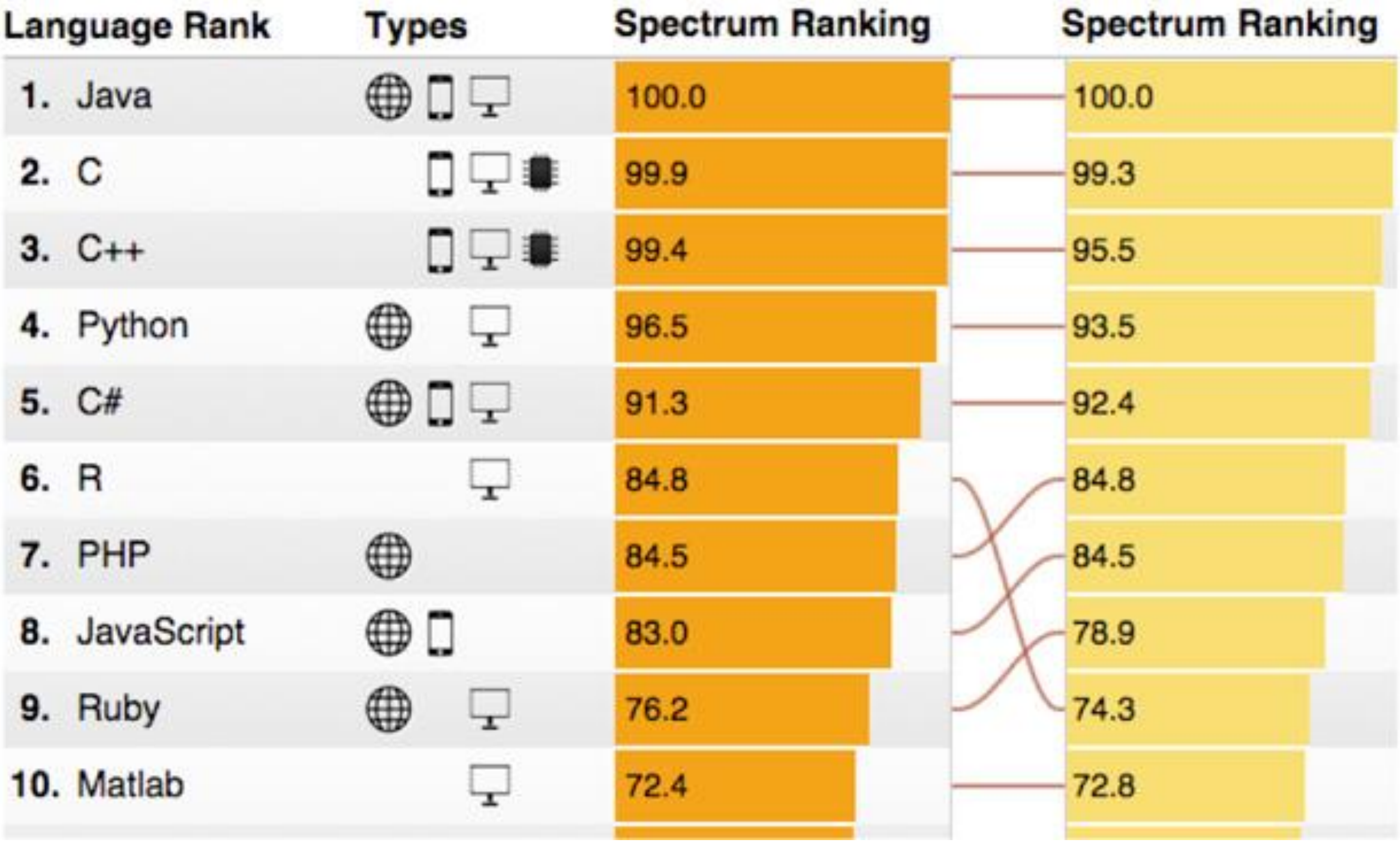
Introduction

Outline

- Introduction
- Features of Java
- Core Mechanisms
- Object oriented programming

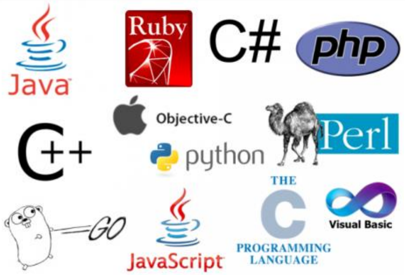
The 2015 Top 10 Programming Languages

- ❑ **IEEE Spectrum**
- ❑ **12 metrics from 10 sources**
  - ❑ **Include IEEE Xplore, Google, and GitHub**

