**JAVA PROGRAMMING-WEEK 2-ASSIGNMENT 4**

1.ANS

class MultithreadedProgramming {

public static void main(String args[]) {

// Get the reference to the current thread (main thread)

Thread t = Thread.currentThread();

// Set the name of the current thread

t.setName("New Thread");

// Print information about the current thread

System.out.println(t);

}

}

2.ANS

class PriorityThread extends Thread {

public PriorityThread(String name) {

super(name);

}

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.println(Thread.currentThread().getName() + " - Count: " + i);

}

}

}

public class ThreadPriorityExample {

public static void main(String[] args) {

// Creating three threads

PriorityThread thread1 = new PriorityThread("Thread 1");

PriorityThread thread2 = new PriorityThread("Thread 2");

PriorityThread thread3 = new PriorityThread("Thread 3");

// Setting priorities for the threads

thread1.setPriority(Thread.MIN\_PRIORITY); // Priority 1

thread2.setPriority(Thread.NORM\_PRIORITY); // Priority 5 (default)

thread3.setPriority(Thread.MAX\_PRIORITY); // Priority 10

// Start the threads

thread1.start();

thread2.start();

thread3.start();

}

}

3.ANS

import java.util.HashSet;

import java.util.Set;

public class SetOperationsExample {

public static void main(String[] args) {

// Create two sets

Set<Integer> set1 = new HashSet<>();

Set<Integer> set2 = new HashSet<>();

// Add elements to the sets

set1.add(1);

set1.add(2);

set1.add(3);

set2.add(2);

set2.add(3);

set2.add(4);

// Perform union and intersection

Set<Integer> unionSet = union(set1, set2);

Set<Integer> intersectionSet = intersection(set1, set2);

// Display the results

System.out.println("Set 1: " + set1);

System.out.println("Set 2: " + set2);

System.out.println("Union: " + unionSet);

System.out.println("Intersection: " + intersectionSet);

}

// Method to calculate the union of two sets

public static <T> Set<T> union(Set<T> set1, Set<T> set2) {

Set<T> unionSet = new HashSet<>(set1);

unionSet.addAll(set2);

return unionSet;

}

// Method to calculate the intersection of two sets

public static <T> Set<T> intersection(Set<T> set1, Set<T> set2) {

Set<T> intersectionSet = new HashSet<>(set1);

intersectionSet.retainAll(set2);

return intersectionSet;

}

}