

### Executing method calls

**Why learn how a method call is executed?**

- Gives you a better understanding of how a Java program is executed.
- You may have to execute a method call by hand during debugging.
- Knowledge of how a call is executed can help in analyzing the time and space requirements of a program.
- Will help you understand how recursion works (not part of this course).

CS101J, Cornell 0

### Executing method calls

a0 v a0 v.num('e')

name Fred String Employee

```

/** = no. times c occurs in name. */
num(char c) {
  1: int s = 0;
  // inv: s = number of c's in s[0..k-1]
  2: for (int k = 0; k < name.length(); k = k+1) {
  3: if (name.charAt(k) == c)
      4: s = s + 1;
  }
  5: return s;
}

```

method name: num  
program counter: m

scope box: where method is located:  
object name or file drawer (class) name

parameters: c

local variables: s, k

frame for the call

CS101J, Cornell 1

### Executing method calls

a0 v a0 v.num('e')

name Fred String Employee

```

/** = no. times c occurs in name. */
num(char c) {
  1: int s = 0;
  // inv: s = number of c's in s[0..k-1]
  2: for (int k = 0; k < name.length(); k = k+1) {
  3: if (name.charAt(k) == c)
      4: s = s + 1;
  }
  5: return s;
}

```

num: 1 a0

c 'e'

s ? k ?

frame for the call

**Memorize!!!!**

1. Draw frame
2. Assign args to pars
3. Execute body
4. Erase frame

CS101J, Cornell 2