Portfolio Project

Sarah LeFlore

Colorado State University Global

CSC450: Programming III

Farhad Bari

03/09/2025

Portfolio Project

This program uses two threads, one that counts up to 20 and one that counts down from 20. The threads are created by extending the Thread class. Deadlock can occur if one thread is stuck waiting for another thread to finish executing. Race conditions can occur if the threads attempt to use the same resource at the same time, to avoid this I used join(), which waits for one thread to finish before the other thread starts.

String variables in Java offer more security than in C++ Strings in Java are not prone to buffer overflow. Strings should not be used for storing sensitive data, such as passwords. Instead, a char array should be used, and then wiped. This is because a String will be stored in memory until it is garbage collected. In Java, an integer can overflow, so when using integers from user input, error handling should be used.

Source code:

```
/*

Main class, uses thread 1, waits for thread 1 to finish, then starts thread 2
 */
public class Main {
public static void main(String[] args) {
    Thread1 t1 = new Thread1();
    Thread2 t2 = new Thread2();
    t1.start();
    try {
        t1.join(); //Waits for thread 1 to finish
    }
    catch (InterruptedException e) { //throws exception if not finished
        throw new RuntimeException(e);
    }
    finally {
        t2.start(); //Starts thread 2
    }
}
}
```

```
/*

This class uses a thread to count up to 20
 */
public class Thread1 extends Thread {
    public void run() {
        System.out.println("Thread 1:");
        for (int i = 1; i <= 20; i++) {
            System.out.print(i + " ");
        }
        System.out.println(); //Start new line
    }
}

/*

This class uses a thread to count down from 20
 */
public class Thread2 extends Thread {
    public void run() {
        System.out.println("Thread 2: ");
        for (int i = 20; i >= 1; i--) {
            System.out.print(i + " ");
        }
    }
}
```