Middlebury

CSCI 201: Data Structures

Spring 2025

Lecture 1M: Introduction

Goals for today

- Introduce ourselves.
- Introduce the course:
  - What will we be doing and how?
  - What will you know at the end of the semester?
- Use a **compiler** to create bytecode from human-readable code.
- Run our first `Java` program!
- **Define** and **assign** `int` variables.
- Use increment (`++`) and decrement (`--`) operators.
- **Print** information using `System.out.println`.
- Group blocks of code using curly braces `{}`.
- Always remember to use semi-colons `;` to end statements!

Getting to know each other

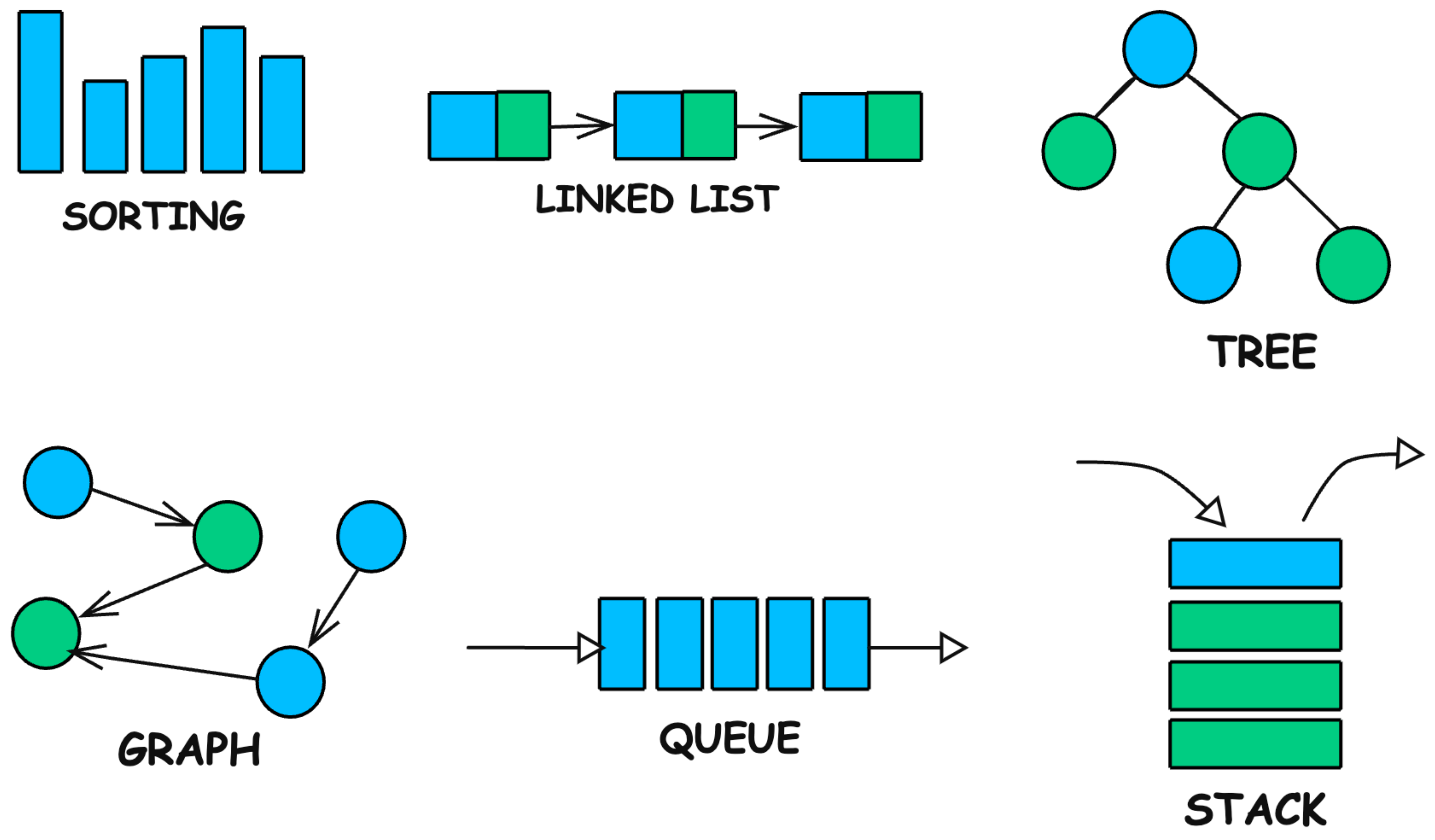
In groups of 3-4:

- Introduce yourselves!
- What is something fun you did this break or last summer?
- **Warmup:** how would you add two positive numbers `a` and `b` without using `a + b` or `a - (-b)`?

