M01.5 - Threads Introduction

YiChen_2020_10_01

Threads - Introduction

Reserved words to know:

Synchronized

Volatile

Use the Java Thread programs shown in class as a starting point for this assignment.

Create static data to represent Dorthy, her favorite character and her favorite color.

```
public class Dorthy {
    public enum OZpeople {SCARECROW, COWARDLYLION, TINMAN}
    public enum OZcolors {BROWN, YELLOW, SILVER}

    private static OZpeople likes;
    private static OZcolors color;
```

Create threads for the Tin Man (Silver), Scarecrow (Brown), and Cowardly Lion (Yellow), that each set Dorthy's favorite character to themselves, and a color.

```
16  public static void main(String[] args) {
17      thread a = new thread(Dorthy.OZpeople.TINMAN, Dorthy.OZcolors.SILVER);     // create thread object  // call the object a
18      a.start();
19
20      thread b = new thread(Dorthy.OZpeople.SCARECROW, Dorthy.OZcolors.BROWN);   // create thread object  // call the object b
21      b.start();
22
23      thread c = new thread(Dorthy.OZpeople.COWARDLYLION, Dorthy.OZcolors.YELLOW);    // create thread object  // call the object c
24      c.start();
```

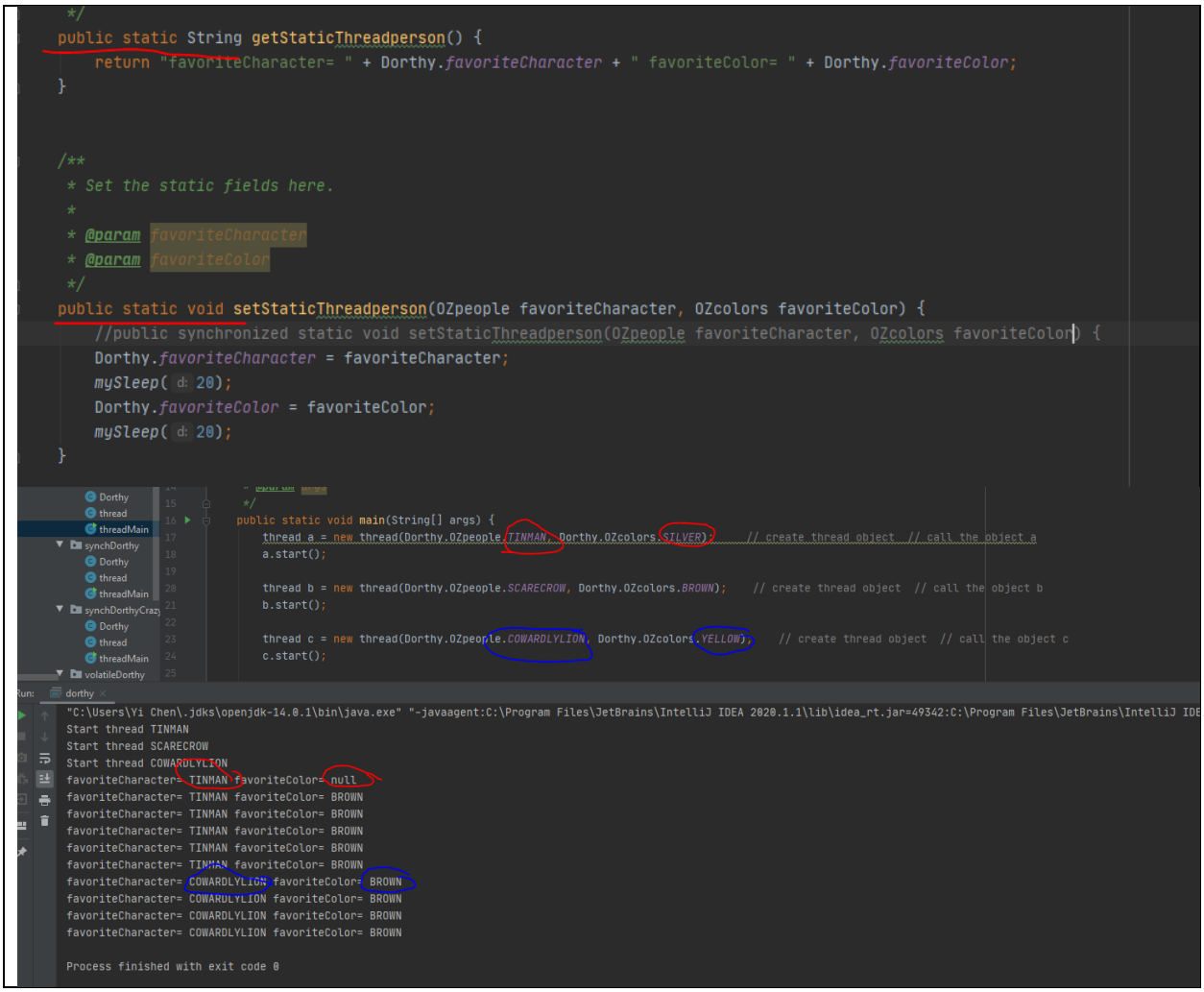
```java
public class thread extends Thread {

    /**
     * This copy of data to use to overwrite
     */
    public Dorthy.OZpeople character;
    public Dorthy.OZcolors color;

    /**
     * thread receives person information
     * @param character character
     * @param color color
     */
    public thread(Dorthy.OZpeople character, Dorthy.OZcolors color) {
        this.character = character;
        this.color = color;
    }

    /**
     * Loop setting character, color etc.. into static area.
     */
    @Override
    public void run() {
        System.out.println("Start thread " + this.character);              //print start() thread when called
        try {
            for (int i = 0; i < 1000; i++) {
                Dorthy.setStaticThreadperson(this.character, this.color);
                //System.out.println("Loop thread " + i + " "+ this.name);
                Thread.sleep( millis: 30);
            }
        } catch (Exception e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
```

