SDET UNIVERSITY

Object-Oriented Programming

- Developed by James Arthur Gosling and the “Green Team”
- 1991: Developed by Sun Microsystems
- 1995: First version of Java was released
- Netscape Navigator was the first Java-enabled web browser
- 2009: Acquired by Oracle

VERSIONS

- JDK 1.0 (1995)
- JDK 1.1 (1997)
- J2SE 1.2 (1998) - Playground
- J2SE 1.3 (2000) - Kestrel
- J2SE 1.4 (2002) - Merlin
- J2SE 5.0 (2004) - Tiger
- Java SE 6 (2006) - Mustang
- Java SE 7 (2011) - Dolphin
- Java SE 8 (2014) – Spider
- Java SE 9 (2017) – coming soon

PROGRAMMING APPROACHES

Structured / Procedural

- Based on functions
- C, C++, COBOL, Pascal

Some disadvantages: No constructs for encapsulation, chances of code repetition, No strong data hiding concept, difficult to debug

Object-Oriented

- Java, C#

OBJECT-ORIENTED PROGRAMMING

Basic Concept:

Write classes from which objects can be created.
Objects contain properties (state and behavior).

CORE PRINCIPLES

1. **Abstraction** – the ability for the client to access only the specific, desired properties
2. **Encapsulation** – binding methods and properties together while hiding complex, inner-workings
3. **Inheritance** – classes are organized in a hierarchy that passes properties down to child classes
4. **Polymorphism** – ability for methods and objects to take on various forms for sake of reusability and convenience

Classes: Object Template

A class is a construct that enables creation of objects

- Also sometimes called **blueprint** or **template** or **prototype** from which objects are created
- It defines members with **variables** and **methods**

Objects: Attributes & Operations

The object's **state** is determined by the value of its properties or attributes.

- Properties or attributes → member variables or data members

The object's **behavior** is determined by the operations that it provides.

- Operations → member functions or methods

Properties

AKA

Attributes

State

Adjectives

Methods

AKA

Operations

Behavior

Verbs

Example: A Light Bulb in Java Terms

CLASS	A manufacturing factory produces many bulbs based on a description or pattern of what a bulb is.
OBJECT	A real-world instance exists.
METHOD	Can be switched on or off
VARIABLES	Has features like glass covering, filament, brightness, and holder.

ABSTRACTION

"Abstraction denotes **essential characteristics** of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, **relative to the perspective of the viewer**."

GRADY BOOCH

ABSTRACTION

Abstraction is the process of taking only a set of essential characteristics from something.

Example: For a Doctor, You are a Patient

What does the Doctor Need to Know about You?
Name, Age, Medical Records

Example: For a Teacher, You are a Student

What does the Teacher Need to Know about You?
Name, Grade, Enrollment

ENCAPSULATION

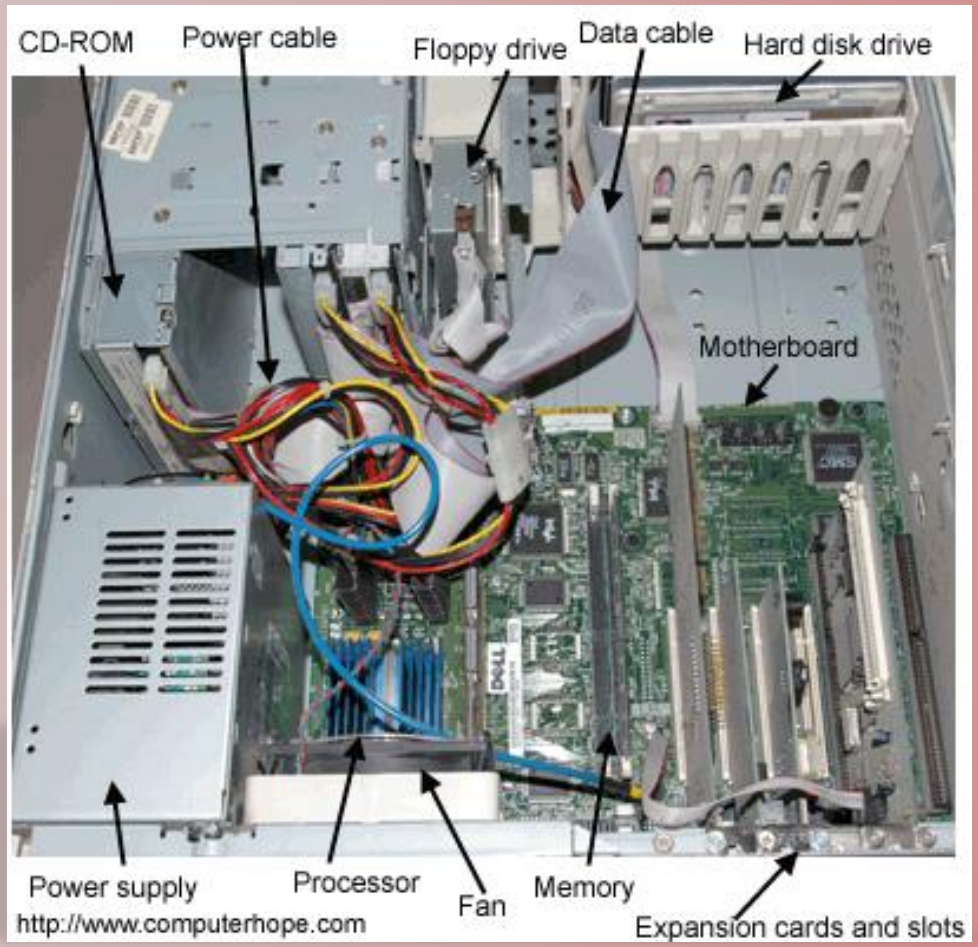
"Encapsulation is the process of **compartmentalizing** the elements of abstraction that constitute its structure and behavior; encapsulation **serves to separate the contractual interface of an abstraction and its implementation.**"

GRADY BOOCH

ENCAPSULATION

Would you like it if your CPU is given to you like this?