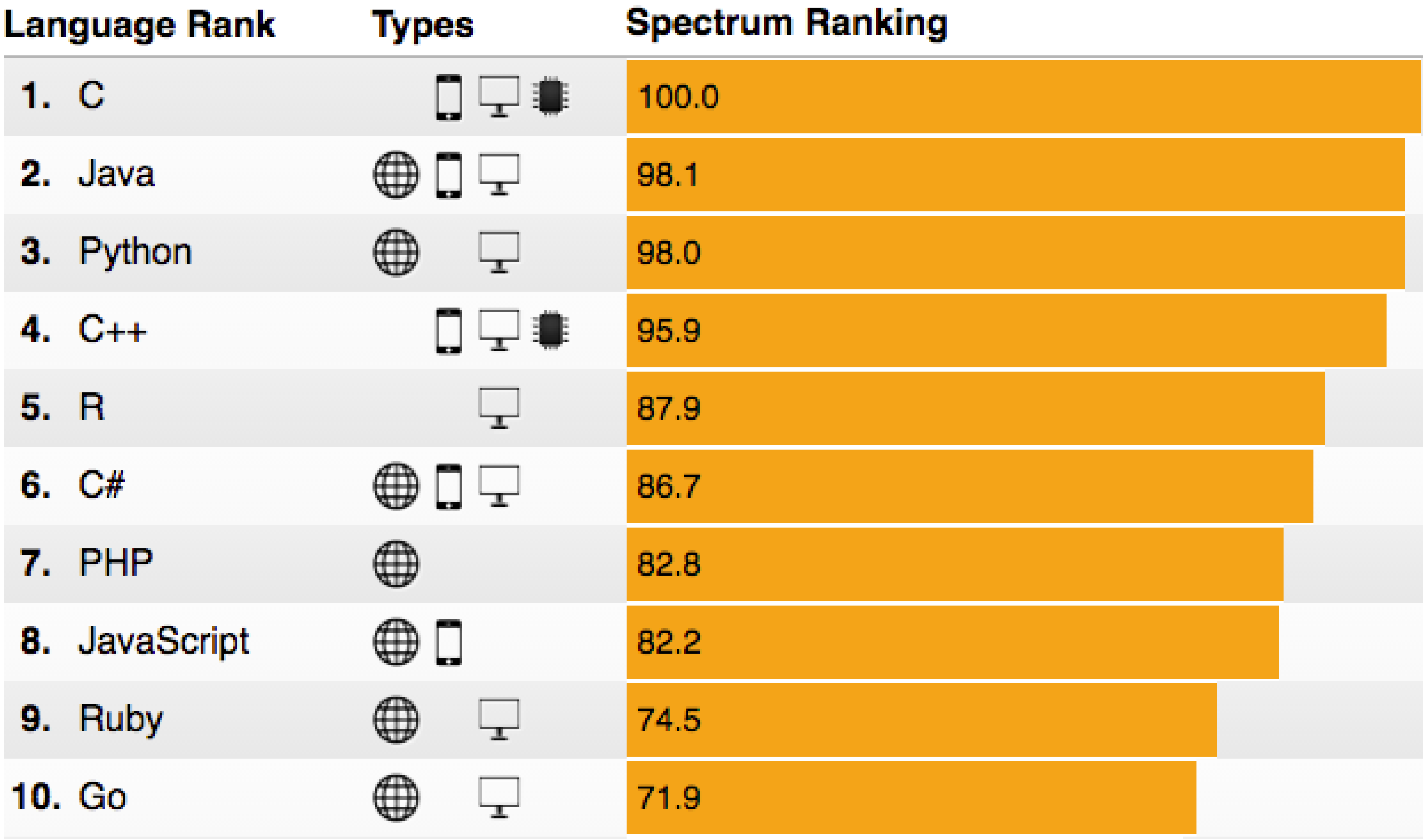


Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java	🌐 📱 🖥	100.0	100.0
2. C	📱 🖥 ▣	99.9	99.3
3. C++	📱 🖥 ▣	99.4	95.5
4. Python	🌐 🖥	96.5	93.5
5. C#	🌐 📱 🖥	91.3	92.4
6. R	🖥	84.8	84.8
7. PHP	🌐	84.5	84.5
8. JavaScript	🌐 📱	83.0	78.9
9. Ruby	🌐 🖥	76.2	74.3
10. Matlab	🖥	72.4	72.8

The 2016 Top 10 Programming Languages

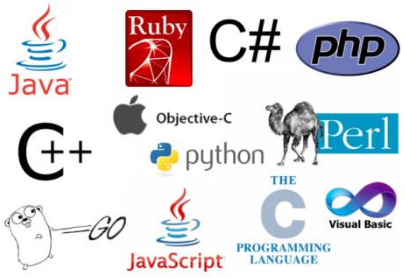


- ❑ **IEEE Spectrum**
- ❑ **12 metrics from 10 sources**
  - ❑ **Include IEEE Xplore, Google, and GitHub**

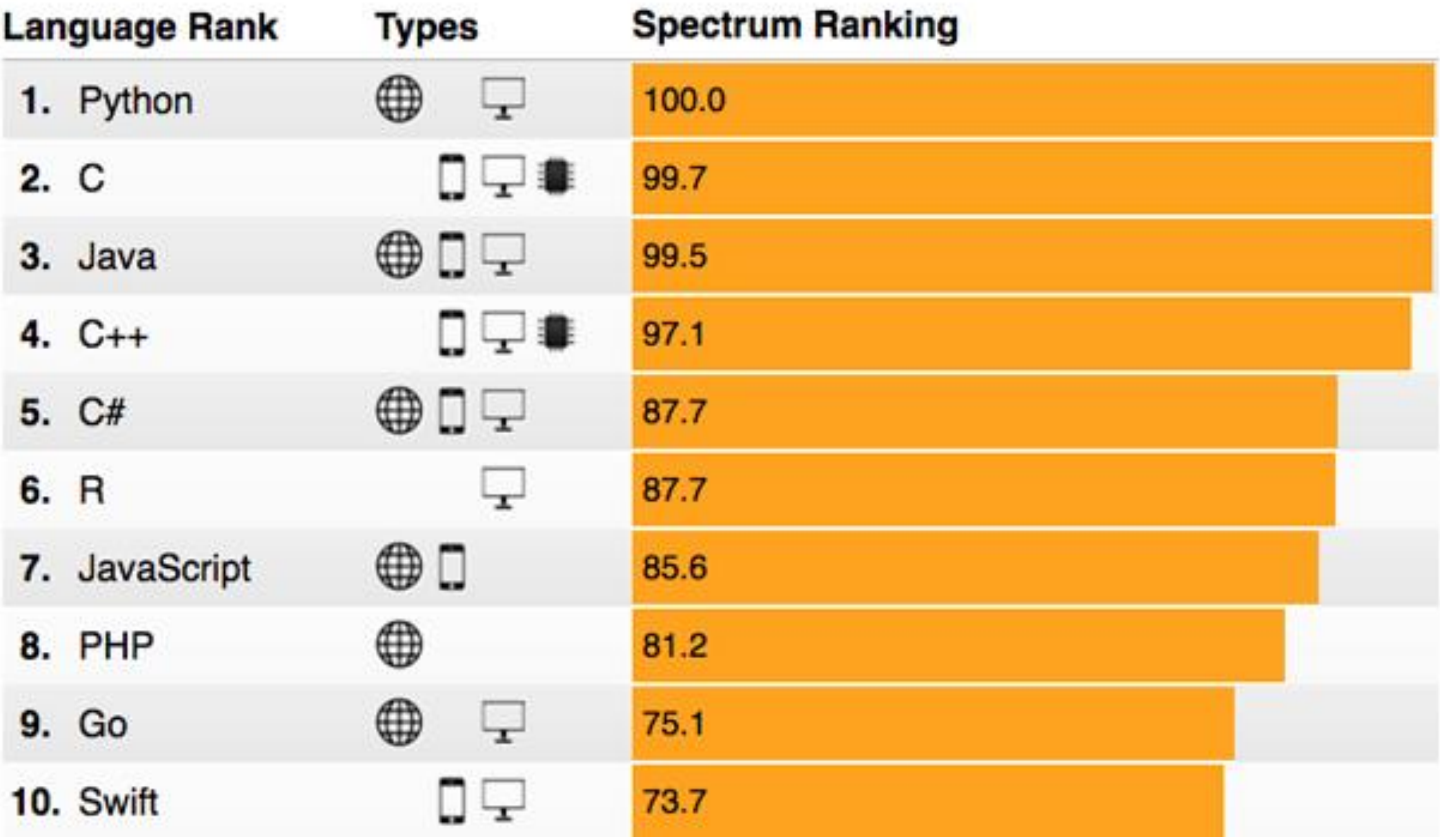


Language Rank	Types	Spectrum Ranking
1. C	📱 🖥 ▣	100.0
2. Java	🌐 📱 🖥	98.1
3. Python	🌐 🖥	98.0
4. C++	📱 🖥 ▣	95.9
5. R	🖥	87.9
6. C#	🌐 📱 🖥	86.7
7. PHP	🌐	82.8
8. JavaScript	🌐 📱	82.2
9. Ruby	🌐 🖥	74.5
10. Go	🌐 🖥	71.9

The 2017 Top 10 Programming Languages

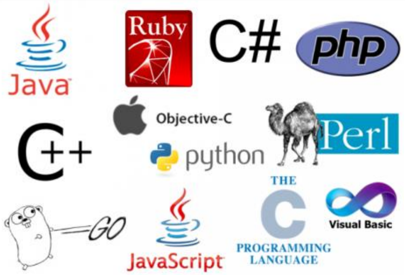


- ❑ **IEEE Spectrum**
- ❑ **12 metrics from 10 sources**
  - ❑ **Include IEEE Xplore, Google, and GitHub**

