. H. (2011, June). Best practices in teaching mobile application development. In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (pp. 333-333).