Lecture 2: Reference Types and Semantics

CS 2110, Matt Eichhorn and Leah Perlmutter

August 28, 2025

Poll Everywhere

PollEv.com/2110fa25 text `2110fa25` to `22333`

Word cloud: What are some of your favorite foods?

(opportunity to set up poll everywhere)

Announcements

- All-gender restrooms on campus!

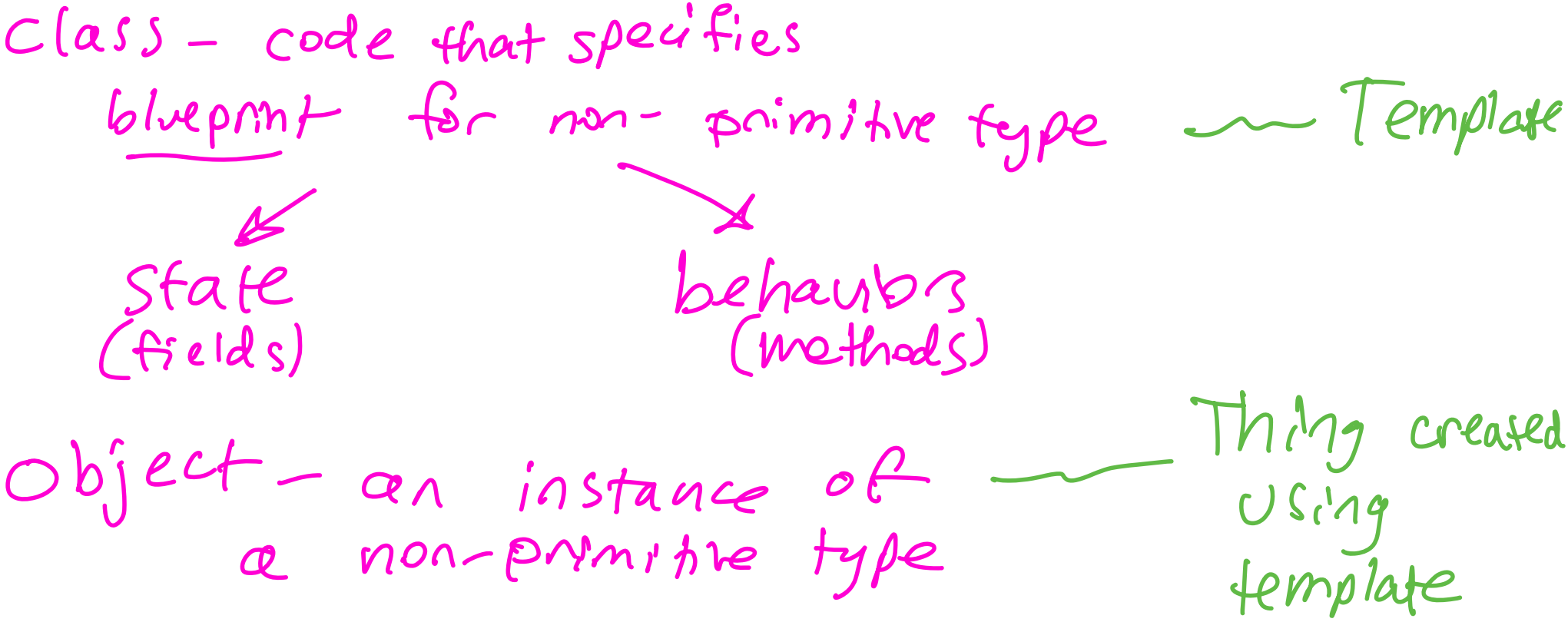
Announcements

- Assignment 1 (A1) released
  - see course website --> Assignments
- Office hours starting today
  - see course website --> About --> Office Hours
  - Campus maps: CIS
- Anonymous feedback form tinyurl.com/2110-feedback
  - Can be found on course homepage

