Fall 2023

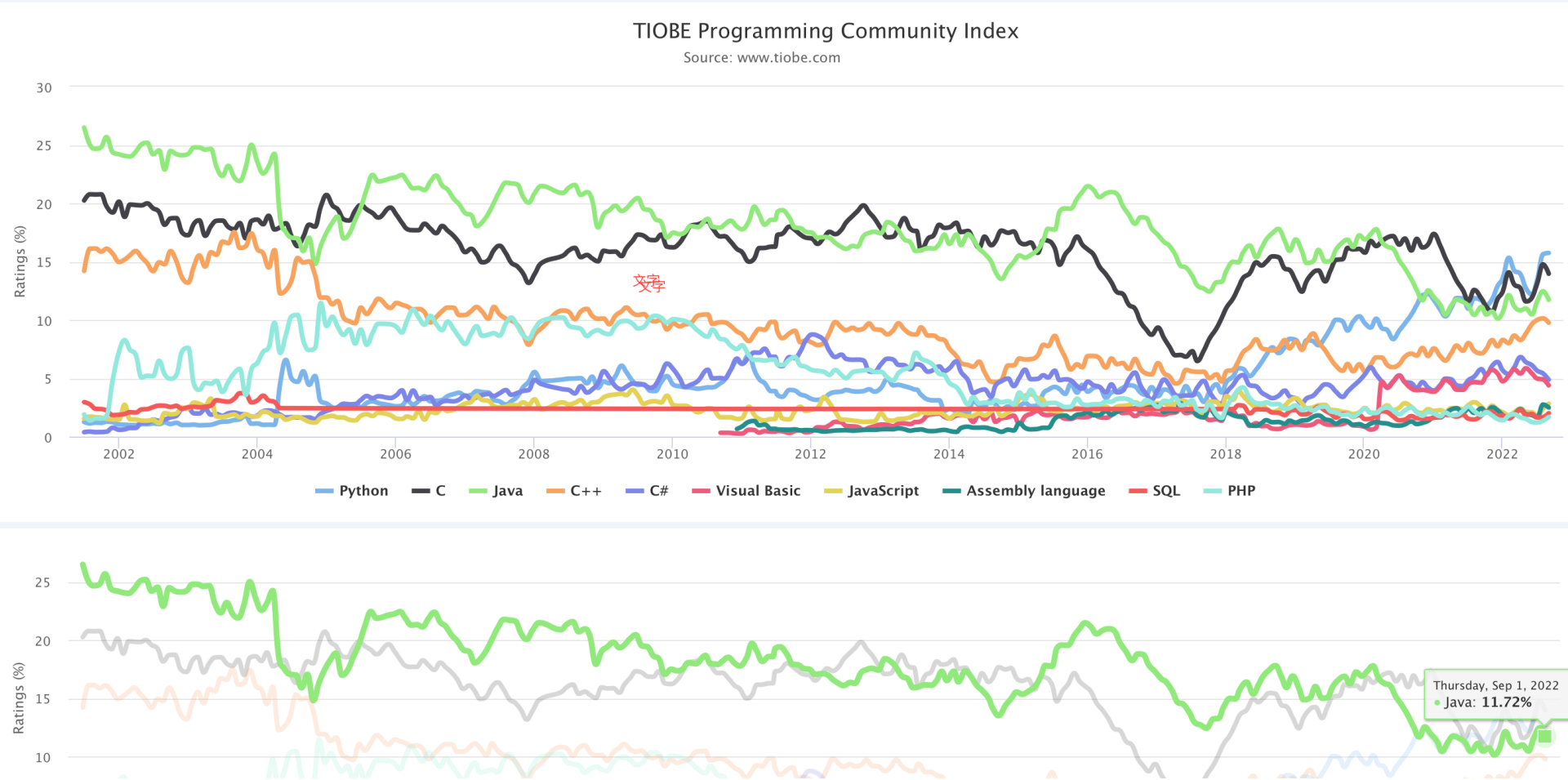
Data Structures Lecture 1

Fang Yu

Software Security Lab.
Dept. Management Information Systems,
National Chengchi University

A brief review of Java programming

Popularity of Programming Languages



Source: https://www.tiobe.com/tiobe-index/

History of PL Popularity

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	↑	Python	15.74%	+4.07%
2	1	↓	C	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		C++	9.76%	+2.63%
5	5		C#	4.88%	-0.89%
6	6		Visual Basic	4.39%	-0.22%
7	7		JavaScript	2.82%	+0.27%
8	8		Assembly language	2.49%	+0.07%
9	10	↑	SQL	2.01%	+0.21%
10	9	↓	PHP	1.68%	-0.17%
11	24	⇈	Objective-C	1.49%	+0.86%
12	14	↑	Go	1.16%	+0.03%
13	20	⇈	Delphi/Object Pascal	1.09%	+0.32%
14	16	↑	MATLAB	1.06%	+0.04%

