Lab9

W.A.C Fernando (26545)

1.

```java
abstract class Container {
    private double radius;
    private double height;

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getHeight() {
        return height;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    public abstract double calculateVolume();
}

//Java Definition of class CylindricalContainer
public class CylindricalContainer extends Container {

    // Constructor
    public CylindricalContainer(double radius, double height) {
        setRadius(radius);
        setHeight(height);
    }

    // Implementing the abstract method to calculate the volume of the cylindrical container
    @Override
    public double calculateVolume() {
        // Volume = PI * Radius^2 * Height
        return Math.PI * Math.pow(getRadius(), 2) * getHeight();
    }

    // You can add additional methods specific to the CylindricalContainer class if needed
}
```

```
// object from the "CylindricalContainer" class and displaying the volume
public class Main {
   public static void main(String[] args) {
      // Create a CylindricalContainer object with radius = 3 and height = 5
      CylindricalContainer cylindricalContainer = new CylindricalContainer(3, 5

      // Calculate and display the volume
      double volume = cylindricalContainer.calculateVolume();
      System.out.println("Volume of the Cylindrical Container: " + volume);
   }
}
```

2.

```
// Interface for PlayerController abstraction
interface PlayerController {
   void moveUp();
   void moveDown();
   void moveLeft();
   void moveRight();
}

// Implementation of the PlayerController for the Life game
class LifePlayerController implements PlayerController {

   @Override
   public void moveUp() {
      System.out.println("Player moves up.");
      // Implement the logic to update the player's position for moving up.
   }

   @Override
   public void moveDown() {
      System.out.println("Player moves down.");
      // Implement the logic to update the player's position for moving down.
   }

   @Override
   public void moveLeft() {
      System.out.println("Player moves left.");
      // Implement the logic to update the player's position for moving left.
   }

   @Override
   public void moveRight() {
      System.out.println("Player moves right.");
      // Implement the logic to update the player's position for moving right.
   }
}
```

```
public class Main {
    public static void main(String[] args) {
        // Create the player controller
        PlayerController playerController = new LifePlayerController();

        // Simulate the game by pressing keys to move the player
        playerController.moveUp();
        playerController.moveLeft();
        playerController.moveDown();
        playerController.moveRight();
    }
}
```