CSE 121 Lesson 9: Conditionals

Matt Wang & Brett Wortzman

Autumn 2024

sli.do #cse121

TAs:

Abby	Afifah	Ailsa	Alice	Aliyan	Arohan
Chloë	Christopher	Dalton	Derek	Elizabeth	Ethan
Hanna	Hannah	Heather	Hibbah	Janvi	Jasmine
Judy	Julia	Kelsey	Lucas	Luke	Mahima
Maitreyi	Maria	Merav	Minh	Neha	Ronald
Ruslana	Sahej	Sam	Samrutha	Sushma	Vivian
Yijia	Zachary				

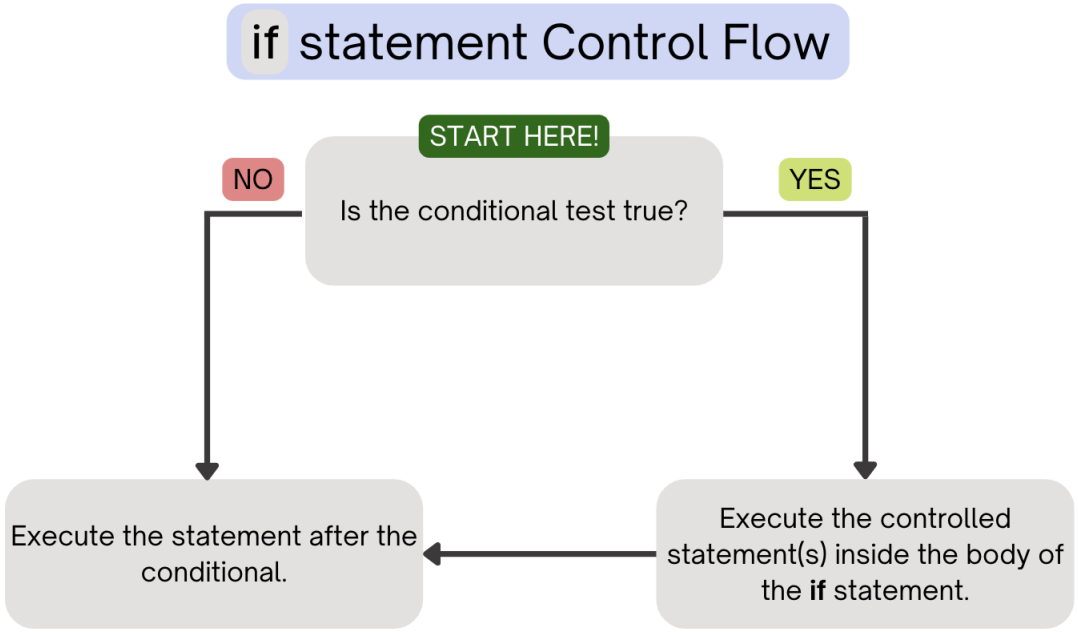
Today's playlist:
121 24au lecture tunes

Announcements, Reminders

- Creative Project 2 released, due Tuesday, Oct 29th
  - note: doable *without* conditionals, but you're free to use them!
- R2 out yesterday, due Thursday Oct 31st
  - Note: this is the last time C0 is eligible for resubmission!
- Almost half-way through – feedback wanted!
  - Friday, Nov 1st – mid-quarter feedback, during lecture
  - next week during quiz section – quiz section & TA feedback

(PCM) Conditionals (1/4)

