Name – **Kunal Dhanawade**

PRN – **230944520039**

1. create a multi-threaded application by using “extends Thread " method. Create 2 threads. they should display characters from A to J.

|  |
| --- |
| **class** Thread1 **extends** Thread{  @Override  **public** **void** run() {  // **TODO** Auto-generated method stub  **for** (**char** c='A'; c<='J'; c++) {  System.***out***.println(Thread.*currentThread*()+"\t"+c);  }  }  }  **public** **class** Demo {  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  Thread1 t1 = **new** Thread1();  Thread1 t2 = **new** Thread1();    t1.setName("First");  t2.setName("Second");    t1.start();  t2.start();  }  } |
| Thread[Second,5,main] A  Thread[Second,5,main] B  Thread[First,5,main] A  Thread[Second,5,main] C  Thread[First,5,main] B  Thread[First,5,main] C  Thread[Second,5,main] D  Thread[First,5,main] D  Thread[First,5,main] E  Thread[First,5,main] F  Thread[First,5,main] G  Thread[First,5,main] H  Thread[First,5,main] I  Thread[First,5,main] J  Thread[Second,5,main] E  Thread[Second,5,main] F  Thread[Second,5,main] G  Thread[Second,5,main] H  Thread[Second,5,main] I  Thread[Second,5,main] J |

1. above program using "implements Runnable" method.

|  |
| --- |
| **class** Thread1 **implements** Runnable{    **public** **void** run() {  // **TODO** Auto-generated method stub  **for** (**char** c='A'; c<='J'; c++) {  System.***out***.println(Thread.*currentThread*()+"\t"+c);  }  }  }  **public** **class** Demo {  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  Thread1 th = **new** Thread1();    Thread t1 = **new** Thread(th);  Thread t2 = **new** Thread(th);    t1.setName("First");  t2.setName("Second");    t1.start();  t2.start();  }  } |
| Thread[First,5,main] A  Thread[First,5,main] B  Thread[First,5,main] C  Thread[First,5,main] D  Thread[First,5,main] E  Thread[First,5,main] F  Thread[First,5,main] G  Thread[First,5,main] H  Thread[First,5,main] I  Thread[First,5,main] J  Thread[Second,5,main] A  Thread[Second,5,main] B  Thread[Second,5,main] C  Thread[Second,5,main] D  Thread[Second,5,main] E  Thread[Second,5,main] F  Thread[Second,5,main] G  Thread[Second,5,main] H  Thread[Second,5,main] I  Thread[Second,5,main] J |

1. create 3 threads in such a way that while one thread is executing, 2 threads cannot interfere. they should display output "Exec 0" to "Exec 5".

|  |
| --- |
| **class** Th1 **implements** Runnable{    **synchronized** **public** **void** run() {  // **TODO** Auto-generated method stub  **for** (**int** i=0; i<=5; i++) {  System.***out***.println(Thread.*currentThread*()+"\tExec. "+i);  }  }  }  **public** **class** Demo {  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  Th1 th = **new** Th1();  Thread t1 = **new** Thread(th);  Thread t2 = **new** Thread(th);  Thread t3 = **new** Thread(th);    t1.start();  t2.start();  t3.start();  }  } |
| Thread[Thread-1,5,main] Exec. 0  Thread[Thread-1,5,main] Exec. 1  Thread[Thread-1,5,main] Exec. 2  Thread[Thread-1,5,main] Exec. 3  Thread[Thread-1,5,main] Exec. 4  Thread[Thread-1,5,main] Exec. 5  Thread[Thread-0,5,main] Exec. 0  Thread[Thread-0,5,main] Exec. 1  Thread[Thread-0,5,main] Exec. 2  Thread[Thread-0,5,main] Exec. 3  Thread[Thread-0,5,main] Exec. 4  Thread[Thread-0,5,main] Exec. 5  Thread[Thread-2,5,main] Exec. 0  Thread[Thread-2,5,main] Exec. 1  Thread[Thread-2,5,main] Exec. 2  Thread[Thread-2,5,main] Exec. 3  Thread[Thread-2,5,main] Exec. 4  Thread[Thread-2,5,main] Exec. 5 |

1. create 2 threads. Write a program which displays number 1 to 10 using class lock. [hint: - use "implements Runnable" and synchronized block]

|  |
| --- |
| **class** Demo **implements** Runnable{    **static** Class *c*;  **static** **void** perform() {    **synchronized** (*c*) {  **for** (**int** i=1; i<=10; i++)  System.***out***.println(Thread.*currentThread*()+"\t"+i);  }  }    **public** **void** run() {  *perform*();  }  **public** **static** **void** main(String[] args) **throws** Exception {  *c* = Class.*forName*("Demo");  Demo th1 = **new** Demo();  Demo th2 = **new** Demo();  Thread t1 = **new** Thread(th1);  Thread t2 = **new** Thread(th2);    t1.start();  t2.start();  }  } |
| Thread[Thread-1,5,main] 1  Thread[Thread-1,5,main] 2  Thread[Thread-1,5,main] 3  Thread[Thread-1,5,main] 4  Thread[Thread-1,5,main] 5  Thread[Thread-1,5,main] 6  Thread[Thread-1,5,main] 7  Thread[Thread-1,5,main] 8  Thread[Thread-1,5,main] 9  Thread[Thread-1,5,main] 10  Thread[Thread-0,5,main] 1  Thread[Thread-0,5,main] 2  Thread[Thread-0,5,main] 3  Thread[Thread-0,5,main] 4  Thread[Thread-0,5,main] 5  Thread[Thread-0,5,main] 6  Thread[Thread-0,5,main] 7  Thread[Thread-0,5,main] 8  Thread[Thread-0,5,main] 9  Thread[Thread-0,5,main] 10 |

1. create 2 threads

one thread will display 1 to 50

second thread will display 50 to 1

both the threads should start simultaneously.

