Australian National University

Introductory Java 1

J1

Imperative programming languages
Java Standard Library
Types
Hello World

10
No.7
GEDORE
MADE IN GERMANY
10

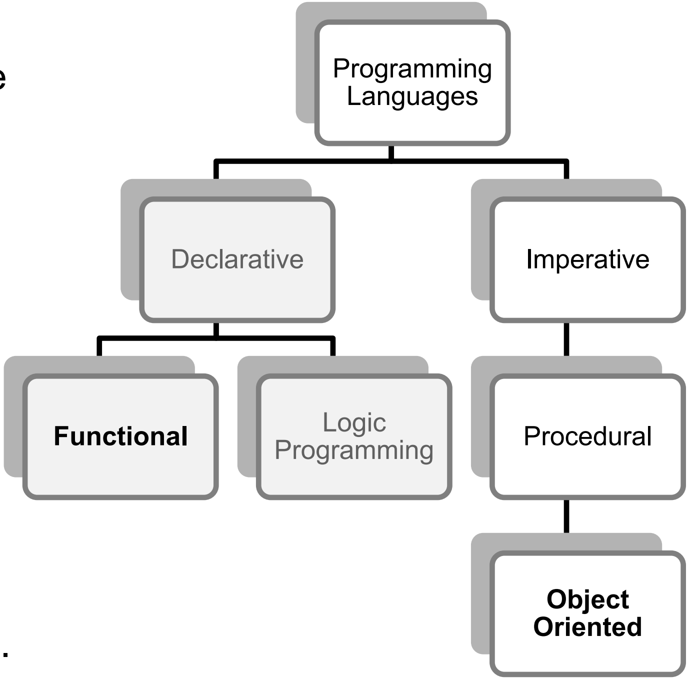
Why Java?

- Learn multiple programming paradigms
- Important example of:
  - Object-oriented programming
  - Large scale programming
  - Programming with a rich standard library

Imperative Programming Languages

Declarative languages describe the desired result without explicitly listing steps required to achieve that goal.

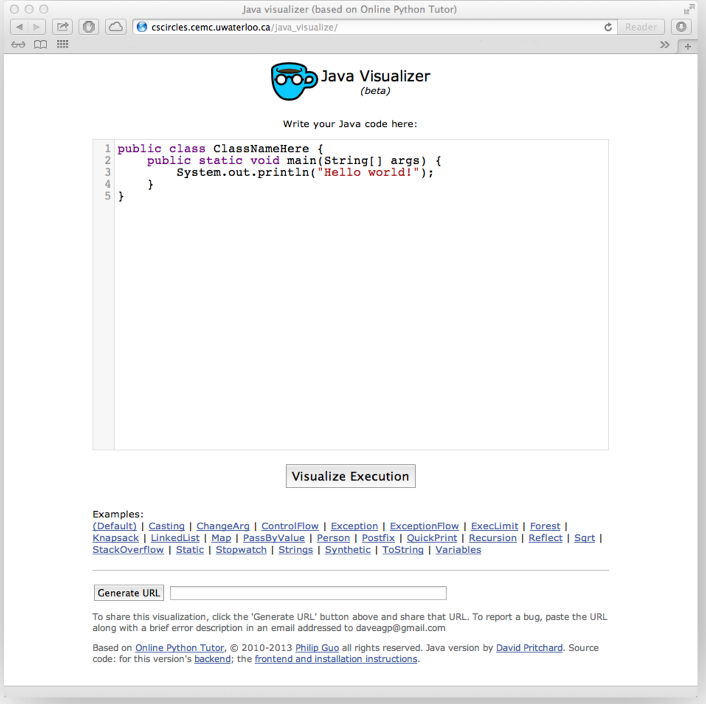
Pure functional languages, like Haskell, will only transform state by using functions without side effects.



Imperative languages describe computation in terms of a series of statements that transform state.

Object-oriented languages use structured (procedural) code, tightly coupling data with the code that transforms it.

The Waterloo Java Visualizer



Java visualizer (based on Online Python Tutor)
cscircles.cemc.uwaterloo.ca/java_visualize/
Reader
Java Visualizer
(beta)

Write your Java code here:

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

Examples:
(Default) | Casting | ChangeArg | ControlFlow | Exception | ExceptionFlow | ExecLimit | Forest | Knapsack | LinkedList | Map | PassByValue | Person | Postfix | QuickPrint | Recursion | Reflect | Sqrt | StackOverflow | Static | Stopwatch | Strings | Synthetic | ToString | Variables

To share this visualization, click the 'Generate URL' button above and share that URL. To report a bug, paste the URL along with a brief error description in an email addressed to daveagp@gmail.com

Based on Online Python Tutor, © 2010-2013 Philip Guo all rights reserved. Java version by David Pritchard. Source code: for this version's backend; the frontend and installation instructions.

A great resource. Type in simple Java programs and watch step-by-step execution. A great way to enhance your understanding of a new language.

The Oracle Java Tutorials

This course follows the structure of the *Oracle Java Tutorials* for the basic introduction to Java.

The tutorials are subject to Oracle's 'Java Tutorial Copyright and License' (Berkeley license).

We will move very fast for the first few weeks. You should use the tutorials to **ensure that you rapidly become proficient in the basics of Java**.

The Java Standard Library

- The Java language is augmented with a large standard library (.NET does the same for C#)
  - IO (accessing files, network, etc)
  - Graphics
  - Standard data structures
  - Much more
- Using and understanding the standard library is part of learning a major language like Java or C#.
- Rich standard libraries are a key software engineering tool.

Data types

The *type* of a unit of data determines the possible values that data may take on.

- Weak v strong
  - Must all data be typed? Can types be coerced or converted?
- Static v dynamic
  - Is checking done at compile-time or run-time?

Haskell: strong, static

Java: strong, static and dynamic

Basic IO (Reading and Writing)

This will be discussed in detail later.

```java
import java.util.Scanner;

public class IOTest {

  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter integer a:  ");
    int a = in.nextInt();
    System.out.print("Enter string b:  ");
    String b = in.next();
    System.out.println("a: " + a + ", b: "+ b);
  }
}
```