```java
public class HurlyBurly extends Thread {

   static int aStaticInt;
   int id;

   HurlyBurly(int id) {
       this.id = id;
       aStaticInt = id;
       if ( id == 1 ) {
           new HurlyBurly(2).run();
       }
   }
   public void run() {
       System.out.println( id + " -----> ");
       System.out.println("id/aStaticInt = " + id + "/" + aStaticInt );
       System.out.println( id + " <----- ");
   }
   public static void main( String[] args ) {
       new HurlyBurly(1).start();
   }
}
```

The order cannot be reversed. The reason is that the run method for threads are executed as if they are a part of the main execution. In this program, since there is only one thread being scheduled in the main method, that one will be executed first. But even before the thread is schedules, the constructor will create a new thread with id 2 and run that thread. This means before the initial thread is even passed on to the scheduler, the thread with id 2 is executed as if it is a part of the main thread. Then, the thread with id 1 is executed, and completed.