Useless Fact of the Day

- The longest foot race in the world is 3,100 miles (more than the distance from New York to Los Angeles) -- the 10th Annual Self-Transcendence 3,100 Mile Race
- Runners circle one city block in Queens, NY, for 51 days straight, lapping the block 5,649 times
- 60 miles per day; 6,000 calories per day
- The winner gets a T-shirt and a plastic trophy

What is Debugging?

- Solving runtime errors
- Also known as “bugs”
- These are errors that allow your program to compile and run, but show up when you’re testing it
- Sometimes, bugs crash the program and show up in the console:

```
Exception in thread "Thread-12"
java.lang.ArrayIndexOutOfBoundsException: 8
  at breakout.Breakout.startGame(Breakout.java:50)
  at fang.GameLoop.begin(GameLoop.java:134)
  at fang.GameWindow$StartGame.run(GameWindow.java:763)
```

- Technically, not solving compilation errors -- errors that keep the code from compiling (the red squiggly line errors in Eclipse)

```
int blah = 3.2;
```
How to Debug

1. Be able to recreate the bug (this can often be rather hard!)
   • This will give you insight into exactly what the program is doing when the bug happens
2. Try to identify the last place the code reaches during correct execution
3. Anticipate what should happen next, and compare this to what actually happens next
4. Determine out exactly where in the code the bug is, and why it happens -- then, fix it!

Breakpoints

• Double-clicking on the thin left bar in Eclipse creates a **breakpoint** at that point in the code execution

```
int r, s;
Area grid = new Area();
for (r = 0; r < getSectors(); r++) {
    for (s = 0; s < getRings(); s++) {
        ...
    }
}
```

• Clicking the “bug” icon in the toolbar will run the program, but **stop at the first breakpoint**

The Console

• Often, runtime errors appear in the console as red text
• Normally, clicking the topmost link to a file you recognize (LunarLander.java, for example) will take you to where the error is in your code
• Examples of common errors:
  • **ArrayIndexOutOfBoundsException**
    • Occurs when you try to index into an array with a negative number or a number >= the size of the array (examples: blocks[-3], or blocks[4] if blocks holds 4 or fewer things)
  • **NullPointerException**
    • Occurs when you try to use an object (normally, using the dot operator with it) without having first created that object with a **new** statement