Today's Learning Outcomes

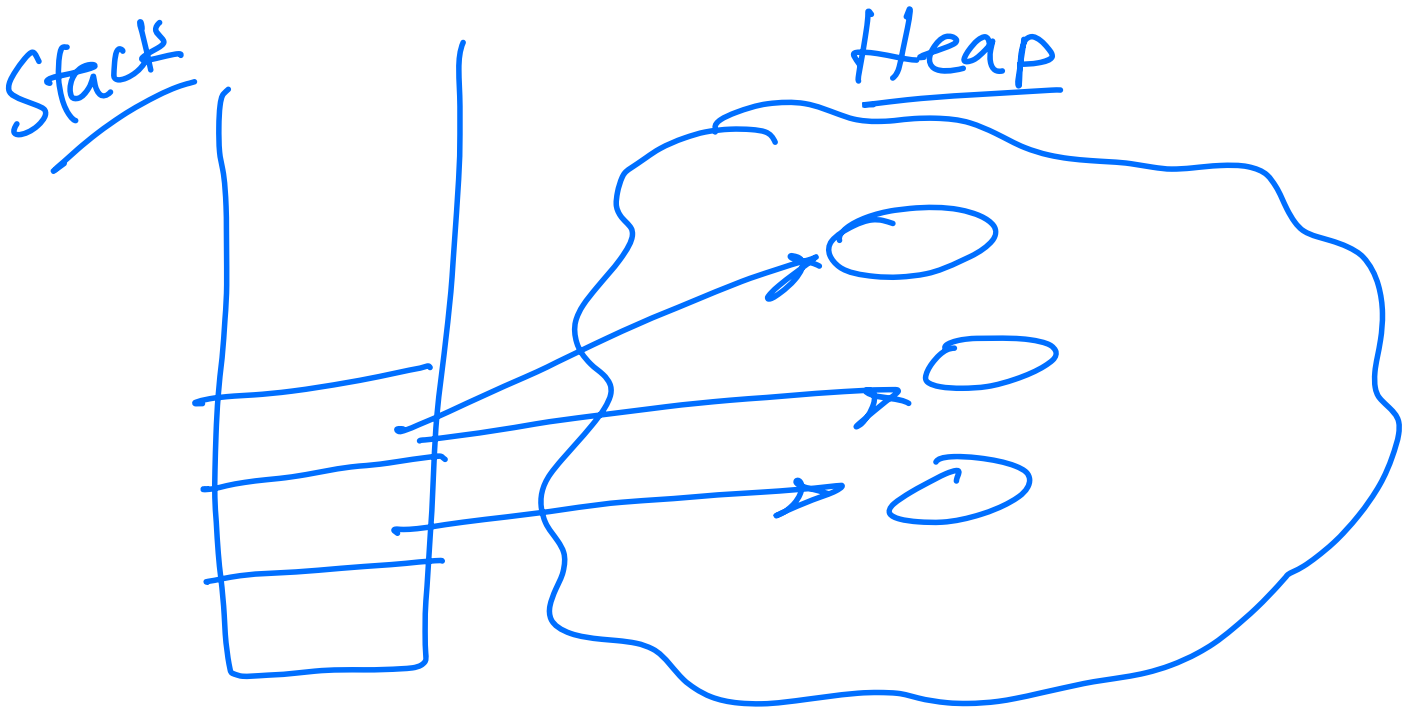
1. Explain how Java stores and references objects in memory.
2. Use the Java documentation to locate a method for a desired behavior and successfully incorporate that method into a program.
3. Draw object diagrams to visualize reference types.
4. Write Java programs involving arrays that utilize syntax for indexing and array literals.
5. Write and execute code in Java that uses program arguments.