You must use an Enumeration for the color.

```java
public class Dorthy {
    public enum OZpeople {SCARECROW, COWARDLYLION, TINMAN}
    public enum OZcolors {BROWN, YELLOW, SILVER}
```

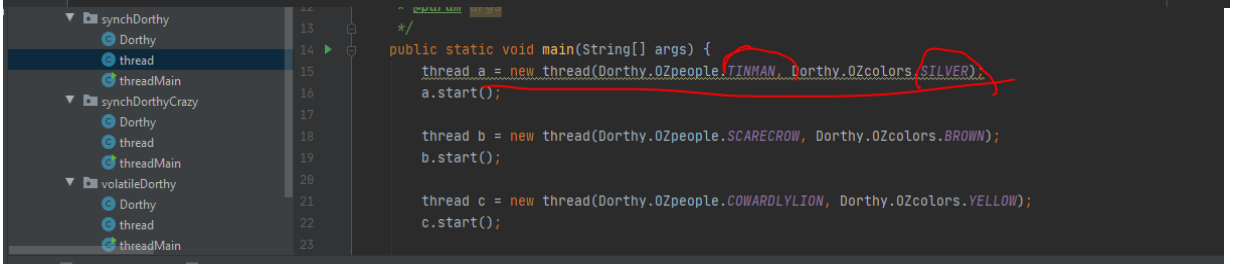
Have 2 versions of the program

Version 1: no thread locking….the data is constantly corrupted when printed from the main line.

