Aim:

Write a Java program to demonstrate the usage of <code>isAlive()</code> and <code>join()</code> methods in threads.

Write a class JoinThreadDemo with a main() method that creates and executes two instances of Counter class which implements Runnable interface.

Let the Counter class take a **String** argument **name** and let its run() method print that message for **10** times along with the current count as given below:

```
System.out.println(name + " : " + i);
```

The JoinThreadDemo.main() method should perform the below tasks in the given order:

- 1. Create the first instance of thread as t1 with an instance of Counter class using "Spain" as the argument.
- 2. Create the second instance of thread as t2 with an instance of Counter class using "UAE" as the argument.
- 3. Print the isAlive() status of t1 as: "t1 before start t1.isAlive(): " + t1.isAlive().
- 4. Print the (isAlive()) status of (t1) as: "t2 before start t2.isAlive(): " + t2.isAlive().
- 5. Start t1 and t2 threads respectively.
- 6. Print a message to the console as: "started t1 and t2 threads".
- 7. Print the <code>isAlive()</code> status of <code>t1</code> as : "t1 after start t1.isAlive() : " + t1.isAlive().
- 8. Invoke the [join()] method on [t2].
- 9. Print the (isAlive()) status of (t1) as: "t2 after start t2.isAlive(): " + t2.isAlive().

Note: Please don't change the package name.

Source Code:

```
q11350/JoinThreadDemo.java
```

```
package q11350;
public class JoinThreadDemo {
  public static void main(String[] args) throws InterruptedException {
      Thread t1 = new Thread(new Counter("Spain"));
      Thread t2 = new Thread(new Counter("UAE"));
      System.out.println("t1 before start t1.isAlive() : "+t1.isAlive());
      System.out.println("t2 before start t2.isAlive() : "+t2.isAlive());
     t1.start();
     t2.start();
      System.out.println("started t1 and t2 threads");
      System.out.println("t1 after start t1.isAlive() : "+t2.isAlive());
     t2.join();
      System.out.println("t2 after start t2.isAlive() : "+t2.isAlive());
   }
}
class Counter implements Runnable {
  private String name;
  public Counter(String name) {
     this.name = name;
   public void run() {
      for (int i = 0; i < 3; i++) {
```

```
System.out.println(name + " : " + i);
     }
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
t1 before start t1.isAlive() : false
t2 before start t2.isAlive() : false
started t1 and t2 threads
t1 after start t1.isAlive() : true
UAE : 0
UAE : 1
UAE : 2
t2 after start t2.isAlive() : false
Spain : 0
Spain : 1
Spain : 2