

LEC 06

CSE 121

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 hobby?*

Instructor: Hayden Feeney**TAs:** Anisha Savannah William
Cayden Tamsyn

Agenda (1/4)

- **Announcements, Reminders** ←
- Math Facts continued
- Scope, class constants review
- Code examples
 - Cumulative sums!

Announcements, Reminders

- Resubmission Cycle 0 (R0) due tomorrow, Thursday, July 16th
- P1: Election Simulator out today, due Tuesday, July 21st
- Quiz 0 is **tomorrow**, Thursday, July 16th (in your registered quiz section)
 - can't make it? email Hayden ASAP!!

A bit more on Quiz 0

- Taken on paper, in your registered quiz section
 - see [Ed post on quiz logistics \(in detail\)](#)
- Broadly: focused on concepts, reading, writing, and debugging code
- Main topics: printing, datatypes, expressions, variables, Strings, for loops
- You get to bring one sheet of notes (8.5x11in, or standard A4)
- [Ed resources tab](#) has reference sheet, practice quizzes, and practice quiz key

A bit more on P1

1. this is a big jump from C1. **Start early!**
2. read the spec carefully (maybe take notes!)
 - make sure you understand what you're doing *before* you code
3. there will be another development slide to help you do *iterative development*, along with a recommended development strategy!
 - you don't have to follow it, but we recommend making *some* plan for how to approach the project (since it's more substantial than previous projects!)

Agenda (2/4)

- Announcements, Reminders
- **Math Facts continued** ←
- Scope, class constants review
- Code examples
 - Cumulative sums!

Agenda (3/4)

- Announcements, Reminders
- Math Facts continued
- **Scope, class constants review** ←
- Code examples
 - Cumulative sums!

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



Practice: Think

sli.do

#cse121

What variables are in scope where the `***` is?

```
int x = 1;
for (int i = 0; i < 10; i++) {
    for (int j = 0; j < 5; j++) {
        int y = 2;
        System.out.println("Inner loop!");
    }
    // ***
    int z = 3;
}
```

A. x, i, j, y

B. x, i, j

C. x, i, z

D. x, i



Practice: Pair

sli.do

#cse121

What variables are in scope where the `***` is?

```
int x = 1;
for (int i = 0; i < 10; i++) {
    for (int j = 0; j < 5; j++) {
        int y = 2;
        System.out.println("Inner loop!");
    }
    // ***
    int z = 3;
}
```

A. x, i, j, y

B. x, i, j

C. x, i, z

D. x, i

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;
```

Agenda (4/4)

- Announcements, Reminders
- Math Facts continued
- Scope, class constants review
- **Code examples** 
 - **Cumulative sums!**

New: Cumulative Sum pattern

Where we *accumulate* information into a variable while a loop is running!

- In order to do this, we need to declare the variable *outside* of the loop

```
int sum = 0;
for (int i = 1; i <= 10; i++) {
    sum += i;
}
System.out.println(sum);
```



sum's scope!

What is the scope of `sum`?