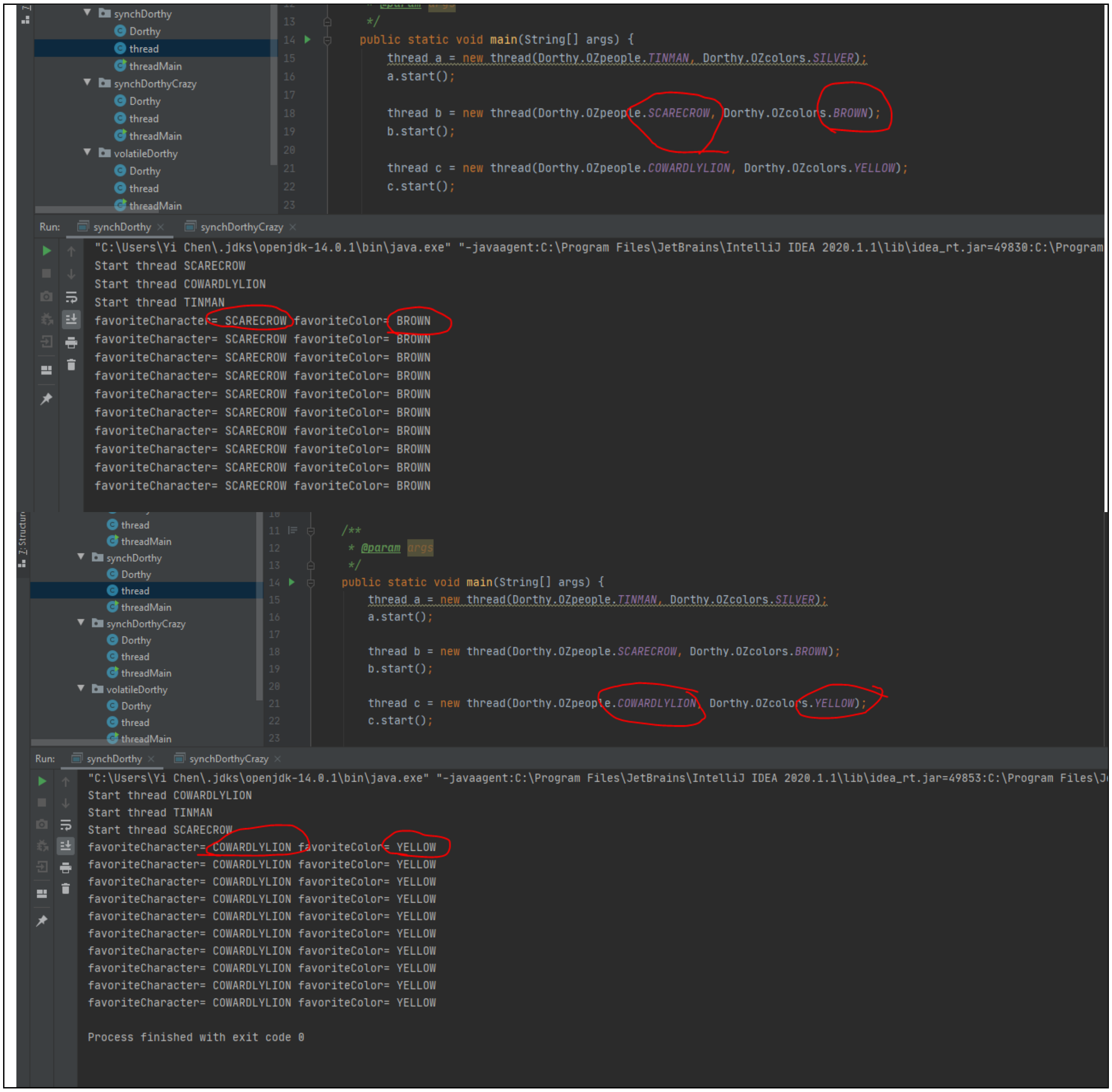


Version 2: thread safe…the data is always fully one set or another, never a combination.

