

LEC 05

CSE 121

Nested Loops, Cumulative Sum, Scope, Class Constants

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbors:**What's your favorite dessert?*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	

Agenda (1/4)

- **Announcements, Reminders**
- Fencepost Review
- Nested for Loop Practice
- Addition Problem

Announcements, Reminders

- **P1** will be released later today, due Tues, Oct 27
- **Computer Quiz 1** (related to P1) will be running Wed, Oct 28 – Wed, Nov 4

Agenda (2/4)

- Announcements, Reminders
- **Fencepost Review**
- Nested for Loops
- Addition Problem

Wed PCM: String Traversals

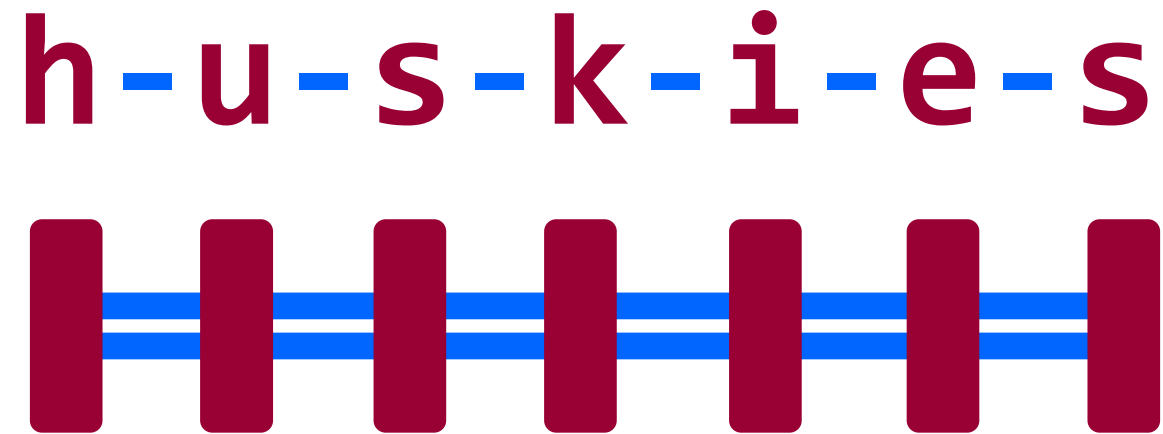
```
// For some String s  
for (int i = 0; i < s.length(); i++) {  
    // do something with s.charAt(i)  
}
```

Go Huskies?

h-u-s-k-i-e-s

The Fencepost Pattern

Some task where one piece is repeated n times, and another piece is repeated $n-1$ times and they alternate



Agenda (3/4)

- Announcements, Reminders
- Fencepost Review
- **Nested for Loops**
- Addition Problem

Fri PCM: for Loops!

For loops are our first **control structure**: a syntax *structure* that *controls* the execution of other statements.

```
for ( initialization ; test ; update ) {  
    body (statements to be repeated)  
}
```

PCM: Nested for Loops

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {  
    System.out.println("outer loop iteration #" + outerLoop);  
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {  
        System.out.println("    inner loop iteration #" + innerLoop);  
    }  
    System.out.println(outerLoop);  
}
```

PCM: Nested for Loops, “Inner Loop”

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {  
    System.out.println("outer loop iteration #" + outerLoop);  
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {  
        System.out.println("    inner loop iteration #" + innerLoop);  
    }  
    System.out.println(outerLoop);  
}
```

inner
loop

outer
loop



Practice: Think

[sli.do](#)

#cse121

What output is produced by the following code?

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(i);  
    }  
    System.out.println();  
}
```

A. 1
12
123
1234
12345

B. i
ii
iii
iiii
iiiii

C. 1
22
333
4444
55555

D. 1
11
111
1111
11111



Practice: Pair

sli.do

#cse121

What output is produced by the following code?

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(i);  
    }  
    System.out.println();  
}
```

A. 1
12
123
1234
12345

B. i
ii
iii
iiii
iiiii

C. 1
22
333
4444
55555

D. 1
11
111
1111
11111



Practice: Think



sli.do

#cse121

What code produces the following output?

A.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(i);  
    }  
    System.out.println();  
}
```

B.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

C.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; i <= j; j++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

D.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; i++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

```
1  
12  
123  
1234  
12345
```



Practice: Pair



sli.do

#cse121

What code produces the following output?

A.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(i);  
    }  
    System.out.println();  
}
```

B.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; j++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

C.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; i <= j; j++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

D.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= i; i++) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

```
1  
12  
123  
1234  
12345
```

PCM: Scope

Scope: the part of a program where a variable exists (and can thus be referenced, modified, or used).

- General Rule: a variable ‘exists’ from its **declaration** to the **closing brace of that indentation level** }
- Exception applies to for loops (header is “part” of the body)

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {  
    System.out.println("outer loop iteration #" + outerLoop);  
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {  
        System.out.println("    inner loop iteration #" + innerLoop);  
    }  
    System.out.println(outerLoop);  
}
```

innerloop's scope

outerloop's scope

Agenda (4/4)

- Announcements, Reminders
- Fencepost Review
- Nested for Loops
- **Addition Problem**

PCM: Class Constants

A fixed value visible (in-scope) to the whole program (the entire *class*).

Value is set at declaration, **cannot** be reassigned – value is ***constant***.

```
public static final type NAME_OF_CONSTANT = expression;
```