ABSTRACT

Flipped learning approach becomes popular in higher education. Many studies have focused on discovering benefits of the flipped learning but fell short on demonstrating the design of flipped learning. Computer programming classes are perfect match for the flipped learning approach. This paper presents a series of detailed flipped learning activities of the nested if statement for an introduction to Java class to first demonstrate the structure of a flipped classroom and to lay the foundation of future research.

Keywords: active learning, flipped classroom, student-centered learning, teaching/learning strategies.

INTRODUCTION

Student engagement is one of the key elements in the student-centered learning environments (Shea, Hayes, Smith, Vickers, Bidjerano, & Pickett, 2012). Strayer (2012) reported that educators often have difficulties managing their finite classroom time to achieve an effective balance between lectures and active learning activities. Flipped classroom approach addresses these challenges by “inverting the classroom”, that is, interchanging "events that have traditionally taken place inside the classroom now take place outside the classroom and vice versa" (Foertsch, J., Moses, G., Strikwerda, J., & Litzkow, M., 2002). In a traditional class, the instructor lectures and students take notes as their learning activities. Students then complete homework post-class. In the flipped environment, in-class lectures are “flipped” with collaborative hands-on activities. A typical flipped classroom approach usually provides online video lectures and homework assignments prior to class sessions. Students are in charge of learning the material on their own pace and to prepare for the hands-on problem solving activities that will be carried out during class time (Bergmann, Overmyer, & Wile, 2012; Davies, Dean, & Ball, 2013; Foertsch, Moses, Strikwerda, & Litzkow, 2002; Fulton, 2012; Hughes, 2012; Zappe, Leicht, Messner, Litzinger, & Lee, 2009). Hands-on activities are designed to replace lectures in class and for students to become more active and interactive. Educators can commit more in-class time to monitor students’ performance and to provide adaptive and instant feedback to students (Fulton, 2012; Herreid & Schiller, 2013; Hughes, 2012).

Prior studies show different evaluations of flipped class approach. Some studies present benefits of positive impact on learning (e.g., Herreid and Schiller, 2013; Zappe et al., 2009) while some did not find significant difference between traditional and flipped approaches (Davies et al., 2013; Strayer, 2012). Many studies focused on the investigation of the strength and weakness of the flip classroom learning but provided little details of the specific flip class room activities. While contributions of these studies are not to be ignored, a demonstration of actual application of the flipped classroom design principles is needed to guide educators who are novice in this approach and to provide backdrop of research of flipped class approach.

Bergmann and Sams (2012) suggested a list of flip design considerations: support from administrators, support from IT department, time, and thoughtful educators. However, their guidance was limited to technological elements. Chen and colleagues (2014) proposed a FLIPPED model for the flipped learning design: flexible environments, learner-centered approach, intentional content, professional educators, progressive networking activities, engaging learning experiences, and diversified and seamless learning platforms. The progressive networking activities refer to "learning by doing" and/or "learning by networking" activities. The engaging learning experience focuses on tracking the effectiveness of students' outside class learning as well as their in-class learning. The diversified and seamless learning platforms are designed and operated to meet the criteria of the course domain knowledge for individualized, differentiated, personalized learning in a flexible, ubiquitous and seamless manner.
In this paper, we present a series of nested if statement activities in an introduction to Java class to demonstrate the design of a flipped class which includes the before, during, and after class activities/assignments. These are described in the following section. We then present future research plan.

**FLIPPED LEARNING ACTIVITIES**

Programming classes are perfect candidates for the flipped learning approach because those classes require a lot of hands-on activities in order to practice programming principles and to master a specific programming language. We designed a series of flipped learning activities targeting an introduction to Java class. We choose the nested if statement to demonstrate the use of flipped learning approach in programming class. We turned to prior instructional studies to select the pre, during, and post class activities for this class.

DeGrazia and colleagues (2012) reported that students who were supplied with optional lecture videos came to class much better prepared than when those who were only been given textbook readings assignments because college students don’t generally complete reading assignments (Sappington et al., 2002). Recommended by students, a required pre-class quiz on the lecture material to enforce the completion of before class activities is considered a “best practice” means and is conducted by many instructors. Zappe et al. (2009) concluded that students preferred interactive class time more than in-person lectures. Applying these suggestions and educational principles, the before class activities are designed mostly related to the Bloom’s (Krathwohl, 2002) lower-level learning such as remembering and comprehending. Both delivery and content of the before class activities are structured to help students acquire content knowledge and prepare them for the application of focus content in the face-to-face class. Instructors use offloaded content relevant to the topic of the class session for the before class activities.

**Before Class**

The before class activities contains various online modules including power point slides, a tutorial video, and a required quiz. Students can complete these modules in any sequence of their choices. Power point slides are used to explain the concepts. A flow chart (Figure 1) demonstrates the decision logic of the nested if statement (Figure 2). Students can either view the power point slides or read the text book to learn the basics and then take the required online quiz (Figure 4). This quiz is a “low-stake” quiz which constitutes only a small percentage of the final course grade.