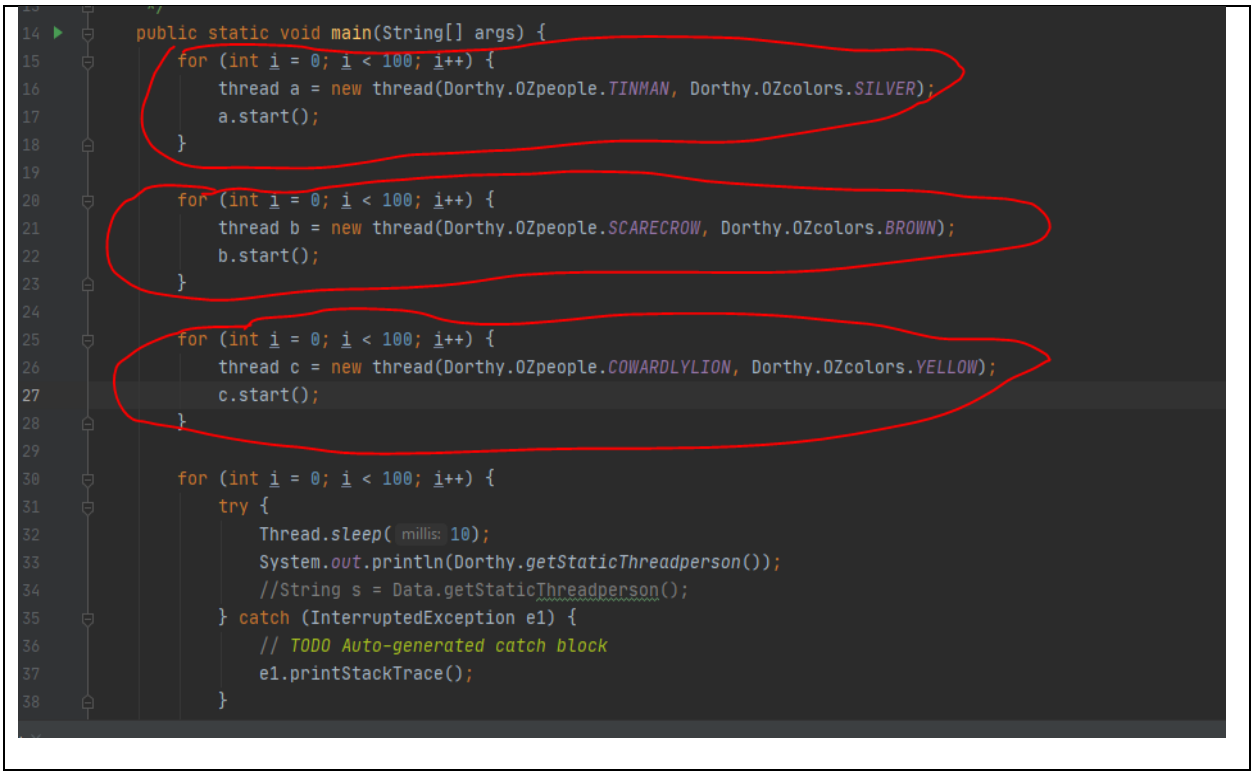
```
     */
    public synchronized static String getStaticThreadperson() {
        return "favoriteCharacter= " + Dorthy.favoriteCharacter + " favoriteColor= " + Dorthy.favoriteColor;
    }


    /**
     * Set the static fields here.
     *
     * @param favoriteCharacter
     * @param favoriteColor
     */
    public synchronized static void setStaticThreadperson(OZpeople favoriteCharacter, OZcolors favoriteColor) {
        //public synchronized static void setStaticThreadperson(String name, long phone, long cell, String email) {
        Dorthy.favoriteCharacter = favoriteCharacter;
        mySleep( d: 20);
        Dorthy.favoriteColor = favoriteColor;
        mySleep( d: 20);
    }

    public static void mySleep(int d) {
        try {
            Thread.sleep(d);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```




```
Run: synchDorthy × synchDorthyCrazy ×
"C:\Users\Yi Chen\.jdks\openjdk-14.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.1.1\lib\idea_rt.jar=49793:C:\Pr
Start thread TINMAN
Start thread COWARDLYLION
Start thread SCARECROW
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER

