



Introduction to Java

What is Java or Java Technology

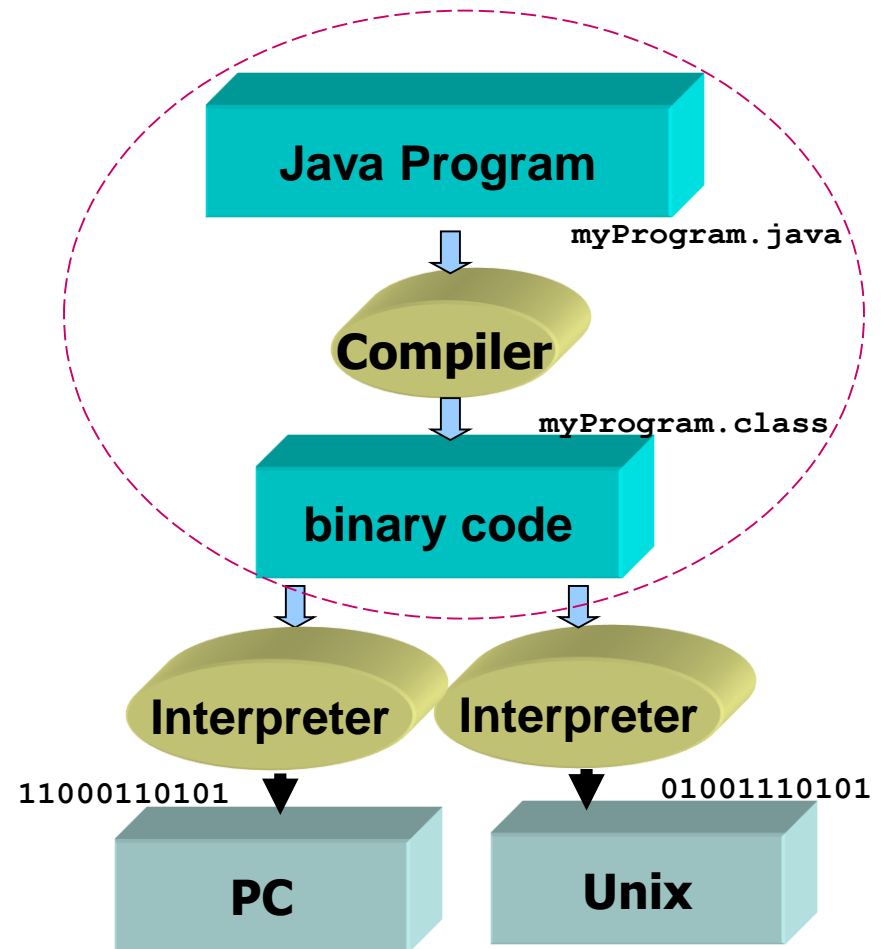
- A Programming Language
 - Syntax of the Java programming language is similar to C++ syntax
- A Development Environment
 - A large suite of tools: compiler (`javac`), interpreter (`java`), documentation generator (`javadoc`), class file packaging tool (`jar`)
- An Application Environment
 - Java applications – standalone programs
 - Java applets – run by a web browser
- A Deployment Environment
 - JRE (Java Runtime Environment)
 - Web browsers supplying a Java interpreter and runtime environment

Comparison

- Not very different from other (OO) languages, “language-wise”
 - Keywords, identifiers, types, expressions, flow controls, class, (functions) methods, inheritance etc
- What makes Java special?
 - Universal glue – connecting users with information wherever it comes from
 - Web servers, databases, information providers
 - Built-in support for advanced programming tasks
 - Network programming, database connectivity and multithreading

