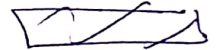


Lab Section #:	
<input checked="" type="radio"/> Lab Sec 1	TR
<input type="radio"/> Lab Sec 2	MW

[Q1 25%]: [20 minutes]

What is the output of the following code?



25

	Code	Output
1	<pre>public class Test { public static void main(String[] args) { int[] list1 = { 1, 2, 3 }; int[] list2 = list1; list2[1] = 10; System.out.println(list1[1]); } }</pre> <p style="text-align: right;">list2 list 1 ↓ ↓ 1 10 3</p>	10
2	<pre>public class Test { public static void main(String[] args) { int[] x = { 1, 2 }; int i = 1; m(i, x); System.out.print(i + "," + x[0]); } public static void m(int i, int[] list) { i = 9; list[0] = 9; } }</pre>	1,9
3	<pre>public class Test { public static void main(String[] args) { int[][] x = { { 2, 1 }, { 1, 7, 1 } }; System.out.println(m(x[1])); } public static int m(int[] m) { int result = 0; for (int i = 0; i < m.length; i++) result += m[i]; return result; } }</pre>	9

0 | 2 | 1
1 | 1 | 7 | 1

4	<pre>public class Test { public static void main(String[] args) { A a1 = new A(); A a2 = new A(); System.out.println(a1.equals(a2)); } } class A { int x = 1; }</pre>	<p>False</p> <p>Because equals here is not overridden, so it compares references</p>
5	<pre>public class Test { public static void main(String[] args) { A a1 = new A(); System.out.print(a1.j); A a2 = new A(); System.out.print(" " + a2.j); } } class A { int i = 1; static int j = 1; A() { i++; j++; } }</pre> <p style="margin-left: 200px;">j = 2 i = 2</p>	<p>2 3</p>
6	<pre>import java.util.; public class Test { public static void main(String[] args) { ArrayList<Integer> list = new ArrayList<>(); list.add(1); list.add(2); list.add(3); list.remove(1); System.out.println(list); } }</pre> <p style="margin-left: 200px;">list.toString()</p>	<p>1 3</p>
7	<pre>public class Test { public static void main(String[] args) { new Person().printPerson(); new Student().printPerson(); } } class Student extends Person { private String getInfo() {return "Student";} } class Person { private String getInfo() {return "Person";} public void printPerson() { System.out.println(getInfo()); } }</pre> <p style="margin-left: 200px;">super method</p>	<p>Person Person Student Person</p>

1 | 1 | 2
2 | 2 | 3

private method

```

8 public class Test {
    public static void main(String[] args) {
        Object circle1 = new Circle();
        Object circle2 = new Circle();
        System.out.println(circle1.equals(circle2));
    }
}

class Circle {
    double radius = 1;

    public boolean equals(Circle circle) {
        return this.radius == circle.radius;
    }
}

```

~~True~~
False

since this method takes Circle object and Circle 2 has an object reference, then it is not passed to this method, it is passed to the super's method (Object) that compares reference.

default

```

9 public class Test {
    public static void main(String[] args) {
        new A();
        new B();
    }
}

class A {
    int i = 7;
    public A() {
        setI(20);
        System.out.println("i from A is " + i);
    }
    public void setI(int i) {
        this.i = 2 * i;
    }
}

class B extends A {
    public B() {
        System.out.println("i from B is " + i);
    }
    public void setI(int i) {
        this.i = 3 * i;
    }
}

```

i from A is 40
i from A is 40
i from B is 60
i from A is 60
i from B is 60

40
 $2 \times 20 = 40$

super
 $3 \times 20 = 60$

```

10 public class Test {
    public static void main(String[] args) {
        String s = "Java";
        StringBuilder builder = new StringBuilder(s);
        change(s, builder);
        System.out.println(s);
        System.out.println(builder);
    }

    private static void change(String s, StringBuilder sb) {
        s = s + " and HTML";
        sb.append(" and HTML");
    }
}

```

Java
Java and HTML

[Q3 20%] [15 minutes]

For each of the following programs, determine the error(s) and highlight/explain them.

	Code	Error
1	<pre> class Test { private double i; public Test(double i) { this.t(); this.i = i; } public Test() { System.out.println("Default constructor"); <u>this(1);</u> } public void t() { System.out.println("Invoking t"); } } </pre>	<p>Invoking the constructor by another constructor must be the first sentence in the block (Syntax Error)</p>
2	<pre> public class Test { public static void main(String[] args) { int n = 2; <u>xMethod(n);</u> System.out.println("n is " + n); } private void xMethod(int n) { n++; } } </pre>	<p>A static method (main) can't invoke an instance method (Syntax Error)</p>
3	<pre> public class Test { int x; public Test(String t) { System.out.println("Test"); } public static void main(String[] args) { <u>Test test = new Test();</u> System.out.println(test.x); } } </pre>	<p>A no-arg constructor is invoked but not implemented. (Syntax Error)</p>
4	<pre> class Test { public static void main(String[] args) { <u>String s;</u> System.out.println("s is " + s); } } </pre>	<p>s is a local variable, thus it must be initialized before using it (Syntax Error)</p>
5	<pre> package p1; public class A { protected int i; void m1() {} } package p2; public class B extends A { public void m2() { System.out.println(i); <u>m1();</u> } } </pre>	<p>B cannot access m1 as it is a default method that cannot be accessed outside the package (Syntax Error)</p>

