LEC 08

# CSE 121

# More Methods, Parameters, and Returns

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

***Talk to your neighbors:***

*What's your favorite book?*

**Respond on sli.do!**

Music: CSE 121 26sp Lecture Tunes

**Instructor:** Matt Wang

**TAs:**

| | | | | |
|---|---|---|---|---|
| Abdul | Amogh | Anant | Anum | Cayden |
| Dalton | Ethan | Hayden | Jesse | Jessica |
| Johnathan | Minh | Navya | Paul | Reese |
| Ruslana | Sam | Savannah | Spencer | Shayna |
| Tamsyn | TJ | Trey | | |

# Agenda (1/4)

- **Announcements, Reminders** ⟵
- Methods, Parameters Warmup
- Returns Review
- Random Characters Example

# Announcements, Reminders

- C2 (Password Protector) releasing later today, due **Thursday** May 7th
  - involves conditionals (Friday's topic)
- R1 due tomorrow; R2 opens tomorrow, due Thursday, May 7th
- Quiz 1: Thursday, May 14th
  - Quiz 0 grades expected *before* Quiz 1

# Agenda (2/4)

- Announcements, Reminders
- **Methods, Parameters Warmup** ⟵
- Returns Review
- Random Characters Example

# Practice: Think

sli.do #cse121

What is the output of this program?

```
public static void main(String[] args) {
  int x = 1;
  int y = 2;
  int z = 5;

  mystery(z, y, x);

  mystery(y, x, z);
}

public static void mystery(int x, int z, int y) {
  z += x;
  System.out.println(z + " and " + (y - x));
}
```

**A.** 7 and -4
7 and 3

**B.** 6 and 3
7 and 3

**C.** 5 and 3
3 and 3

**D.** 7 and -4
3 and 3

# Practice: Pair

sli.do #cse121

What is the output of this program?

```
public static void main(String[] args) {
  int x = 1;
  int y = 2;
  int z = 5;

  mystery(z, y, x);

  mystery(y, x, z);
}

public static void mystery(int x, int z, int y) {
  z += x;
  System.out.println(z + " and " + (y - x));
}
```

**A.** 7 and -4
7 and 3

**B.** 6 and 3
7 and 3

**C.** 5 and 3
3 and 3

**D.** 7 and -4
3 and 3

# Agenda (3/4)

- Announcements, Reminders
- Methods, Parameters, Warmup
- **Returns Review** ⟵
- Random Characters Example

# PCM: Returns

**Returns** allow us to send values *out of a method*

```
public static <type> myMethod(int num) {
    System.out.print(num + " is the best!");
    ...
    return <value of correct type>
}
```

`return` does 3 things:

1. **Evaluate** the expression
2. **Return** this value to where the method was called
3. **Exit** the method

Calling a method that returns a value…

```
<type> result = myMethod(42);
```

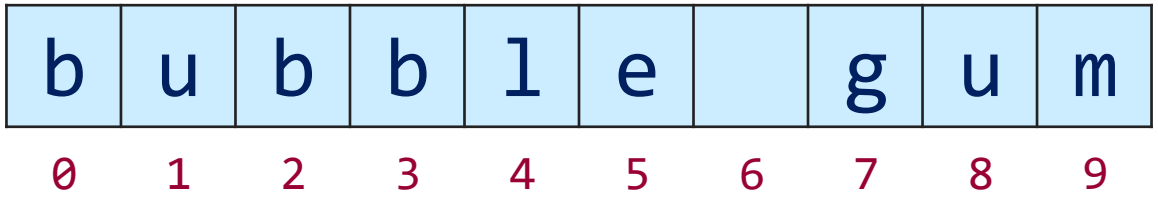
# Recall: Math

| Method | Description |
|---|---|
| `Math.abs(value)` | **Returns** the absolute value of value |
| `Math.ceil(value)` | **Returns** value rounded up |
| `Math.floor(value)` | **Returns** value rounded down |
| `Math.max(value1, value2)` | **Returns** the larger of the two values |
| `Math.min(value1, value2)` | **Returns** the smaller of the two values |
| `Math.round(value)` | **Returns** value rounded to the nearest whole number*<br>note: need to cast result to int (it's complicated!) |
| `Math.sqrt(value)` | **Returns** the square root of value |
| `Math.pow(base, exp)` | **Returns** base raised to the exp power |

# Recall: String Methods

| Method | Description |
|---|---|
| `length()` | **Returns** the length of the string. |
| `charAt(i)` | **Returns** the character at index i of the string |
| `indexOf(s)` | **Returns** the index of the first occurrence of s in the string; returns -1 if s doesn't appear in the string |
| `substring(i, j)` or `substring(i)` | **Returns** the characters in this string from i (inclusive) to j (exclusive); if j is omitted, goes until the end of the string |
| `contains(s)` | **Returns** whether or not the string contains s |
| `equals(s)` | **Returns** whether or not the string is equal to s (case-sensitive) |
| `equalsIgnoreCase(s)` | **Returns** whether or not the string is equal to s ignoring case |
| `toUpperCase()` | **Returns** an uppercase version of the string |
| `toLowerCase()` | **Returns** a lowercase version of the string |

# Recall: chaining methods in expressions




```
s.substring(7, 8).toUpperCase() + s.substring(8) + "ball"

"g".toUpperCase() + s.substring(8) + "ball"

"G" + s.substring(8) + "ball"

"G" +      "um"      + "ball"
```

# Remember: Method Comments

Each method you write (except main) should have a short comment!

```
// Behavior:
// - Calculates net profit using monthly income and daily spending.
// Parameters:
// - income: user's income this month (non-negative)
// - spending: amount spent each day this month (non-negative)
// Returns:
// - int: the net profit or loss. Positive if profit, negative if loss.
public static int calculateNetExpenses(int income, int spending) {
  return income - (spending * DAYS_IN_MONTH);
}
```

# Practice: Think

sli.do #cse121

What value is returned from this method?

```
public static int returnExample() {
  for (int i = 0; i < 5; i++) {
    return i;
  }
  return -1;
}
```

A. -1

B. 0

C. 4

D. 5

# Practice: Pair

sli.do #cse121

What value is returned from this method?

```
public static int returnExample() {
  for (int i = 0; i < 5; i++) {
    return i;
  }
  return -1;
}
```

A. -1

B. 0

C. 4

D. 5

# Agenda (4/4)

- Announcements, Reminders
- Methods, Parameters Warmup
- Returns Review
- **Random Characters Example** ⟵

# Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
  - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
  - What is it doing?
  - What do you expect it to do?
  - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?