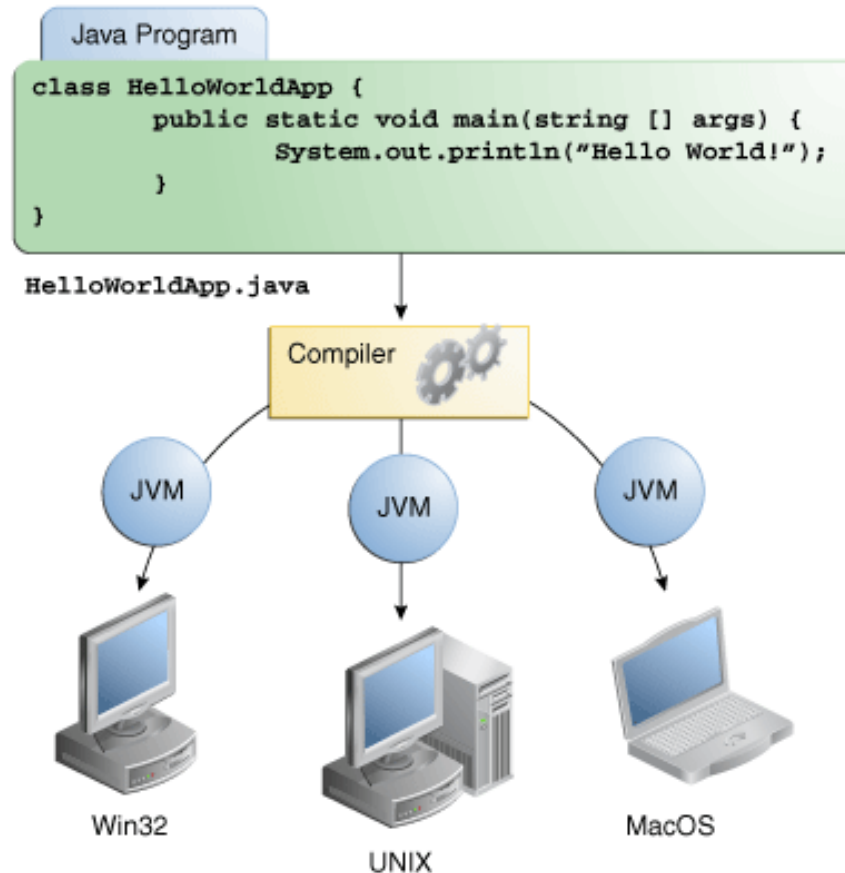
Development Process

- Java compiler:
 - `javac myProgram.java`
- Java interpreter:
 - `java myProgram`
- Java applet:
 - run in an Internet browser
 - `appletviewer`



What are the differences from other programming languages ?

Java is Architecture-Neutral

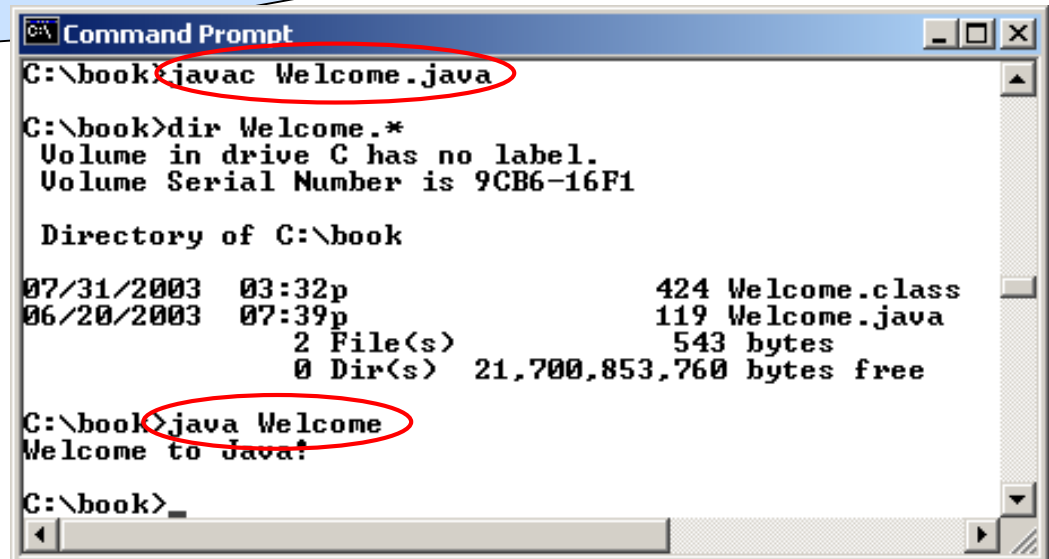


What makes it possible for Java to be architecture-neutral?

