Java Programming 2 Introduction

Mary Ellen Foster
MaryEllen.Foster@glasgow.ac.uk
Semester 1 2020/2021

Who am I?

Originally from Canada

Degrees from

- Waterloo (Canada)
- Toronto
- Edinburgh

Been programming in Java since ~1996 (Java 1.0 beta)

Current research focus: social robotics

Official specification for JP2

http://www.gla.ac.uk/coursecatalogue/course/?code=COMPSCI2001

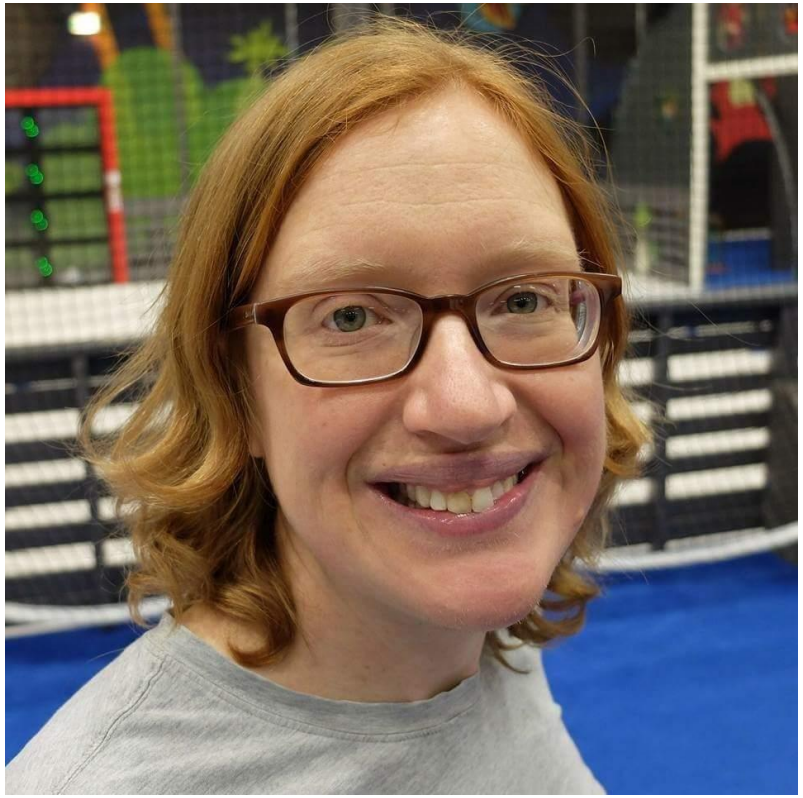
Short description:

"This course extends students' experience in programming using a strongly typed language (Java) and strengthens their problem solving skills. Students will learn the ideas that underpin object-oriented programming and will apply those concepts in developing small and medium sized software systems. Students will also learn to select and re-use existing software components and libraries, and will gain experience in concurrent programming and elementary graphical user-interface (GUI) development."

Assessment (TBC):

- 1 hour examination in December (60%)
- 2 hour laboratory examination in Week 11 (20%)
- 8 individual lab exercises (20% total – based on best 5)

Logistics



Instructor: Dr Mary Ellen Foster

Email: MaryEllen.Foster@Glasgow.ac.uk
Webpage: http://maryellenfoster.uk/
Office: Room S134, 18 Lilybank Gardens
Student hours: Friday 12-1pm (or by appointment)

Structure

Two lectures / week
Monday 11am
Wednesday 1pm
One tutorial & questions session / week
Friday 1pm
One lab session / week
Monday/Tuesday, starting in Week 2

What will we do during the time slots?

Monday/Wednesday “Lectures”
- Live synchronous lectures (like this one)
- “Watch party” of short recorded talks (with Q&A + examples)
- … Other things to be determined?
- **Details will be revisited over the next few weeks as we figure out what works best!**

Friday “Tutorial”
- Extended live coding examples
- Solutions and discussion of previous lab exercise
- Discussion and examples relevant to upcoming lab exercise

Monday/Tuesday “Labs”
- One-on-one and group support from your tutor and demonstrators for the lab exercises

Details of lab exercises

Weekly – beginning in Week 2, based on previous week's lectures

Each exercise is worth **4%** of your final grade – mark will be based on best 5 exercises

Schedule:

- Lab distributed through Moodle on or before **Thursday evening** in week N-1
- Lab is due (through Moodle) on **Thursday 5pm** in week N
- Solutions discussed in **Friday tutorial** in week N

You may submit work that is **incorrect** or **incomplete**

- In order to stretch the stronger members of the class, some of the laboratory exercises are quite challenging …
- … so don't worry if you can't complete all of them.

