COS Crash Course: Week 1

intro and data types

welcome!

- Bummed out by COS 125’s cancellation? (Same.)
- Want to familiarize yourself with the structure of Princeton’s intro COS course, 126, before the fall semester starts?
- Want to brush up on coding interview problems before the next recruitment cycle? (*Same* same.)
- Not planning on majoring in COS but still want to familiarize yourself with the code that makes up this Virtual Insanity?
- Just *really* need a HALPS point?

You’re in the right place!

name game

Yacoub Kahkajian ***he/him***

Graduating in 2026

A.B. Computer Science

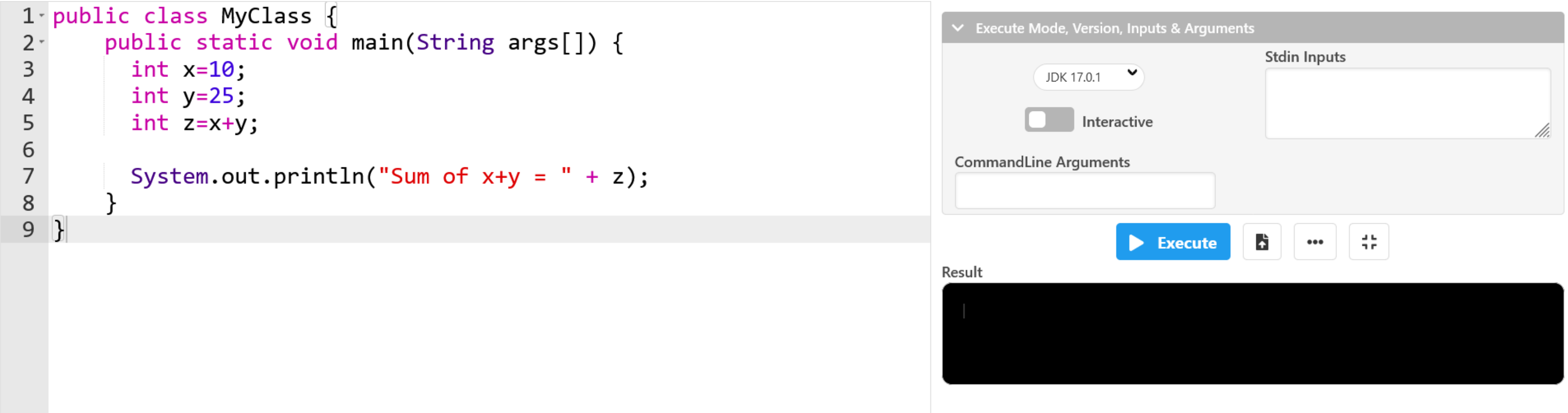
yacoub.xyz

but first, a word on stacks

- A technical stack refers to the programming languages, libraries, and IDEs (code editors) that are used to create a project.
- These sessions will use Java and an online editor called JDoodle.
- If you want to replicate the more professional tools COS 125/126 use, download IntelliJ IDEA from lift.cs.princeton.edu/java/windows

jdoodle.com/online-java-compiler

Feel free to play around with it while I yap.



```java
public class MyClass {
    public static void main(String args[]) {
      int x=10;
      int y=25;
      int z=x+y;

      System.out.println("Sum of x+y = " + z);
    }
}
```

JDK 17.0.1

lecture

strict typing

strings, ints, doubles

boolean logic

variables

- Variables are a key feature that make programming languages just so dang versatile.
  - Shorthand for longer values; makes code easier to maintain.
  - Can be altered based on user input while keeping the rest of the code the same.
  - Stores values which can be changed throughout the course of a loop or based on conditional statements (we'll come back to this another week!)

```
int foobar = 2763;
```

a variable...