Objects


class - code that specifies
blueprint for non-primitive type
Template
state
(fields)
behaviors
(methods)
object - an instance of
a non-primitive type
Thing created
using
template

Interlude: Computer Memory

- Memory is a physical place in the computer hardware where data is stored

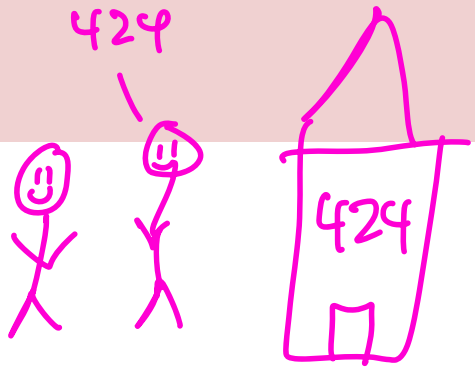


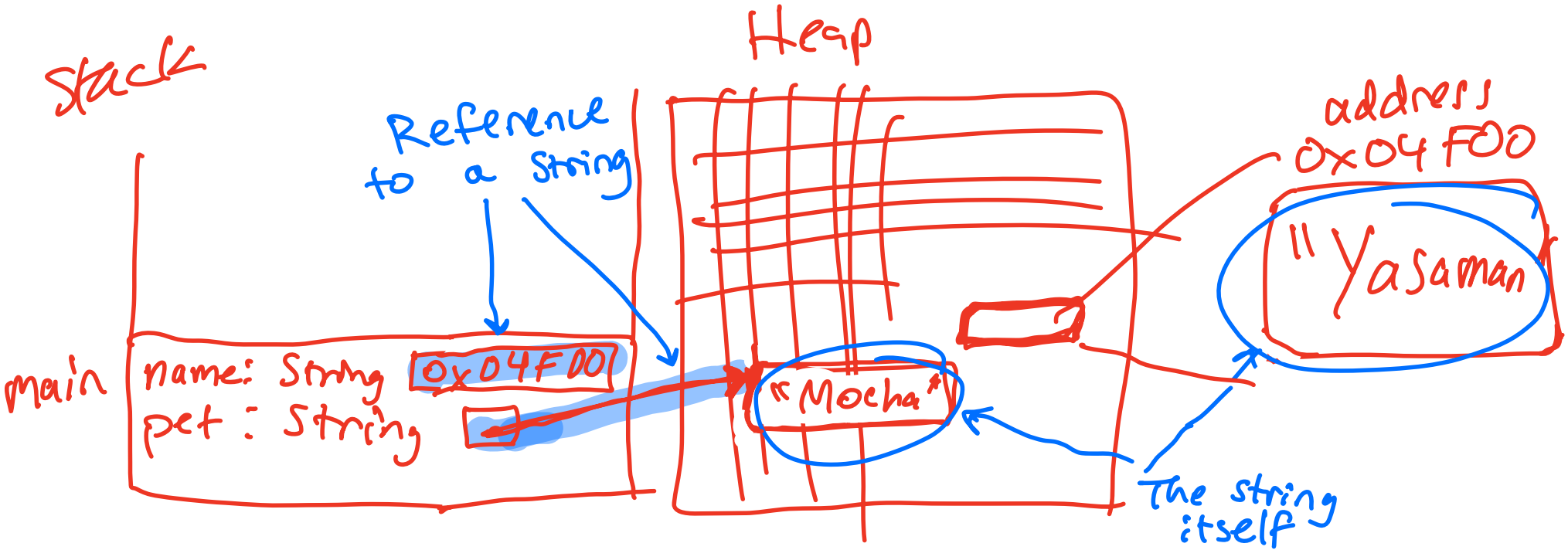
Java Virtual Machine (JVM)

↳ program that executes java programs

Construction

```
String name = new String("Yasaman");
String pet = new String("Mocha");
```





Java Library Documentation

```
// get the string "Computer" from "Computer Science"
public static void main(String[] args) {
   String str1 = "Computer Science";
   int pos = str1.indexOf(' ');
   String str2 = ____________________________________;
   System.out.print(str2);
}
```

Instance Methods

- Virtual field trip: Java Library Documentation!

Java Library Documentation

```
// get the string "Computer" from "Computer Science"
public static void main(String[] args) {
   String str1 = "Computer Science";
   int pos = str1.indexOf(' ');
   String str2 =                                    ;
   System.out.print(str2);
}
```

Interactive!

`public String substring(int beginIndex, int endIndex)`
Returns a string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.