Course objectives

Primary goal: enhance skills to design, develop, debug, and test programs that are **efficient** in *time* and *memory*.

- We'll look at arrays, maps, linked lists, stacks, queues, trees and graphs.
- Work with your peers to solve programming problems.
- Debug your code when it doesn't work.



Course content

Data Structures	Algorithms	Java
- Arrays - Lists: ArrayList and LinkedList - Sets: HashSet and TreeSet - Maps: HashMap and TreeMap - Stacks, Queues - Priority Queues, Heaps - Trees: Binary Search Trees - Graphs	- Iterative - Hashing - Big-O Analysis - Recursive - Sorting - Greedy	- Java API - Objects, Classes - Interfaces, Inheritance - Testing, Debugging

Course site

go/cs201

See Syllabus and Calendar

Course logistics

- **Lectures:** please bring a computer to each class and lab
- **Homeworks** (40%, most weeks):
  - Longer programming problems: **start early!**
  - Submitted on Gradescope, due on Thursdays.
- **Labs** (10%, most Fridays):
  - Short exercises usually completed in pairs to practice with content from the week.
  - Submitted on Gradescope, due by Monday night.
- **Exams:**
  - Midterm (20%): in class portion Wed 4/2, programming portion due Thur 4/3.
  - Final (25%): during final exam week.
- **Participation** (5%, informed by self-evaluation):
  - Consider what your participation goals are this semester in CS 201 (attendance, engagement in exercises, asking questions, meeting with study group, etc.).

Late Policy

For homeworks and labs:

- No deduction if submitted within 24 hours of the due date.
- 20% deduction for 2 days late (eg, homework by Saturday midnight).
- 50% deduction for 3-7 days late
  (eg, homework submitted late on Sunday through Thursday).
- No credit after 1 week.

Extensions for extenuating circumstances, such as
medical emergencies must be requested **before the due date**.

You are expected to attend the lab, which will usually be completed in pairs.
If you are unable to attend (e.g. illness or scheduled event), please email me beforehand.

Honor Code and Generative AI (see Syllabus for full policy)

