1. **Write a Java program to perform a runnable interface, take two threads t1 and t2 and fetch the names of the thread using getName() method.**

**CODE :**

class MyRunnable implements Runnable {

public void run() {

String threadName = Thread.currentThread().getName();

System.out.println("Thread Name: " + threadName);

}

}

public class ThreadNameExample {

public static void main(String[] args) {

MyRunnable myRunnable = new MyRunnable();

Thread t1 = new Thread(myRunnable);

Thread t2 = new Thread(myRunnable);

t1.start();

t2.start();

}

}

**2.Given an integer N, the task is to write program to print the first N natural numbers in increasing order using two threads.**

***Input: N = 10  
Output: 1 2 3 4 5 6 7 8 9 10***

***Input: N = 18  
Output: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18***

**CODE :**

class PrintNumbers implements Runnable {

private int n;

public PrintNumbers(int n) {

this.n = n;

}

@Override

public void run() {

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

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

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class PrintNumbersInOrder {

public static void main(String[] args) {

int N = 18;

PrintNumbers printNumbers = new PrintNumbers(N);

Thread numbersThread = new Thread(printNumbers);

numbersThread.start();

}

}

**3.  Write a two-threaded program, where one thread finds all prime numbers (in 0 to 10) and another thread finds all palindrome numbers (in 10 to 50). Schedule these threads in a sequential manner to get the results.**

**Palindrome numbers from 10 to 50 : 11 22 33 44**

**Prime numbers from 0 to 10 : 2 3 5 7**

**CODE :**

public class SequentialExecution {

public static void main(String[] args) {

Thread primeThread = new Thread(() -> {

System.out.print("Prime numbers from 0 to 10: ");

for (int i = 0; i <= 10; i++) {

if (isPrime(i)) {

System.out.print(i + " ");

}

}

System.out.println();

});

Thread palindromeThread = new Thread(() -> {

System.out.print("Palindrome numbers from 10 to 50: ");

for (int i = 10; i <= 50; i++) {

if (isPalindrome(i)) {

System.out.print(i + " ");

}

}

System.out.println();

});

primeThread.start();

try {

primeThread.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

palindromeThread.start();

}

private static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

private static boolean isPalindrome(int num) {

int originalNum = num;

int reversedNum = 0;

while (num > 0) {

int digit = num % 10;

reversedNum = reversedNum \* 10 + digit;

num /= 10;

}

return originalNum == reversedNum;

}

}