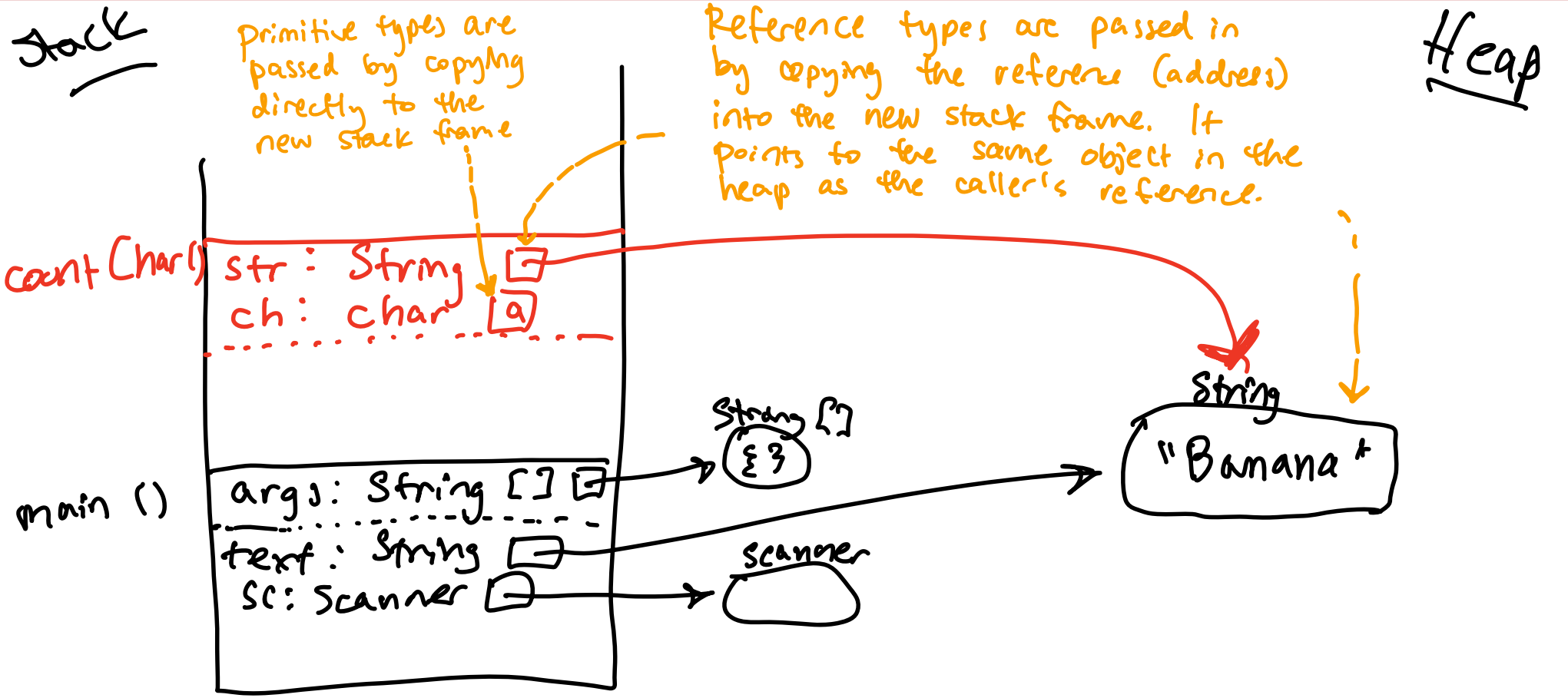
Diagramming (exercise for the student!)

```
// get the string "Computer" from "Computer Science"
public static void main(String[] args) {
  String str1 = "Computer Science";
  int pos = str1.indexOf(' ');
  String str2 = str1.substring(0,pos);
  System.out.print(str2);
}
```