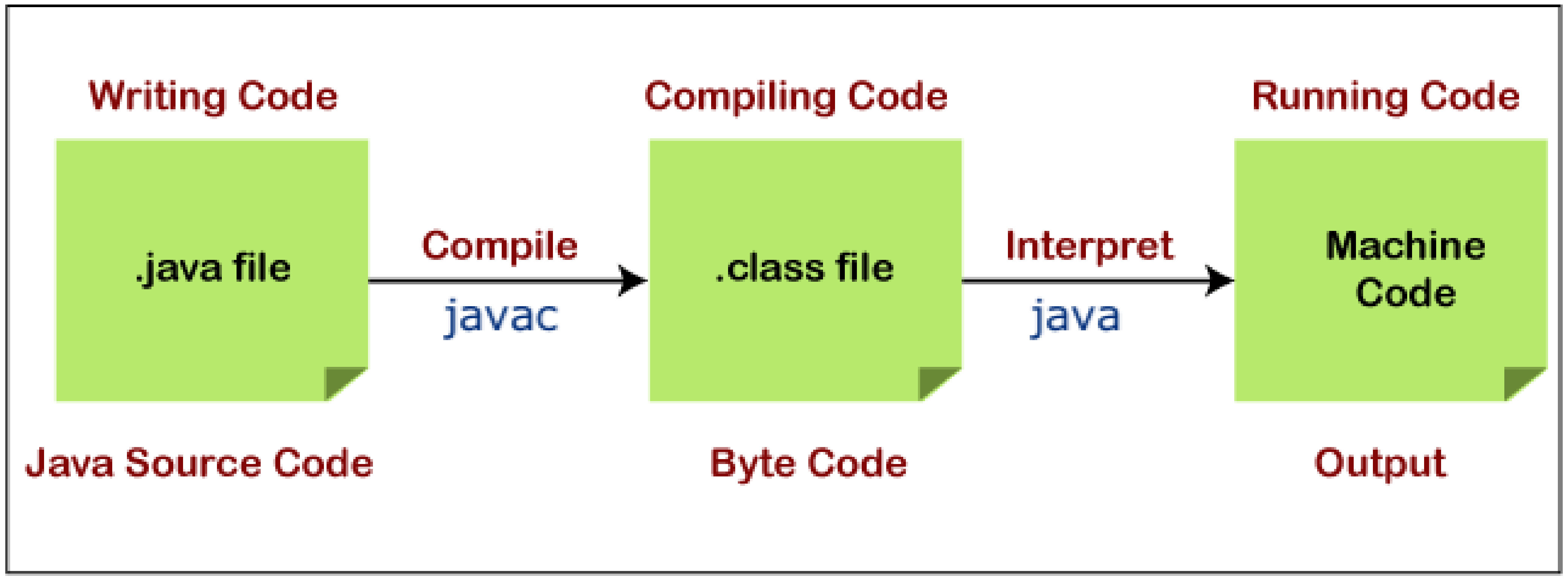
- You can map out ideas and chat with your peers about assignments.
- You should never write code for someone else or let someone else write code for you (or exchange code in email or texts).

Generative AI

- You can use AI to help with your understanding of theoretical concepts.
- You are not allowed to use AI to write code for you.

We need to compile our Java source code.

(from javatpoint.com)



You will learn how to parse compiler error messages.

The compiler is your friend!

compiling advantages:

- performance
- portability

Anatomy of a Java program

Just for today (since we're not set up with VS Code yet), we'll use onecompiler.com.

Visit go/cs201 and in the row for today, click on exercise.

```java
import java.util.*;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- Indentation is for humans (unlike `Python` where it defines blocks of code).
- Blocks of code grouped by curly braces `{}`.
- `main` is the entry-point of the program.
- `public`, `static`: we'll talk about these soon.

END EVERY STATEMENT WITH A SEMICOLON.

;

Some built-in types: `int` and `String`

- We need to **declare** the `type` of every variable (unlike `Python`).
- These declarations occur **before** the variable name.
- A `String` is a sequence of characters (`char`).
- A `char` is also a built-in type and is delimited by single quotes, e.g. `'H'`.
- `String`s are delimited by double-quotes, e.g. `"Hello, World!"`.
- Can use `+` to concatenate two `String`s.
- `System.out.println` will also know what to do with `int` variables.

```java
import java.util.*;

public class HelloWorld {
    public static void main(String[] args) {
        int year = 2025;
        String name = "World";
        System.out.println("Hello, " + name + "! It's " + year);
    }
}
```

Exercise: add a new line to print the next year

```java
import java.util.*;

public class HelloWorld {
    public static void main(String[] args) {
        int year = 2025;
        String name = "World";
        System.out.println("Hello, " + name + "! It's " + year);
    }
}
```

```java
import java.util.*;