Process finished with exit code 0
```



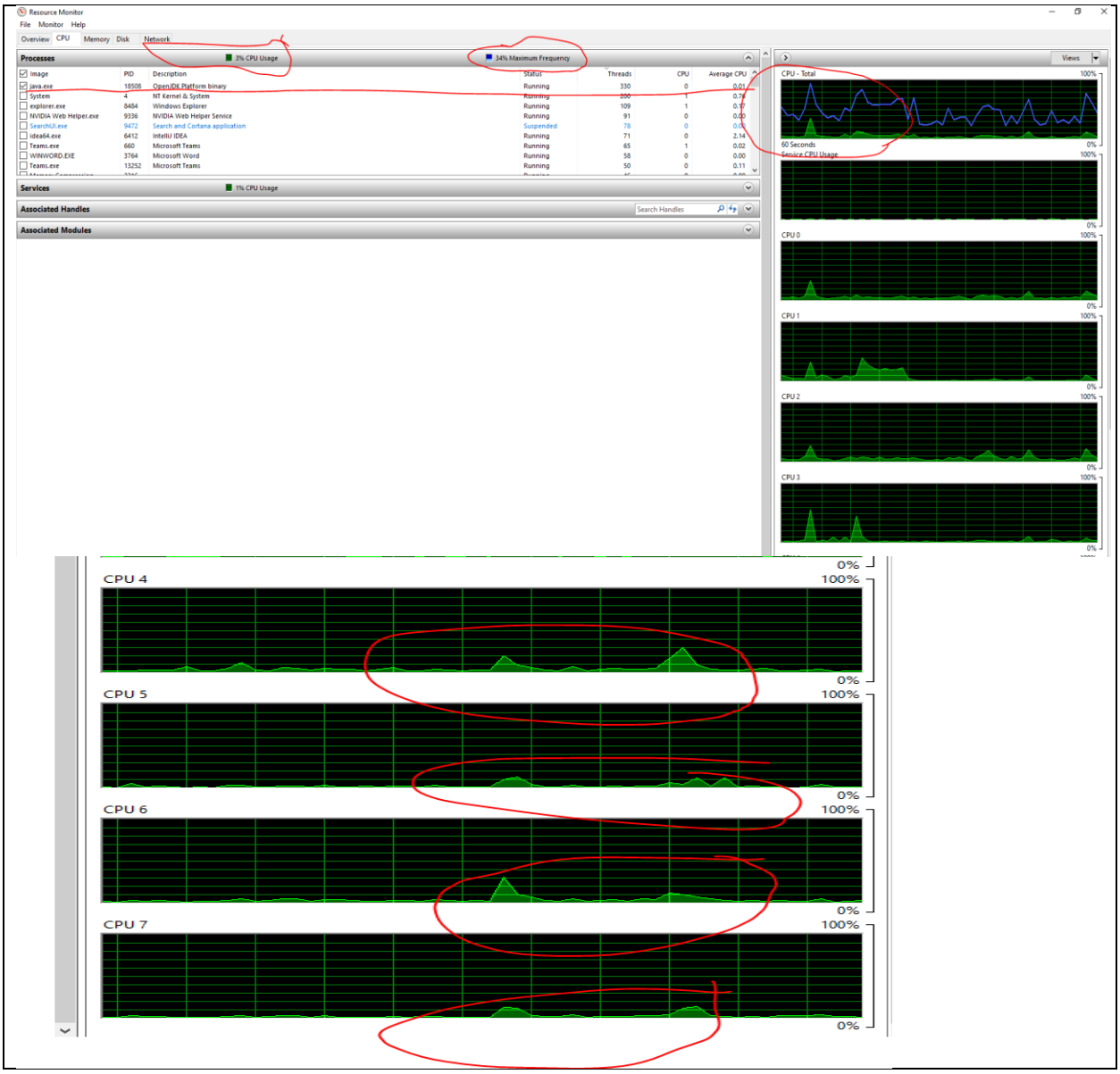
Just three threads is not enough, you'll need about 100 or more of each (Tin Man, Scarecrow, and Cowardly Lion), all trying to be Dorthy's favorite. Use an appropriate collection(s) for this.



```
public static void main(String[] args) {
    for (int i = 0; i < 100; i++) {
        thread a = new thread(Dorthy.OZpeople.TINMAN, Dorthy.OZcolors.SILVER);
        a.start();
    }

    for (int i = 0; i < 100; i++) {
        thread b = new thread(Dorthy.OZpeople.SCARECROW, Dorthy.OZcolors.BROWN);
        b.start();
    }

    for (int i = 0; i < 100; i++) {
        thread c = new thread(Dorthy.OZpeople.COWARDLYLION, Dorthy.OZcolors.YELLOW);
        c.start();
    }