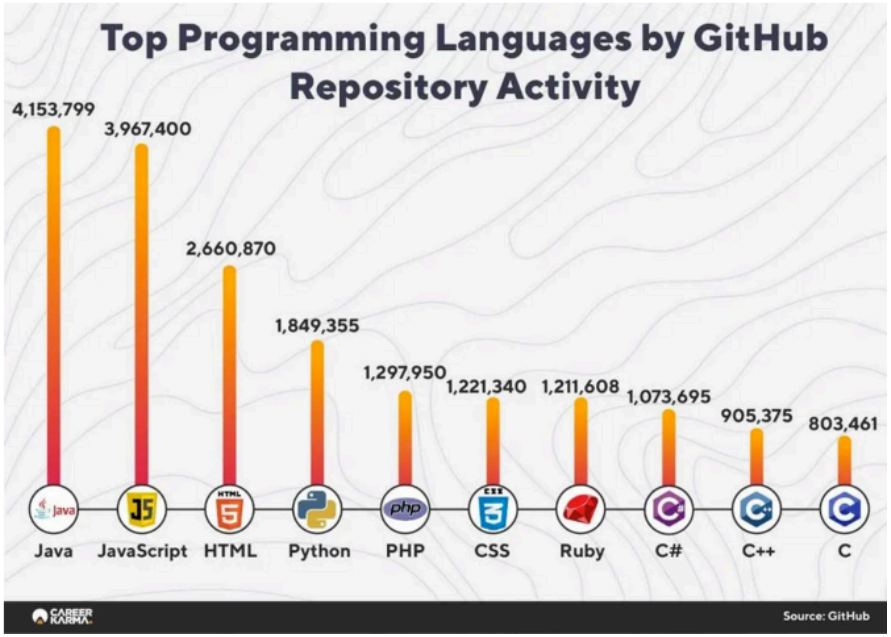
Source: https://www.tiobe.com/tiobe-index/

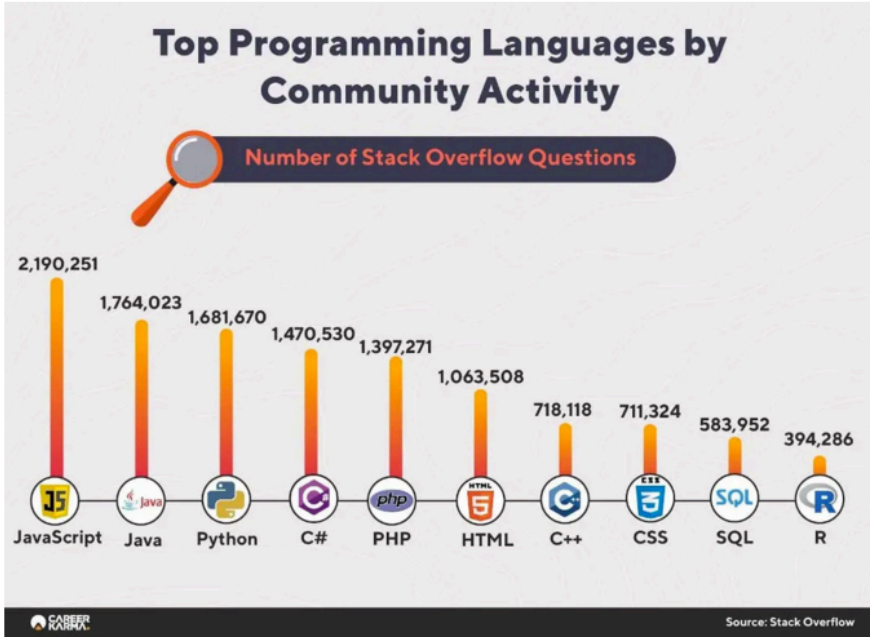
Very Long History of PL Popularity

Programming Language	2022	2017	2012	2007	2002	1997	1992	1987
Python	1	5	8	7	12	28	-	-
C	2	2	1	2	2	1	1	1
Java	3	1	2	1	1	16	-	-
C++	4	3	3	3	3	2	2	6
C#	5	4	4	8	14	-	-	-
Visual Basic	6	14	-	-	-	-	-	-
JavaScript	7	8	10	9	8	24	-	-
Assembly language	8	10	-	-	-	-	-	-
SQL	9	-	-	-	7	-	-	-
PHP	10	7	6	5	6	-	-	-
Prolog	24	32	33	27	17	21	12	3
Lisp	33	31	13	16	13	10	4	2
Pascal	270	114	16	22	99	9	3	5