public class HelloWorld {
    public static void main(String[] args) {
        int year = 2025;
        String name = "CS 201";
        System.out.println("Hello, " + name + "! It's " + year);
        year = year + 1;
        System.out.println("Next year, it will be " + year);
    }
}
```

We can also use increment (++) and decrement (−−) operators.

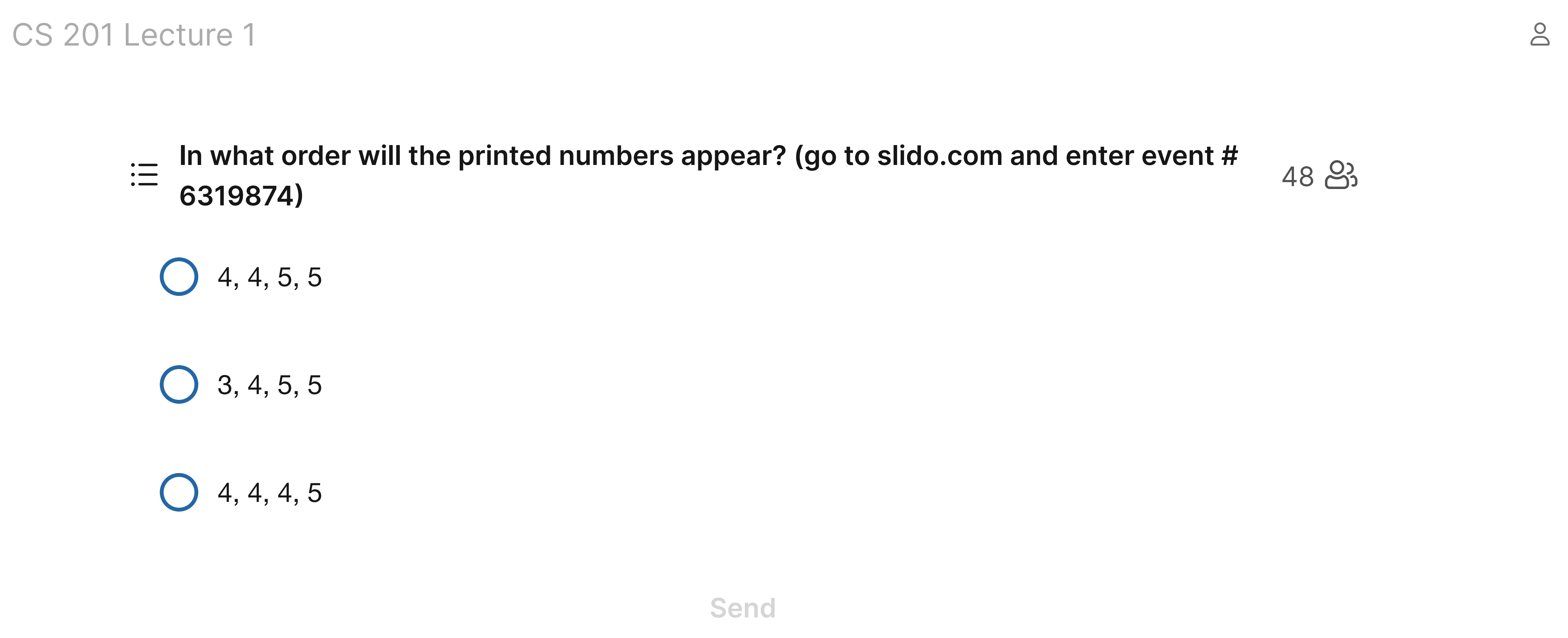
```java
import java.util.*;

public class HelloWorld {
    public static void main(String[] args) {
      int year = 2025;
      String name = "CS 201";
      System.out.println("Hello, " + name + "! It's " + year);
      year++; // same as year = year + 1
      // ++year // <-- we can also write this!
      System.out.println("Next year, it will be " + year);
   }
}
```

but year++ and ++year are not equivalent.

- `newYear = year++` increments `year` *after* assigning it to `newYear`.
- `newYear = ++year` increments `year` *before* assigning it to `newYear`.

What is printed here?