...and its value (or literal)

strict typing

- Java is an example of a strictly typed language. Whenever you declare a new variable, you must specify what kind of values you want to associate to it.
- This may be different from other programming languages you might be familiar with such as Python or JavaScript, which are dynamically typed and automatically infer what types variables should take on.

```
int foobar = 2763;
```

this variable should only ever be associated with a number

built-in data types

variable type	what it can store	example values	example operations
String	Character sequences	“abcdefg” “four bees”	Concatenation (joining strings together), length, emptiness
char	Single characters	‘F’ ‘s’ ‘@’	String and integer conversions
int	Numbers	2763 -240	Arithmetic
double	Decimals	3.333 -0.126	Arithmetic
boolean	Binary truth values	true false	Logic operations (or, and, not)

the importance of types

In your code editor, try declaring two int values and two String values and using **System.out.println** to see how Java handles…

- Adding the two int values.
- Adding the two String values.
- Adding an int and a String, in that order.
- Adding a String and an int, in that order.
- Adding two int values, then the String.

type conversions

- In the previous examples, we witnessed automatic type conversion, where Java treated an int as a String to concatenate because it thought it was best for us.

```
“My favorite number is “ + 1746 + “.”
```

- In other cases, you may want to force Java to do the opposite and treat a String as an int or double using explicit type conversion.

```
Integer.parseInt(“1984”)
```

“WOW THIS ALL SOUNDS LIKE A LOT OF PRIMITIVE BOILERPLATE SYNTAX”

- Firstly, rude.
- Secondly, while Java *does* have a lot of that, strict typing is far from primitive.
  - Reduces “miscommunication” between the programmer and computer.
  - Helpful for debugging.

lecture

strict typing

strings, ints, doubles

boolean logic

strings…

- …treat whitespace (such as spaces) literally.

```
"I got a blank " + "space" ▶
"I got a blank space"
```

- …are not integers.

“1“ + “2” ▶ “12”

- …support Unicode, meaning they support emoji!

“👁👄👁”

