HW 10.2

1. In this code, the "HurlyBurly" class extends the "Thread" class to enable the main thread to create new threads.
2. We create a new object using a constructor that takes parameters, initializing the "id" member variable.
3. This new object initiates a new thread using the "start" method.
4. Within the constructor of this object, after setting the "id" variable, we create another object with an "id" value of 2.
5. This new object directly invokes the "run" method as an overridden method, bypassing the full thread lifecycle.
6. It proceeds to execute the code inside the "run" method sequentially, displaying the values of "staticInt" and "id" set during object creation.
7. In the regular execution flow, "aStaticInt" is 2, and the output reflects this accordingly.

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

The provided output is not feasible because it suggests a linear execution order. The object instantiation occurs before the thread starts. The code containing "aStaticInt" set to 2 is located within the constructor, executed during object creation, and precedes the thread transitioning to the "ready" state through the "start" method, which subsequently runs the "run" method.