We used Adobe Connect software to create the tutorial video. Video links are provided so students can access them throughout the semester. Students follow the instruction and demonstration of the tutorial to create a JAVA class, in this case, a payroll program. Figure 3 shows the model codes and steps shown in the tutorial video. There are build-in individual components to prevent plagiarism. In this case, students must replace “FML” shown in the example with their own initials. Students are required to follow the video, write a Java class similar to Figure 3, and submit the .java file by the deadline. All the before-class activities are due one day before the scheduled class time.
**Figure 1.** Flowchart of Nested if Statements

**Figure 2.** Nested if JAVA Statements of the Flowchart
Figure 3. Nested if Statements: Lab Tutorial
During Class

With the topic objectives in mind, the face-to-face class is designed for the application of the knowledge acquired by students from the online before-class activities using the built-in active learning strategies. Active learning strategies enable students to achieve higher levels of learning of Bloom's taxonomy (e.g., applying, analyzing, and synthesizing). There are many active learning strategies for educators to use in the face-to-face class. When designing the face-to-face component, it is important to choose only a few active learning strategies to use throughout the course rather than a different one for each class time. This will allow students to become familiar with the active learning strategy and avoid the risk of students focusing on the process of the strategy rather than the learning related to the content.

The best way to learn programming is by emulating sample programs. In the nested if statement topic, a coding activity was used (Figure 5). The instructor demonstrates and explains step by step and students follow. Students are encouraged to change the codes as long as their final produced program meets all the requirements. While student busy typing the Java program, the instructor helps students by trouble shooting simple common errors such as typos or other issues. After all students completed the Figure 5 Java class without any problem, the instructor then starts the discussion component to solicit feedback or reflections such as lessons learned. The instructor can also explain the concepts within this program and modify the program with different statements and quiz the students of the possible outcome after changes.

**Figure 4. Sample Quiz Questions**
After all students completed the Java program and all questions were answered, a pop quiz (Figure 6) can be used to assess students’ learning, specifically in checking whether students can apply the concepts/statements learned from the in-class coding as well as providing another opportunity for the instructor’s feedback.
Assessments are an integral part of the after-class component of the flipped classroom and should be aligned with the objectives of the offloaded content and the in-class activities. A lab assignment related to the nested if statement will be a good repetitive assignment to reinforce students' learning of this topic. The following description is the nested if statement assignment. Students need to analyze the problem, synthesize the concepts learned, and apply specific Java statements to complete this assignment. Figure 7 presents the solution of the after-class lab assignment.
Lab Assignment

1. Use BlueJ or other text editor to create a new Java class called `SalesDiscountFML.java`. In this assignment, you will use Scanner to capture user inputs and assign to variables. **Note: In terms of FML, F is your First Name initial, M is your Middle Name initial, and L is your Last Name initial.**

2. In the description part (the comments area), you should include your name, date created/modify this class, and the purpose of this class.

3. Import `java.util.Scanner` class to your Java class.

4. Declare the following variables:
   a. `String` variables: `nameFML`, `inputStringFML`
   b. `int` variable: `quantityFML`
   c. `double` variable: `salesPriceFML`, `totalSalesAmountFML`

5. Use in-line comments to explain your Java statements (see Lab Tutorial).

6. Write Java statements to read the user's input using `Scanner` (the console window) as the followings.
   a. create a `Scanner` object to get the user inputs,
   b. use the `Scanner` object to read the product’s name and assign to the `String` variable: `nameFML`.
   c. use the `Scanner` object to read the product's sales price and assign to the `double` variable: `salesPriceFML`.
   d. use the `Scanner` object to read product’s sales quantity and assign to the `int` variable: `quantityFML`.
   e. calculate the user's total sales amount and store the result to the `double` variable: `totalSalesAmountFML`.

```
# BlueJ: Terminal Window - temp
Options
What is the product name? LED Monitor
What is the sales price of the product? 249.5
What is the quantity of the product? 6
```

7. Write a nested if statement to determine the sales discount based on the total sales amount the following flow chart.

```
      totalSalesAmountFML > 1000
            v
        totalSalesAmountFML > 500
            v
       display the total sales amount with 15% discount
```

After running the main method of your `SalesDiscountFML` class, your results will be similar to the following screen shots.
You can run with a 5% discount scenario as the following screen shots:

Run with no discount scenario as the following screen shots:

8. Submit your `SalesDiscountFML.java` file before the deadline.
FUTURE RESEARCH

This paper proposed a series of detailed flipped learning activities for an introduction to Java class. The preliminary result shows that about 70% student prefer the flipped approach to the traditional. The authors plan to develop other series of activities for all the topics covered in the introduction to Java class. After developing the flipped learning activities, the authors intend to apply the flipped classroom approach to the introduction to Java class in the Fall of 2017 using all the activities designed. We then will assess students’ learning and survey students in the Java class for their reflections of this approach. We will continue collecting qualitative and quantitative data to get better insight of the flipped classroom approach.

REFERENCES


