

Week 06 : Programming Assignment 4

Due on 2025-03-06, 23:59 IST

Execution of two or more threads occurs in a random order. The keyword `synchronized` in Java is used to control the execution of threads in a specific sequence. In the following program, some numbers are expected to be printed. Use the `synchronized` keyword appropriately to ensure that the program prints the output in the following order:

-----OUTPUT-----  
5  
10  
15  
20  
25  
100  
200  
300  
400  
500  
-----

Private Test cases used for evaluation

Test Case 1

Input	Expected Output	Actual Output	Status
	5\n10\n15\n20\n25\n100\n200\n300\n400\n500	5\n10\n15\n20\n25\n100\n200\n300\n400\n500	Passed

The due date for submitting this assignment has passed.  
1 out of 1 tests passed.  
You scored 100.0/100.

Assignment submitted on 2025-03-05, 19:46 IST

Your last recorded submission was :

```
1 class Execute{
2     synchronized void print(int n){
3         for (int i = 1; i <= 5; i++){
4             System.out.println(n * i);
5             try {
6                 Thread.sleep(500);
7             } catch (InterruptedException e) {
8                 System.out.println(e);
9             }
10        }
11    }
12 } // Ending Execute class
13
14 class Thread1 extends Thread{
15     Execute t;
16     Thread1(Execute t){
17         this.t=t;
18     }
19     public void run(){
20         t.print(5);
21     }
22 }
23
24 class Thread2 extends Thread{
25     Execute t;
26     Thread2(Execute t){
27         this.t=t;
28     }
29     public void run(){
30         t.print(100);
31     }
32 }
33
34 public class W06_P4{
35     public static void main(String args[]){
36         Execute ob = new Execute(); //only one object
37         Thread1 t1=new Thread1(ob);
38         Thread2 t2=new Thread2(ob);
39         t1.start();
40         t2.start();
41     }
42 }
43 }
```

Sample solutions (Provided by instructor)

```
1 class Execute{
2     // Just add 'synchronized' in the method
3     synchronized void print(int n){
4         for(int i=1;i<=5;i++){
5             System.out.println(n*i);
6             try{
7                 Thread.sleep(400);
8             }catch(Exception e){
9                 System.out.println(e);
10            }
11        }
12    }
13 } // Ending Execute class
14
15 class Thread1 extends Thread{
16     Execute t;
17     Thread1(Execute t){
18         this.t=t;
19     }
20     public void run(){
21         t.print(5);
22     }
23 }
24
25 class Thread2 extends Thread{
26     Execute t;
27     Thread2(Execute t){
28         this.t=t;
29     }
30     public void run(){
31         t.print(100);
32     }
33 }
34
35 public class W06_P4{
36     public static void main(String args[]){
37         Execute ob = new Execute(); //only one object
38         Thread1 t1=new Thread1(ob);
39         Thread2 t2=new Thread2(ob);
40         t1.start();
41         t2.start();
42     }
43 }
```