    for (int i = 0; i < 100; i++) {
        try {
            Thread.sleep( millis: 10);
            System.out.println(Dorthy.getStaticThreadperson());
            //String s = Data.getStaticThreadperson();
        } catch (InterruptedException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
```

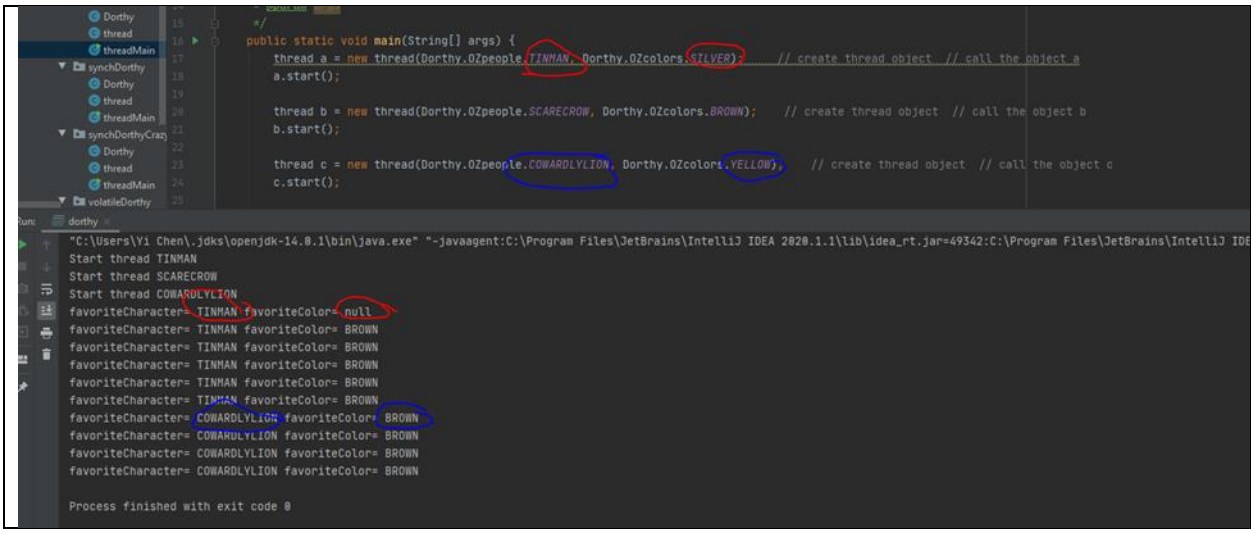
SHOW:

Run Resource Monitor or Performance Monitor (PERFMON) or Task Manager and show how much the PC resources are being stressed. Place screen capture in git folder.

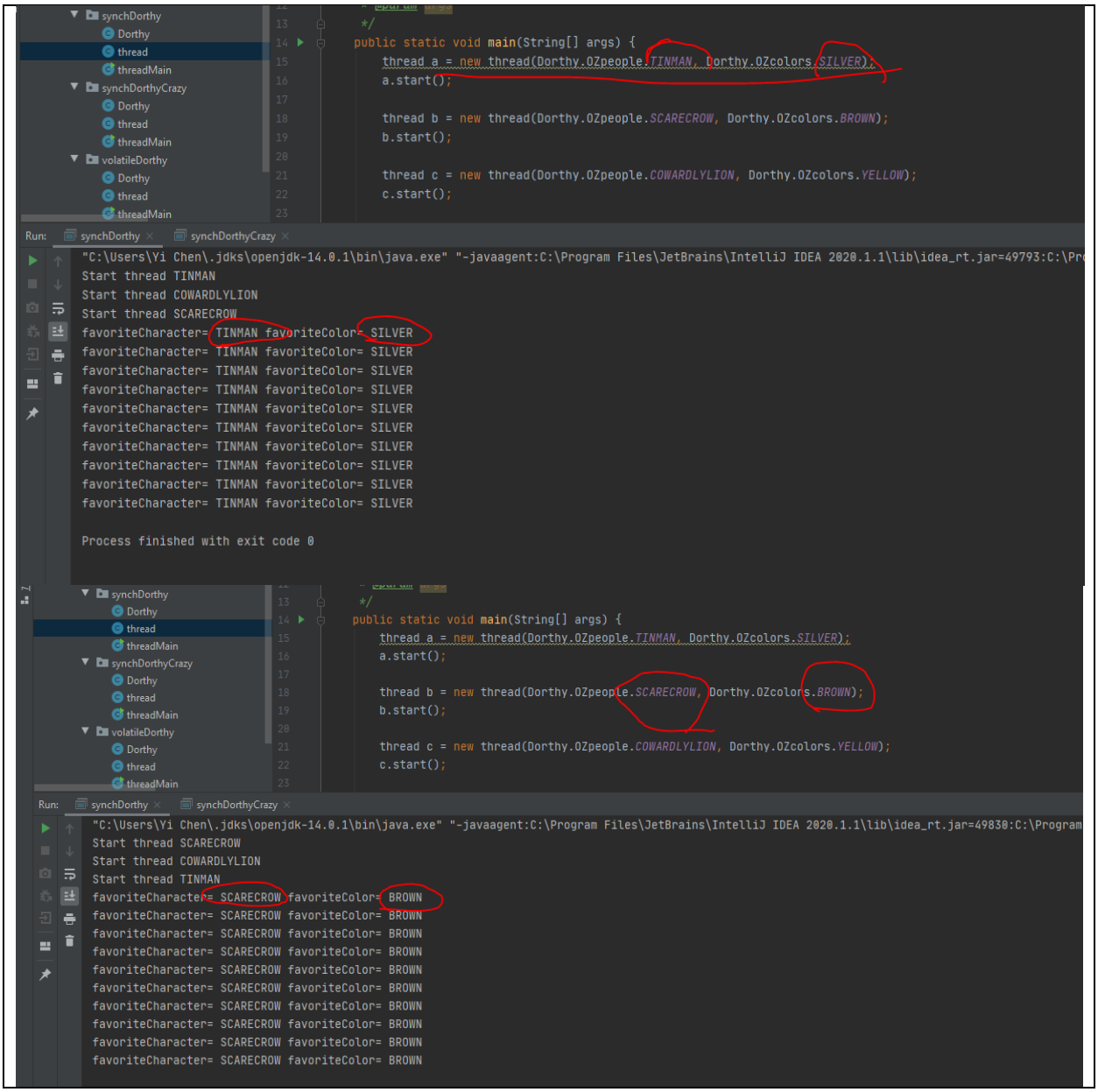


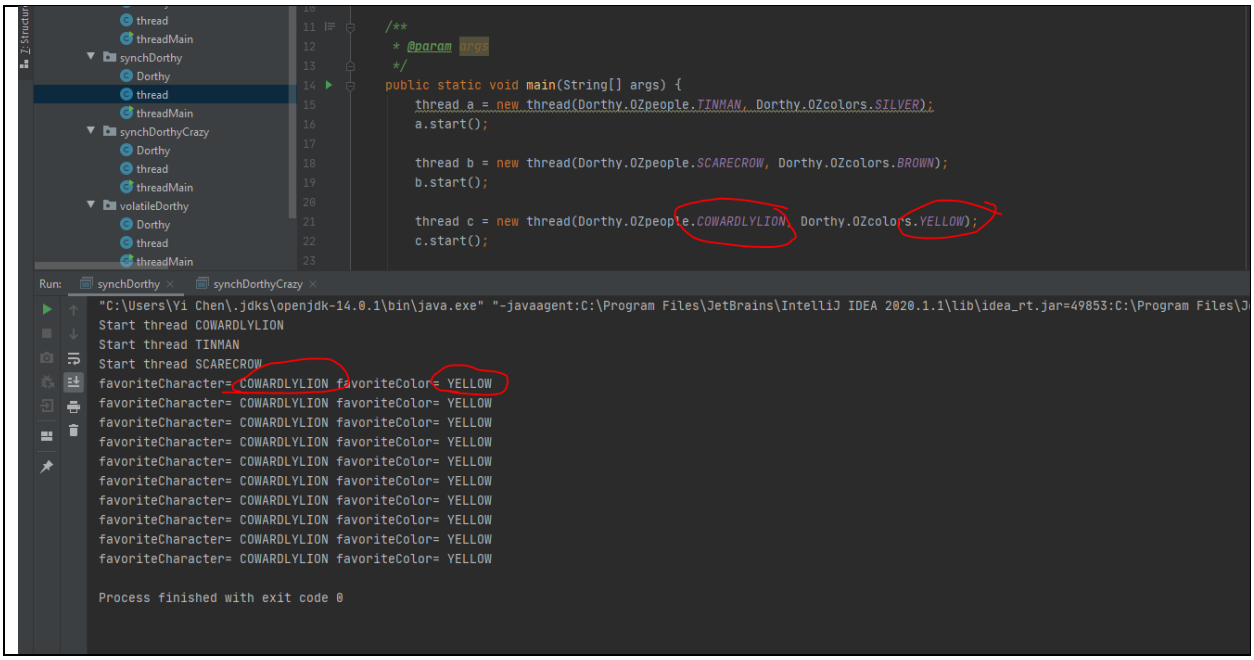
It isn't easy, but show that the unprotected Dorthy gets corrupted, and the protected Dorthy is not corrupted.

Dorthy gets corrupted



Dorthy is not corrupted


synchDorthy
Dorthy
thread
threadMain
synchDorthyCrazy
Dorthy
thread
threadMain
volatileDorthy
Dorthy
thread
threadMain
*/
public static void main(String[] args) {
thread a = new thread(Dorthy.OZpeople.TINMAN, Dorthy.OZcolors.SILVER);
a.start();
thread b = new thread(Dorthy.OZpeople.SCARECROW, Dorthy.OZcolors.BROWN);
b.start();
thread c = new thread(Dorthy.OZpeople.COWARDLYLION, Dorthy.OZcolors.YELLOW);
c.start();
Run:
synchDorthy
synchDorthyCrazy
"C:\Users\Yi Chen\.jdks\openjdk-14.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.1.1\lib\idea_rt.jar=49793:C:\Pr
Start thread TINMAN