Source: https://www.tiobe.com/tiobe-index/

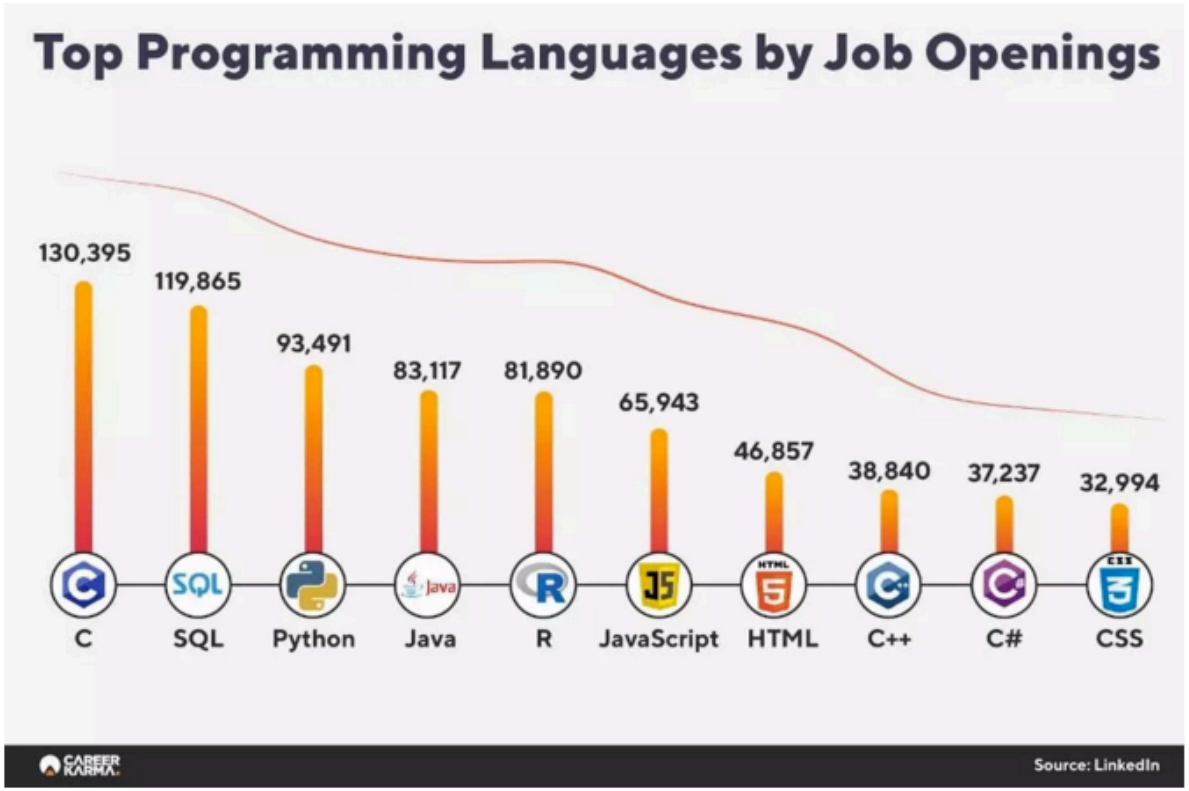
Java is very popular in community





https://www.geeksforgeeks.org/top-10-programming-languages-to-learn-in-2022/

Java still has its needs in job markets



https://www.geeksforgeeks.org/top-10-programming-languages-to-learn-in-2022/

Java is worth to learn

- One of the most popular languages in the past years
- Platform independence and Object-oriented programming
- Enhanced productivity, performance, and security
- It is the most secure language
- Companies working on Java are Amazon, Adobe, Flipkart, Instagram, etc.