Reference Semantics

```
public class CountChar {
  static int countChar(String str, char ch) {
    int count = 0;
    for (int i = 0; i < str.length(); i++) {
      if (str.charAt(i) == ch) {
        count += 1;
      }
    }
    return count;
  }
  public static void main(String[] args) {
    System.out.print("Enter some text: ");
    String text;
    try (Scanner sc = new Scanner(System.in)) {
        text = sc.nextLine();
    }
    int count = countChar(text,'a');
    System.out.print("The text you entered contains " + count + " 'a'");
    System.out.print((count == 1) ? "." : "s.");
  }
}
```

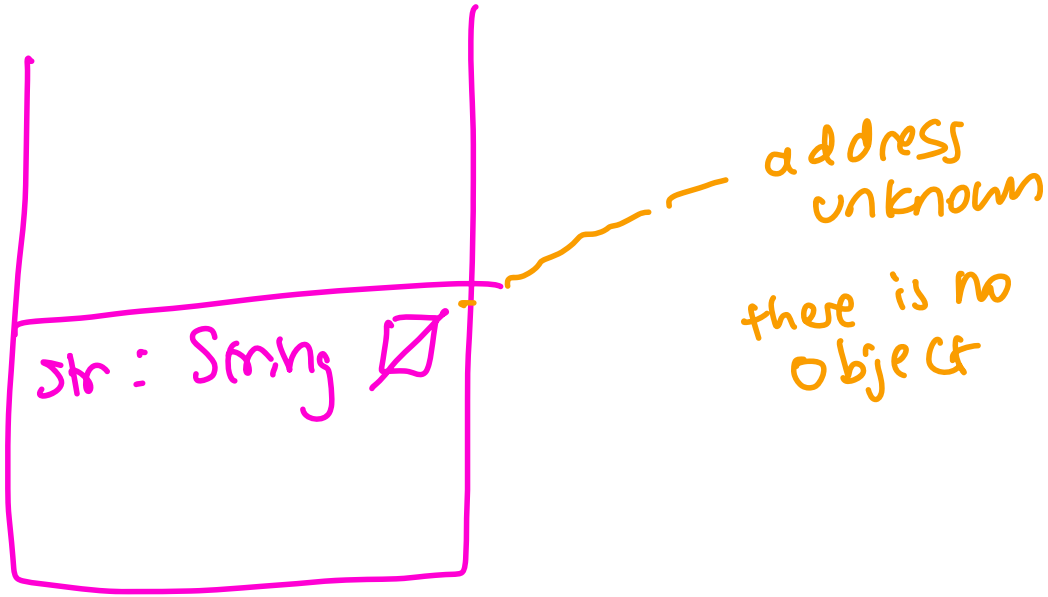
Reference Semantics of method calls



null

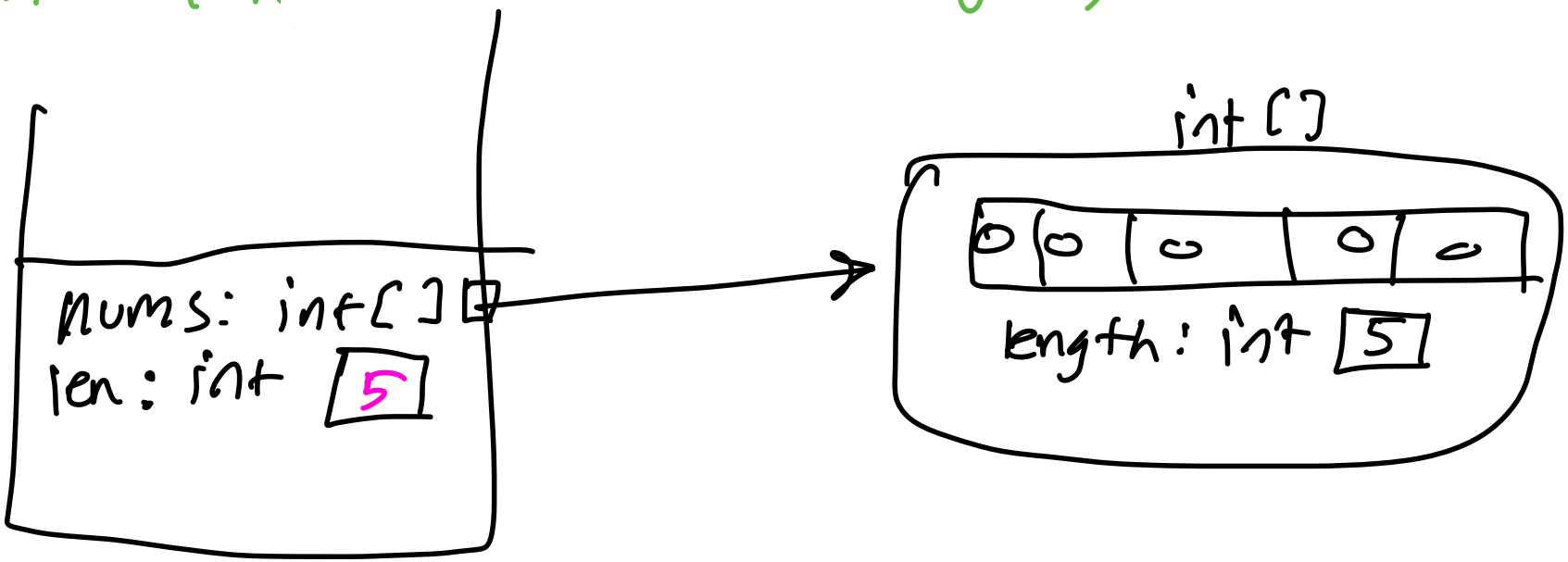
```
String str = null;
int len = str.length();
```

← throws exception!



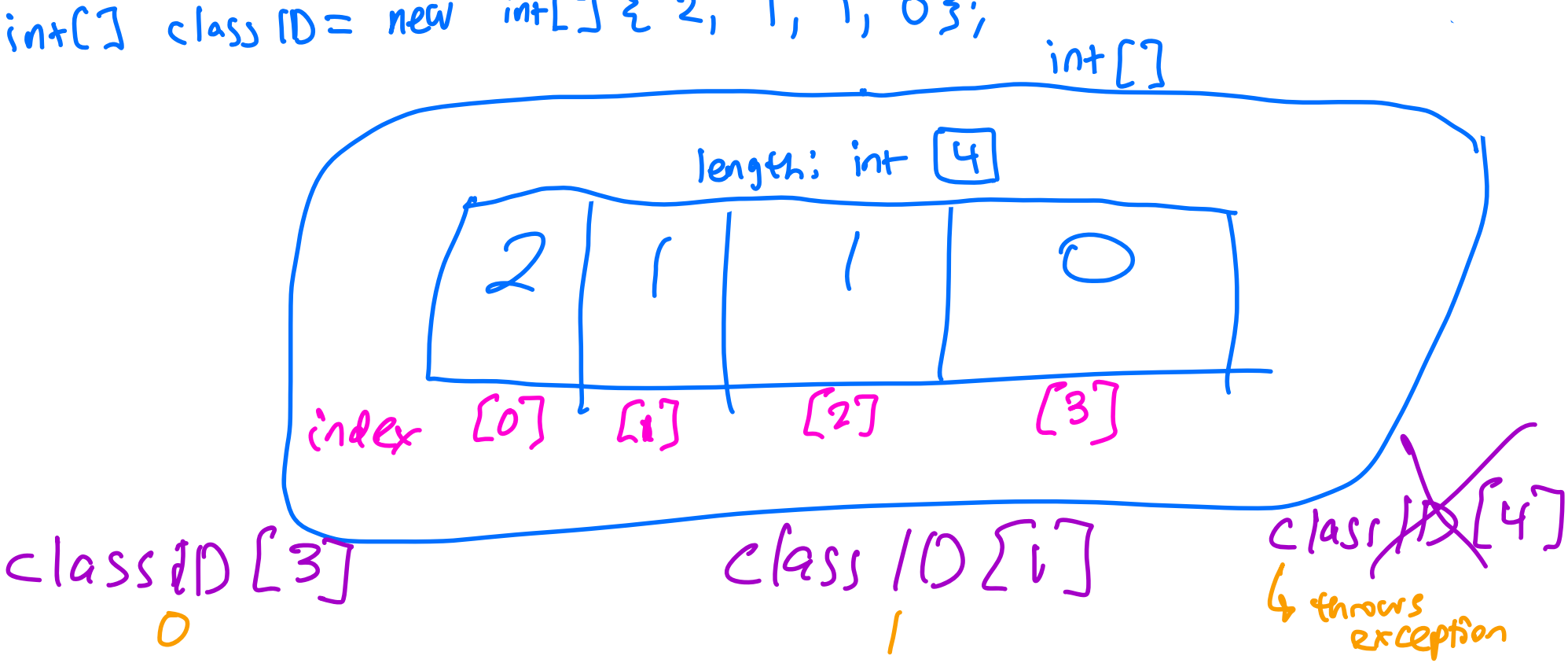
Arrays

```
int[] nums = new int[5];
int len = nums.length;
```