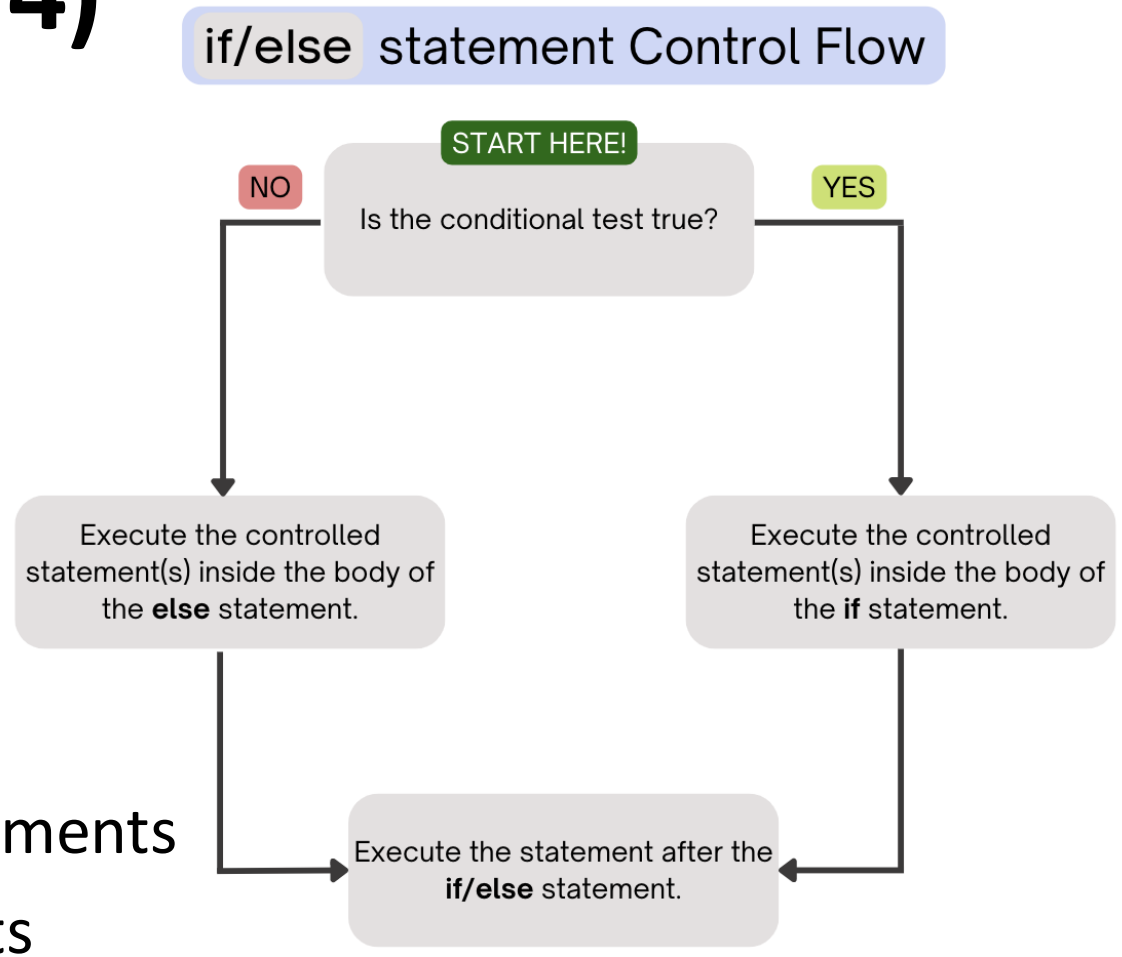
```
if ( test ) {
    body (statements to be executed)
}
```



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

(PCM) Conditionals (2/4)

