

LEC 09

CSE 121

Random, 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 26au Lecture Tunes](#) 🍂**Instructor:** Miya Natsuhara and Anjali Pal

TAs:	Samrutha	Tamsyn	Owen	Jennie
	William	Cedric	Jameson	Anisha
	Samvedha	Anum	Paul	TJ
	Syreeta	Navya	Gurbax	Ivory
	Ryan	Gene	Tony	Sarah W
	Anant	Margot	Lucas	Avery
	Henry	Sarah K	Alex	
	Reese	Ethan	Jesse	
	Savannah	Evelyn	Matthew	

Announcements, Reminders

- **CQ0 : Today through next Wednesday**
 - Schedule through Canvas
 - You **must** sign up for a timeslot by 11:59pm the night before
- **PQ0 : Tuesday, October 20th**

Randomness

Having a computer generate truly random numbers is hard!

[LavaRand](#): CloudFlare's
Wall of Lava Lamps



Pseudo-Randomness

Having a computer generate truly random numbers is hard!

Instead, computers generate numbers that “look random” in a predictable way, using mathematical formulas

- can use “external” variables like time, mouse position, etc.
- if we “fix” these variables, we can reproduce the same behavior
 - very important for testing!

Randomness is very important: applications in cryptography, security, machine learning, and more!

PCM: Random

The Random class helps us generate pseudo-random numbers.

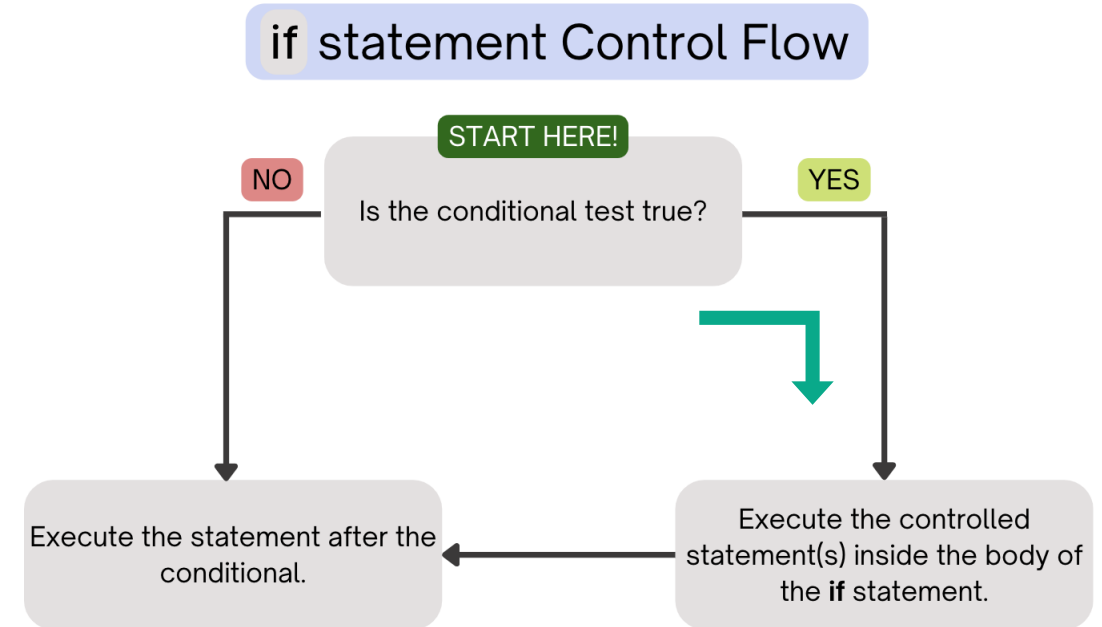
- We create a new Random number generator by calling `new Random()`
- First, we need to **import** the Random class with `import java.util.*;`
- We can “seed” the generator to make it behave *deterministically*

Method	Description
<code>nextInt()</code> <code>nextInt(max)</code> <code>nextInt(min, max)</code>	Returns a random integer Returns a random integer in the range [0, max) Returns a random integer in the range [min, max)
<code>nextDouble()</code> <code>nextDouble(max)</code> <code>nextDouble(min, max)</code>	Returns a random double in the range [0.0, 1.0) Returns a random double in the range [0.0, max) Returns a random double in the range [min, max)