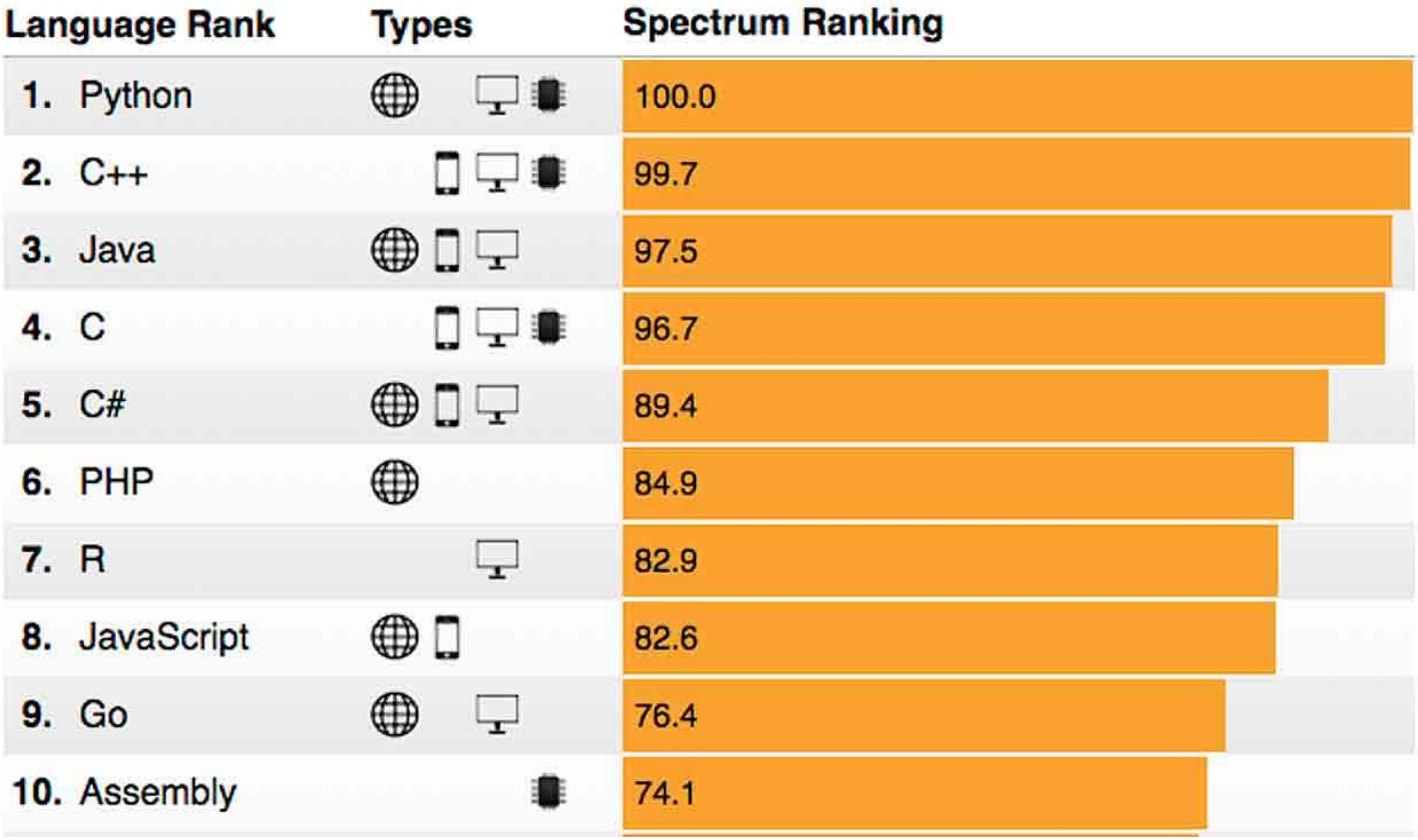


Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥	100.0
2. C	📱 🖥 ▦	99.7
3. Java	🌐 📱 🖥	99.5
4. C++	📱 🖥 ▦	97.1
5. C#	🌐 📱 🖥	87.7
6. R	🖥	87.7
7. JavaScript	🌐 📱	85.6
8. PHP	🌐	81.2
9. Go	🌐 🖥	75.1
10. Swift	📱 🖥	73.7

The 2018 Top 10 Programming Languages



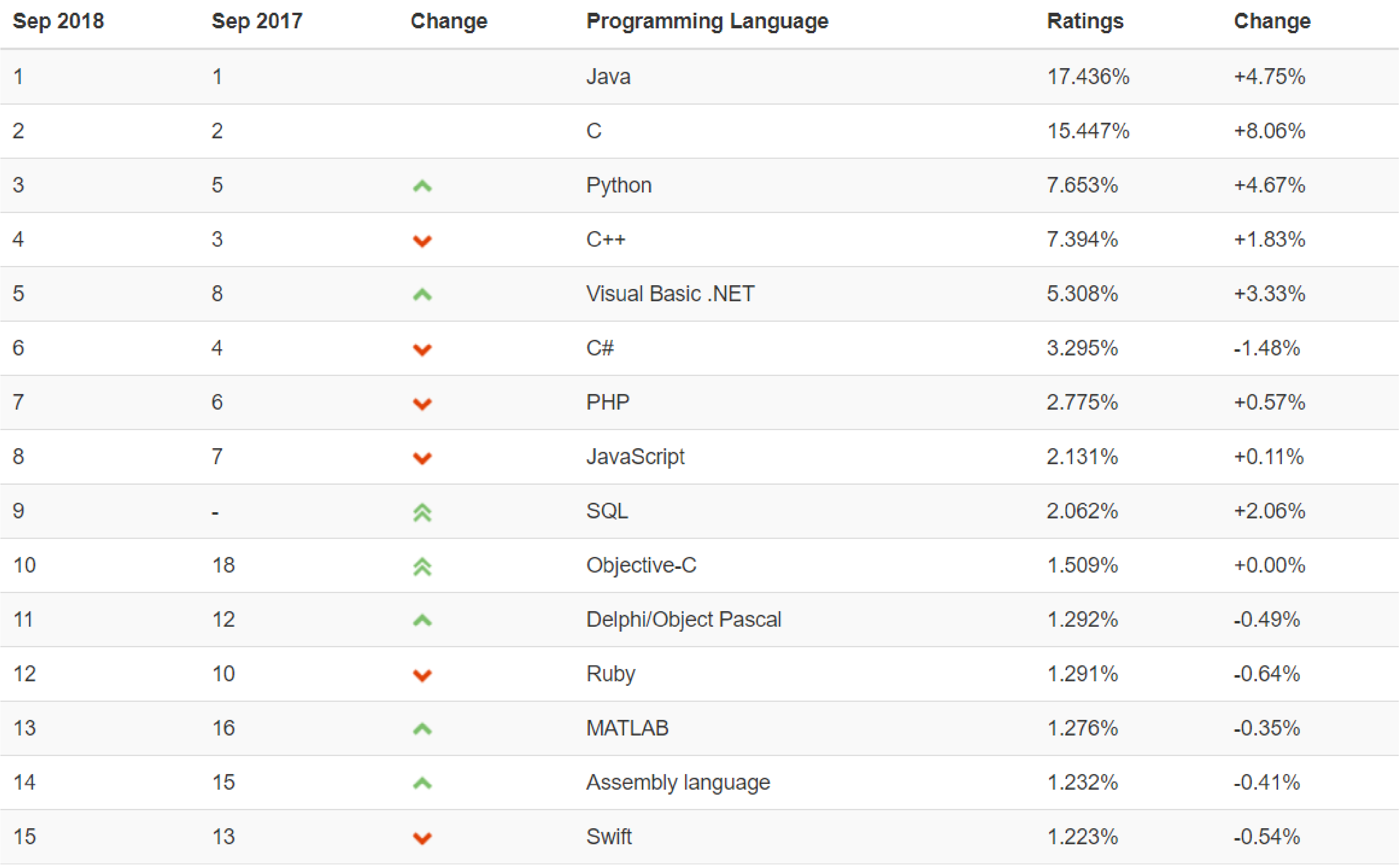
- ❑ IEEE Spectrum
- ❑ 12 metrics from 10 sources
  - ❑ Include IEEE Xplore, Google, and GitHub



