JAVA HW 10.2

```
1   public class HurlyBurly extends Thread {
2
3       static int aStaticInt;
4       int id;
5
6       HurlyBurly(int id) {
7           this.id = id;
8           aStaticInt = id;
9           if ( id == 1 )
10                  new HurlyBurly(2).run();
11      }
12      public void run()   {
13          System.out.println( id + " -----> ");
14          System.out.println("id/aStaticInt = " + id + "/" + aStaticInt );
15          System.out.println( id + " <----- ");
16      }
17      public static void main( String[] args ) {
18          new HurlyBurly(1).start();
19      }
20  }
```

1. In this code, a new HurlyBurly object is created which initializes the aStaticInt and id to 1 then because this id satisfies the if statement, the new object is created with id 2 and run method is called on it.
2. Because of this, 2 -----> 2 <----- will always be printed first.
3. Then start is called upon this object, scheduler creates a new thread which is ready to run.
4. Now only one sequence of output is left to be printed which will b, 1-----> 1 <-----

```
1 ----->
id/aStaticInt = 1/2
1 <-----
2 ----->
id/aStaticInt = 2/2
2 <-----
```

This output is not possible at all because the run method is called inside the constructor. So, no other thread is created. There are only two threads and it will run sequentially, which is firstly 2 -----> 2 <----- and then 1-----> 1 <-----.

Only changing the .run() to .start() can make this output possible.

We cannot construct the code anyhow with sleep statements to make this output very likely.