United International University
Department of Computer Science and Engineering

CSE 4495: Software Testing and Quality Assurance
Final Examination : Spring 2023
Total Marks: 40 Time: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1 Unit Testing(15 marks)

1. Imagine that you plan to build a Ed-tech platform based out of UIU where different people can come together and offer to sell their particular skill by the means of selling courses. Other Users can signup into the system and subscribe to courses and watch the content.Now, consider the following class diagrams for a simplified version of the system. The three diagrams below represent the user, course and video contents of a course. Your job is to write unit test cases for the methods of this classes according to the given specifications.

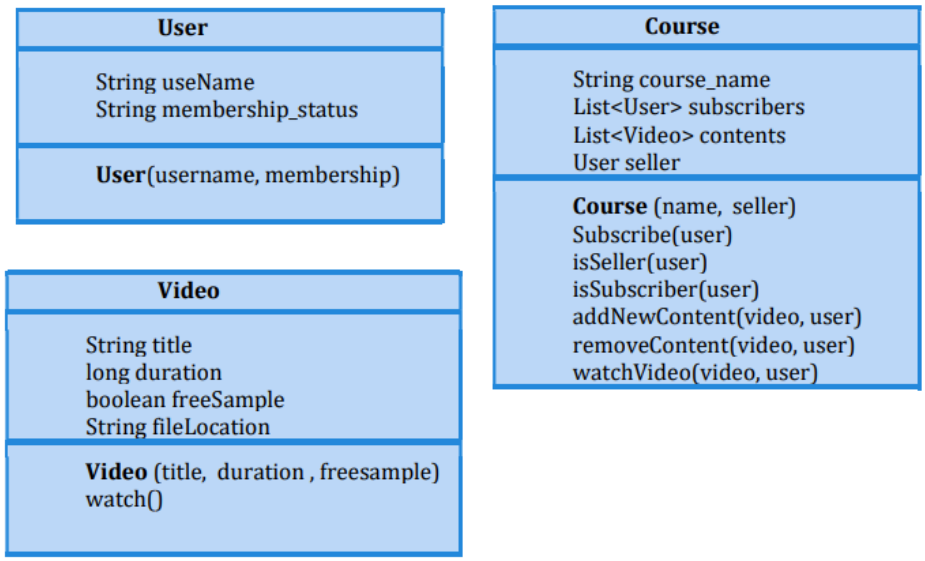

Figure 1: Class diagrams for Ques-1

(a) First, you need to test the *watchVideo* method from Course class. Each user has a membership status which is a String that can be either 'Free' or 'Premium'. If a user is a premium member he should be able to watch any video if he is subscribed to the course. if the user is a free member he can enjoy only the free sample videos. So, you need to write three unit tests to check the following three scenarios. $[4 + 4 + 4 = 12]$

- Test Case 1: Check if the *watchVideo* works properly when it is called by a 'premium' user who is subscribed to the course.Set up a course and add two videos(one is a free sample another one is not) to it. The user under test should be able to watch both videos(i.e. watchvideo() returns *true*).
- Test Case 2: Test the same method for a user with a 'Free' user. Again set up a course with two videos like test case 1. This time however, the user is able to only watch the freesample video.
- Test Case 3: Now run test case 2 without subscribing to the course. The system should throw *NotaSubscriber.class* error with message *"You cannot add a watch a video without subscribing first"* .

Devise executable test cases for all of the above specifications for this method in the *JUnit* notation.

(b) Now you need to conduct a performance test on the same *watchVideo* method which returns true if the video is able to successfully and returns false otherwise. Write an executable test case in JUnit notation that tests whether the method is able to load the video within a 3 second time limit. [3]

2 Model Based Testing (10 marks)

2. (a) Explain why transition coverage is not enough to uncover all the faults in a finite state model. Show with an illustrative example that single transition path coverage criteria can uncover faults in a model that may be otherwise overlooked even if we achieve transition coverage. [2]

(b) UIU cafeteria has recently implemented an mobile app to control and organize lunch time traffic. From now on, during the lunch hours to buy something students will first have to raise an e-ticket from the app. Each e-ticket has a serial number. After raising a ticket the customer will either choose the daily set menu or choose to create his own meal.If the customer goes for the set menu he/she can immediately checkout with the food and collect their food by paying for it on cafe stall no. 1. On the other hand if a customer chooses to create his/her own dish, they will be redirected to the active menu page where one can add or remove items to customize their own meal. After they finish customizing they will be inserted into a queue until their ticket number is announced on the app. Once notified the customer can then collect their meal by completing payment from stall no.2. You must design a state machine that simulates a customers journey through the mentioned app. [3]

(c) Consider the following class implementing a simple game which has two levels. A player starts his journey from game menu with four lives. The methods of this class are described below - [5]

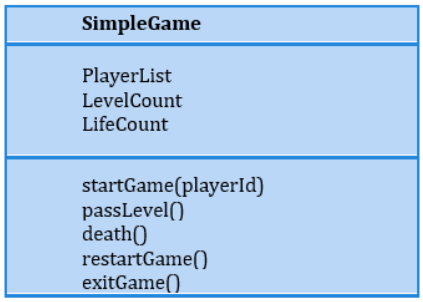

Figure 2: Class diagram for Ques-2(b)

- ***startGame()*** – This method can only be called from the 'game menu' screen. This call initiates a new game by rendering the first level to the player and inserts the player to **PlayerList**.
- ***passLevel()*** – Advances player to the next level and increments the **LevelCount** . If called from the second level renders the 'Game Over' Screen
- ***death()*** – Invoked when the player dies, decrements **LifeCount** by one and restarts the level. If the player had a single life left then he will be shown the 'Game Over' Screen.
- ***restartGame()*** – can be called from any level. Resets the **LevelCount**, **LifeCount** and returns the player to the first level.
- ***exitGame()*** – can be called from any level or the 'Game Over' Screen. Resets the **LevelCount**, **LifeCount** and returns the player to the 'game menu'. Also removes the player from active **PlayerList**.

Now identify the states and design a Finite State Model for the above class with all the relevant labeled transitions. [Hint : Think of each level as a state].

3 Structural Testing (15 marks)

(a)

```
int findMax(int a, int b, int c) {
    int temp;

    if (a>b) temp=a;
    else temp=b;
    if (c>temp)
    temp = c;

    return temp;
}
```

i. Draw the control flow graph for the program ***findMax***, which finds the maximum of three integers. [3]

ii. Develop test input that will provide statement coverage. [2]

iii. Develop test input that will provide branch coverage. [2]

iv. Develop test input that will provide path coverage. [3]

(b)

```
int binarySearch(int arr[], int x){
   int l = 0, r = arr.length - 1;
   while (l <= r) {
       int m = l + (r - l) / 2;

       if (arr[m] == x) return m;

       if (arr[m] < x) l = m + 1;

       else  r = m - 1;
   }
   return -1;
}
```

i. Draw the control flow graph for the ***binarySearch*** method. [3]

ii. In your CFG highlight the paths exercised by the following test inputs. [$2 \times 1 = 2$]

A. *binarySearch([-5,10],10)*

B. *binarySearch([-5,10],5)*