string methods

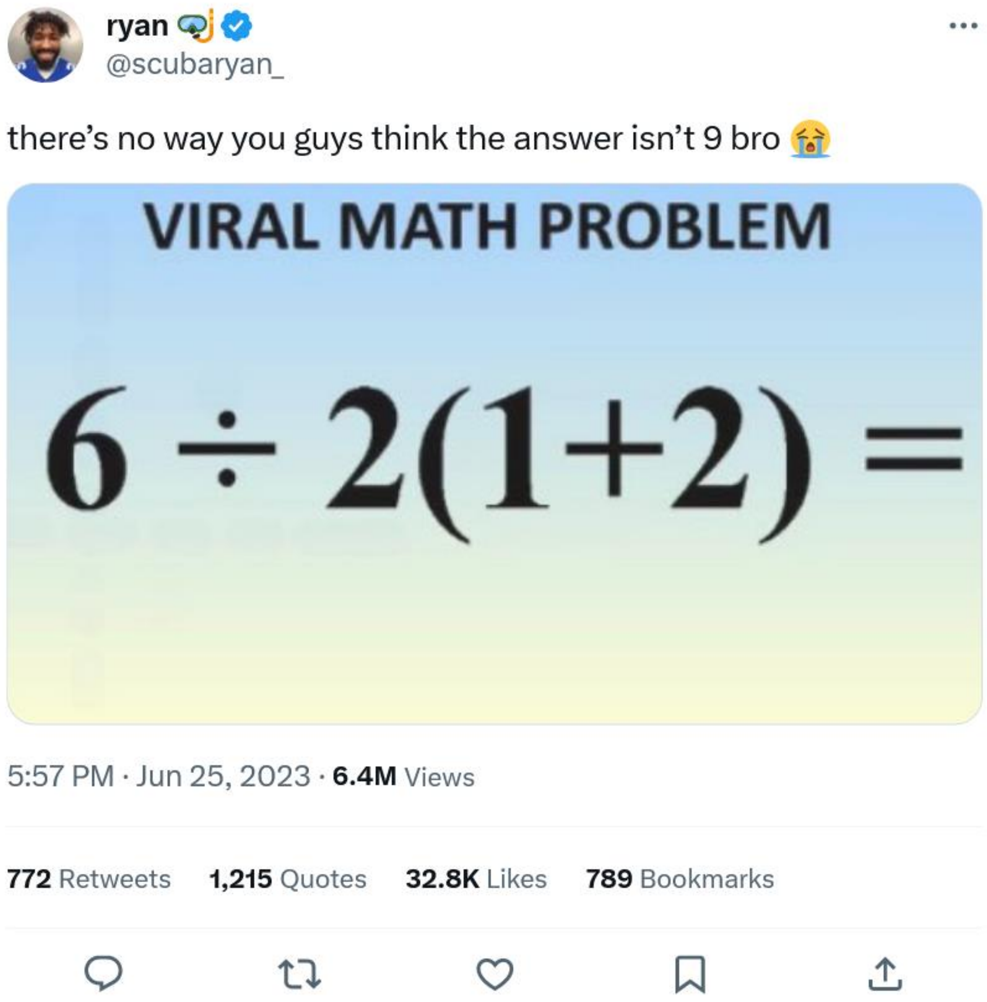
```
String x = “Princeton”;
```

method	what it does	example	output

charAt(int n)	Returns the nth character, starting at 0.	x.charAt(0)
		x.charAt(4)
			‘P’
			‘c’

length()	Returns the string length.	x.length()	9
isEmpty()	Returns if the string is empty or not	x.isEmpty()	false
indexOf(char n)	Finds where n is in a string,	x.indexOf(‘o’)	7

You’ll learn an efficient way in COS 226!



ryan 🤿 ✓
@scubaryan_

there's no way you guys think the answer isn't 9 bro 😭

VIRAL MATH PROBLEM

$$6 \div 2(1+2) =$$

5:57 PM · Jun 25, 2023 · **6.4M** Views

772 Retweets **1,215** Quotes **32.8K** Likes **789** Bookmarks

but what does java think?

ints and doubles...

- ...use PEMDAS for arithmetic.

  `6 / 2 * (1 + 2)` ▶ Try it yourself!

- ...use automatic type conversion when using ints and doubles in the same expression.