A Simple Application

```
/**
 * The Welcome class implements an application that
 * displays "Welcome to Java!" to the standard output.
 */
public class Welcome{
    public static void main(String[] args){
        // Display "Welcome to Java!"
        System.out.println("Welcome to Java!");
    }
}
```

- Everything in a Java program must be inside a class
- Class – a container for the program logic that defines the behaviour of which all Java applications and applets are built



The screenshot shows a Windows Command Prompt window with the following text:

```
C:\book>javac Welcome.java

C:\book>dir Welcome.*
Volume in drive C has no label.
Volume Serial Number is 9CB6-16F1

Directory of C:\book

07/31/2003  03:32p                424 Welcome.class
06/20/2003  07:39p                119 Welcome.java
                2 File(s)                543 bytes
                0 Dir(s)  21,700,853,760 bytes free

C:\book>java Welcome
Welcome to Java!

C:\book>
```

In the original image, the command `javac Welcome.java` and the output `Welcome to Java!` are circled in red.

Anatomy of a Java Program

- Packages
- Classes
- Variable declarations
- Methods
- Statements

```
packages

public class car {
    variable declarations

    public method1{
        local variables
        statements
    }

    private method2{
        statements
    }
    ...
}

public class engine{
    ...
}
```

Entry Points of Java Programs

- Application: `main()` method

The `main` method provides the control of program flow.

The Java interpreter executes the application by invoking the `main` method.

```
public static void main(String[] args) {  
    // Statements;  
}
```

- Applet: `init()` and `start()` methods

Example of Java Applet

```
import java.applet.*;
import java.awt.*;

/**
 * The Welcome class implements an applet that
 * simply displays "Welcome to Applet!".
 */
public class Welcome extends Applet {
    public void paint(Graphics g) {
        // Display " Welcome to Applet!"
        g.drawString("Hello world!", 50, 25);
    }
}
```

Example of C Application

```
#include <stdio.h>

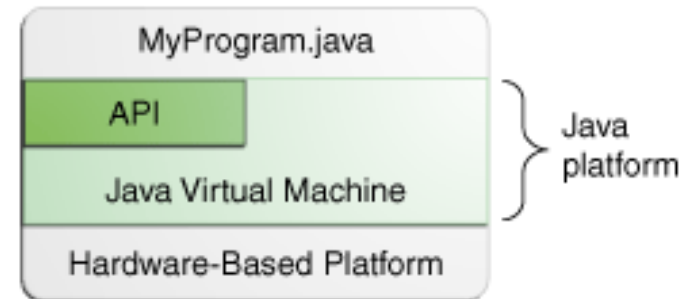
int main()
{
    printf ("Welcome to C!\n");
    return 0;
}
```

Example of Python Application

```
print "Welcome to Python!"
```

Java Technology

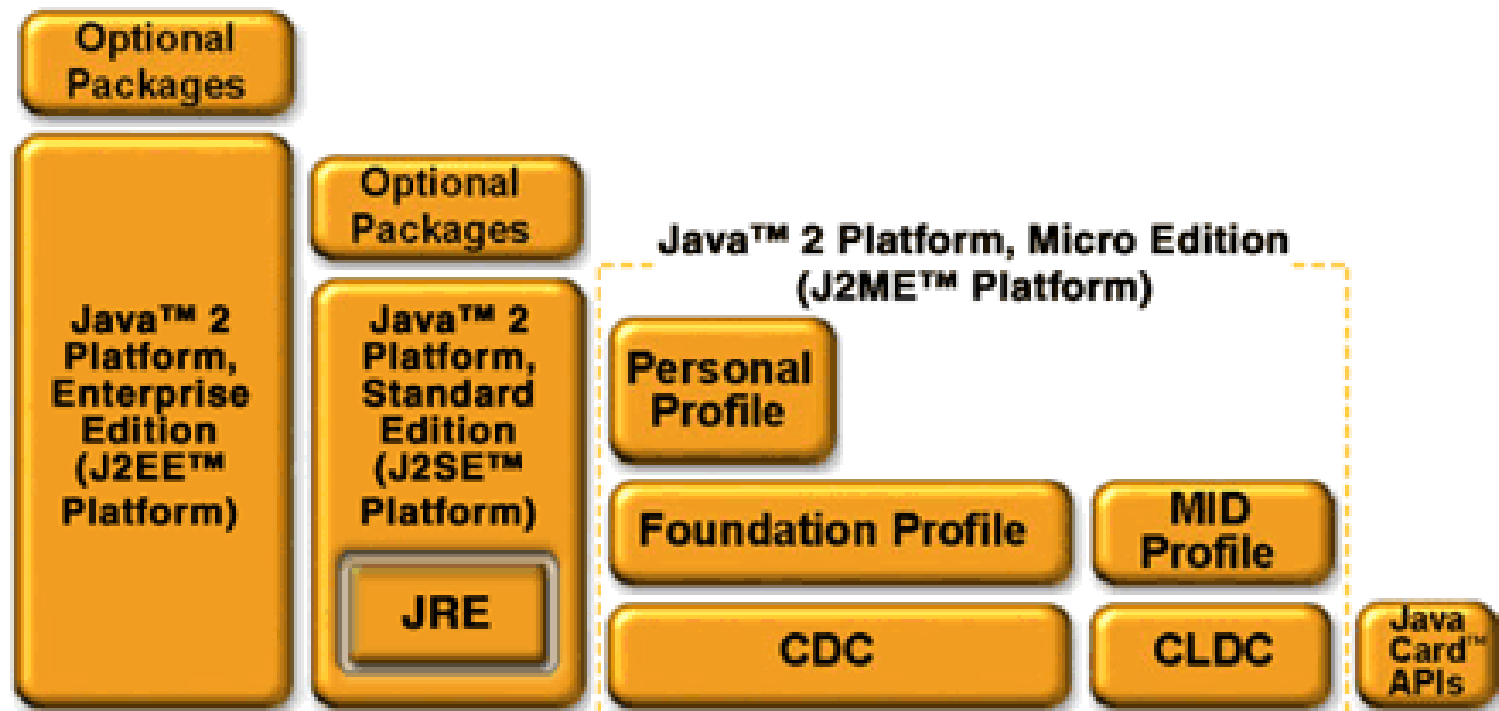
- Java Language
- Development Environment
 - Java Development Kit (JDK)
- Java Platform