You should spend around **3 hours per week** on programming exercises

Official specification revisited

Short description:

“This course extends students' experience in programming using a strongly typed language (Java) and strengthens their problem solving skills. Students will learn the ideas that underpin object-oriented programming and will apply those concepts in developing small and medium sized software systems. Students will also learn to select and re-use existing software components and libraries, and will gain experience in concurrent programming and elementary graphical user-interface (GUI) development.”

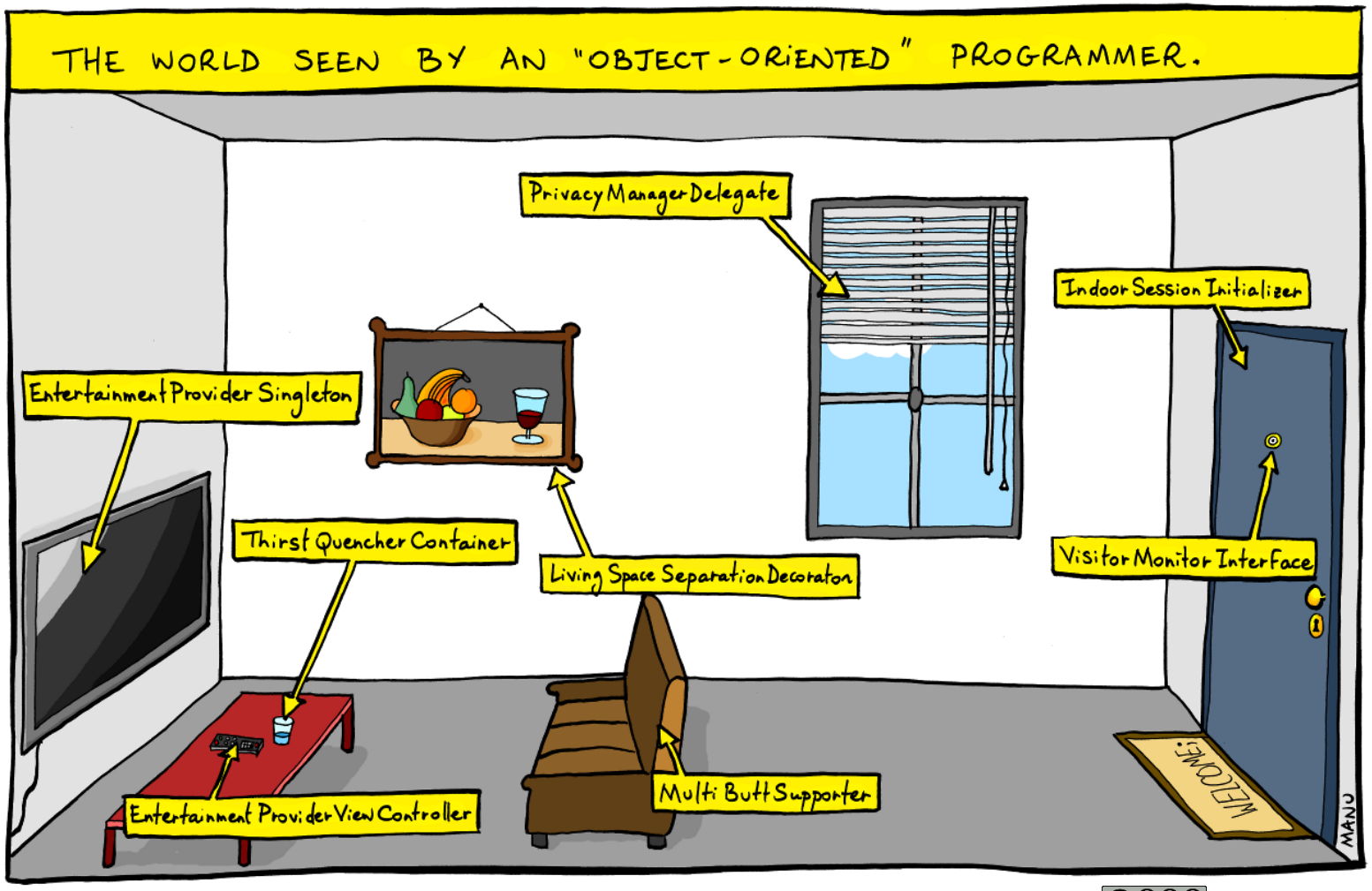
My assumptions

You **probably** know how to program in Python

You **may or may** not know anything about Java

Java is ...