PCM Review: if Statements

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

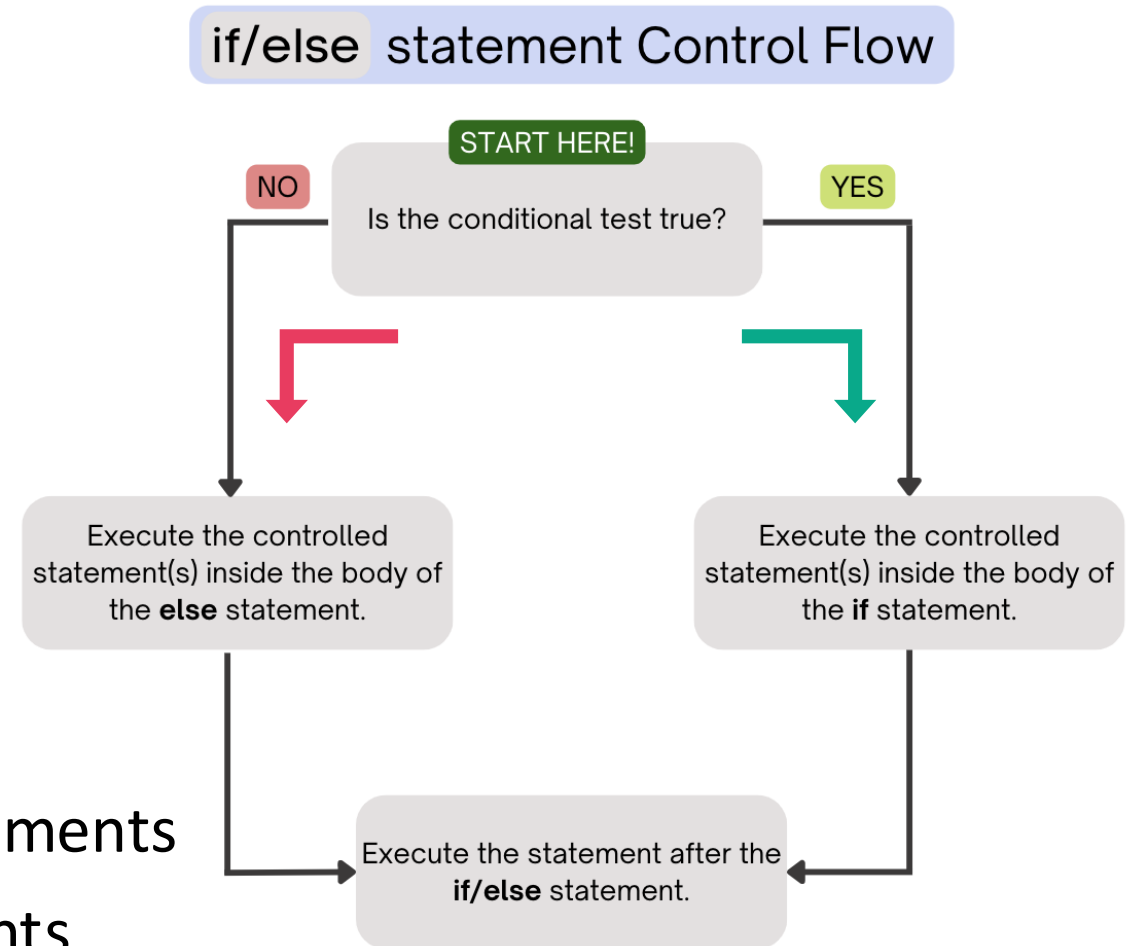
Executes a block of statements
if and only if the test is **true**