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ▪	100.0
2. C++	📱 🖥 ▪	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ▪	96.7
5. C#	🌐 📱 🖥	89.4
6. PHP	🌐	84.9
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	▪	74.1

The 2018 Top 10 Programming Languages

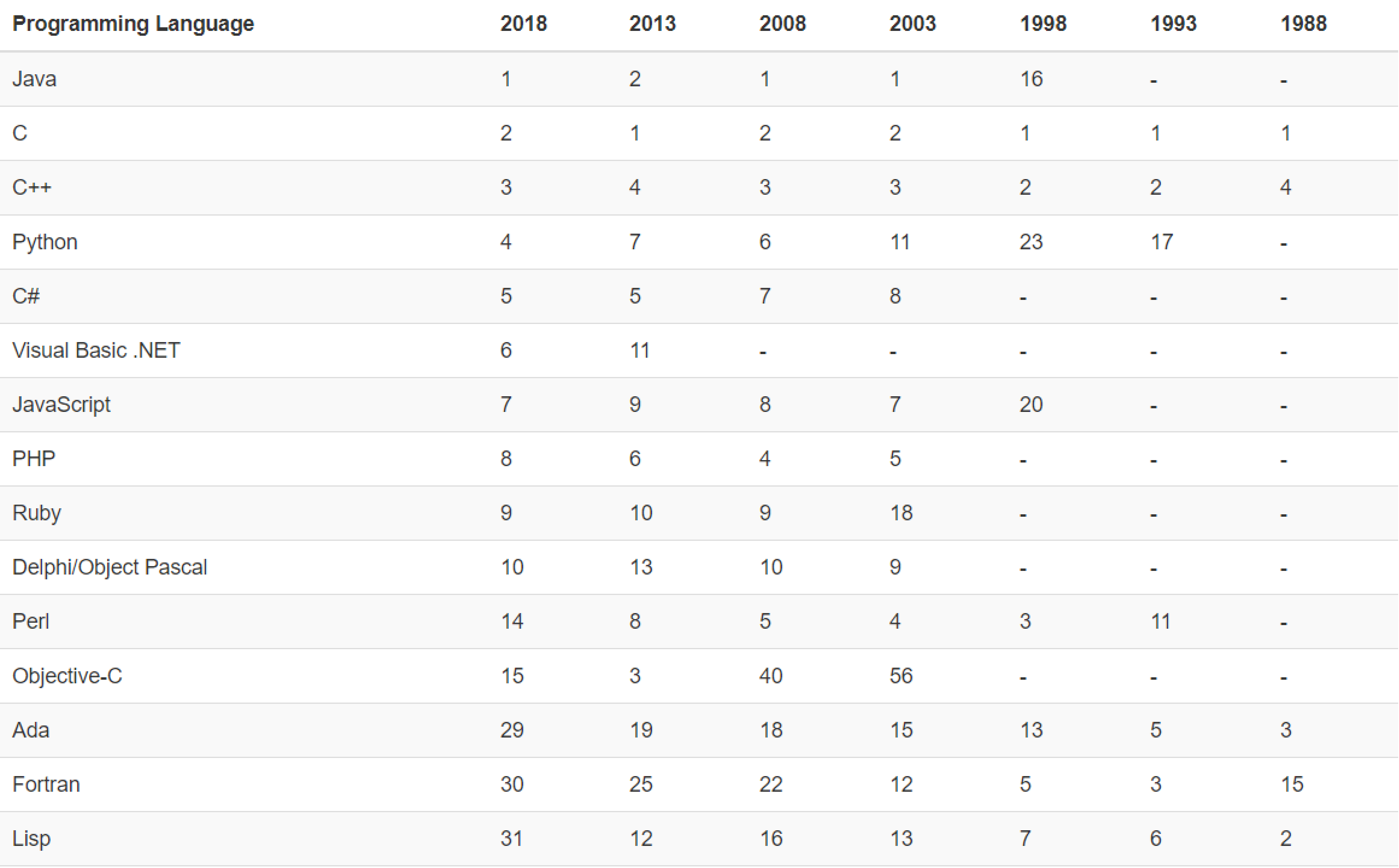
❑ TIOBE Index for September 2018



Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲▲	SQL	2.062%	+2.06%
10	18	▲▲	Objective-C	1.509%	+0.00%
11	12	▲	Delphi/Object Pascal	1.292%	-0.49%
12	10	▼	Ruby	1.291%	-0.64%
13	16	▲	MATLAB	1.276%	-0.35%
14	15	▲	Assembly language	1.232%	-0.41%
15	13	▼	Swift	1.223%	-0.54%

The 2018 Top 10 Programming Languages

❑ TIOBE Index for September 2018



Programming Language	2018	2013	2008	2003	1998	1993	1988
Java	1	2	1	1	16	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	4
Python	4	7	6	11	23	17	-
C#	5	5	7	8	-	-	-
Visual Basic .NET	6	11	-	-	-	-	-
JavaScript	7	9	8	7	20	-	-
PHP	8	6	4	5	-	-	-
Ruby	9	10	9	18	-	-	-
Delphi/Object Pascal	10	13	10	9	-	-	-
Perl	14	8	5	4	3	11	-
Objective-C	15	3	40	56	-	-	-
Ada	29	19	18	15	13	5	3
Fortran	30	25	22	12	5	3	15
Lisp	31	12	16	13	7	6	2

Java is everywhere!

- 97% of Enterprise Desktops Run Java
- 89% of Desktops (or Computers) in the U.S. Run Java
- 9 Million Java Developers Worldwide
- #1 Choice for Developers
- #1 Development Platform
- 3 Billion Mobile Phones Run Java
- 100% of Blu-ray Disc Players Ship with Java
- 5 Billion Java Cards in Use
- 125 million TV devices run Java
- 5 of the Top 5 Original Equipment Manufacturers Ship Java ME

Java is everywhere! (Cont.)

□ **Mobile Computing**

android

□ **Data Mining**

WEKA
The University
of Waikato

□ **Computer Vision**

OpenCV

□ **Natural Language Processing**

openNLP™

□ **Big Data**

hadoop

Spark

STORM

□ **Machine Learning**

TensorFlow

Java is everywhere! (Cont.)