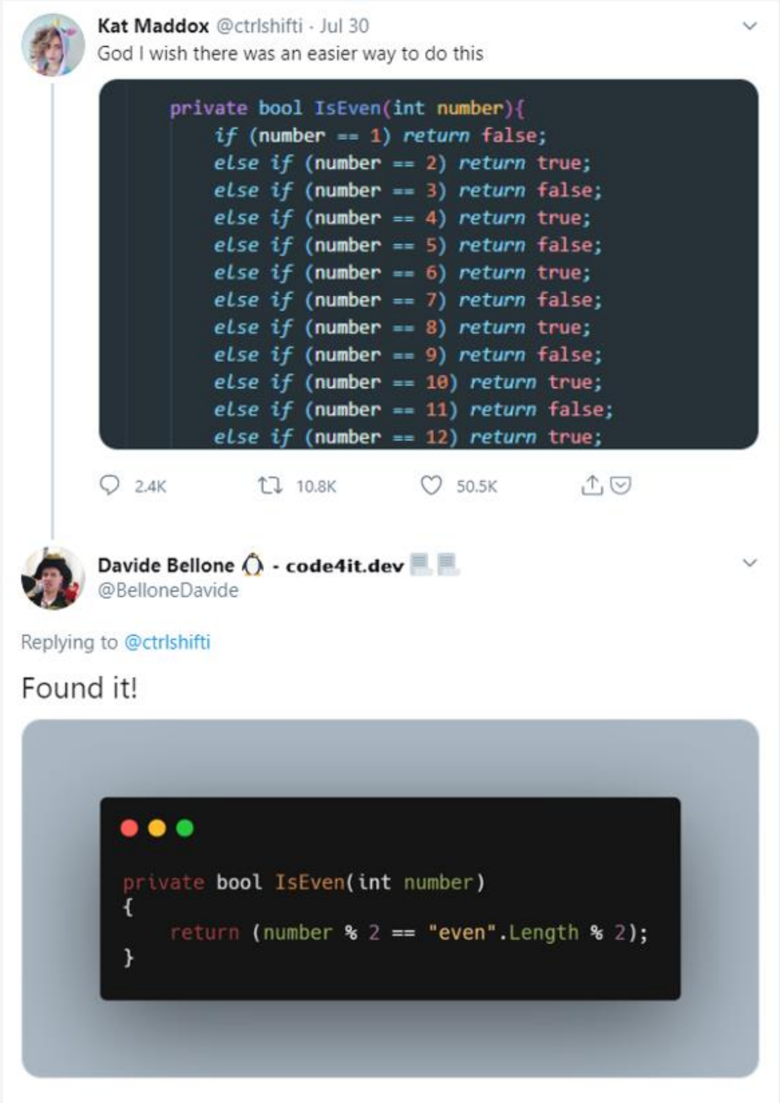
  `2.0 / 4` ▶ 2.0

- ...can be positive or negative.

  `-2 + 3` ▶ 1

- ...support the modulo operator, which calculates the remainder of a number when divided by the number after it.

  `5 % 2` ▶ 1



Kat Maddox @ctrlshifti · Jul 30

God I wish there was an easier way to do this

```
private bool IsEven(int number){
    if (number == 1) return false;
    else if (number == 2) return true;
    else if (number == 3) return false;
    else if (number == 4) return true;
    else if (number == 5) return false;
    else if (number == 6) return true;
    else if (number == 7) return false;
    else if (number == 8) return true;
    else if (number == 9) return false;
    else if (number == 10) return true;
    else if (number == 11) return false;
    else if (number == 12) return true;
```

Davide Bellone 🐧 - code4it.dev
@BelloneDavide

Replying to @ctrlshifti

Found it!

```
private bool IsEven(int number)
{
    return (number % 2 == "even".Length % 2);
}
```

oh dear.

division with ints is weird

- When you divide two ints together, Java assumes that you still want a result of value type int.

```
2 / 4 ▶ 0
```

- There are two ways to fix this:
  - Make one of the digits a double.

```
2 / 4.0 ▶ 0.5
```

  - Explicitly convert the ints to doubles using type casting.

```
(double) 2 / 4 ▶ 0.5
```

int and double methods

method	what it does	example	output
Math.abs(n)	Returns absolute value of an int or double.	Math.abs(-2)	2

Math.max(a, b)
Math.min(a, b)
	Returns the max or min of two ints or doubles.	Math.max(5, 7)	7

Math.pow(a, b)	Raises a to the power of b, returns a double.	Math.pow(2, 3)	8.0
Math.random()	Returns a double from [0, 1)	Math.random()	0.87113234

Can you think of a way to return random whole numbers using this method?

lecture

strict typing

strings, ints, doubles

boolean logic

operators

Booleans have their own wacky set of operators.

- And (&&) outputs true if all conditions are true.

```
false && true ▶  false
 true && true ▶  true
```

- Or (||) outputs true if at least one condition is true.

```
false || true ▶  true
 true || true ▶  true
```

- Not (!) returns the opposite result of the expression.

```
!false ▶ true
```

```
!(true && false || !false)
!(true && false ||   true)
!(false         ||   true)
!(true)
false
```

some boolean math

comparison operators

You can also use comparison operators to get a boolean output from comparing two values.

operator	meaning
==	Equal
!=	Not equal
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Single equal sign and double equal signs are different breeds!

Singles are for assigning variables to new values.

Doubles are for comparing two existing values.

```
1 <= 4 <= 12
(1 <= 4) <= 12
(true) <= 12
```

