**How to perform single task by multiple threads?**

[**https://www.javatpoint.com/multitasking-in-multithreading**](https://www.javatpoint.com/multitasking-in-multithreading)

# How to perform single task by multiple threads?

|  |
| --- |
| If you have to perform single task by many threads, have only one run() method.For example: |

***Program of performing single task by multiple threads***

1. **class** TestMultitasking1 **extends** Thread{
2. **public** **void** run(){
3. System.out.println("task one");
4. }
5. **public** **static** **void** main(String args[]){
6. TestMultitasking1 t1=**new** TestMultitasking1();
7. TestMultitasking1 t2=**new** TestMultitasking1();
8. TestMultitasking1 t3=**new** TestMultitasking1();
10. t1.start();
11. t2.start();
12. t3.start();
13. }
14. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMultitasking1)

Output:task one

task one

task one

***Program of performing single task by multiple threads***

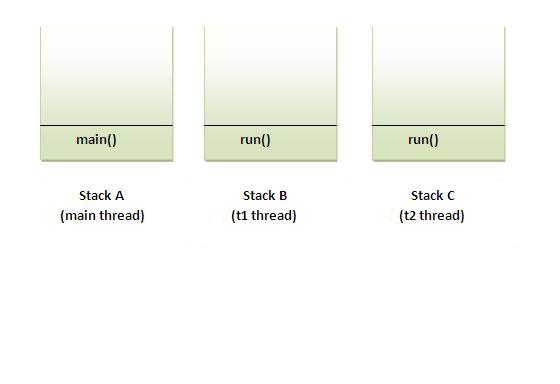
1. **class** TestMultitasking2 **implements** Runnable{
2. **public** **void** run(){
3. System.out.println("task one");
4. }
6. **public** **static** **void** main(String args[]){
7. Thread t1 =**new** Thread(**new** TestMultitasking2());//passing annonymous object of TestMultitasking2 class
8. Thread t2 =**new** Thread(**new** TestMultitasking2());
10. t1.start();
11. t2.start();
13. }
14. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMultitasking2)

Output:task one

task one

#### Note: Each thread run in a separate callstack.



### How to perform multiple tasks by multiple threads (multitasking in multithreading)?

|  |
| --- |
| If you have to perform multiple tasks by multiple threads,have multiple run() methods.For example: |

***Program of performing two tasks by two threads***

1. **class** Simple1 **extends** Thread{
2. **public** **void** run(){
3. System.out.println("task one");
4. }
5. }
7. **class** Simple2 **extends** Thread{
8. **public** **void** run(){
9. System.out.println("task two");
10. }
11. }
13. **class** TestMultitasking3{
14. **public** **static** **void** main(String args[]){
15. Simple1 t1=**new** Simple1();
16. Simple2 t2=**new** Simple2();
18. t1.start();
19. t2.start();
20. }
21. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMultitasking3)

Output:task one

task two

### Same example as above by annonymous class that extends Thread class:

***Program of performing two tasks by two threads***

1. **class** TestMultitasking4{
2. **public** **static** **void** main(String args[]){
3. Thread t1=**new** Thread(){
4. **public** **void** run(){
5. System.out.println("task one");
6. }
7. };
8. Thread t2=**new** Thread(){
9. **public** **void** run(){
10. System.out.println("task two");
11. }
12. };

15. t1.start();
16. t2.start();
17. }
18. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMultitasking4)

Output:task one

task two

### Same example as above by annonymous class that implements Runnable interface:

***Program of performing two tasks by two threads***

1. **class** TestMultitasking5{
2. **public** **static** **void** main(String args[]){
3. Runnable r1=**new** Runnable(){
4. **public** **void** run(){
5. System.out.println("task one");
6. }
7. };
9. Runnable r2=**new** Runnable(){
10. **public** **void** run(){
11. System.out.println("task two");
12. }
13. };
15. Thread t1=**new** Thread(r1);
16. Thread t2=**new** Thread(r2);
18. t1.start();
19. t2.start();
20. }
21. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMultitasking5)

Output:task one

task two