LEC 08

CSE 121

Returns

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

Talk in chat:

Are you watching any good tv shows right now?

Music: ♣ CSE 121 25su Lecture Tunes ♣

Instructor: Hannah Swoffer

TAs: Abby, Hannah, Julia, Merav, Trey

Agenda

- **Announcements, Reminders** ⟵
- Common Problem-Solving Strategies
- Methods & Parameters Warmup
- Returns
- Temperatures Example
- Miscellaneous Tasks

Announcements, Reminders

- C2 releasing later today, due **Tuesday, July 29th**
- R3 releasing later today, due **Tuesday, July 29th**
- Quiz 1 on **Thursday, July 31st**
  - Can’t make it? Email Hannah ASAP

Agenda

- Announcements, Reminders
- **Common Problem-Solving Strategies** ⟵
- Methods & Parameters Warmup
- Returns
- Temperatures Example
- Miscellaneous Tasks

Common Problem-Solving Strategies (1/2)

- **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?

Common Problem-Solving Strategies (2/2)

- **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?

Agenda

- Announcements, Reminders
- Common Problem-Solving Strategies
- **Methods & Parameters Warmup** ⟵
- Returns
- Temperatures Example
- Miscellaneous Tasks

Practice: Think

sli.do #cse121

What is the output of this program?

```
public static void main(String[] args) {
  int x = 9;
  int y = 2;
  int z = 5;

  mystery(z, y, x);

  mystery(y, x, z);
}

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

A. 2 and 4
9 and 3

B. 5 and -7
5 and -7

C. 9 and -3
5 and -7

D. I'm lost

Practice: Pair

sli.do #cse121

What is the output of this program?

```
public static void main(String[] args) {
  int x = 9;
  int y = 2;
  int z = 5;

  mystery(z, y, x);

  mystery(y, x, z);
}

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

A. 2 and 4
9 and 3

B. 5 and -7
5 and -7

C. 9 and -3
5 and -7

D. I'm lost

Agenda

- Announcements, Reminders
- Common Problem-Solving Strategies
- Methods & Parameters Warmup
- **Returns** ⟵
- Temperatures Example
- Miscellaneous Tasks

PCM: Returns

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

```
public static <type> myMethod(...) {
    ...
    return <value of correct type>
}
```

Evaluates the expression

Returns this value to where the method is called from

Method immediately exits

Calling a method that returns a value…

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

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* 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

Reminder: Gumball & Strings

```
String s = "bubblegum";
s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";

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

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

s =                                           "G" + "um" + "ball";
```

Agenda

- Announcements, Reminders
- Common Problem-Solving Strategies
- Methods & Parameters Warmup
- Returns
- **Temperatures Example** ⟵
- Miscellaneous Tasks

Agenda

- Announcements, Reminders
- Common Problem-Solving Strategies
- Methods & Parameters Warmup
- Returns
- Temperatures Example
- **Miscellaneous Tasks** ⟵

Revisiting 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`