Our PrelimEvent class just returns an instance of CircleSeeding,

```
public class PrelimEvent extends Event {

//class describes an event that will be swum twice
    public PrelimEvent(String filename, int lanes) {
        super(filename, lanes);
    }
    //return circle seeding
    public Seeding getSeeding() {
        return new CircleSeeding(swimmers, numLanes);
    }
    public boolean isPrelim() {
        return true;
    }
    public boolean isFinal() {
        return false;
    }
    public boolean isTimedFinal() }
        return false;
    }
}
```

Straight Seeding

In actually writing this program, we'll discover that most of the work is done in straight seeding; the changes for circle seeding are pretty minimal. So we instantiate our StraightSeeding class and copy in the Vector of swimmers and the number of lanes.

```
public class StraightSeeding extends Seeding {
      protected Vector     swimmers;
      protected Swimmer[] swmrs;
      protected int        numLanes;
      protected int[]      lanes;
      protected int        count;
      protected int        numHeats;

public StraightSeeding(Vector sw, int lanes) {
      Swimmers = sw;
      numLanes = lanes;
      count = sw.size();
      calcLaneOrder();
      seed();
    }
```

Then, as part of the constructor, we do the basic seeding:

```
protected void seed() {
         //loads the swmrs array and sorts it
         sortUpwards();

         int lastHeat = count % numLanes;
         if(lastHeat < 3)
            lastHeat = 3;     //last heat must have 3 or more
```