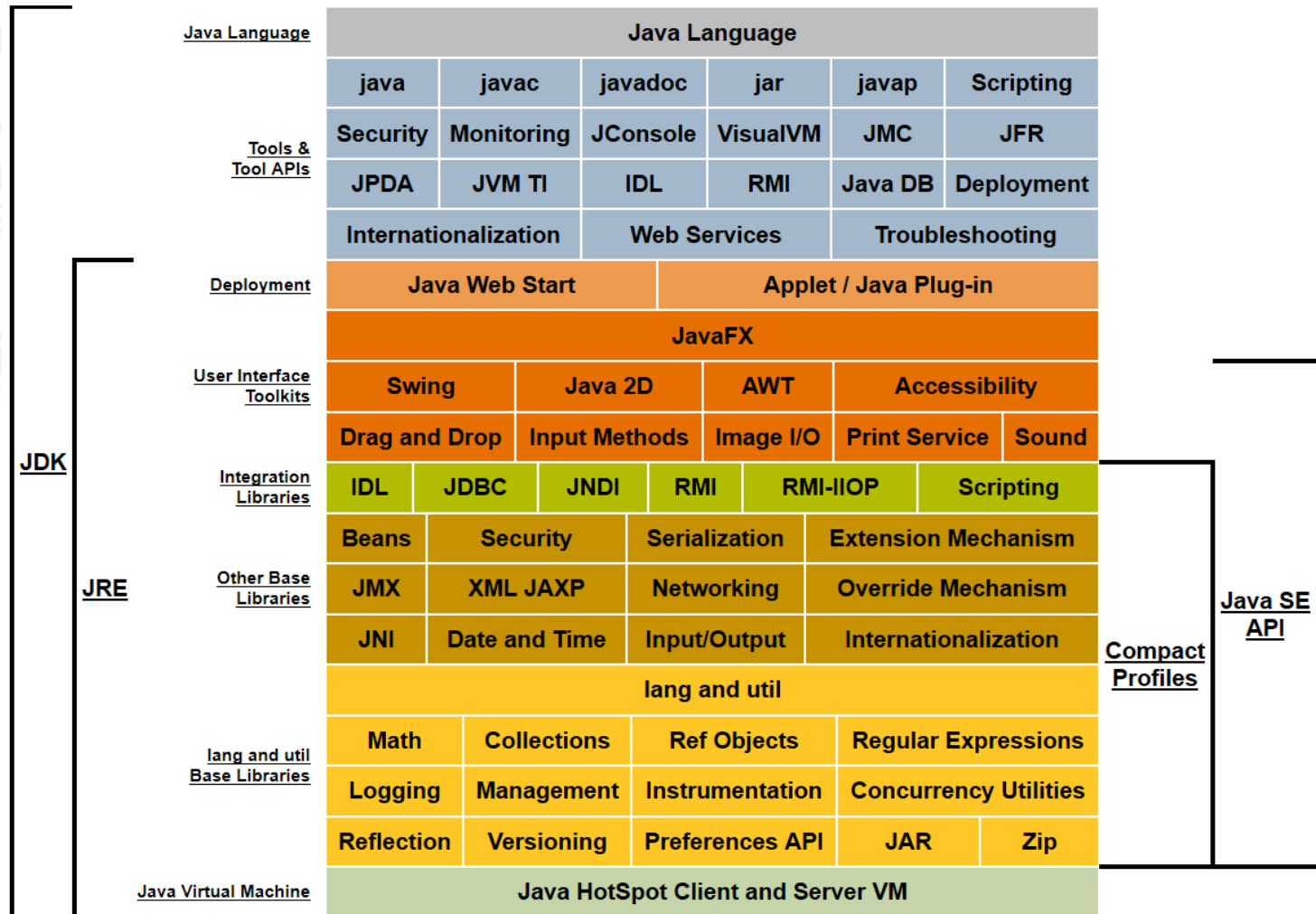
The API and the JVM make the program independent from the underlying hardware

