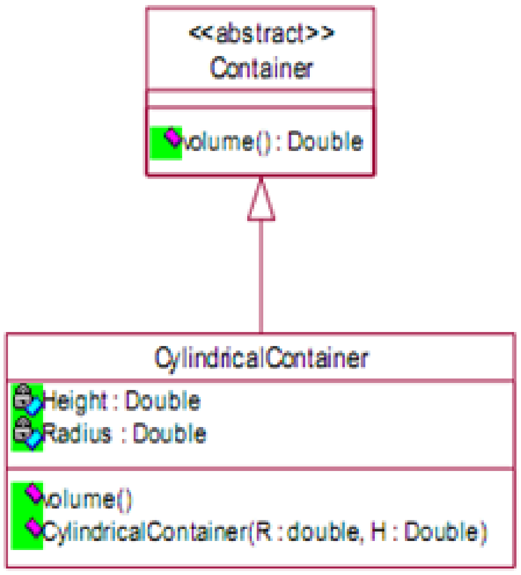
Question 01



The Volume of a Cylinder can be found with the following formula:

Volume = PI * Radius*Radius*Height   where PI=3.14159

It is required to map the above class diagram to Java code.

Note: Container is an abstract class.
Height & Radius are private variables
All the methods are public

(i) Write down the Java definition of class container

```
public abstract class container
{
        public abstract double volume ();
 }
```

(ii) Write the Java Definition of class CylindricalContainer. (Implement the Methods)

```
public class CylindericalContainer extends container
{
   //variables
   private double height,radius;

   //methods
   public CylindericalContainer (double radius,double height)
   {
      this.radius=radius;
      this.height=height;
   }

    public double volume ()
   {
        return 3.14159f*radius*height*radius;
    }
 }
```

(iii) Create an object from CylindricalContainer and display the volume.

```
public class AbstractExample
{
    public static void main (String [] args)
    {
        CylindericalContainer c1=new CylindericalContainer(8.75,12.50);

        System.out.println("Volume is "+c1.volume());
    }
 }
```

Question 02

A Student wants to create a game called “Life”, ‘life’ is a RPG game in which a player can move up, down, left & Right. In order to implement this game assume that you need to create an abstraction of the player controllers. Make sure to print the directions of the player when keys are pressed.

1. Implement a new player extended from movements. When each movement method is executed print out the direction that the player has moved.

```
public abstract class Movements
{
    public abstract void up();
    public abstract void down();
    public abstract void left();
    public abstract void right();
}

public class Player extends Movements
{
    int Sp;

    Player(int i)
    {
        this.Sp=i;
    }

    @Override
    public void up()
    {
        System.out.println("Player moved up by "+this.Sp+" spaces.");
    }
    @Override
    public void down()
    {
       System.out.println("Player moved down by "+this.Sp+" spaces.");
    }

    @Override
    public void left()
    {
       System.out.println("Player moved left by "+this.Sp+" spaces.");
    }

    @Override
    public void right()
    {
       System.out.println("Player moved right by "+this.Sp+" spaces.");
    }

}
```

II. Assume that there are three types of players, those who move in regular directions and opposite directions when a movement is performed and others who jump by 5 spaces at a time except when crouched, where he moves by only two spaces. Those who move in the opposite direction move 2 spaces at a time. Implement separate players and print their movements.

```
public class Opposites extends Movements
{
    int Sp;

    Opposites(int i)
    {
       this.Sp=i;
    }

    @Override
    public void up()
    {
       System.out.println("Player moved down by "+this.Sp+" spaces.");
    }

    @Override
    public void down()
    {
       System.out.println("Player moved up by "+this.Sp+" spaces.");
    }

    @Override
    public void left()
    {
```

```java
        System.out.println("Player moved right by "+this.Sp+" spaces.");
    }

    @Override
    public void right()
    {
        System.out.println("Player moved left by "+this.Sp+" spaces.");
    }

}


public class Jumpers extends Player
{
    public Jumpers(int i)
    {
        super(i);
    }

    public void CrouchDown()
    {
        System.out.println("Player moved down by "+(this.Sp-3)+" spaces.");
    }

    public void CrouchUp()
    {
        System.out.println("Player moved up by "+(this.Sp-3)+" spaces.");
    }
}



public class Lab9
{
    public static void main(String[] args)
    {
        Player p1=new Player(2);
        p1.up();
        p1.down();
        p1.left();
        p1.right();

        Jumpers j1=new Jumpers(5);
        j1.up();
        j1.down();
        j1.right();
        j1.left();

        Opposites o1=new Opposites(1);
        o1.up();
        o1.down();
        o1.right();
        o1.left();
    }
}
```