An **object-oriented** language



Comic by Manu Comet -- http://www.bonkersworld.net/object-world/

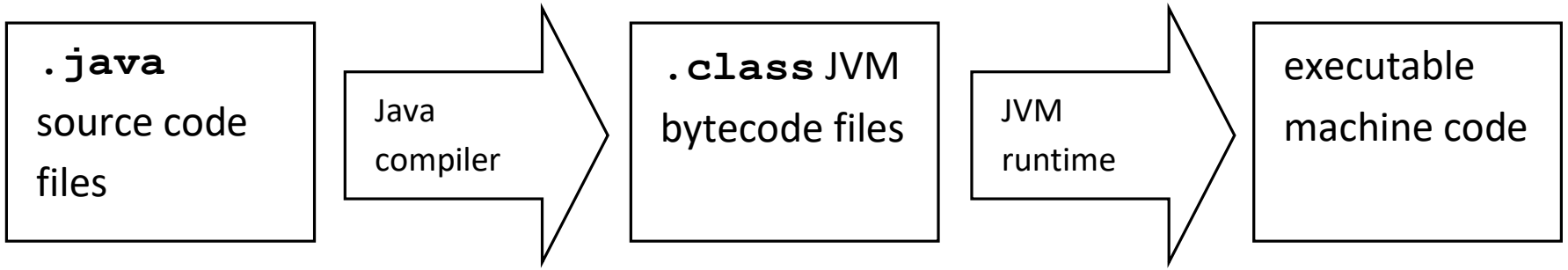
Java is ...

A **platform-neutral language** with a "**write once run anywhere**" philosophy

Supported by the **Java Virtual Machine (JVM)**

Source programs compiled to JVM **bytecode** .class files

Bytecode files converted to **native, platform-specific machine code** for execution



Java is …

Currently one of the **top programming languages** according to most popularity metrics

- Familiar programming syntax (like C/C++)
- Good support for modularity
- Relatively safe features (garbage collection)
- Comprehensive library support