- Mobile Computing

ANDROID

- Data Mining
- Computer Vision

OpenCV

- Natural Language Processing
- Big Data
- Machine Learning

History of Programming Language

- ❑C language
  - ❑Developed in 1972 by Dennis Ritchie
  - ❑Be tedious with its structural syntax

- ❑C++
  - ❑Developed in 1979 by Bjarne Stroustrup
  - ❑An enhancement to the C language with included OOP fundamentals and features

- ❑Engineer Patrick Naughton
  - ❑He is frustrated with the state of Sun's C++ and C APIs and tools
  - ❑He is offered a chance to work on new technology and the *Stealth Project* was started in Dec. of 1990

History of Java

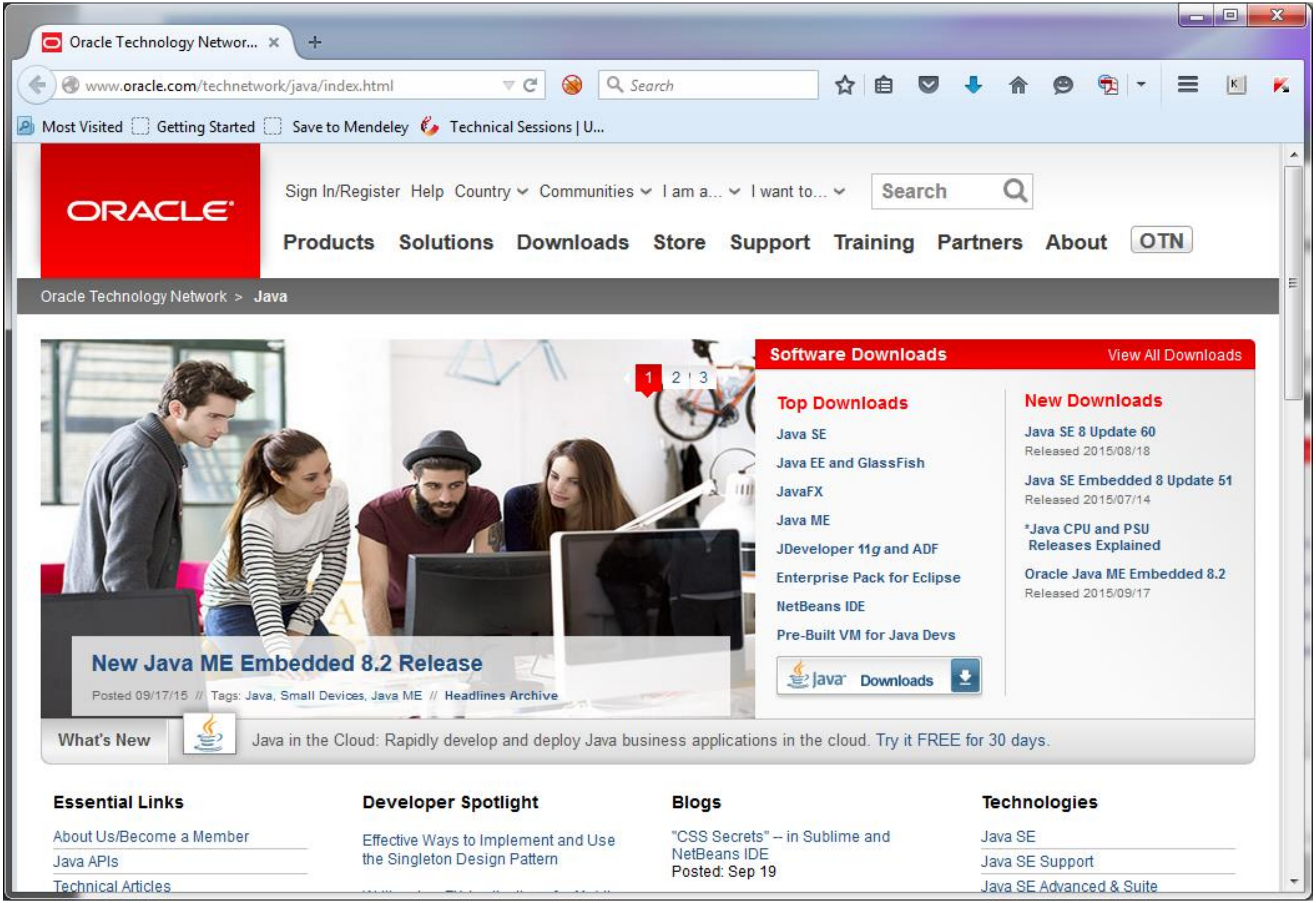
- ❑January 1991, Stealth Project
  - ❑Green Project (Green team)
  - ❑Build software for consumer electronics
  - ❑Gosling works on the "Oak" interpreter
  - ❑Develop for an embedded system with limited resources

History of Java (Cont.)

- **Green Team was a failure**
  - **Too advanced for the digital cable television industry at the time**
- **Bill Joy, one of the co-founders of Sun**
  - **Saw an opportunity for Oak in the emergence of the World Wide Web**
  - **Re-targeted the platform for the World Wide Web**
- **1995, Oak was renamed Java**
  - **Release Java for free over the Internet (JDK 1.0)**
  - **Netscape announces its intention to license Java for use in Netscape browser**

James Gosling

Java Homepage – java.sun.com



Top Downloads

- Java SE
- Java EE and GlassFish
- JavaFX
- Java ME
- JDeveloper 11g and ADF
- Enterprise Pack for Eclipse
- NetBeans IDE
- Pre-Built VM for Java Devs

New Downloads

Java SE 8 Update 60
Released 2015/08/18

Java SE Embedded 8 Update 51
Released 2015/07/14

*Java CPU and PSU Releases Explained

Oracle Java ME Embedded 8.2
Released 2015/09/17

New Java ME Embedded 8.2 Release

Posted 09/17/15 // Tags: Java, Small Devices, Java ME // Headlines Archive

Java in the Cloud: Rapidly develop and deploy Java business applications in the cloud. Try it FREE for 30 days.

Essential Links

- About Us/Become a Member
- Java APIs
- Technical Articles

Developer Spotlight

Effective Ways to Implement and Use the Singleton Design Pattern

Blogs

"CSS Secrets" -- in Sublime and NetBeans IDE
Posted: Sep 19

Technologies

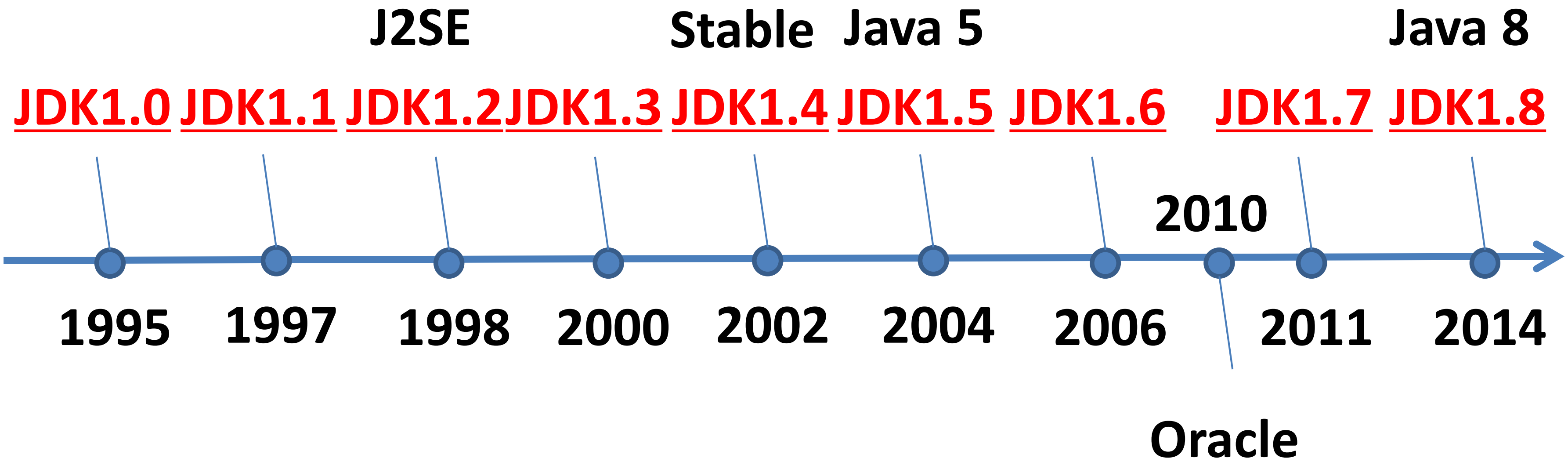
- Java SE
- Java SE Support
- Java SE Advanced & Suite