Data structure

- To design an object in the digital world
- That means, in this class you will learn how to define a class in Java

Three main elements: Class, Type, and Object

- An object is the basic unit in Java
- A class defines the type of an object

Java Programming Basics

- A class consists of
  - fields (to store data)
  - methods (to define operations that can act on data)

The class name (Save this code as **HelloWorld**.java)

A field

A method

```
public class HelloWorld {
    public static string prefix = "Hello ";
    public static void say(String s) {
        System.out.print(prefix+s);
    }
    public static void main(String[] argv) {
        say("World!");
    }
}
```

The main method (The entry point while executing the program)

Modifiers

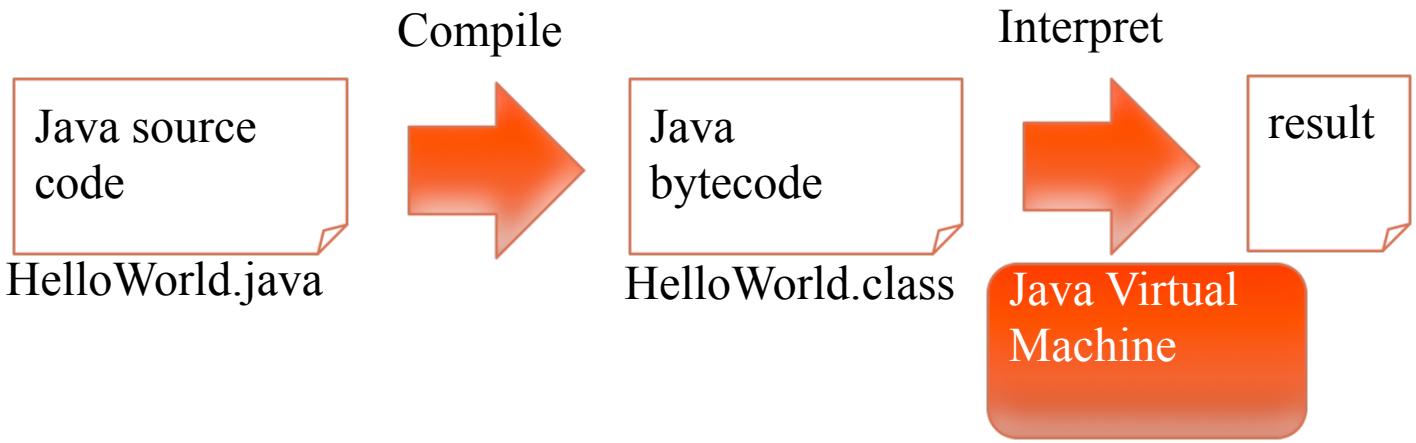
"public" indicates that anyone can run/extend/ import this class

"static" indicates the field/method belongs to the class, not objects

```
public class HelloWorld {
    public static string prefix = "Hello ";
    public static void say(String s) {
        System.out.print(prefix+s);
    }
    public static void main(String[] argv) {
        say("World!");
    }
}
```

"void" indicates that the method returns nothing

How Java works



- Execute your code in command lines
  - "javac HelloWorld.java" to generate HelloWorld.class
  - "java HelloWorld" to execute the bytecode

Example: Operator

- Operators are similar to C++
  - E.g., =, +, - , *, /, %
- A simple example:
- Sum 1 to 100 using a formula

```
public class Example {
        public static void main(String[] argv) {
                int n = 100;
                System.out.println("1+2+...+" + n + " = " + ( n * (n + 1) / 2));
        }
}
```

```
javac Example.java
java Example

1+2+...+100 = 5050
```

Example: Method

- Sum 1 to 100 using a method

```
public class Example {
        public static int sum(int n) {
                return n*(n+1)/2;
        }
        public static void main(String[] argv) {
                int n = 100;
                System.out.println("1+2+...+"+n+" = " + sum(n));
        }
}
```

```
javac Example.java
java Example

1+2+...+100 = 5050
```

Example: Loop

- Sum 1 to n using a method with for-loop

```
public class Example {
        public static int sum(int n) {
                int total = 0;
                for (int i = 1; i <= n; i++) { total += i; }
                return total;
        }
        public static void main(String[] argv) {
                int n = 100;
                System.out.println("1+2+...+"+n+" = " + sum(n));
        }
}
```

```
javac Example.java
java Example

1+2+...+100 = 5050
```

Example: Loop

- Sum n1 to n2 using a method with for-loop

```
public class Example {
        public static int sum(int n) {
                int total = 0;
                for (int i = 1; i <= n; i++) { total += i; }
                return total;
        }

        public static int sum2(int n1, int n2) {
                int total = 0;
                for (int i = n1; i <= n2; i++) { total += i; }
                return total;
        }

        public static void main(String[] argv) {
                int n1 = 100, n2 = 200;
                System.out.println(n1+“+...+"+n2+" = " + sum2(n1,n2));
        }
}
```

javac Example.java
java Example

100+...+200 = 15000

Use pre-defined class library

- Use java.util.Scanner to get inputs
- The Scanner class reads the input stream and divides it into tokens by delimiters (whitespace)
- The Scanner class includes the following methods:

hasNext()	Return true if there is another token
next()	Return the next token
hasNext**Type**()	Return true if there is another token that can be interpreted as the **Type**
next**Type**()	Return the next token that can be interpreted as the **Type**

