

LEC 09

CSE 121

Conditionals

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbors:**How are you doing? (like, actually)*Music:  [CSE 121 25au Lecture Tunes](#) **Instructors:** Brett Wortzman & James Weichert

TAs:	Trey	Ava	Caleb	Elden	Anya
	Amogh	Reese	Anum	Suyash	Minh
	Samrutha	Hayden	Abdul	Sthiti	TJ
	Dalton	Aki	Janvi	Paul	Zach
	Ailsa	Spencer	Navya	Shayna	Cayden
	Ryan	Savannah	Sam	Jesse	Johnathan
	Anant	Tamsyn	Jessica	Nhan	

Announcements, Reminders

- **C2** released, due Thursday, October 30th
- **R2** out yesterday, due **Thursday, October 30th**
 - this is the last time C0 is eligible for resubmission
- Quiz reminders:
 - Quiz 0 grades will be released *end* of next week at the earliest
 - Quiz 1 is on **Thursday, November 6th**

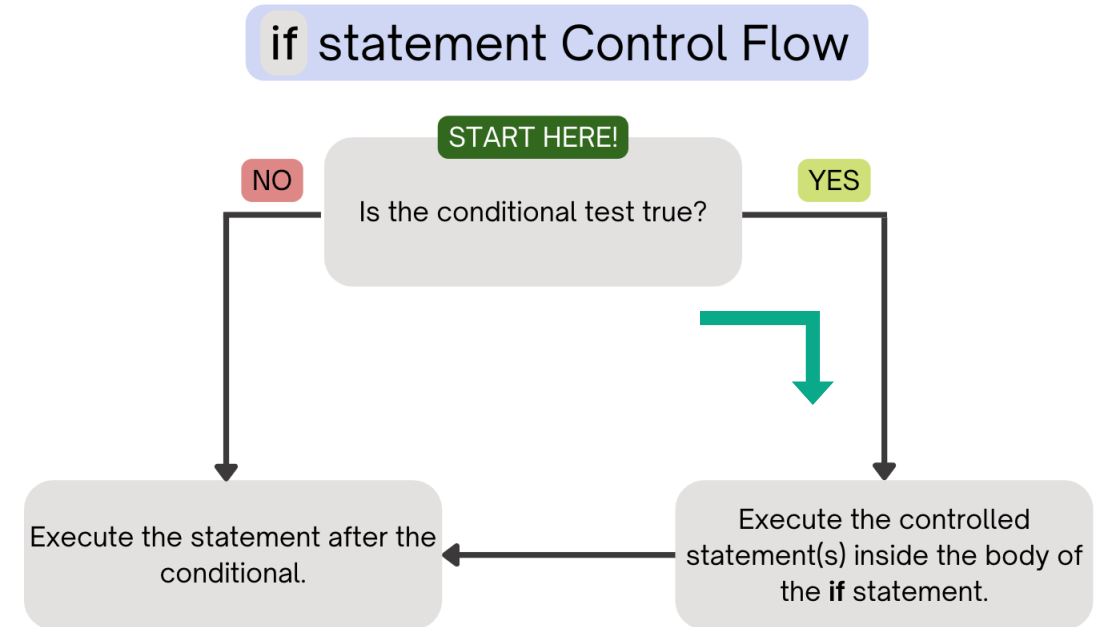
More Reminders

- **Getting help and support** — needing help is *expected*!
 - TAs are awesome and very helpful in the [IPL](#)!
 - Instructor office hours
 - we **love** when students come to office hours!
 - James has special “coffee chats” on Mondays — come talk about 121 or *anything* else
 - Asynchronously via Ed and email
- **Course expectations**
 - Copying *anything* from AI into assignments is **academic dishonesty**
 - results in: student conduct case, “U” grade (can’t resubmit)