Java Platforms

- Three Platforms
  - Java SE - J2SE, Java 2 Platform Standard Edition
  - Java EE - J2EE, Java 2 Platform Enterprise Edition
  - Java ME - J2ME, Java 2 Platform Micro Edition

- Java Development Kit (JDK)

Java Version History



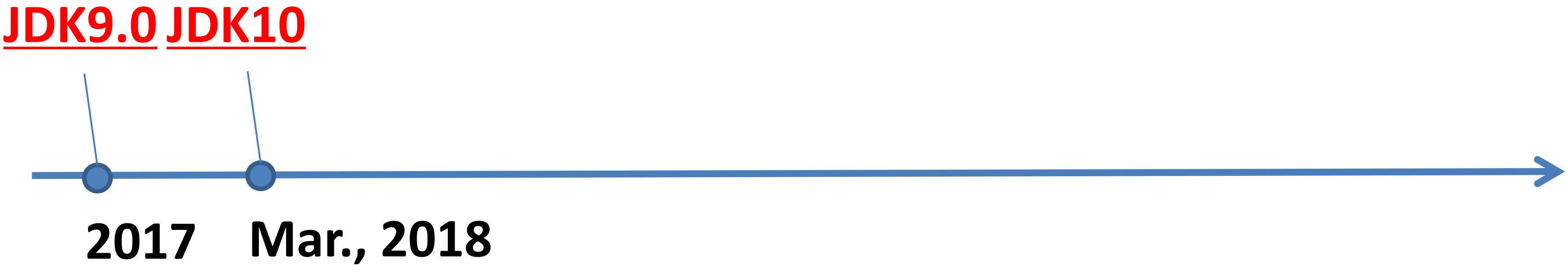
J2SE
Stable
Java 5
Java 8

JDK1.0 JDK1.1 JDK1.2 JDK1.3 JDK1.4 JDK1.5 JDK1.6 JDK1.7 JDK1.8

2010

1995 1997 1998 2000 2002 2004 2006 2011 2014

Oracle



JDK9.0 JDK10

2017 Mar., 2018

Six-month release model

Java Enhancements

- JDK1.4
  - Assert, Logging, Java2D, NIO, etc.
- Java 5
  - Generics, Enhanced for Loop, Autoboxing/Unboxing, Typesafe Enums, Varargs, Static Import, Annotations, etc.
- Java 6
  - Compiler API, Scripting, WebService, etc.
- Java 7
  - Improved exception handling, Strings in switch Statements, Improved Compiler Warnings and Errors
  - http://docs.oracle.com/javase/7/docs/technotes/guides/language/enhancements.html

Java Enhancements

- JDK 8
  - forEach() method in Iterable interface, default and static methods in Interfaces, Functional Interfaces and Lambda Expressions, etc.
- Java 9
  - Enhancements to the Streams API, Better JavaScript backing, HTTP/2 client API, Improved HTML5 and Unicode support, DTLS security API, etc.
- Java 10
  - Local-Variable Type Inference, Garage Collector Interface, Parallel Full GC for G1, Thread-Local Handshakes, etc.

JCP and JSR

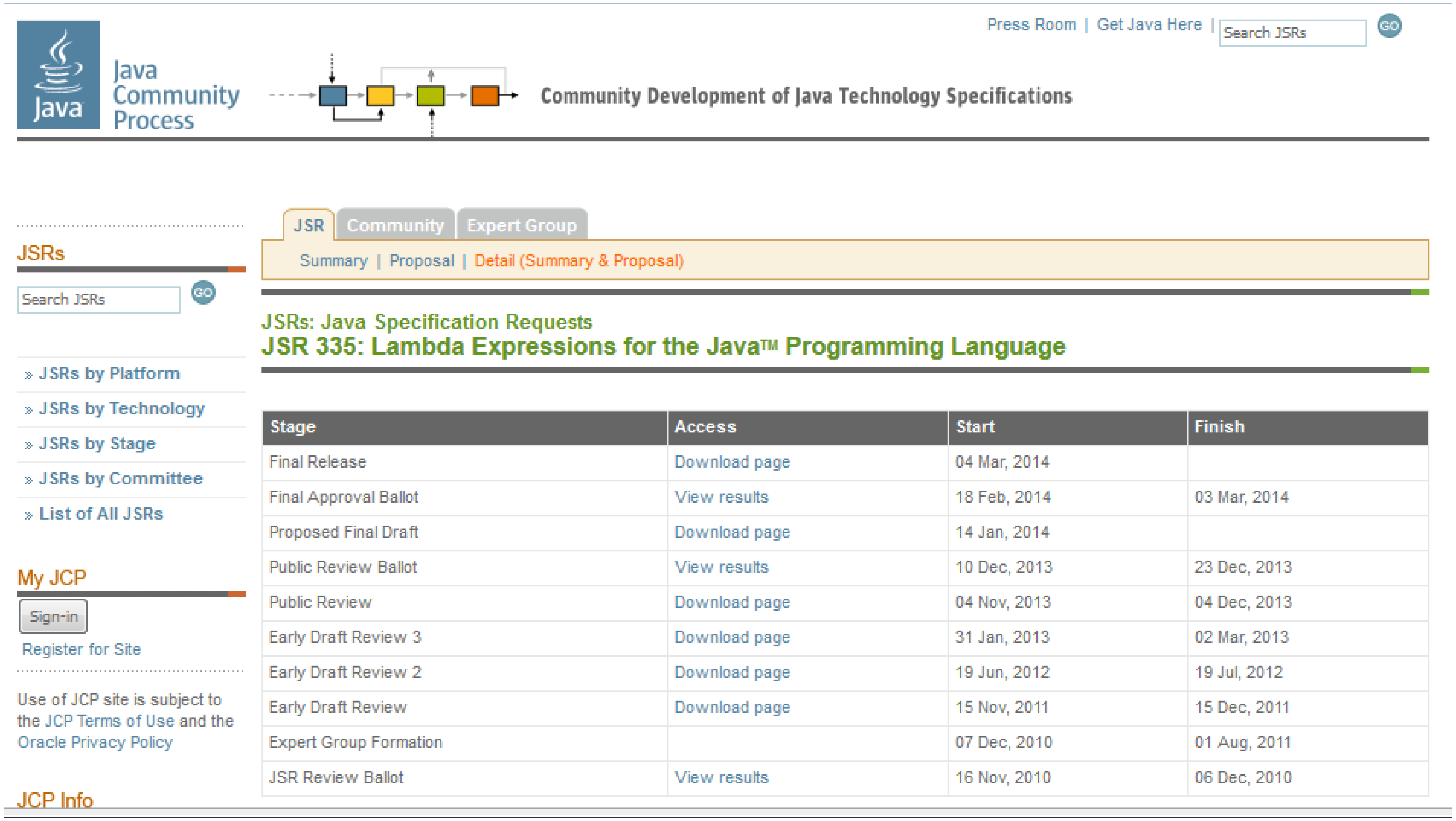
- ❑ **http://jcp.org**
  - ❑ **Welcome to jcp.org, home of the Java Community Process^{SM} (JCP^{SM}) Program. The JCP is the mechanism for developing standard technical specifications for Java technology. Anyone can register for the site and participate in reviewing and providing feedback for the Java Specification Requests (JSRs), and anyone can sign up to become a JCP Member and then participate on the Expert Group of a JSR or even submit their own JSR Proposals.**