20

default

subclass

Error

[Q4 40%] [30 minutes]

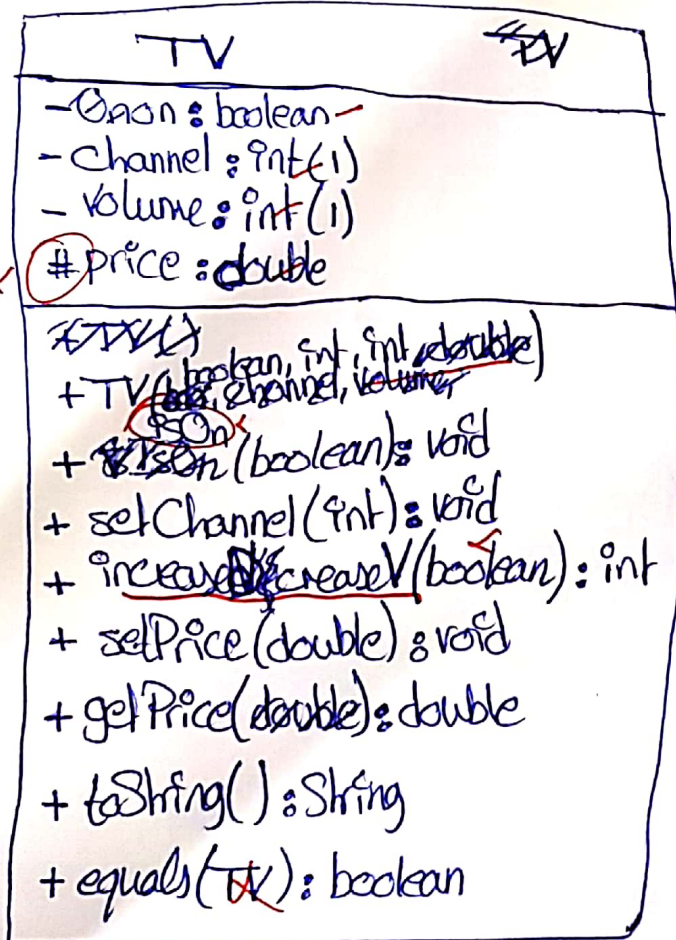
a) You are asked to implement a TV (television) class. The TV behaves as follows:
 The television remembers (knows) whether it is **on or off**, the **channel** number it is currently set to (channel number is an integer between **1 and 120**), the **volume** it is currently set to (volume is an integer between **1 and 10**), and the price (positive double).

A TV instance can be **only** constructed using 3 arguments (**on/off**, **channel**, and **volume**).

The following are the actions that can be performed on a TV instance:

- The TV can be turned **on or off**.
- The channel can be set to any channel between 1 and 120. Setting the channel to another integer causes no change to the TV.
- The volume can be increased by 1; But if the volume is already at maximum, then no change occurs.
- The volume can be decreased by 1; But if the volume is already at minimum, then no change occurs.
- A set and get methods for the price.
- Override **toString** method to return TV instance specific information.
- Override **equals** method to compare TVs based on price.

Draw a UML diagram for the TV class and then implement the class:



4

```

public class TV {
    private boolean on;
    private int channel;
    private int volume;
    private double price;
    protected

    public TV(boolean on, int channel, int volume) {
        this.on = on;
        this.channel = channel;
        this.volume = volume;
    }

    public void setIsOn(boolean b) {
        if (b == 1)
            on = b;
    }

    public void setChannel(int ch) {
        channel = ch;
    }

    public void setPrice(double price) {
        price = price;
    }

    public int increaseVolume() {
        volume++;
    }

    public int decreaseVolume() {
        volume--;
    }

    public boolean isOn() {
        return on;
    }

    public double getPrice() {
        return price;
    }

    public String toString() {
        return "TV: on=" + on + ", channel=" + channel + ", volume=" + volume + ", price=" + price;
    }

    public boolean equals(Object obj) {
        if (obj instanceof TV) {
            TV tv = (TV) obj;
            return price == tv.price;
        }
        return false;
    }
}
    
```

Question 4 solution :

```
package FormFive;

public class QuestionFour {

    private boolean isOn;
    private int channelNumber = 1;
    private int volume = 1;
    private double price;
    QuestionFour(boolean isOn,int channelNumber,int volume){
        this.isOn = isOn;
        if(!(channelNumber >= 120) && !(channelNumber < 1))
            this.channelNumber = channelNumber;

        if(volume > 10) {
            this.volume = 10;
        }
        else if(volume < 1 ) {
            this.volume = 1;
        }
        else {
            this.volume = volume;
        }

    }
    public boolean isOn() {
        return isOn;
    }
    public void setOn(boolean isOn) {
        this.isOn = isOn;
    }
    public int getChannelNumber() {
        return channelNumber;
    }
    public void setChannelNumber(int channelNumber) {
        if(!(channelNumber < 1) || !(channelNumber > 120))
            this.channelNumber = channelNumber;
    }
    public int getVolume() {
        return volume;
    }
    public void setVolumeUp() {
        volume = (volume == 10) ? 10 : volume++;
    }
    public void setVolumeDown() {
        volume = (volume == 1) ? 1 : volume--;
    }
}
```

```
public double getPrice() {
    return price;
}
public void setPrice(double price) {
    this.price = price;
}
@Override
public String toString() {
    return "Im too tired to do that";
}
}
```