Use pre-defined class library

- Import the package

  import java.util.Scanner;

- Construct a Scanner object:

  Scanner in = **new** Scanner(System.in);

- Call its method:

  e.g., in.nextInt() or in.hasNext()

Example: Get a user input from the screen

- Sum using java.util.Scanner class

```
import java.util.Scanner;
public class Example {
        public static int sum(int n) {
                int total = 0;
                for (int i = 1; i <= n; i++) { total += i; }
                return total;
        }
        public static void main(String[] argv) {
                Scanner in = new Scanner(System.in);
                int n = in.nextInt();
                System.out.println("1+2+...+"+n+" = " + sum(n));
        }
}
```

```
javac Example.java
java Example
100
1+2+...+100 = 5050
```

Example: Print something on the screen

- Sum using java.util.Scanner class

```
import java.util.Scanner;
public class Example {
        public static int sum(int n) {
                int total = 0;
                for (int i = 1; i <= n; i++) { total += i; }
                return total;
        }
        public static void main(String[] argv) {
                Scanner in = new Scanner(System.in);
                System.out.print(“Enter n: ”);
                int n = in.nextInt();
                System.out.println("1+2+...+"+n+" = " + sum(n));
        }
}
```

```
javac Example.java
java Example

Enter n: 100
1+2+...+100 = 5050
```

Example: Get user inputs

- Sum using java.util.Scanner class

```
import java.util.Scanner;
public class Example {
        public static int sum(int n) {
                int total = 0;
                for (int i = 1; i <= n; i++) { total += i; }
                return total;
        }
        public static int sum2(int n1, int n2) {
                int total = 0;
                for (int i = n1; i <= n2; i++) { total += i; }
                return total;
        }
        public static void main(String[] argv) {
                Scanner in = new Scanner(System.in);
                System.out.print(“Enter n1 n2: ”);
                int n1 = in.nextInt();
                int n2 = in.nextInt();
                System.out.println(n1+”+...+"+n2+" = " + sum2(n1,n2));
        }
}
```

```
javac Example.java
java Example

Enter n1 n2: 10 100
10+...+100 = 5005
```

Homework 1 (Due on 9/21)

- BMI Calculator:
  - BMI = (Weight in Kilograms / (Height in Meters x Height in Meters))
- Enter Height and Weight, return BMI and
  - "You are not in shape. Actually, you are not even close." if BMI >= 30
  - "To be honest, you are not in shape." if 30 > BMI >= 26
  - "You are in shape" if 26 > BMI >= 20
  - "You are under shape" if 20 > BMI
- Use Eclipse to write/execute/debug your java code
- Upload your code using WM5 (**no** direct copy accepted)
- TAs will show you "clear" hints to do so on Monday's lab (Sep. 19)

About Eclipse

Eclipse is

- An Integrated Development Environment (IDE) for Java and also many other languages
- An open source platform (free!)
- Maintained by many software development leaders like IBM and Borland

Eclipse Extension

Furthermore, Eclipse

- provides a common environment that companies can modify and customize by creating **plug-ins**
- These plug-ins can **add functionality** to Eclipse like svn, github, modeling, UML, XML, metrics, reliability reports, and other information.
- The Eclipse web site has a list of links to many popular **plug-in repositories**

Learn Eclipse and Java

- Eclipse and Java tutorials. Watch this if you are a total beginner.
  http://eclipsetutorial.sourceforge.net/index.html

- Another nice java/eclipse tutorial on Youtube:
  https://www.youtube.com/watch?v=r59xYe3Vyks&list=PLS1QulWo1RIbfTjQvTdj8Y6yyq4R7g-Al

  To install Eclipse and Java:
  https://www.youtube.com/watch?v=dwNLNk2LKBE

Coming up…

- The first lab is scheduled on Monday, **Sep 18, 12:00-2:00pm**.
- TAs will talk about Eclipse and HW1 (We have limited PCs, so bring your own laptops)
- We will continue our discussion on object oriented design and abstract data type on Sep. 21
- Read TB Chapter 1 and Chapter 2