UNITED INTERNATIONAL UNIVERSITY (UIU)

Dept. of Computer Science & Engineering

Trimester: Summer 2022

Course No: CSE 4495 Title: Software Quality Assurance and Testing

Section: A Class Test-4

Time: 25 minutes **Marks: 20**

Name		**ID**	

1. Consider a simple microwave controller modeled as a finite state machine using the following state variables: **[3+4+3 = 10]**
   - Door: {Open, Closed} -- sensor input indicating state of the door
   - Button: {None, Start, Stop} -- button press (assumes at most one at a time)
   - Timer: 0...999 -- (remaining) seconds to cook
   - Cooking: Boolean -- state of the heating element

   Formulate the following informal requirements in LTL:
   a) It will always be true that, if the microwave is cooking it will finish cooking at some point.
   b) It will always be true that, cooking cannot start until the button is in start state and the door is closed and the remaining cook time is not zero.
   c) It will always be true that when the stop button is pressed the microwave will stop cooking.

2. 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 in this class is described below **[10]**

SimpleGame
PlayerList LevelCount LifeCount
startGame(playerId) passLevel() death() restartGame() exitGame()

- **startGame()** – This method can only be called from the game menu 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 **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].