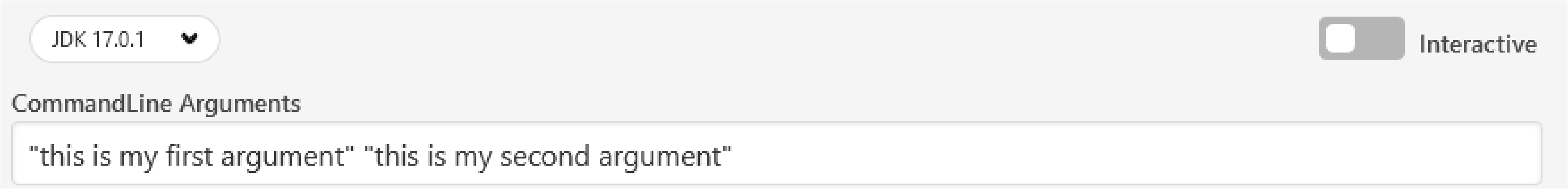
💀💀💀💀💀😭

some regular math

One more thing…

command line arguments

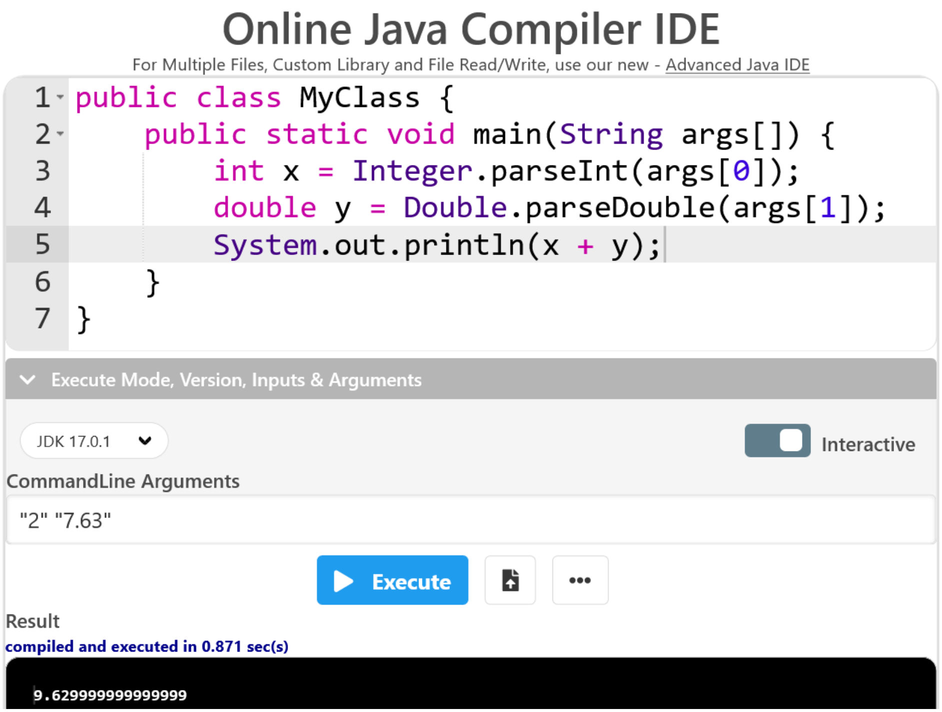
- Throughout the course of this crash course, we will be sending input to our programs primarily through command line arguments.
- JDoodle provides a graphical way of sending command-line arguments, just by filling in a text box.
- Command line arguments are sent as an array of strings titled args. To use them in your program, use args[0] to refer to your first string, args[1] to refer to your second string, and so on.



CommandLine Arguments: "this is my first argument" "this is my second argument"

string to int

All command line arguments have the string value type. If you want Java to interpret your argument as a number or decimal value, use the `Integer.parseInt()` and `Double.parseDouble()` methods.



Online Java Compiler IDE

For Multiple Files, Custom Library and File Read/Write, use our new - Advanced Java IDE

```
public class MyClass {
    public static void main(String args[]) {
        int x = Integer.parseInt(args[0]);
        double y = Double.parseDouble(args[1]);
        System.out.println(x + y);
    }
}
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1

Interactive

CommandLine Arguments

"2" "7.63"

Result

compiled and executed in 0.871 sec(s)

```
9.629999999999999
```

©2023 Yacoub Kahkajian (yacoub.xyz)

exercises

unit conversion

You are given a temperature in Fahrenheit, a weight in pounds, and a measurement in inches as command line arguments. Convert them to NORMAL units (Celsius, kilograms, centimeters). You are free to look up the equations if you'd like.

teen numbers

Three numbers are given to you as command line arguments. If all three of them are “teen” numbers, return true. Otherwise, return false.

Follow up: Can you do this using a modulo expression?