- Java Virtual Machine
 - Execution environment
 - Security, portability and automatic garbage collection
- Java APIs
 - Java packages
 - Over 4000 classes

Java Editions



Java Conceptual Diagram



Java Virtual Machine

- Provides hardware platform specifications
- Reads compiled byte codes that are platform-independent
- Is implemented as software or hardware
- Is implemented in a Java technology development tool or a web browser

Using Java API Documentation

- A set of HTML files
- One package contains links to information on all of the classes
- A class documentation includes the class hierarchy, a description of the class, a list of member variables, a list of constructors and so on

The screenshot shows the Java Platform Standard Edition 7 API Specification web page. The page has a navigation bar at the top with links: Overview, Package, Class, Use, Tree, Deprecated, Index, and Help. Below the navigation bar, there is a sidebar on the left with a search box and a list of packages and classes. The main content area displays the 'Overview' page for the Java Platform, Standard Edition 7 API Specification. It includes a description of the document and a table of packages.

Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.

Java language & OO Style

- Java is “just another Object Oriented language”.
- Its syntax is similar to C++
- Java supports only the OO style
- Libraries defining elaborate hierarchies of classes are an intrinsic part of these languages
- Programs have a “**main()**” that creates the principal object (this **main()** may be hidden, provided by the compiler system), all subsequent control flow involve object interactions.

Java History

- Java comes from Sun research Labs
- Started around 1990/1991, to develop a software for consumer durables
 - Limited memory
 - Limited CPU power
 - Many difference architectures
 - Relatively low computational demands
 - Typically waiting for user input or external events
 - Limited amount of processing for each input/event
 - Nothing too time critical
- Something simpler than C++
- Support for the programmers
 - Garbage collection
- Something easier to implement – Java Virtual Machine
 - Generate code for a single idealized computer
 - Simulate that ideal computer on all real machines
- Web and Client side computing

What can Java Technology Do

- **Development Tools**
 - `javac` compiler, the `java` launcher, and the `javadoc` documentation tool
- **Application Programming Interface (API)**
 - From basic objects, to networking and security, to XML generation and database access
- **Deployment Technologies**
 - Web Start and Plug-in in JDK
- **User Interface Toolkits**
 - AWT, Swing and 2D toolkits
- **Integration Libraries**
 - Database access and manipulation of remote objects
 - IDL, JDBC, JNDI, RMI, RMI-IIOP

How will Java Tehcnology Help

- Get started quickly
 - Easy to learn, especially for programmers familiar with C/C++
- Write less code
 - 4 times less than C++
- Write better code
 - Good coding practices, automatic garbage collection, object orientation, JavaBeans™, extendible API (tested code and fewer bugs)
- Develop programs more quickly
 - 2 as fast (simpler than C++), fewer lines of code
- Avoid platform dependencies with 100% Pure Java
 - Portable programs
- ***Write once, run anywhere***
 - On any Java platform
- Distribute software more easily
 - Web Start (automatic version check and update)