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

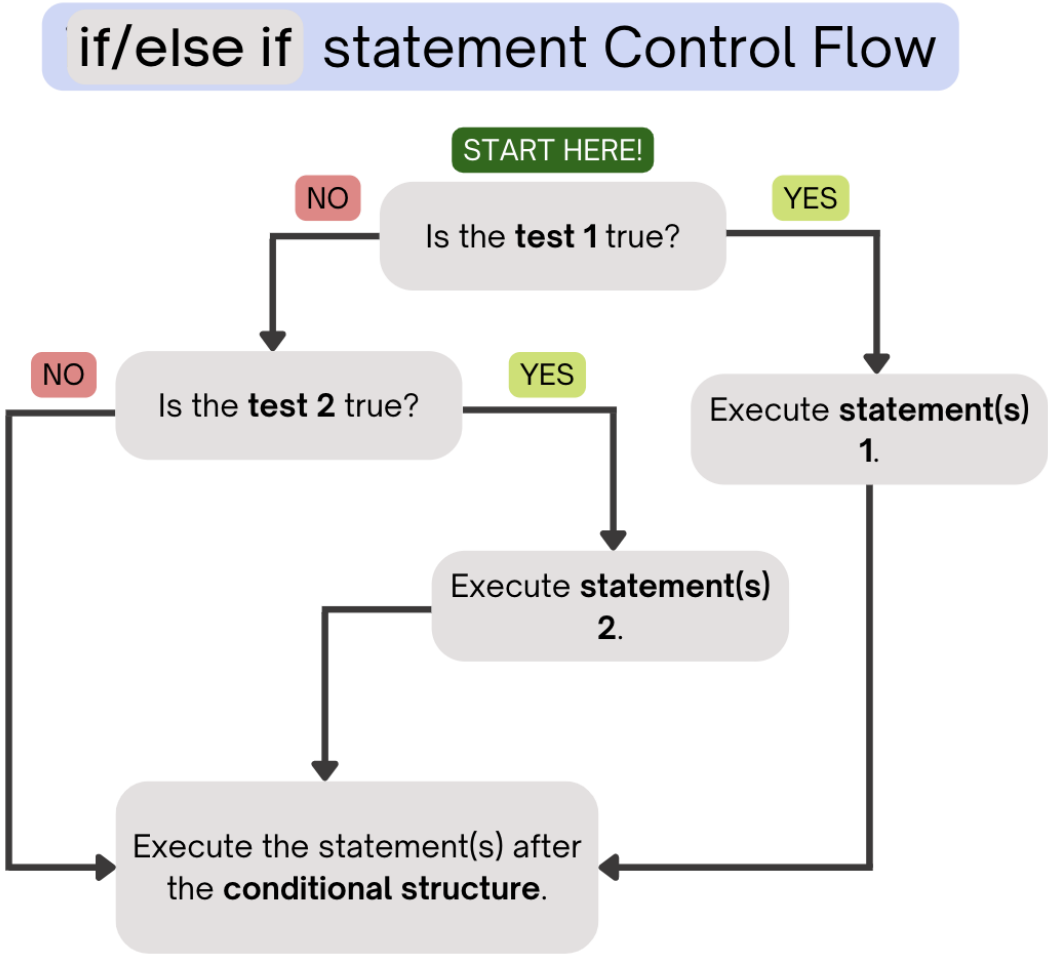


1. If the test is true: execute block of statements
2. If not, execute other block of statements

(PCM) Conditionals (3/4)

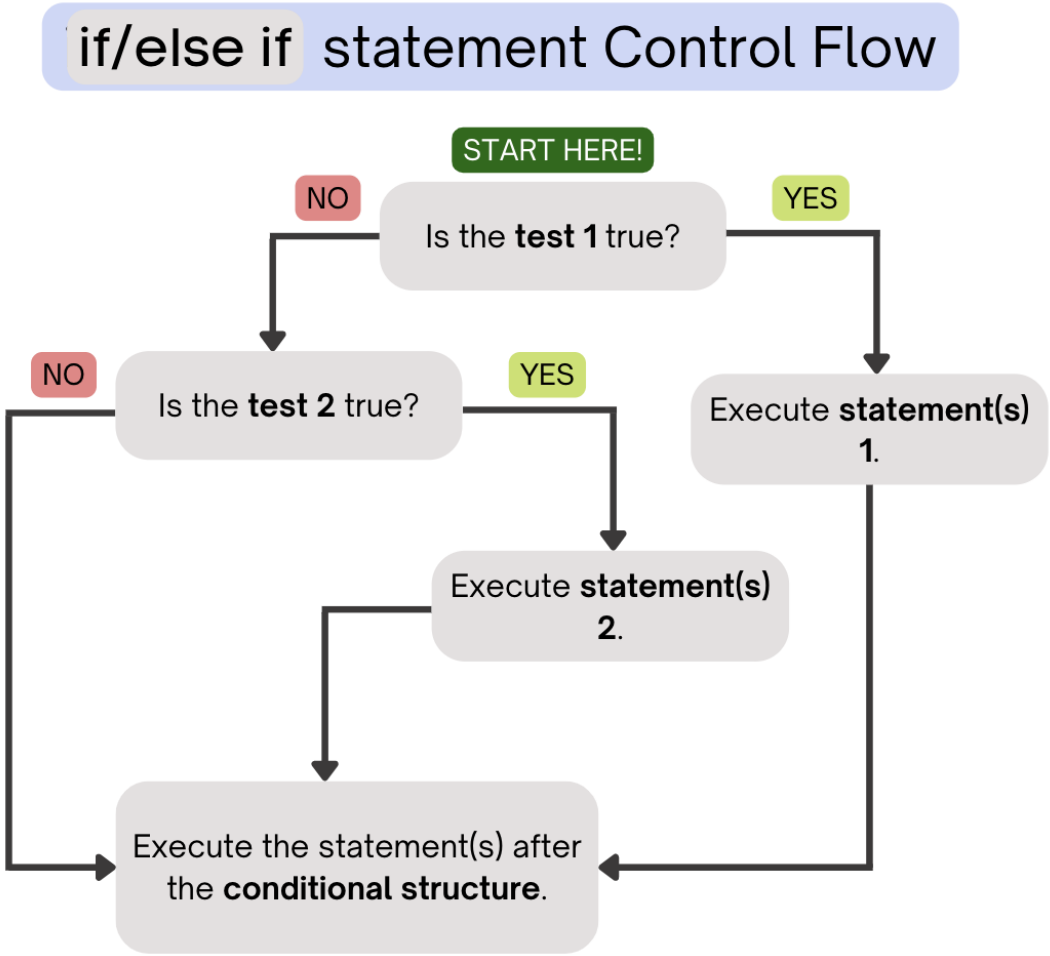
```
if ( test ) {
        statement(s)
} else if (test) {
        statement(s)
}
```

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



(PCM) Conditionals (4/4)

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



With a large if-else-if-else chain,

- if there is an ending else, exactly one block will execute
- if there is no ending else, zero or one blocks will execute

Poll in with your answer!

```
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

sli.do #cse121

Poll in with your answer!

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;
}
```

This else if statement never runs!

Common Problem-Solving Strategies (1/4)

- **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 (2/4)

- **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 (3/4)

- **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 (4/4)

- **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?