

Question 01:

Container.java (abstract class):

```
public abstract class Container {  
    protected double volume;  
  
    public abstract double calculateVolume();  
}
```

CylindricalContainer.java (subclass of Container):

```
public class CylindricalContainer extends Container {  
    private double height;  
    private double radius;  
  
    public CylindricalContainer(double radius, double height) {  
        this.radius = radius;  
        this.height = height;  
    }  
  
    public double getHeight() {  
        return height;  
    }  
  
    public void setHeight(double height) {  
        this.height = height;  
    }  
  
    public double getRadius() {  
        return radius;  
    }  
  
    public void setRadius(double radius) {
```

```

        this.radius = radius;
    }

    @Override
    public double calculateVolume() {
        return Math.PI * radius * radius * height;
    }
}

```

TestCylindricalContainer.java (for testing the CylindricalContainer class):

```

public class TestCylindricalContainer {
    public static void main(String[] args) {
        CylindricalContainer container = new CylindricalContainer(5.0, 10.0);
        double volume = container.calculateVolume();
        System.out.println("Volume of Cylindrical Container: " + volume);
    }
}

```

Question 02:

PlayerController.java (interface):

```

public interface PlayerController {
    void moveUp();
    void moveDown();
    void moveLeft();
    void moveRight();
}

```

Then, in the Life game implementation, you can have a class that implements the PlayerController interface to control the player's movements:

```

import java.util.Scanner;

public class LifeGame implements PlayerController {

```

```
@Override
```

```
public void moveUp() {  
    System.out.println("Player moves UP.");  
}
```

```
@Override
```

```
public void moveDown() {  
    System.out.println("Player moves DOWN.");  
}
```

```
@Override
```

```
public void moveLeft() {  
    System.out.println("Player moves LEFT.");  
}
```

```
@Override
```

```
public void moveRight() {  
    System.out.println("Player moves RIGHT.");  
}
```

```
public static void main(String[] args) {
```

```
    LifeGame game = new LifeGame();  
    Scanner scanner = new Scanner(System.in);
```

```
    // Assuming that the player controls are through keyboard inputs
```

```
    System.out.println("Press W to move UP.");  
    System.out.println("Press S to move DOWN.");  
    System.out.println("Press A to move LEFT.");  
    System.out.println("Press D to move RIGHT.");
```

```
    while (true) {
```

```
char input = scanner.next().charAt(0);
switch (input) {
    case 'W':
    case 'w':
        game.moveUp();
        break;
    case 'S':
    case 's':
        game.moveDown();
        break;
    case 'A':
    case 'a':
        game.moveLeft();
        break;
    case 'D':
    case 'd':
        game.moveRight();
        break;
    default:
        System.out.println("Invalid input. Try again.");
        break;
}
}
}
```