Java Lab Practice Programs

1. Write a program to create a thread and find the sum of odd numbers from 1 to 100 in this thread. Find the sum of even numbers for the same range in the main thread.

Program:

```
class NewThread implements Runnable {
   Thread t;

   NewThread() {
      t = new Thread(this, "New Thread");
      System.out.println("CT:" + t);
      t.start();
   }

   public void run() {
      int sum = 0, i;
      try {
         for (i = 1; i <= 100; i++) {
            if (i % 2 == 1) {
               sum = sum + i;
            }
         }
         System.out.println("Sum of odd numbers " + sum);
         Thread.sleep(1000);
      } catch (InterruptedException ie) {
         System.out.println("Child Thread Interrupted");
```

```
    }
  }
}


class MainSum {
  public static void main(String args[]) {
    int sum = 0, i;
    NewThread n1 = new NewThread();
    try {
      for (i = 1; i <= 100; i++) {
        if (i % 2 == 0) {
          sum = sum + i;
        }
      }
      Thread.sleep(2000);
      System.out.println("Sum of even numbers " + sum);
    } catch (InterruptedException ie) {
      System.out.println("Child Thread Interrupted");
    }
  }
}
```

Output:

```
C:\Users\Deepesh\Desktop\java>javac MainSum.java

C:\Users\Deepesh\Desktop\java>java MainSum
CT:Thread[New Thread,5,main]
Sum of odd numbers 2500
Sum of even numbers 2550
```

2. Develop a multithreaded Java program to create three threads. First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints. If the value is odd, the third thread will print the value of cube of number.

Program:

```
import java.util.Random;

class RandomNumberThread extends Thread {
	public void run() {
		Random random = new Random();
		for (int i = 0; i < 10; i++) {
			int randomInteger = random.nextInt(100);
			System.out.println("Random Integer generated : " + randomInteger);
			if((randomInteger%2) == 0) {
				SquareThread sThread = new SquareThread(randomInteger);
				sThread.start();
			}
			else {
				CubeThread cThread = new CubeThread(randomInteger);
				cThread.start();
			}
			try {
				Thread.sleep(1000);
			}
			catch (InterruptedException ex) {
				System.out.println(ex);
			}
```

```
                }
        }
}

class SquareThread extends Thread {
        int number;

        SquareThread(int randomNumbern) {
                number = randomNumbern;
        }

        public void run() {
                System.out.println("Square of " + number + " = " + (number * number));
        }
}

class CubeThread extends Thread {
        int number;

        CubeThread(int randomNumber) {
                number = randomNumber;
        }

        public void run() {
                System.out.println("Cube of " + number + " = " + number * number * number);
        }
}

public class MultiThreadingTest {
```

```
        public static void main(String args[]) {
                RandomNumberThread rnThread = new RandomNumberThread();
                rnThread.start();
        }
}
```

Output:

```
C:\Users\Deepesh>cd desktop

C:\Users\Deepesh\Desktop>cd java

C:\Users\Deepesh\Desktop\java>javac MultiThreadingTest.java

C:\Users\Deepesh\Desktop\java>java MultiThreadingTest
Random Integer generated : 92
Square of 92 = 8464
Random Integer generated : 6
Square of 6 = 36
Random Integer generated : 44
Square of 44 = 1936
Random Integer generated : 93
Cube of 93 = 804357
Random Integer generated : 99
Cube of 99 = 970299
Random Integer generated : 54
Square of 54 = 2916
Random Integer generated : 63
Cube of 63 = 250047
Random Integer generated : 17
Cube of 17 = 4913
Random Integer generated : 49
Cube of 49 = 117649
Random Integer generated : 76
Square of 76 = 5776
```