

The background of the slide is a grayscale image of a circuit board. It features a complex network of black lines representing traces and several solid black circles representing vias or components. The circuitry is arranged in a somewhat symmetrical, horizontal pattern. A solid black horizontal band runs across the middle of the image, partially obscuring the circuit board design. In the center of this dark band, the text 'CMSC 131' is written in a large, white, sans-serif font. Below it, the text 'Fall 2018' is written in a smaller, green, monospace-style font.

# CMSC 131

Fall 2018

# Announcements

- Many new students...
  - Class webpage
  - Copy someone's notes
- First project (Hello World) is now due Wednesday 9/12

# Announcements

## Maryland Center for Women in Computing (MCWIC) “Welcome to Computing” Event

TONIGHT (9/5) from 6:00-7:30pm, CSIC 3117

All women currently enrolled in CMSC 131 and 132 are invited.

# Free Tutoring (from MCWIC)

Open to EVERYONE

- 1 on 1 tutoring by appointment:

<http://go.umd.edu/TutorRequest>

- Guided Study Sessions (CMSC 131)

Mondays 2:00 – 3:00

Tuesdays 1:00 – 2:00

# Computer Systems Overview (Software)

(This slide was supposed to be earlier!)

- Applications
- Operating system
  - Process management
  - Memory management
  - Primitive I/O
  - Windowing
  - Network control
  - Security

# Example: SimpleProgram.java

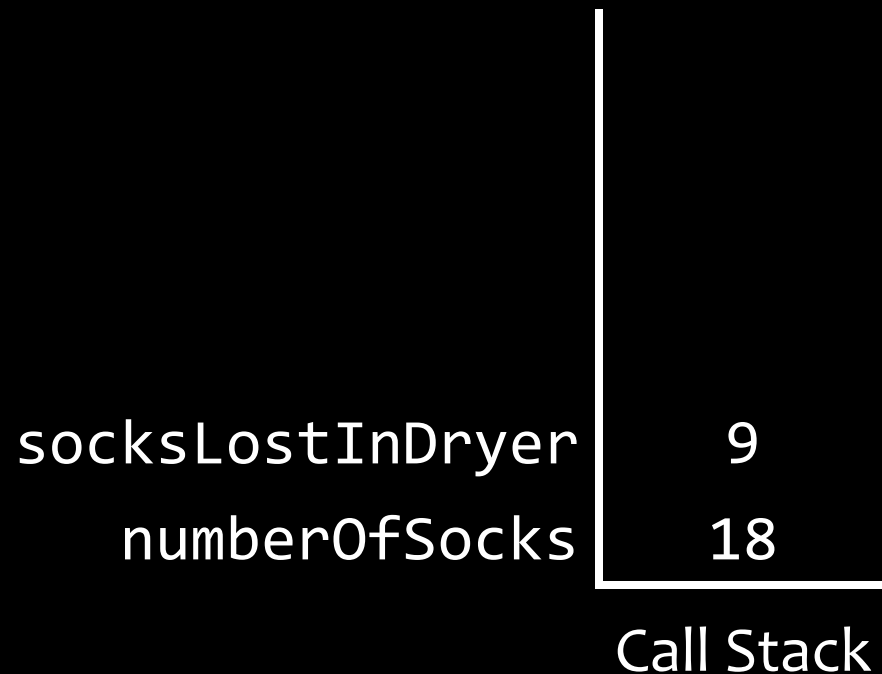
Things to observe:

- We are looking at a class called “SimpleProgram”
- There is just one method, called main
- The main method contains a few statements
- There are two kinds of “comments”
- Most statements end with semi-colons
- System.out.print is a primitive tool for text output
  - Note the difference in print vs. println

# Example: VariablesExample.java

Things to observe:

- Two “local” variables are declared (their type is “int”)
- The “assignment operator” stores values inside the variables
- Memory diagram (at the end):



# Variable Types

- Primitives (basic “atoms”)
- References to Objects (later)



# Primitive Types (Whole Numbers First)

- An `int` variable takes up 4 bytes of memory. What range of values can be stored?

Type	Memory used	Range of values that can be stored
<code>long</code>	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<code>int</code>	4 bytes	-2,147,483,648 to 2,147,483,647
<code>short</code>	2 bytes	-32,768 to 32,767
<code>byte</code>	1 byte	-128 to 127