```java
public class PrintExample {
    public static void main(String[] args) {
        int x = 3;
        System.out.println(x++);
        System.out.println(x);
        System.out.println(++x);
        System.out.println(x);
    }
}
```



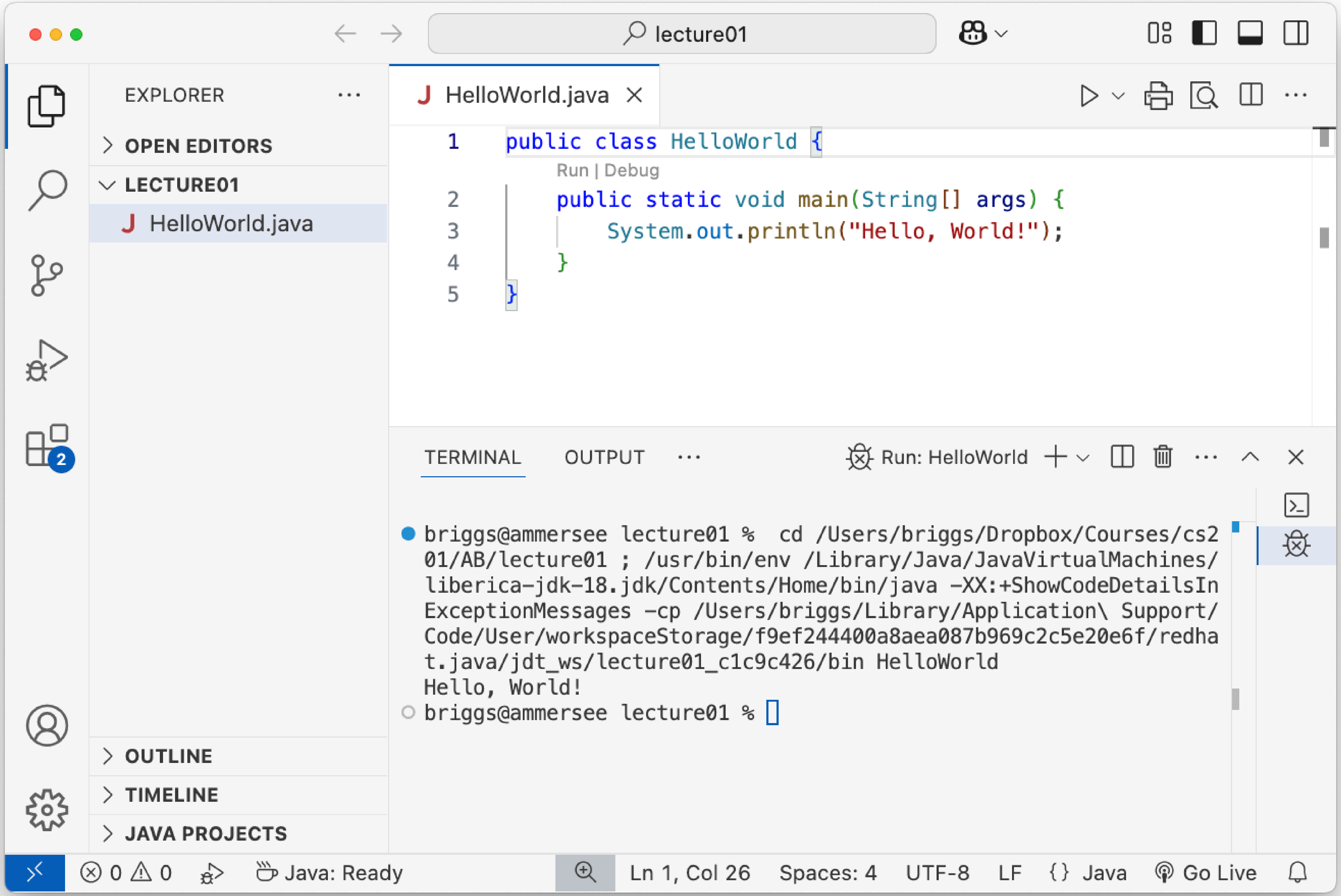
CS 201 Lecture 1

In what order will the printed numbers appear? (go to slido.com and enter event # 6319874)

48

- ○ 4, 4, 5, 5
- ○ 3, 4, 5, 5
- ○ 4, 4, 4, 5

Your main task before Wednesday: setup VS Code



```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

```
briggs@ammersee lecture01 %  cd /Users/briggs/Dropbox/Courses/cs2
01/AB/lecture01 ; /usr/bin/env /Library/Java/JavaVirtualMachines/
liberica-jdk-18.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsIn
ExceptionMessages -cp /Users/briggs/Library/Application\ Support/
Code/User/workspaceStorage/f9ef244400a8aea087b969c2c5e20e6f/redha
t.java/jdt_ws/lecture01_c1c9c426/bin HelloWorld
Hello, World!
briggs@ammersee lecture01 %
```

See you on Wednesday!

- Bookmark go/cs201!
- Familiarize yourself with syllabus, calendar, notes from today.
- Complete steps on Setup page.
- Sign-up for our Campuswire discussion board.
- Complete the Introduction Form.