Learning Java

- Java Language
 - Yes, you have to learn a new language; but if you know C++ there isn't really much new to learn (just which parts of C++ to leave out)
- Java Libraries (API)
 - “learning Java” will, to a larger extent, involve **learning about the libraries**, and design ideas for using libraries, etc

OO Design With Libraries

- You design your programs in the context of the class libraries
 - *“We need to communicate with the user, we will use*
 - *a scrolling (output only) text pane*
 - *a set of standard action buttons for processing options ...*
 - *a “pop-up” menu to deal with choices for ...*
 - *...*
 - *“we also need to contact an Oracle/Access database, we will use an instance of class ...*
- When making the initial breakdown of a problem, you think in terms of the reusable classes from the library

Java Programming Map

The Java Programming Language Basics

Getting Started (Java programming environment)	Object-Oriented Concept (class and methods)	Java Basics (identifiers, keywords, types)
Expression and Flow control		Arrays

More Object-Oriented Programming

Inheritance	Advanced Class Features (interface)
--------------------	---

Building Applications

Exceptions	Text-based Applications
-------------------	--------------------------------

Developing Graphic User Interface

AWT and Swing	GUI Event Handling	Java Applet	GUI-based Applications
----------------------	---------------------------	--------------------	-------------------------------

Advanced Java Programming

I/O Streams	Threads	Networking	etc. etc...
RMI	JDBC	Java Security	JavaBeans, Java2D/3D, Jini Servlets, JAI, JMF/QTJ, JSP

Object-Oriented language

Toolkit / Frameworks Object APIs

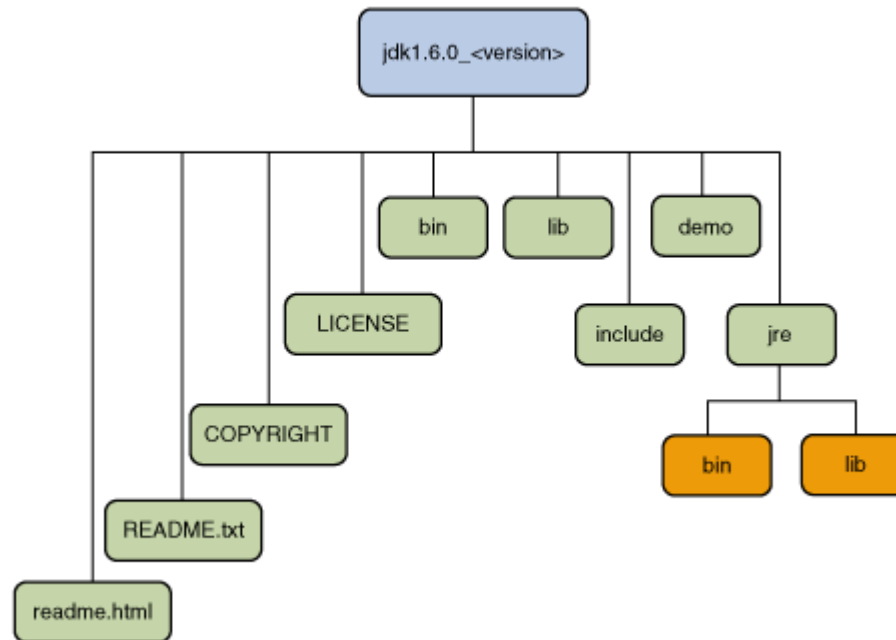
Practical Work – Prepare your study

- Install Java (JDK) to your own PC?
 - If possible, DO IT! (make sure to check the installation docs)
- Java Development Kits (JDKs) for Linux, OSX & Windows:
 - Free download from Oracle Technology Network
 - Java Runtime Environment (JRE)
 - Java Compiler
 - Command line tools - essential skills
 - API classes
- Remember to take a copy of JDK documentation

For instance:

- JRE: 40 MB ([Java SE JRE 7u25 for Windows x86](#))
- JDK: 89.09 MB ([Java SE JDK 7u25 for Windows x86](#))
- API docs: 60.2 MB ([Java SE 7u25](#))

JDK Directory Tree



Java Optional Packages

- Java Advanced Imaging
- Java (2D and) 3D
- Java Media Framework
- QuickTime for Java
- etc.

No optional packages are required in this subject