Title of the journal article “*Formative Assessment of Programming Language Learning based on Peer Code Review: Implementation and Experience Report*”, authored by *Qing Sun, Ji Wu\*, Wenge Rong, and Wenbo Liu*.

**Ability to analyze the article against the theories learnt**

In the peer-reviewed paper, they designed a formative peer assessment approach based on the peer code review (PCR) module and implemented it in a programming learning course. The model is supported by the e-learning system OOCourse. The assessment process begins with a new assignment from a teacher. Assignment guidelines that state the detailed explanation of requirements were also released as soon as a new assignment is published. This aspect will help the students to understand the clear interpretations of the programming requirements and fair assessment opportunities to all the students. As we all know that the teaching-learning is enhanced when the students get the opportunity of being both the examiner and the observer. Hence, in this paper, the students played the role of both author and reviewer. As an author, students submitted a manuscript code and as a reviewer, they evaluated the manuscript code of their peers to promote collaborative learning. Each student can see the bugs reported by the reviewer student and the bugs that they reported as a reviewer. Such assessment activities led the student to identify their strengths and weaknesses to promote life-long learning. To focus on quality assurance in the assessment process, they implemented three functions in the OOCourse system which I felt was a critical challenge in programming language assessment. The first one being the Arbitration for non-consensus on reported bugs. The solving of non-consensus on reported bugs is very important to the reliability of peer assessment as it might lead to an argument between the reviewer and the authors, ultimately rising the distrust among the students. Therefore, they provided an appeal module in the OOCourse system where if the author disagrees with the reported bug, they can initiate the appeal to the respective reviewer anonymously. Students are more engaged in the peer assessment given the anonymity. Secondly, for fair and transparent assessments being important during the learning process, they have incorporated an anti-plagiarism mechanism. The manuscript code submitted by the students will be rejected if the similarity of code is above the threshold setting. Finally, they implemented the dynamic reviewer appointment strategy according to student’s programming performance which means the higher-ranked student review programs written by lower-ranked students. Teachers review the program written by the highest-ranked students while the students of equal rank review each other’s work. This allowed balanced assessment for diverse types of learners which allowed them to have the maximum exercise of each student’s ability.

**Ability to analyze the article against one’s teaching experience.**

I taught programming language modules to first-year students. I normally give the same assignment question to the whole batch of students. A set of guidelines including the rubrics will also be displayed in the VLE and the students need to submit within two weeks of time. We use the Urkund tool to detect the plagiarism of the work submitted but students will be allowed to submit the work even though the work is 100% plagiarized.

Like in the article, I didn’t implement the code rejection against the threshold setting because in the programming module, most of the programs would have the same logic with different implementations and it will be unfair to implement the code rejection before the work submission. When checking the submitted assignment, I look for similarities in their work and the most obvious thing that I discovered was the plagiarized work. The whole resource gets wasted when we check for such work and would have been effective if I did the same as in the article.

Having said that, as a tutor we also need that level of competency to build the system which checks for plagiarism in programming code, and I need experience and time to reach that level and implement. In addition, I didn’t assign any peer assessment strategy in my module. After I gave feedback to their work, if the students argue that they have done good work yet haven’t received good feedback, they can come, and we sort out things. Mutual agreements were drawn between the students and me.