

LEC 06

CSE 121

Methods and Parameters

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

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

Agenda

- **Announcements, Reminders** ←
- Methods, Parameters Review
- Code example

Announcements, Reminders

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

A bit more on Quiz 0

- Taken on paper, in your registered quiz section
 - [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)
- See Ed for [provided reference sheet](#), [practice quizzes](#), and [practice quiz solutions](#)

A bit more on P1

1. this is a big jump from C1. **Start early!**
2. Read the spec carefully, take notes
 - Make sure you understand what you're doing *before* you start coding
3. P1 does *not* require you to use methods (though you can!).
 - advice: feeling shaky on writing methods? don't do it for P1

Agenda

- Announcements, Reminders
- Practice Problem
- **Method, Parameter Review** ←
- Code example

PCM: Methods

Writing our own **methods** allows us to define our own commands!
(naming conventions for methods are same as variables: camelCased)

```
public static void myMethod() {  
    /**  
    Your code here  
    **/  
}
```



Practice: Think



sli.do

#cse121

```
public class HelloGoodbye {  
    public static void main(String[] args) {  
        welcome();  
        hello();  
        goodbye();  
    }  
  
    public static void hello() {  
        System.out.print("Hello! ");  
        glad();  
    }  
  
    public static void goodbye() {  
        System.out.println("Goodbye!");  
    }  
  
    public static void welcome() {  
        System.out.print("Welcome! ");  
        glad();  
    }  
  
    public static void glad() {  
        System.out.println("Glad you're here.");  
    }  
}
```

What is the output of this program?

A.

Welcome! Glad you're here.
Hello! Glad you're here.
Goodbye!

C.

Welcome! Hello! Goodbye!

B.

Welcome!
Hello!
Goodbye!

D.

Welcome!
Glad you're here.
Hello!
Glad you're here.
Goodbye!



Practice: Think



sli.do

#cse121

```
public class HelloGoodbye {  
    public static void main(String[] args) {  
        welcome();  
        hello();  
        goodbye();  
    }  
  
    public static void hello() {  
        System.out.print("Hello! ");  
        glad();  
    }  
  
    public static void goodbye() {  
        System.out.println("Goodbye!");  
    }  
  
    public static void welcome() {  
        System.out.print("Welcome! ");  
        glad();  
    }  
  
    public static void glad() {  
        System.out.println("Glad you're here.");  
    }  
}
```

What is the output of this program?

A.

Welcome! Glad you're here.
Hello! Glad you're here.
Goodbye!

C.

Welcome! Hello! Goodbye!

B.

Welcome!
Hello!
Goodbye!

D.

Welcome!
Glad you're here.
Hello!
Glad you're here.
Goodbye!

PCM: Parameters

Definition: a value passed to a method by its caller. “Like” a variable!

```
public static void myMethod(String musicalAct) {  
    System.out.print(musicalAct + " is the best!");  
    ...  
}
```

Calling a method with a parameter...

```
myMethod("Laufey"); // prints: Laufey is the best!
```

PCM: Scope, Redux

Our scope rules also apply to methods and parameters!

- General rule: from its **declaration to the next closing brace, }**
- a variable declared in a method only exists in that method!

```
public static void example(int n) {  
    System.out.println("hello");  
    int x = 3;  
    for (int i = 1; i <= n; i++) {  
        System.out.print(x);  
    }  
}
```

i's scope

x's scope

n's scope

Agenda

- Announcements, Reminders
- Practice Problem
- Method, Parameter Review
- **Code example** ←

New: Method Comments

Each method you write (except main) should have a short comment!

```
// Randomly generates an addition problem where the  
// operands are in the range 1-10 (inclusive),  
// and prints the result, rounded to two decimal places.
```

```
public static void addTwoRandomNumbers() {  
    Random randy = new Random();  
    int num1 = randy.nextInt(10) + 1;  
    int num2 = randy.nextInt(10) + 1;  
    int sum = num1 + num2;  
    ...  
}
```

New: 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;
```