Experiment 9

Threads and Collection

Name: Aditya Goyat

SAP ID: 500080575

1) Write a program to implement the concept of threading by extending Thread Class and Runnable interface.

CODE:

**package** exp9;

**class** T1 **extends** Thread{

**public** **void** run() {

System.***out***.println("The thread is running.");

}

}

**class** T2 **implements** Runnable{

**public** **void** run() {

System.***out***.println("The runnable thread is running");

}

}

**public** **class** MainQ1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

T1 t1 = **new** T1();

t1.start();

T2 r2 = **new** T2();

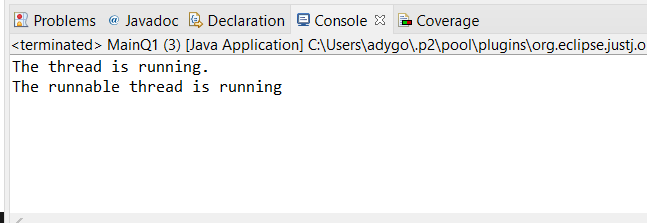
Thread t2 = **new** Thread(r2);

t2.start();

}

}

OUTPUT:



2) Write a program for generating 2 threads, one for printing even numbers and the otherfor printing odd numbers.

CODE:

**package** exp9;

**class** R1 **extends** Thread{

**public** **void** run() {

**for**(**int** i = 0; i<=10; i+=2)

System.***out***.println(i);

}

}

**class** R2 **extends** Thread{

**public** **void** run() {

**for**(**int** i = 1; i<=10; i+=2)

System.***out***.println(i);

}

}

**public** **class** MainQ2 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

R1 t1 = **new** R1();

R2 t2 = **new** R2();

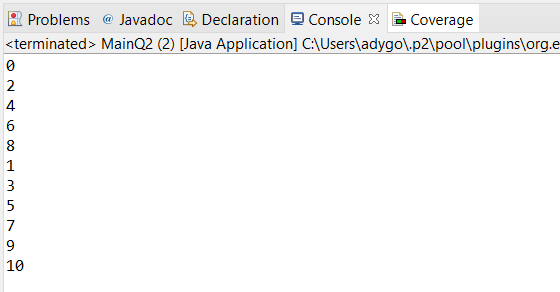
t1.start();

t2.start();

}

}

OUTPUT:



3) Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.

CODE:

**package** exp9;

**class** Counter **extends** Thread{

**static** **int** *n* = 0;

**public** **void** run() {

*n* = inc(*n*);

}

**public** **synchronized** **int** inc(**int** i) {

i++;

**return** i;

}

}

**public** **class** MainQ3 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Counter t1 = **new** Counter();

Counter t2 = **new** Counter();

Counter t3 = **new** Counter();

Counter t4 = **new** Counter();

Counter t5 = **new** Counter();

Counter t6 = **new** Counter();

Counter t7 = **new** Counter();

Counter t8 = **new** Counter();

Counter t9 = **new** Counter();

Counter t10 = **new** Counter();

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

t6.start();

t7.start();

t8.start();

t9.start();

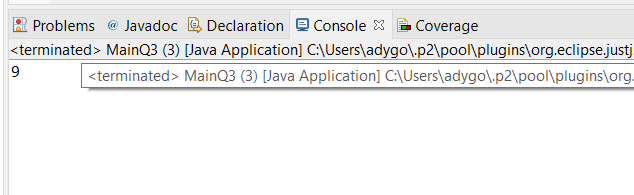
t10.start();

System.***out***.println(Counter.*n*);

}

}

OUTPUT:



4) Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.

CODE:

**package** exp9;

**class** T **extends** Thread{

**public** **void** run() {}

}

**public** **class** MainQ4 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

T t1 = **new** T();

T t2 = **new** T();

T t3 = **new** T();

T t4 = **new** T();

T t5 = **new** T();

t1.setPriority(Thread.***NORM\_PRIORITY*** + 3);

t2.setPriority(Thread.***MIN\_PRIORITY***);

t3.setPriority(Thread.***NORM\_PRIORITY*** + 1);

t4.setPriority(Thread.***NORM\_PRIORITY*** - 1);

t5.setPriority(Thread.***NORM\_PRIORITY*** + 2);

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

System.***out***.println("Is thread one alive? " + t1.isAlive());

System.***out***.println("Is thread one alive? " + t2.isAlive());

System.***out***.println("Is thread one alive? " + t3.isAlive());

System.***out***.println("Is thread one alive? " + t4.isAlive());

System.***out***.println("Is thread one alive? " + t5.isAlive());

**try** {

t1.*sleep*(1000);

t5.*sleep*(1000);

}

**catch**(InterruptedException e) {

System.***out***.println(e);

}

t1.~~stop~~();

t2.~~stop~~();

t3.~~stop~~();

t4.~~stop~~();

t5.~~stop~~();

}

}