| Lab Section #: | |
|---|---|
| ☑ Lab Sec 1 | TR |
| ○ Lab Sec 2 | MW |

25

**[Q1 25%]: [20 minutes]**

What is the output of the following code?

| | Code | Output |
|---|---|---|
| 1 | public class Test {<br>        public static void main(String[] args) {<br>                int[] list1 = { 1, 2, 3 };<br>                int[] list2 = list1;<br>                list2[1] = 10;<br>                System.*out*.println(list1[1]);<br>        }<br>}<br><br>list2 list1<br>1 \| ~~2~~10 \| 3 | 10 |
| 2 | public class Test {<br>        public static void main(String[] args) {<br>                int[] x = { 1, 2 };<br>                int i = 1;<br>                *m*(i, x);<br>                System.*out*.print(i + "," + x[0]);<br>        }<br><br>        public static void m(int i, int[] list) {<br>                i = 9;<br>                list[0] = 9;<br>        }<br>} | ~~8~~ 1 , 9 |
| 3 | public class Test {<br>        public static void main(String[] args) {<br>                int[][] x = { { 2, 1 }, { 1, 7, 1 } };<br>                System.*out*.println(*m*(x[1]));<br>        }<br><br>        public static int m(int[] m) {<br>                int result = 0;<br>                for (int i = 0; i < m.length; i++)<br>                        result += m[i];<br>                return result;<br>        }<br>} | 9 |

0 | 2 1
1 | 1 7 1

Scanned by CamScanner

4

```
 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;
}
```

False

Because equals here is not overridden, so it compares references

5

```
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++;
        }
}
```

j=2
i=2

2 3

*6

1 | 2
2 | 3

```
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);
        }
}
```

list.toString()

1 3

7

```
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());
        }
}
```

private method

super method

Person
~~Person~~
Student
Person

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;
        }
}
```

default

~~Fase~~

False

Since this method takes Circle object and Circle2 has an object reference, then it is not passed to this method, it is passed to the super's method (object) that compares reference

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;
        }
}
```

010

super()

$20*2=40$

$2*20=40$

$3*20=60$

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

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

20

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

```
TV
-------------------------------------------
- on : boolean
- channel : int(1)
- volume : int(1)
# price : double
-------------------------------------------
+ TV(boolean, int, int, double)
+ isOn(boolean) : void
+ setChannel(int) : void
+ increaseDecreaseV(boolean) : int
+ setPrice(double) : void
+ getPrice(double) : double
+ toString() : String
+ equals(TV) : boolean
```

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

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

  public void isOn(boolean b){
      on = b; }

  public void setChannel(int ch){
      if (ch >= 1 && ch <= 120)
         channel = ch;
      else
       System.out.println("The entered number channel is not valid");
  }
```

4

6

## 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";
	}

}
```