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

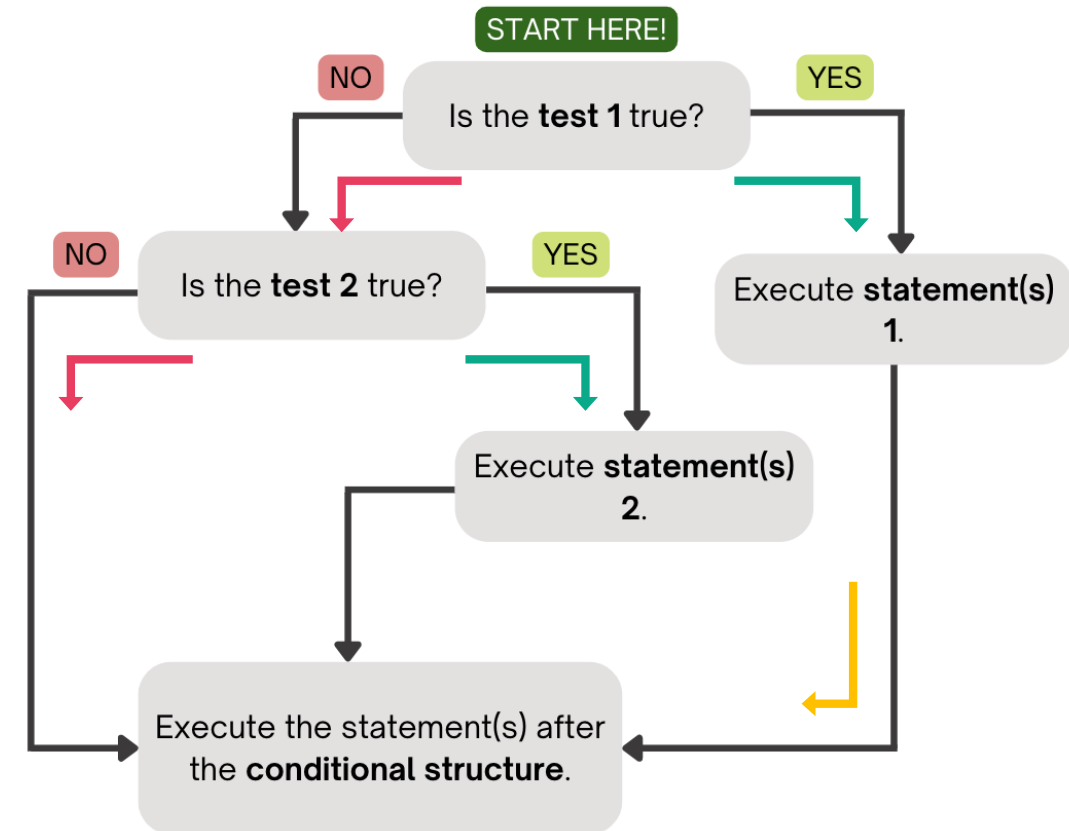


PCM Review: if-else if

```
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.

Reminder: Boolean Operators!

- ! (NOT)
- && (AND)
- || (OR)
- relational: <, >, <=, >=, ==, !=

Very common with conditionals

```
if (day.equals("Wed") || day.equals("Fri")) {  
    System.out.println("121 lecture!");  
} else if (day.equals("Tue") || day.equals("Thu")) {  
    System.out.println("121 quiz section!");  
} else {  
    System.out.println("no 121!");  
}
```



Practice: Pair

sli.do

#cse121

Files + Dice.java

```
1 // Anjali Pal
2 // 10/14/2026
3 // CSE 121
4 // TA: Samrutha Babu
5 // Think-Pair-Share
6 // Generates two random dice rolls and determines whether the sum is seven.
7
8 public class Dice {
9     public static void main(String[] args) {
10         // 1. Make a new Random instance
11
12         // 2. Declare two integer variables with randomly generated integers
13         // between 1 and 6 (inclusive) to simulate rolling two dice
14
15         // 3. Declare a variable for the sum of the two dice rolls
16
17         // 4. Print out a string that says
18         // "2 Dice Roll: __ | Is Equal to Seven? __"
19         // (fill in the blanks with your variables)
20
21         // Hint: Remember == is the boolean operator that
22         // checks equality between two values.
23     }
24 }
```

1. Work with a neighbor to fill in the program.

2. Keep clicking Run until the program prints **true.**

3. On slido: how many times does it take?



Practice: Think



sli.do #cse121

```
Random rand = new Random();
int a = 2;
int b = rand.nextInt(10);
if (a % 2 == 0) {
    System.out.print("even ");
} else if (a == 2) {
    System.out.print("2 ");
} else if (b == 10) {
    System.out.print("10 ");
}
System.out.print(a + " ");
if (a * b % 2 == 1) {
    System.out.print("odd");
}
```

What does this program output?

- A. even 2
- B. even 2 2
- C. even 2 10 2 odd
- D. even 2 odd
- E. The output depends on what random integer b is.
- F. None of the above



Practice: Pair



sli.do #cse121

```
Random rand = new Random();
int a = 2;
int b = rand.nextInt(10);
if (a % 2 == 0) {
    System.out.print("even ");
} else if (a == 2) {
    System.out.print("2 ");
} else if (b == 10) {
    System.out.print("10 ");
}
System.out.print(a + " ");
if (a * b % 2 == 1) {
    System.out.print("odd");
}
```

What does this program output?

- A. even 2
- B. even 2 2
- C. even 2 10 2 odd
- D. even 2 odd
- E. The output depends on what random integer b is.
- F. None of the above

“Useless” Conditionals

```
public static void main(String[] args) {  
    int n = 1;  
    String response = "even";  
    if (n % 2 == 1) {  
        response = "odd ";  
    } else if (n == 1) {  
        response = "one ";  
    }  
    System.out.println(response);  
}
```

← This else if statement never runs!