LEC 09

CSE 121

Conditionals

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

Talk to your neighbours:

What's your favourite book?

Music: [121 25wi lecture playlist ❄️]

Instructor: Matt Wang

TAs:
Ailsa, Alice, Chloë, Christopher,
Ethan, Hanna, Hannah, Hibbah,
Janvi, Judy, Julia, Kelsey,
Lucas, Luke, Maitreyi, Merav,
Ruslana, Samrutha, Sam, Shayna,
Sushma, Vivian

Announcements, Reminders

- C2 released, due **Thursday, Feb 13th**
- R2 out yesterday, due **Thursday, Feb 13th**
  - note: this is the last time C0 is eligible for resubmission!
- Almost half-way through – feedback wanted!
  - next Wed, Feb 12th – mid-quarter feedback, during lecture
  - next Tue, Feb 11th – quiz section feedback, during section
- Quiz reminders:
  - Quiz 0: next **Thursday, Feb 13th**
  - Quiz 1: the following **Thursday, Feb 20th**

Oh, and a quick note on class comments

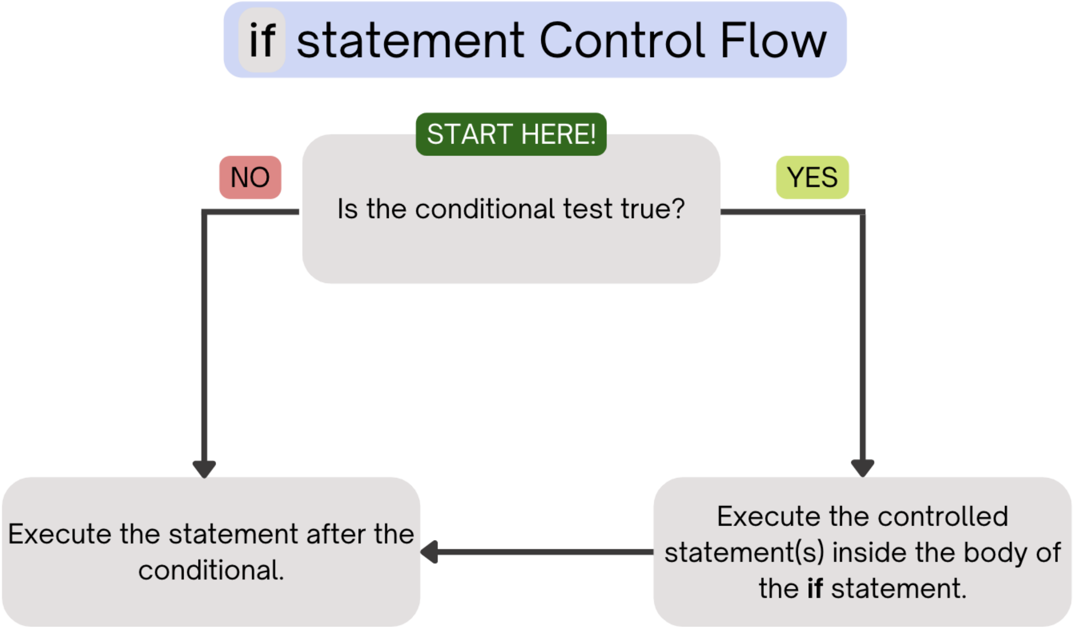
tl;dr: Matt points out that we've adjusted our class comment guidance in the commenting guide.

We'll retroactively apply this from P1 onwards.

(it's generally in your favour!)

PCM Review: if statements

```
if ( test ) {
    body (statements to be executed)
}
```