Start thread COWARDLYLION
Start thread SCARECROW
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
favoriteCharacter= TINMAN favoriteColor= SILVER
Process finished with exit code 0
synchDorthy
Dorthy
thread
threadMain
synchDorthyCrazy
Dorthy
thread
threadMain
volatileDorthy
Dorthy
thread
threadMain
*/
public static void main(String[] args) {
thread a = new thread(Dorthy.OZpeople.TINMAN, Dorthy.OZcolors.SILVER);
a.start();
thread b = new thread(Dorthy.OZpeople.SCARECROW, Dorthy.OZcolors.BROWN);
b.start();
thread c = new thread(Dorthy.OZpeople.COWARDLYLION, Dorthy.OZcolors.YELLOW);
c.start();
Run:
synchDorthy
synchDorthyCrazy
"C:\Users\Yi Chen\.jdks\openjdk-14.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.1.1\lib\idea_rt.jar=49830:C:\Program
Start thread SCARECROW
Start thread COWARDLYLION
Start thread TINMAN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN
favoriteCharacter= SCARECROW favoriteColor= BROWN



Upload your movie showing (and explaining) your running output and code.

Submit Artifacts for marking:

You will be asked to explain your code in class during a code review. Be prepared to demonstrate your code, and answer questions.

your code into the git server, using a branch labeled PROG2200-Mxx (where xx is the module number)

Submit a simple PDF with code and running output (no TOC, paragraphs, ...)

labeled PROG2200-Mxx