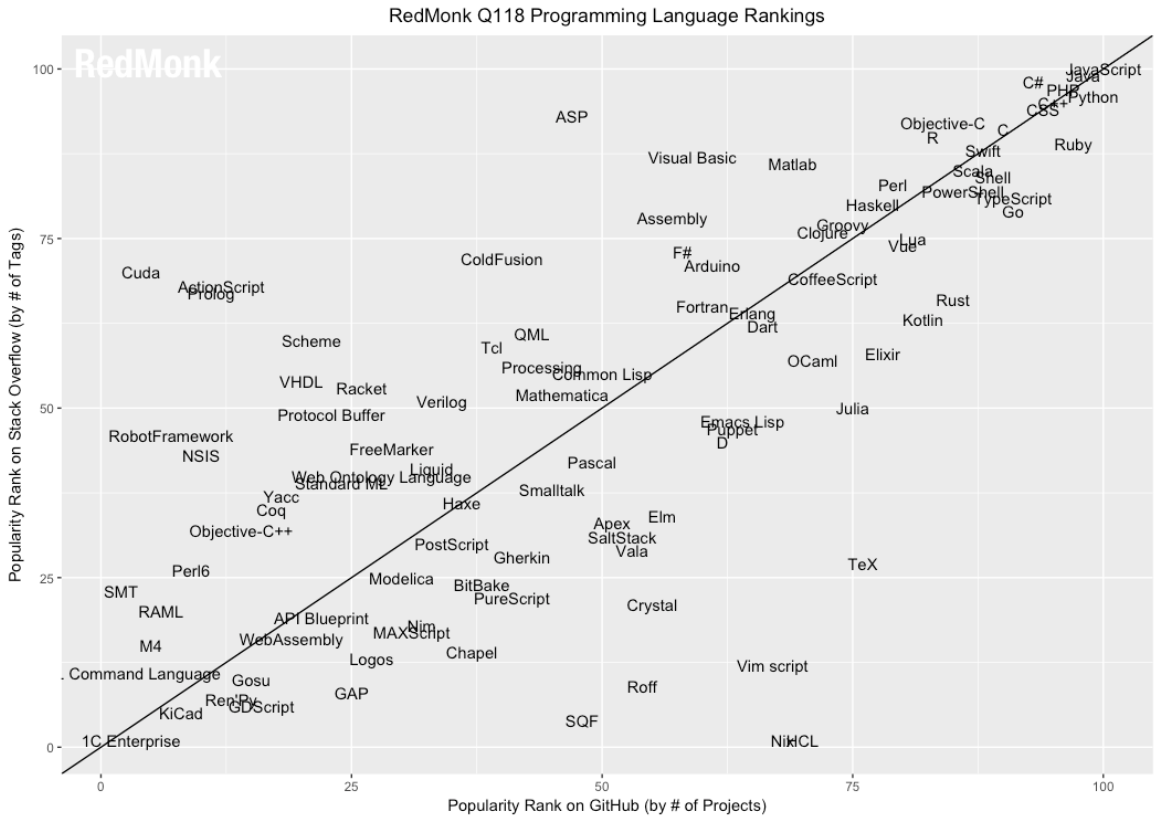
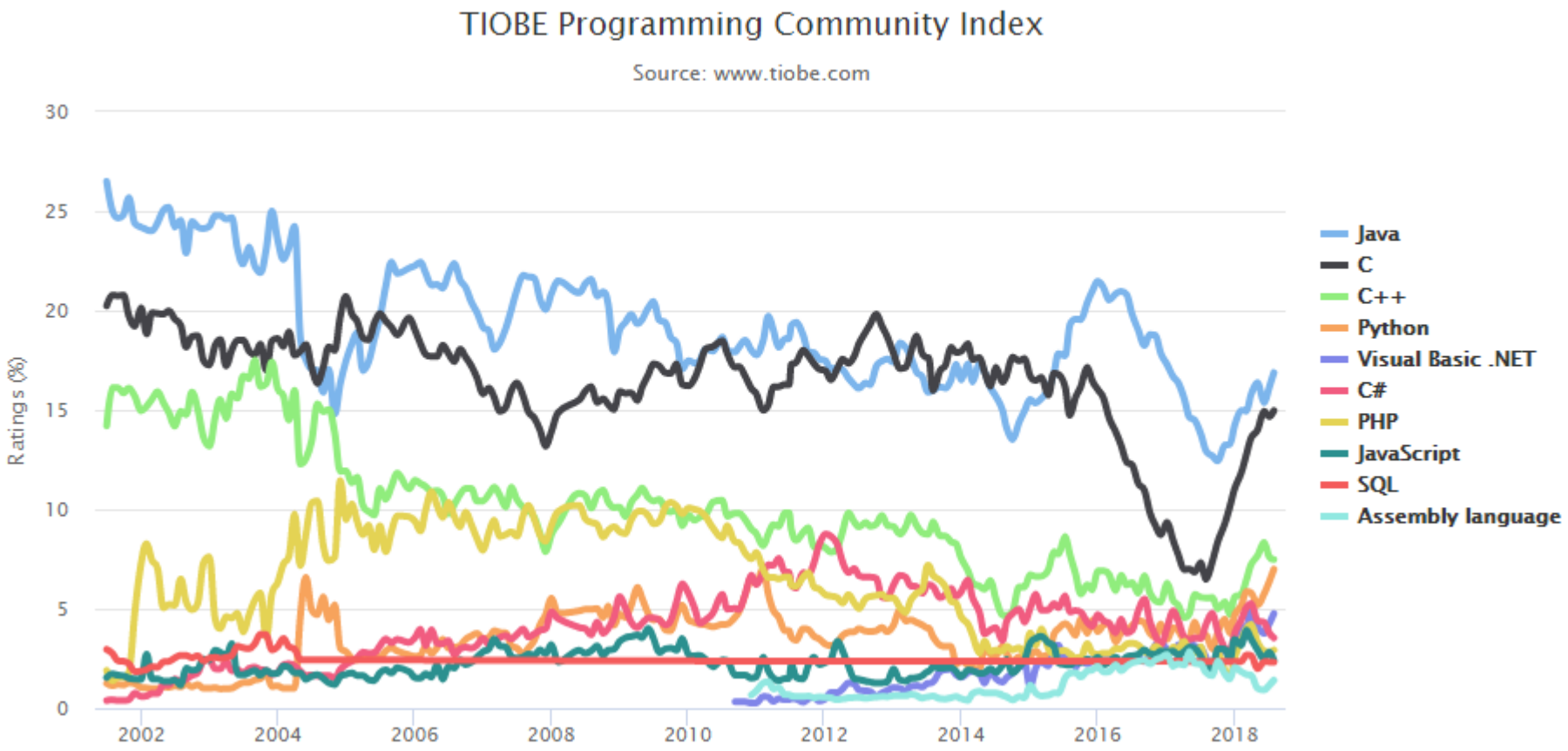


Image from https://redmonk.com/sogrady/2018/03/07/language-rankings-1-18/

Java is ...