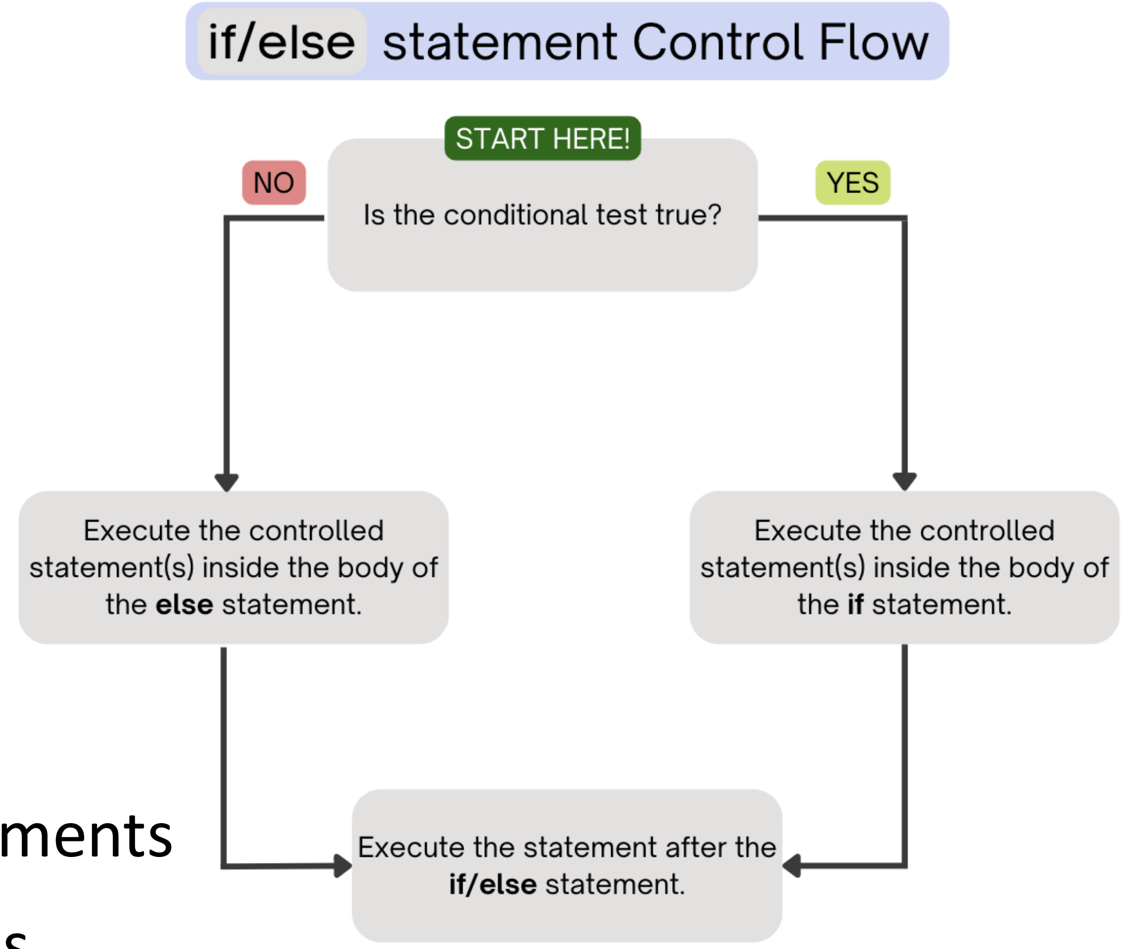

if statement Control Flow
START HERE!
Is the conditional test true?
NO
YES
Execute the statement after the conditional.
Execute the controlled statement(s) inside the body of the **if** statement.

Executes a block of statements if and only if the test is true

PCM Review: if-else

```
if ( test ) {
        statement(s)
} else {
        statement(s)
}
```