```
Yi Chen@DESKTOP-HUATEIC MINGW64 /c/git/w0443276/PROG2200/M01.5 - Threads Introduction ((364ce62...))
$ git branch -l
* (HEAD detached at origin/master)
  M01.5-ThreadsIntroduction
  master
```

MINGW64:/c/git/w0443276/PROG2200/M01.5 - Threads Introduction

```
$ git add -A

Yi Chen@DESKTOP-HUATEIC MINGW64 /c/git/w0443276/PROG2200/M01.5 - Threads Introduction (master)
$ git commit -m " git branch M01.5 - Threads Introduction"
[master 364ce62]  git branch M01.5 - Threads Introduction
 38 files changed, 704 insertions(+)
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/.idea/.gitign
ore
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/.idea/misc.xm
l
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/.idea/modules
.xml
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/.idea/uiDesig
ner.xml
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/M01.5 - Threa
ds Introduction.iml
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/META-INF/M01.5 - Threads Introduction.kotlin_module
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/dorthy/Dorthy$OZcolors.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/dorthy/Dorthy$OZpeople.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/dorthy/Dorthy.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/dorthy/thread.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/dorthy/threadMain.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthy/Dorthy$OZcolors.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthy/Dorthy$OZpeople.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthy/Dorthy.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthy/thread.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthy/threadMain.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthyCrazy/Dorthy$OZcolors.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthyCrazy/Dorthy$OZpeople.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthyCrazy/Dorthy.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthyCrazy/thread.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/synchDorthyCrazy/threadMain.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/volatileDorthy/Dorthy$OZcolors.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/volatileDorthy/Dorthy$OZpeople.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/volatileDorthy/Dorthy.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/volatileDorthy/thread.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/out/productio
n/M01.5 - Threads Introduction/com/volatileDorthy/threadMain.class
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/dorth
y/Dorthy.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/dorth
y/thread.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/dorth
y/threadMain.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/synch
Dorthy/Dorthy.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/synch
Dorthy/thread.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/synch
Dorthy/threadMain.java
 create mode 100644 PROG2200/M01.5 - Threads Introduction/M01.5 - Threads Introduction/src/com/synch
DorthyCrazy/Dorthy.java
```