Used by 90% of Fortune 500 companies as a server-side language

Used in all Android applications



“Hello world” in Java

```
/**
 * A first example program to print Hello world
 */

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

Basic Java syntax

Every Java **statement** needs to end with a **semicolon**

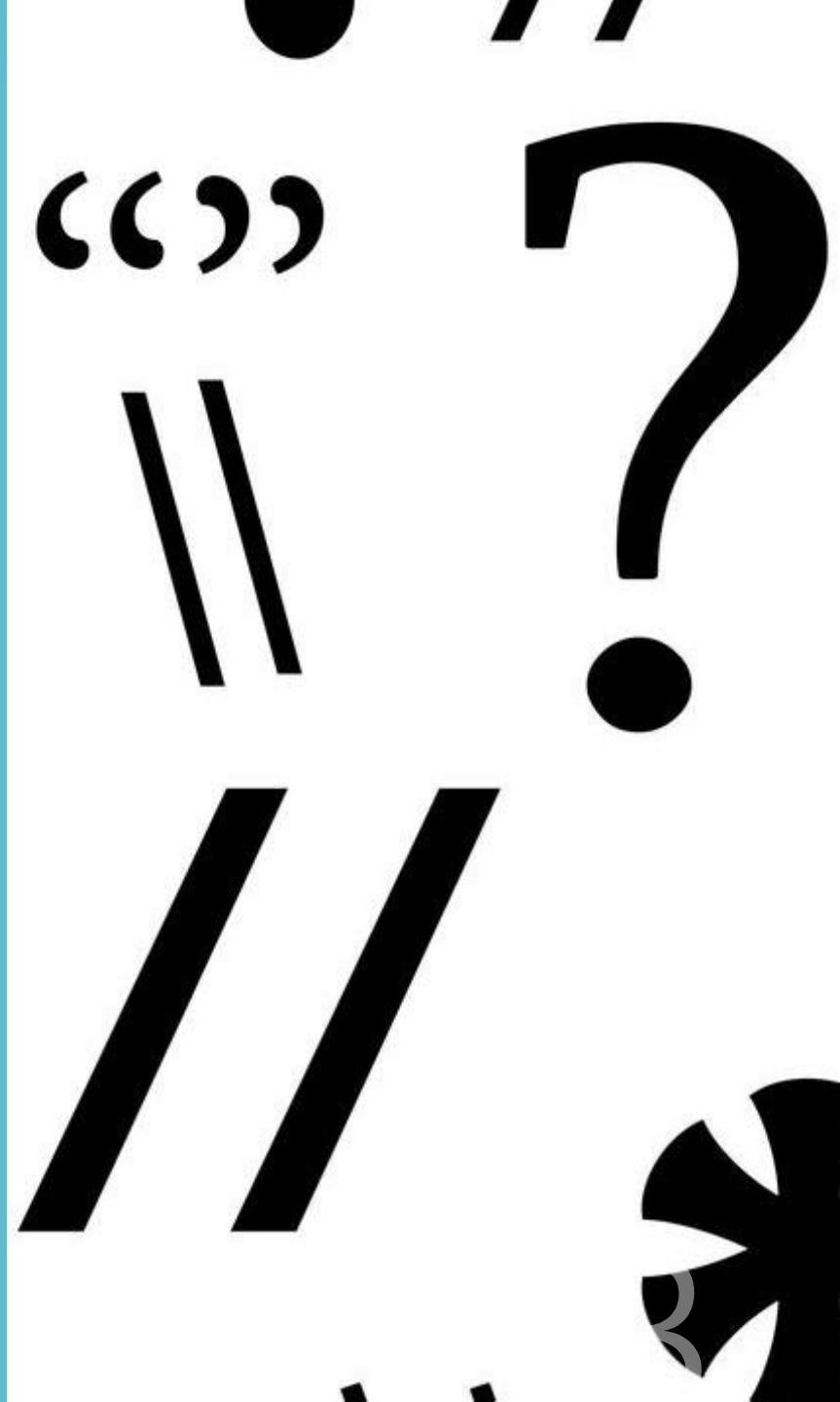
Every **block** is surrounded by **curly brackets**

Producing output: Use **System.out.println()**

```
System.out.println("hello");
System.out.println(5);
System.out.println("hello" + 5);
```

Commenting:

- Single line comments: start with //
- Multi-line comments: start with /*, end with */



Java without objects -- JShell

JShell is a Read-Evaluate-Print Loop ("REPL") for Java

Included in Java since Java 9 (September 2017)

In JShell, you can

- Enter program elements one at a time
- Immediately see the result
- Make adjustments as needed

Image from http://www.importnew.com/16353.html

“Hello world” in ~~Java~~ JShell

```
/**
 * A first example program to print Hello world
 */

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