Array indexing

- index – a number that specifies a specific cell of an array



Array literals

```
int[] classID = new int[] {2, 1, 1, 0};
```

array literal

Reference Semantics (revisited)

Not covered in class. Try it!

```
import java.util.Arrays;
public class Arguments {
    // assume nums has at least one element
    public static void assignArray(int[] nums) {
        nums[0] = 99;
    }
    public static void assignNum(int num) {
        num = 99;
    }
    public static void main(String[] args) {
        int num = 0;
        assignNum(num);
        int[] nums = new int[]{0, 0, 0};
        assignArray(nums);
        // print out num and a string representation of nums
        System.out.println("num: " + num + ", nums: "
                           + Arrays.toString(nums));
    }
}
```

Reference Semantics (revisited)

```java
import java.util.Arrays;
public class Arguments {
    // assume nums has at least one element
    public static void assignArray(int[] nums) {
        nums[0] = 99;
    }
    public static void assignNum(int num) {
        num = 99;
    }
    public static void main(String[] args) {
        int num = 0;
        assignNum(num);
        int[] nums = new int[]{0, 0, 0};
        assignArray(nums);
        // print out num and a string representation of nums
        System.out.println("num: " + num + ", nums: "
                           + Arrays.toString(nums));
    }
}
```

PollEv.com/2110fa25
Or text 2110fa25 to 22333

Poll Everywhere

PollEv.com/2110fa25 text `2110fa25` to `22333`

Placeholder: Pass by reference vs pass by value

num: 99, nums: [99, 0, 0] **(A)**

num: 99, nums: [0, 0, 0] **(B)**

num: 0, nums: [99, 0, 0] **(C)**

num: 0, nums: [0, 0, 0] **(D)**

```java
import java.util.Arrays;
public class Arguments {
    // assume nums has at least one element
    public static void assignArray(int[] nums) {
        nums[0] = 99;
    }
    public static void assignNum(int num) {
        num = 99;
    }
    public static void main(String[] args) {
        int num = 0;
        assignNum(num);
        int[] nums = new int[]{0, 0, 0};
        assignArray(nums);
        // print out num and a string representation of nums
        System.out.println("num: " + num + ", nums: "
                           + Arrays.toString(nums));
    }
}
```

Space for Diagramming ?

Reference Semantics (revisited)

Arrays of reference types

Not covered in class. Please see lecture notes.

```
String[] words = new String[3];
words[1] = "apple";
words[0] = words[1];
words[1] = "banana";
words[2] = words[0].substring(0,3);
```

Multidimensional arrays

Not covered in class. Please see lecture notes.

