CSE 121 Lesson 6: Methods & Parameters

Matt Wang & Brett Wortzman

Autumn 2024

sli.do #cse121

TAs:

Abby	Afifah	Ailsa	Alice	Aliyan	Arohan
Chloë	Christopher	Dalton	Derek	Elizabeth	Ethan
Hanna	Hannah	Heather	Hibbah	Janvi	Jasmine
Judy	Julia	Kelsey	Lucas	Luke	Mahima
Maitreyi	Maria	Merav	Minh	Neha	Ronald
Ruslana	Sahej	Sam	Samrutha	Sushma	Vivian
Yijia	Zachary				

Today's playlist:
121 24au lecture tunes

Announcements & Reminders

- Resubmission Cycle 0 (R0) due tomorrow, October 17
- Programming Assignment 1 (P1) out today, due October 22
  - note: a big jump from C1. Start early!
- Quiz 0 next week in section (Thursday, October 24)
  - Topics: everything up to and including today's lecture
  - More review in upcoming sections & lectures
- Post-section work required for extra resub: 12 -> 10

Last Time 1

- Nested for loops
  - Syntax & conventions: (i , j , k)
  - Applications: “doing the same thing for multiple iterations”

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {
    System.out.println("outer loop iteration #" + outerLoop);
    for (int innerLoop = 1; innerLoop <= 7; innerLoop++) {
        System.out.println("    inner loop iteration #" + innerLoop);
    }
    System.out.println(outerLoop);
}
```

Last Time 2

- Random
  - A `Random` object generates *pseudo*-random numbers
  - `nextInt(max)` returns random `int` value [0, max)
    i.e. between 0 and max-1


```
rand.nextInt(6) + 1
```

(PCM) Methods

Writing our own **methods** allow us to define our own statements / commands in Java!

- Naming conventions for methods are the same as variables: `camelCased`

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

Poll in with your answer!

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!
```

B.
```
Welcome!
Hello!
Goodbye!
```

C.
```
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

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

Calling a method with a parameter…

```
myMethod("Rush"); // Rush is the best!
```

(PCM) Scope

- Definition: The part of a program where a variable exists (and can thus be referenced/modified/used).
  - From its **declaration to the end of the { } braces** (kind of)
  - Ex: a variable declared in a for loop only exists <u>in that loop</u>!

innerloop's scope

```
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);
}
```

outerloop's scope

(PCM) Scope

- Definition: The part of a program where a variable exists (and can thus be referenced/modified/used).
  - From its **declaration to the end of the { } braces** (kind of)
  - Ex: a variable declared in a method exists only in that method!

```
public static void example() {
    System.out.println("hello");
    int x = 3;
    for (int i = 1; i <= 10; i++) {
        System.out.print(x);
    }
}
```

i's scope

x's scope

Method Comments!

- Now that we know how to write methods, we have a new form of documentation (using comments) to write.
- Each method you write (except for main) should be accompanied by a short comment that describes what it does.

```
// 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;
    ...
}
```

Class Constants

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

- Value can be set only at declaration; **cannot** be reassigned (so the value is constant)

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