Java

Java
Community
Process

JSR 335

- ❑ **JSR**
  - ❑ **Lambda Expressions for the Java Programming Language**
  - ❑ **One of Enhancements in Java SE 8**



Java
Java Community Process
Community Development of Java Technology Specifications
Press Room | Get Java Here | Search JSRs GO
JSR | Community | Expert Group

Summary | Proposal | Detail (Summary & Proposal)

JSRs: Java Specification Requests

JSR 335: Lambda Expressions for the Java™ Programming Language

Stage	Access	Start	Finish
Final Release	Download page	04 Mar, 2014	
Final Approval Ballot	View results	18 Feb, 2014	03 Mar, 2014
Proposed Final Draft	Download page	14 Jan, 2014	
Public Review Ballot	View results	10 Dec, 2013	23 Dec, 2013
Public Review	Download page	04 Nov, 2013	04 Dec, 2013
Early Draft Review 3	Download page	31 Jan, 2013	02 Mar, 2013
Early Draft Review 2	Download page	19 Jun, 2012	19 Jul, 2012
Early Draft Review	Download page	15 Nov, 2011	15 Dec, 2011
Expert Group Formation		07 Dec, 2010	01 Aug, 2011
JSR Review Ballot	View results	16 Nov, 2010	06 Dec, 2010

JSRs

Search JSRs GO

- » JSRs by Platform
- » JSRs by Technology
- » JSRs by Stage
- » JSRs by Committee
- » List of All JSRs

My JCP

Sign-in

Register for Site

Use of JCP site is subject to the JCP Terms of Use and the Oracle Privacy Policy

JCP Info

Outline

- Introduction
- **Features of Java**
- Core Mechanisms
- Object oriented programming

What is Java

- Java is a programming language, very similar to C++

- Java Programming Language not too large of itself

- Large growing set of utilities and other library packages in accompanying development kit

Features of Java

- ❑Simple
- ❑Object-oriented
- ❑Robust
- ❑Portable
- ❑Secure
- ❑Multithreaded

Libraries

- ❑Language package
- ❑Utility package
- ❑I/O package
- ❑Network package
- ❑User interface package
- ❑And more

What can Java do

- ❑Network Application
- ❑Web Application
- ❑Visualization
- ❑Database Operation
- ❑And more

Java is C++ --

- No pointer
- No malloc, delete etc.
    - Automated memory administration
- Fixed data size
- No includes
- No structure and union
- No macros/preprocessor
- No GOTO

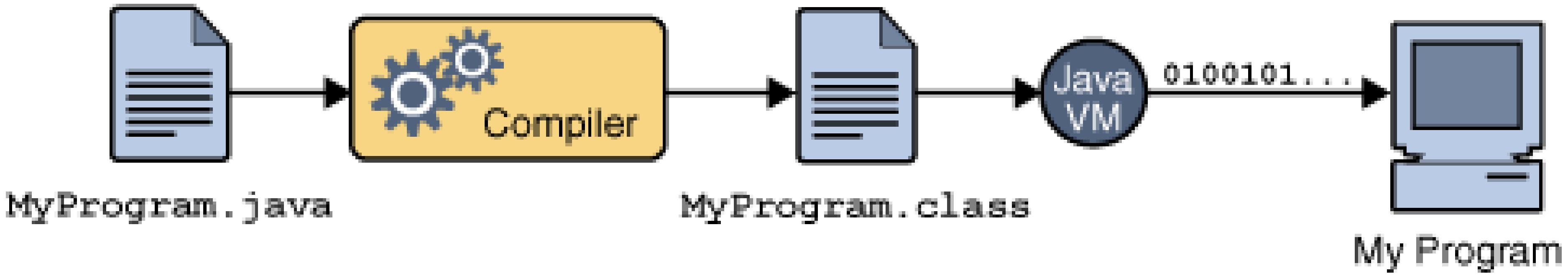
Outline

- Introduction
- Features of Java
- **Core Mechanisms**
- Object oriented programming

Core Mechanisms

- Java Virtual Machine
- Code Security
- Garbage Collection

Compile and run



- ❑ **Edit ASCII source code in MyProgram.java**

- ❑ **Compile to byte-code in MyProgram.class**

- ❑ **Load and run in JVM (as stand-alone or in WWW browser)**

Java Virtual Machine

- Java Virtual Machine (JVM) constitutes
  - Java Runtime Environment (JRE)
  - Garbage Collector (GC) - as a separate thread
  - Security model

- JVM Defines
  - Instruction set
  - Register set
  - Class file format
  - Stack
  - Garbage collected heap
  - Memory area

Java Runtime Environment

- ❑JRE
  - ❑ JRE = JVM + API (Lib)

- ❑Three main functions
  - ❑Load code: class loader
  - ❑Check code: bytecode verifier
  - ❑Execute code: runtime interpreter