1. If the test is true: execute block of statements
2. If not, execute other block of statements



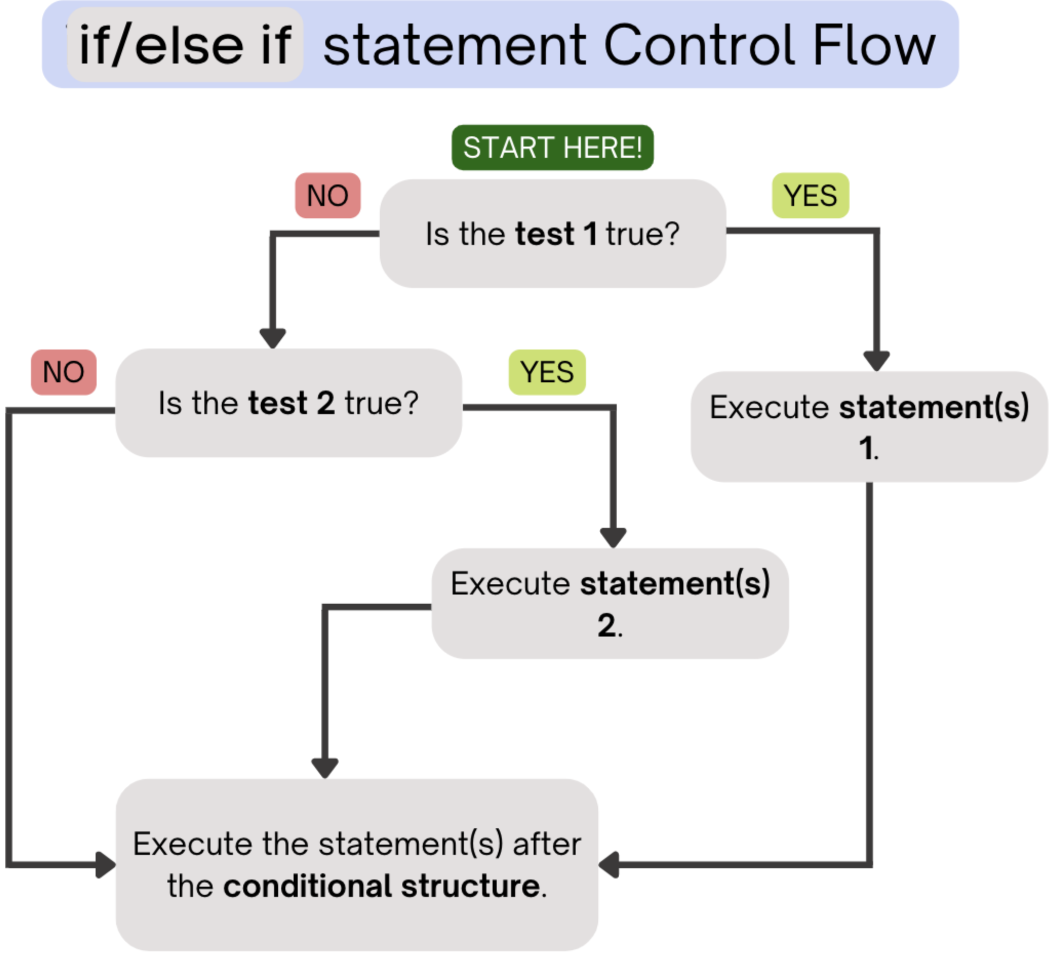
if/else statement Control Flow

START HERE!
Is the conditional test true?
NO
YES
Execute the controlled statement(s) inside the body of the **else** statement.
Execute the controlled statement(s) inside the body of the **if** statement.
Execute the statement after the **if/else** statement.

PCM Review: if-else-if

```
if ( test ) {
        statement(s)
} else if (test) {
        statement(s)
}
```

1. If the first test is true, execute that block
2. If not, proceed to the next test, and repeat
3. If none were true, don't execute any blocks


if/else if statement Control Flow
START HERE!
NO
Is the **test 1** true?
YES
NO
Is the **test 2** true?
YES
Execute **statement(s) 1**.
Execute **statement(s) 2**.
Execute the statement(s) after the **conditional structure**.

Practice: Think

sli.do #cse121

```
public static void main(String[] args) {
  for (int i = 1; i <= 3; i++) {
    System.out.print(mystery(i));
  }
}
public static String mystery(int n) {
  String response = "even ";
  if (n % 2 == 1) {
    response = "odd ";
  } else if (n == 1) {
    response = "one ";
  }
  return response;
}
```

What does this program output?

A. odd even odd

B. one even odd

C. one even even

D. even even even

Practice: Pair

sli.do #cse121

```
public static void main(String[] args) {
  for (int i = 1; i <= 3; i++) {
    System.out.print(mystery(i));
  }
}
public static String mystery(int n) {
  String response = "even ";
  if (n % 2 == 1) {
    response = "odd ";
  } else if (n == 1) {
    response = "one ";
  }
  return response;
}
```

What does this program output?

A. odd even odd

B. one even odd

C. one even even

D. even even even

“Useless” Conditionals (and “Boolean Zen”)

```
public static void main(String[] args) {
  for (int i = 1; i <= 3; i++) {
    System.out.print(mystery(i));
  }
}
public static String mystery(int n) {
  String response = "even ";
  if (n % 2 == 1) {
    response = "odd ";
  } else if (n == 1) {
    response = "one ";
  }
  return response;
}
```

This else if statement <u>never runs!</u>

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?

Announcements, Reminders (again)

- C2 released, due **Thursday, Feb 13th**
- R2 out yesterday, due **Thursday, Feb 13th**
    - note: this is the last time C0 is eligible for resubmission!
- Almost half-way through – feedback wanted!
    - next Wed, Feb 12th – mid-quarter feedback, during lecture
    - next Tue, Feb 11th – quiz section feedback, during section
- Quiz reminders:
    - Quiz 0: next **Thursday, Feb 13th**
    - Quiz 1: the following **Thursday, Feb 20th**