CSE 121 – Lesson 6

Elba Garza & Matt Wang

Winter 2024

TAs: Abby Aishah Anju Annie Archit Ayesha Christian
Hannah Heather Hibbah Jacob James Janvi Jasmine
Jonus Julia Lucas Luke Maria Nicole Shananda
Shayna Trey Vidhi Vivian

sli.do #cse121-6

Today’s playlist:
CSE 121 24wi lecture beats :D
Announcements, Reminders

• Resubmission Cycle 0 (R0) due tomorrow, Jan 25th
  • Eligible for submission: C0 & P0
  • Even if you're not resubmitting – read your feedback!!

• Programming Assignment 1 (P1) releasing today (due January 30th)

• Quiz 0 is coming up fast! February 1st in Quiz Section
  • Topics: Printing, Data Types, … , Methods, Parameters, Returns
Last Time 1

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

```java
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 \([0, \text{max})\)
    i.e. between 0 and \(\text{max}-1\)

```java
Random rand = new Random();
rand.nextInt(6) + 1
```
Last Time 3

• Pseudo-Randomness

Randomness drives:
• Encryption
• OS Security
• Protocols

For more information, read this linked blog post by CloudFlare
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

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

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

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

Calling a method with a parameter...

```java
myMethod("Olivia Rodrigo"); // Olivia Rodrigo is the best!
```
Scope 1

- **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**
  - Ex: a variable declared in a `for` loop only exists **in that loop**!

```java
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);
}
```
Scope 2

- 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**
  - Ex: a variable declared in a method exists only in that method!

```java
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
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;
```
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;
    ...
}