- ❑Understand
  - ❑Cross-Platform
  - ❑Secure

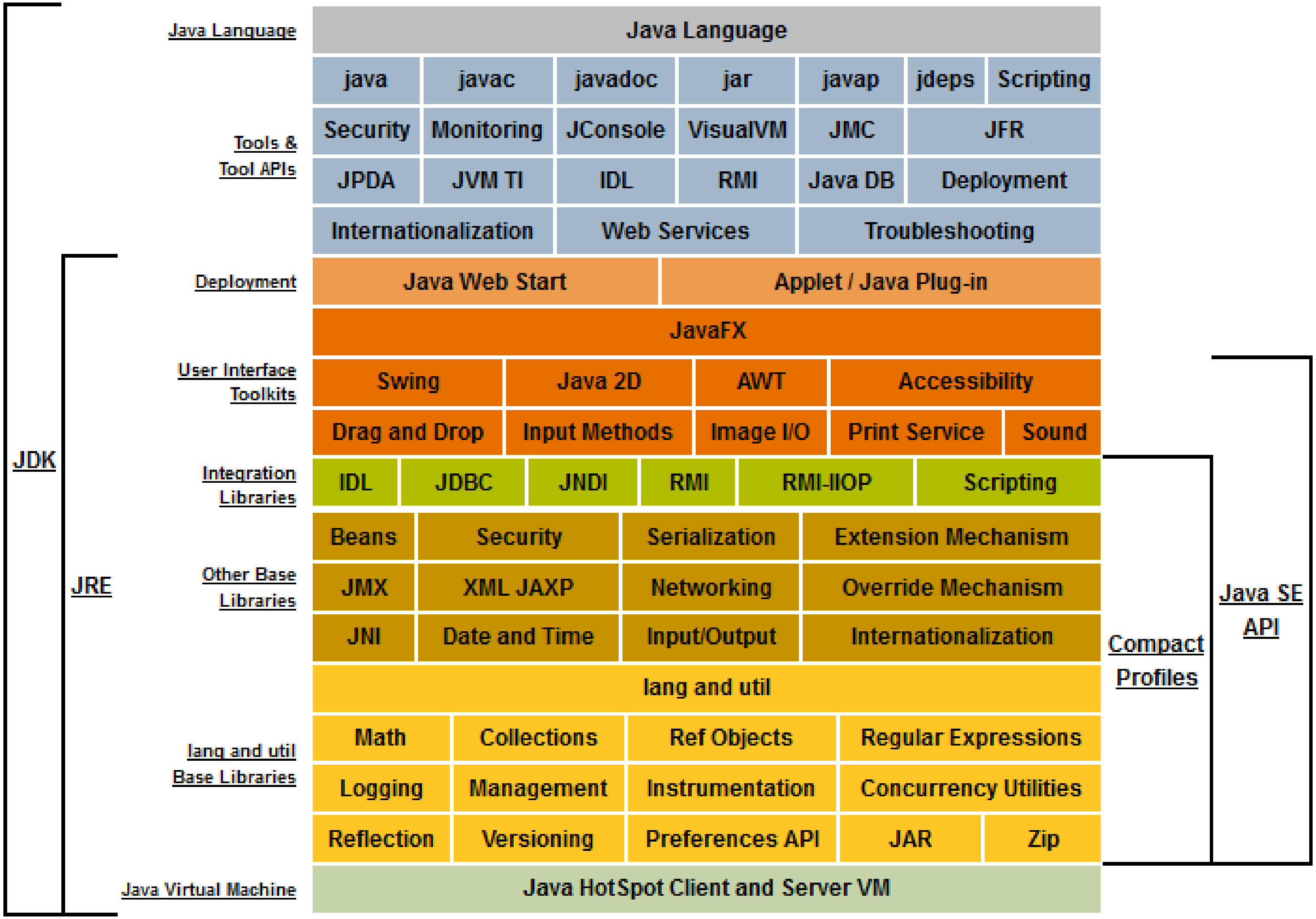
Java Garbage Collection

- ❑C/C++，developer needs to collect the memory

- ❑Java auto garbage collection
  - ❑A System thread tracks memory assignment
  - ❑Release the memory when JVM is idle
  - ❑Developer cannot control garbage collection

JDK (Java Development Kit)

- JDK = JRE+Tools JRE = JVM + API
  - http://docs.oracle.com/javase/8/docs/



JDK

JRE

Java Language: Java Language

Tools & Tool APIs:
java | javac | javadoc | jar | javap | jdeps | Scripting
Security | Monitoring | JConsole | VisualVM | JMC | JFR
JPDA | JVM TI | IDL | RMI | Java DB | Deployment
Internationalization | Web Services | Troubleshooting

Deployment: Java Web Start | Applet / Java Plug-in

User Interface Toolkits:
JavaFX
Swing | Java 2D | AWT | Accessibility
Drag and Drop | Input Methods | Image I/O | Print Service | Sound

Integration Libraries: IDL | JDBC | JNDI | RMI | RMI-IIOP | Scripting

Other Base Libraries:
Beans | Security | Serialization | Extension Mechanism
JMX | XML JAXP | Networking | Override Mechanism
JNI | Date and Time | Input/Output | Internationalization

lang and util Base Libraries:
lang and util
Math | Collections | Ref Objects | Regular Expressions
Logging | Management | Instrumentation | Concurrency Utilities
Reflection | Versioning | Preferences API | JAR | Zip

Java Virtual Machine: Java HotSpot Client and Server VM

Compact Profiles

Java SE API

More Java Techniques

- To develop Java code, we need to download JDK
  - http://java.sun.com
  - Download JavaSE or NetBeans
- Note: JRE only, if you just run a Java program
  - Http://java.com
- JDK
  - javac.exe
  - java.exe
  - javadoc.exe
  - jar.exe
  - jdb.exe

More Java Techniques

- ❑ Java SE
- ❑ Java SE Advanced & Suite
- ❑ Java Embedded
- ❑ Java EE
- ❑ Java ME
- ❑ JavaFX
- ❑ Java Card
- ❑ Java TV
- ❑ Java DB
- ❑ Developer Tools

Outline

- Introduction
- Features of Java
- Core Mechanisms
- **Object oriented programming**

Object

- Object has two meanings
  - In the real world
    - Object in the real world
  - In the computer world
    - Memory space

Class

- ❑Class
  - ❑Field
  - ❑Method

Class

```
class Person{
        int age;
        String name;
        void sayHello(){…}
}
```

Object

```
Person p = new Person()
```

- ❑Class and Object
  - ❑Class is a template of objects
  - ❑Object is an instance of class

Features of OOP

- ❑Encapsulation
- ❑Inheritance
- ❑Polymorphism

Encapsulation

- Template
  - Encapsulate field and method into a class
- Hide information
  - Hide the details
  - Access the class via interface

```
class Person{
    private int age;
    public int getAge(){ return age; }
    public void setAge(int a){ age=a;}
    String name;
    void sayHello(){…}
}
```

Inheritance

- Parent Class and child class share data and method
- Abstract and classify
- Reuse the code
- Easy to maintain the code

```
class Person{
        int age;
        String name;
        void sayHello(){…}
}
class Student extends Person{
        String school;
        double score;
        void meetTeacher(){ … }
}
```

Polymorphism

- Examples
  - foo(Person p){ p.sayHello(); }
  - foo(new Student());
  - foo(new Teacher());

Design of OOP

- ❑Everything is an object
- ❑Design
  - ❑Abstract the Classes
  - ❑Method, field of a class
  - ❑Relation between classes (inheritance, etc.)
  - ❑Send messages between objects (methods)

Learning Pattern

- ❑ **Lecturer: theory with sample programs**

- ❑ **Labs (135 mins): Practice theory using Java programs**

- ❑ **Workshops/seminars: discussion/reflection**

Assessment Pattern

- Components of Course Grade
  - Attendance ---- 10%
  - Project ---- 20%
  - Final Exam ---- 70%
  - No Midterm Exam

Course Schedule

Week	Monday	Wednesday
1	J2-308(8-9)	J2-308(3-5)
2	J2-308(8-9)	COSE235(3-5)
3	J2-308(8-9)	J2-308(3-5)
4	J2-308(8-9)	COSE235(3-5)
5	J2-308(8-9)	J2-308(3-5)
6	J2-308(8-9)	COSE235(3-5)
7	J2-308(8-9)	J2-308(3-5)
8	J2-308(8-9)	COSE235(3-5)
9	J2-308(8-9)	COSE235(3-5)
10	J2-308(8-9)	COSE235(3-5)
11	J2-308(8-9)	COSE235(3-5)
12	J2-308(8-9)	COSE235(3-5)
13	J2-308(8-9)	COSE235(3-5)
14	J2-308(8-9)	COSE235(3-5)
15		COSE235(3-5)

Materials

- The best textbook is Javadoc
- Textbook
  - Java Programming Language 4ed
- Readings
  - Thinking in Java
  - Head First Java
  - Better, Faster and Lighter Java
  - Beyond Java
  - Inside Java Virtual Machine
  - etc.

Thank you

zhenling@seu.edu.cn