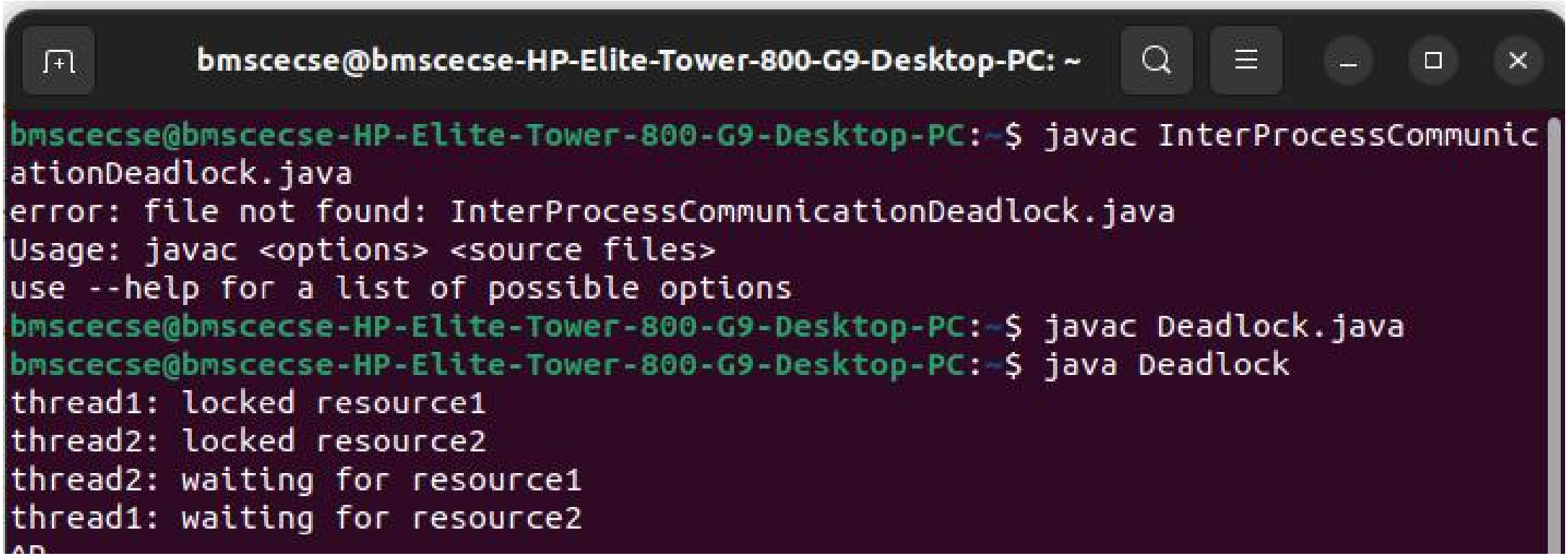
```java
public class Deadlock {
    public static void main(String[] args) {
        // Declare resources as Objects
        final Object resource1 = "resource1";
        final Object resource2 = "resource2";

        // Thread 1
        Thread thread1 = new Thread(() -> {
            synchronized (resource1) {
                System.out.println("thread1: locked resource1");
                try {
                    Thread.sleep(50); // Simulate some work
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                System.out.println("thread1: waiting for resource2");

                // Attempt to acquire resource2
                synchronized (resource2) {
                    System.out.println("thread1: locked resource2");
                }
            }
        });

        // Thread 2
        Thread thread2 = new Thread(() -> {
            synchronized (resource2) {
                System.out.println("thread2: locked resource2");
                try {
                    Thread.sleep(50); // Simulate some work
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                System.out.println("thread2: waiting for resource1");

                // Attempt to acquire resource1
                synchronized (resource1) {
                    System.out.println("thread2: locked resource1");
                }
            }
        });

        // Start the threads
        thread1.start();
        thread2.start();
    }
}

```


bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: ~
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ javac InterProcessCommunic
ationDeadlock.java
error: file not found: InterProcessCommunicationDeadlock.java
Usage: javac <options> <source files>
use --help for a list of possible options
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ javac Deadlock.java
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ java Deadlock
thread1: locked resource1
thread2: locked resource2
thread2: waiting for resource1
thread1: waiting for resource2