PCM Review: if Statements

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

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

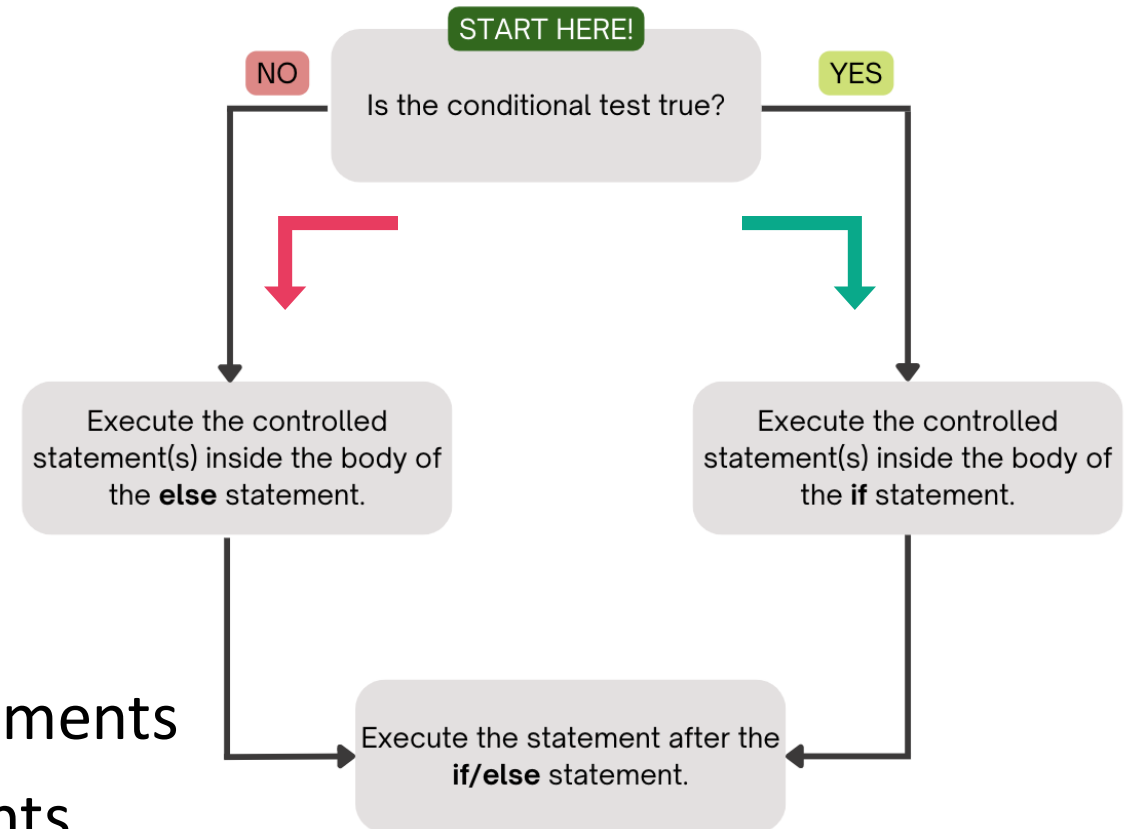


PCM Review: if-else

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

1. If the test is **true**: execute block of statements
2. If **false**, execute other block of statements

if/else statement Control Flow

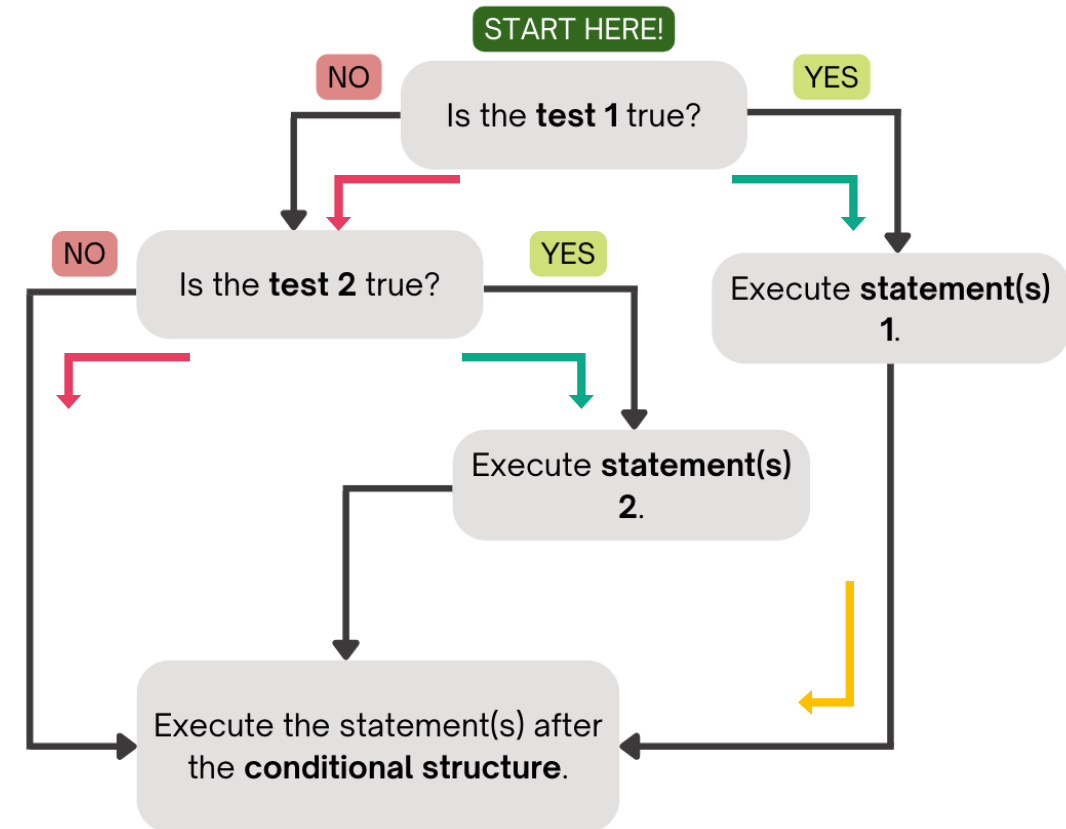


PCM Review: if-else if

```
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



Note After entering an if /else if block, **skip** all remaining blocks.



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

```
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 (1/2)

- **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", then 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", then 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?