Title: Create and demonstrate program for the concept of Multithreading

Objective: Understand the concepts of multithreading in Java.

Problem Statement: Create a Java program to illustrate multithreading concepts. The program should consist of two parts, each addressing different aspects of multithreading.

Instructions:

1. Use proper comments for clarity and understanding.
2. Ensure that the interaction between threads is well-organized and demonstrates the specified aspects of multithreading.

Java Program:

```
// File: MultithreadingBasicsDemo.java

class NumberPrinter extends Thread {
    private static final Object lock = new Object(); // Object for synchronization

    private void printNumbers() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getName() + " - Count: " + i);
        }
    }


    private void printEvenOddNumbers() {
        for (int i = 2; i <= 10; i += 2) {
            System.out.println(Thread.currentThread().getName() + " - Even Number: " + i);
            try {
                // Introducing a delay of 500 milliseconds
                Thread.sleep(500);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                e.printStackTrace();
            }
        }

        for (int i = 1; i <= 9; i += 2) {
            System.out.println(Thread.currentThread().getName() + " - Odd Number: " + i);
            try {
                // Introducing a delay of 500 milliseconds
                Thread.sleep(500);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                e.printStackTrace();
            }
        }
    }

    // Entry point for each thread
    public void run() {
        printNumbers();
        printEvenOddNumbers();
    }
}

class MultithreadingBasicsDemo {
    public static void main(String[] args) {
        // Creating two threads
        Thread thread1 = new NumberPrinter();
        Thread thread2 = new NumberPrinter();

        // Naming the threads for clarity
        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        // Starting the threads
        thread1.start();
```

```
        thread2.start();
    }
}
```

Explanation:

- The `NumberPrinter` class extends `Thread` and defines methods for printing numbers in two different ways.
- The `run` method serves as the entry point for each thread, executing the specified sub-questions.
- The `MultithreadingBasicsDemo` class creates two threads, names them for clarity, and starts their execution.

The provided code focuses on understanding multithreading concepts, with well-commented sections for clarity and comprehension.