(use implements method)

|  |
| --- |
| **class** Demo **implements** Runnable {    **void** ascending() {  **for**(**int** i=1; i<=50; i++) {  System.***out***.println(Thread.*currentThread*()+"\t"+i);  }  }    **void** descending() {  **for**(**int** i=50; i>0; i--) {  System.***out***.println(Thread.*currentThread*()+"\t"+i);  }  }    **public** **void** run() {  **if**(Thread.*currentThread*().getName().equalsIgnoreCase("ascending"))  ascending();  **else**  descending();  }  **public** **static** **void** main(String[] args) **throws** Exception {    Demo th1 = **new** Demo();  Thread t1 = **new** Thread(th1);  Thread t2 = **new** Thread(th1);    t1.setName("ascending");  t2.setName("descending");    t1.start();  t2.start();  }  } |
| Thread[ascending,5,main] 1  Thread[ascending,5,main] 2  Thread[ascending,5,main] 3  Thread[ascending,5,main] 4  Thread[ascending,5,main] 5  Thread[descending,5,main] 50  Thread[descending,5,main] 49  Thread[descending,5,main] 48  Thread[ascending,5,main] 6  Thread[descending,5,main] 47  Thread[descending,5,main] 46  Thread[ascending,5,main] 7  Thread[descending,5,main] 45  Thread[descending,5,main] 44  Thread[ascending,5,main] 8  Thread[descending,5,main] 43  Thread[descending,5,main] 42  Thread[descending,5,main] 41  Thread[descending,5,main] 40  Thread[descending,5,main] 39  Thread[descending,5,main] 38  Thread[descending,5,main] 37  Thread[descending,5,main] 36  Thread[descending,5,main] 35  Thread[descending,5,main] 34  Thread[descending,5,main] 33  Thread[descending,5,main] 32  Thread[descending,5,main] 31  Thread[descending,5,main] 30  Thread[descending,5,main] 29  Thread[descending,5,main] 28  Thread[descending,5,main] 27  Thread[descending,5,main] 26  Thread[descending,5,main] 25  Thread[ascending,5,main] 9  Thread[ascending,5,main] 10  Thread[ascending,5,main] 11  Thread[ascending,5,main] 12  Thread[descending,5,main] 24  Thread[descending,5,main] 23  Thread[descending,5,main] 22  Thread[descending,5,main] 21  Thread[ascending,5,main] 13  Thread[ascending,5,main] 14  Thread[ascending,5,main] 15  Thread[ascending,5,main] 16  Thread[ascending,5,main] 17  Thread[ascending,5,main] 18  Thread[ascending,5,main] 19  Thread[ascending,5,main] 20  Thread[ascending,5,main] 21  Thread[ascending,5,main] 22  Thread[ascending,5,main] 23  Thread[descending,5,main] 20  Thread[descending,5,main] 19  Thread[descending,5,main] 18  Thread[ascending,5,main] 24  Thread[ascending,5,main] 25  Thread[ascending,5,main] 26  Thread[ascending,5,main] 27  Thread[ascending,5,main] 28  Thread[ascending,5,main] 29  Thread[ascending,5,main] 30  Thread[ascending,5,main] 31  Thread[ascending,5,main] 32  Thread[ascending,5,main] 33  Thread[ascending,5,main] 34  Thread[ascending,5,main] 35  Thread[descending,5,main] 17  Thread[descending,5,main] 16  Thread[descending,5,main] 15  Thread[descending,5,main] 14  Thread[descending,5,main] 13  Thread[descending,5,main] 12  Thread[descending,5,main] 11  Thread[descending,5,main] 10  Thread[descending,5,main] 9  Thread[descending,5,main] 8  Thread[descending,5,main] 7  Thread[descending,5,main] 6  Thread[descending,5,main] 5  Thread[descending,5,main] 4  Thread[descending,5,main] 3  Thread[descending,5,main] 2  Thread[descending,5,main] 1  Thread[ascending,5,main] 36  Thread[ascending,5,main] 37  Thread[ascending,5,main] 38  Thread[ascending,5,main] 39  Thread[ascending,5,main] 40  Thread[ascending,5,main] 41  Thread[ascending,5,main] 42  Thread[ascending,5,main] 43  Thread[ascending,5,main] 44  Thread[ascending,5,main] 45  Thread[ascending,5,main] 46  Thread[ascending,5,main] 47  Thread[ascending,5,main] 48  Thread[ascending,5,main] 49  Thread[ascending,5,main] 50 |