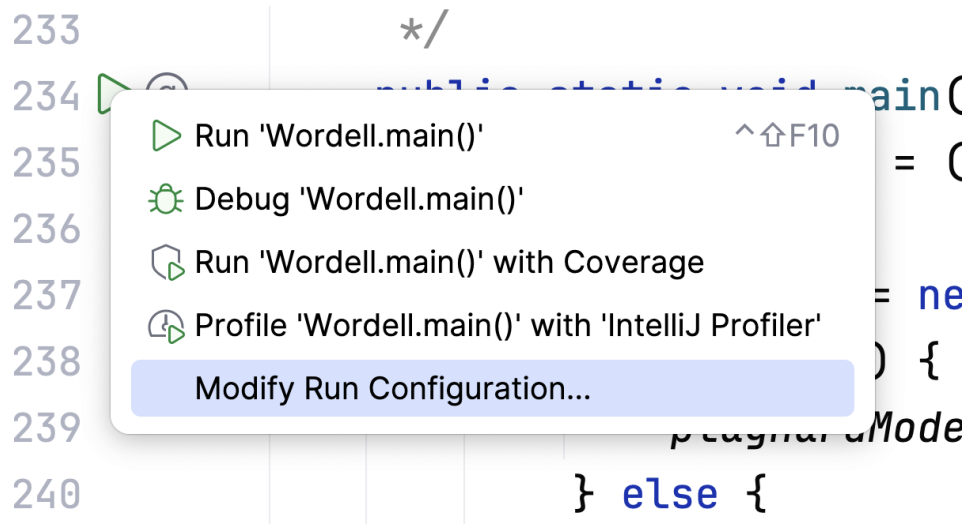
Metacognition

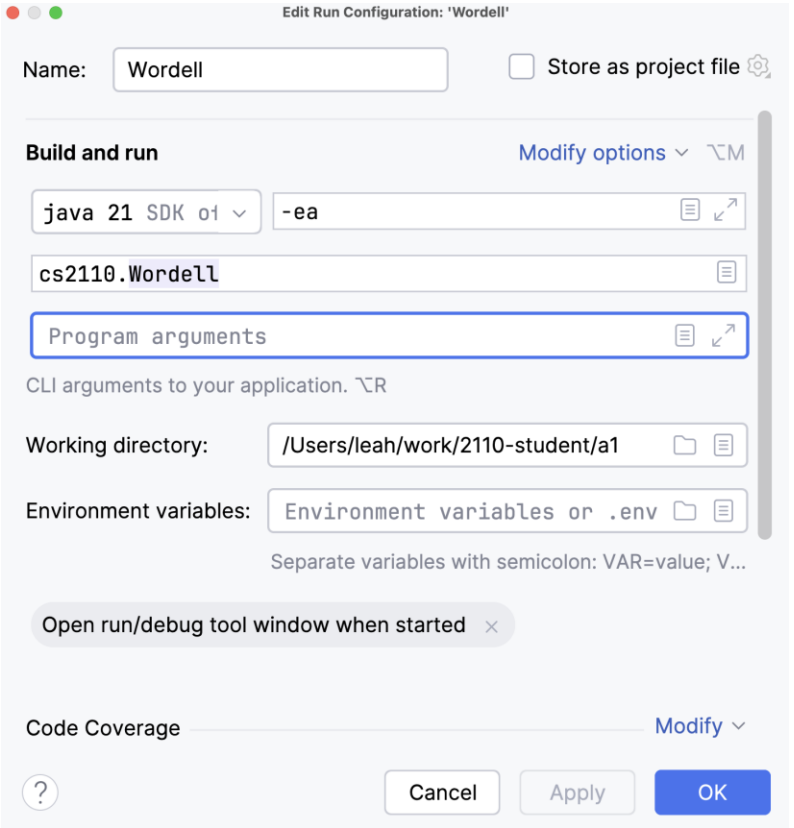
- Take 1 minute to write down a brief summary of what you have learned today

closing announcements to follow…

Coding Demo: Program Arguments

- (also known as command line arguments)
- Text entered as program arguments will be passed in to the main method.
- Main takes in a parameter **`String[] args`**
- Length of the array will be the number of words in program arguments
- Each array element is one of the words





Announcements

- Assignment 1 (A1) released
  - see course website --> Assignments
- Office hours starting today
  - see course website --> About --> Office Hours
  - Campus maps: CIS
- Anonymous feedback form tinyurl.com/2110-feedback
  - Can be found on course homepage