Encapsulation hides the complex inner works from the client, by data with operations



Data and Implementation Hiding in Java Classes

Java Supports Four Access Modifiers:

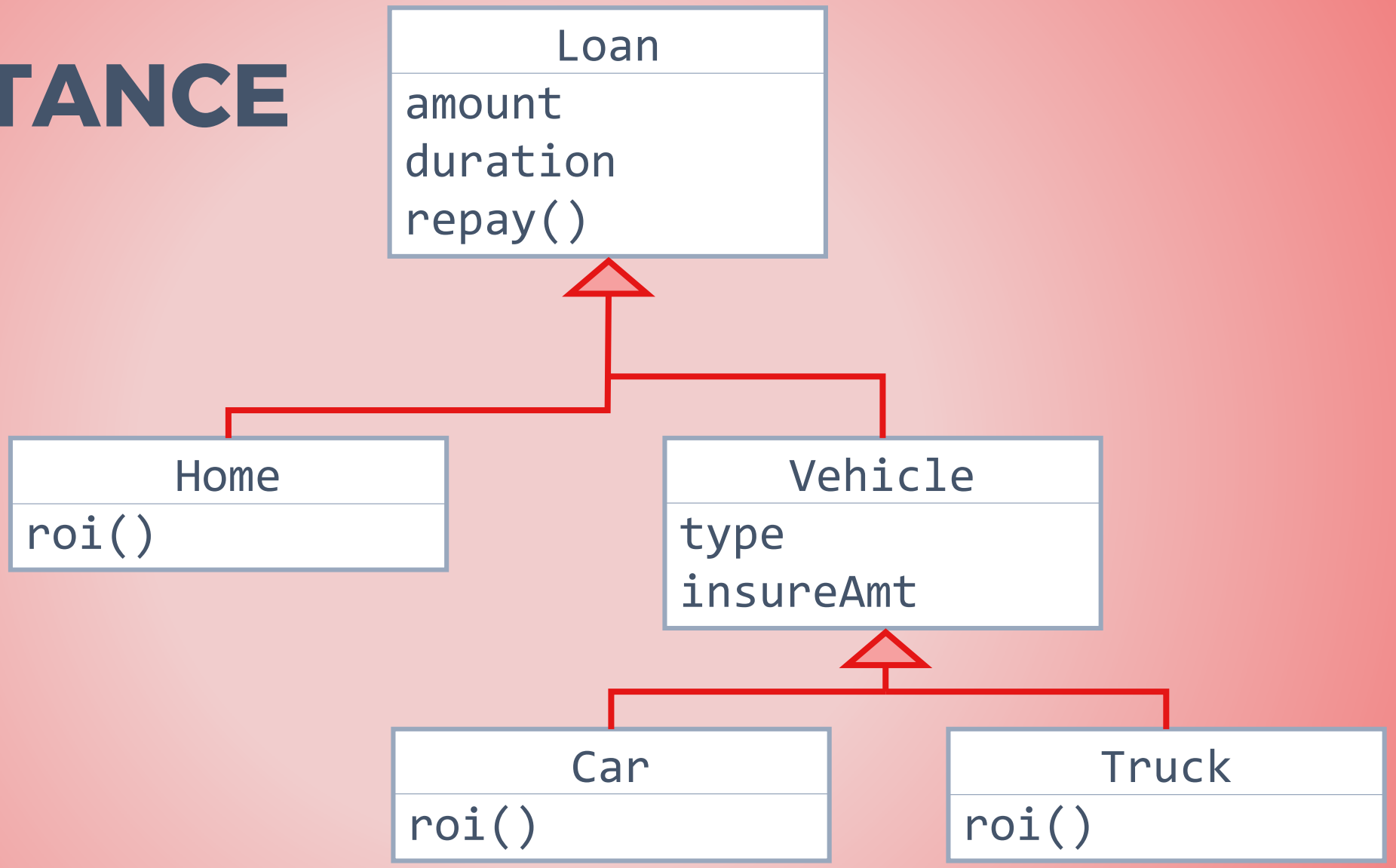
1. Public
2. Private
3. Default
4. Protected

INHERITANCE

“Inheritance defines relationship among classes, wherein one class share structure or behavior defined in one or more classes.”

GRADY BOOCH

INHERITANCE



Portable & Platform Independent

Before we understand portability and platform independence, we need to understand a few concepts.

- Java Code can be compiled anywhere
- Bytecode can be executed anywhere

"Write Once / Run Anywhere"

Features

Portable

- Java source code can be compiled in any Java-aware system (with JDK).
- When Java code executes, its behavior is exactly same in any Java-aware system.
- There are no platform-specific code in Java programs that causes compilation problems in any other OS.

Java programs are portable, which implies that they behave the same way when executed in any system and produce the same result.

Features

Platform Independent

- A Java program requires JVM (part of JRE) to execute Java code. When java application starts to executes, that Java Virtual Machine also starts.
- Bytecode has instructions that Java Virtual Machine can understand and execute.
- JVM converts the Bytecode to machine specific code.
- Java Bytecode can be copied on to any machine that has JVM and executed. This is what makes Java Platform Independent.
- “Write Once, Run Anywhere”

Is JVM Platform Independent?

No, since JVM needs to convert the byte code to machine specific code, it is different for each machine or OS, since each OS has its own native language.

That is the reason why JDK/JRE is available for different platforms.

Java SE Development Kit 8u121

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Solaris x64	96.9 MB	jdk-8u121-solaris-x64.tar.gz
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

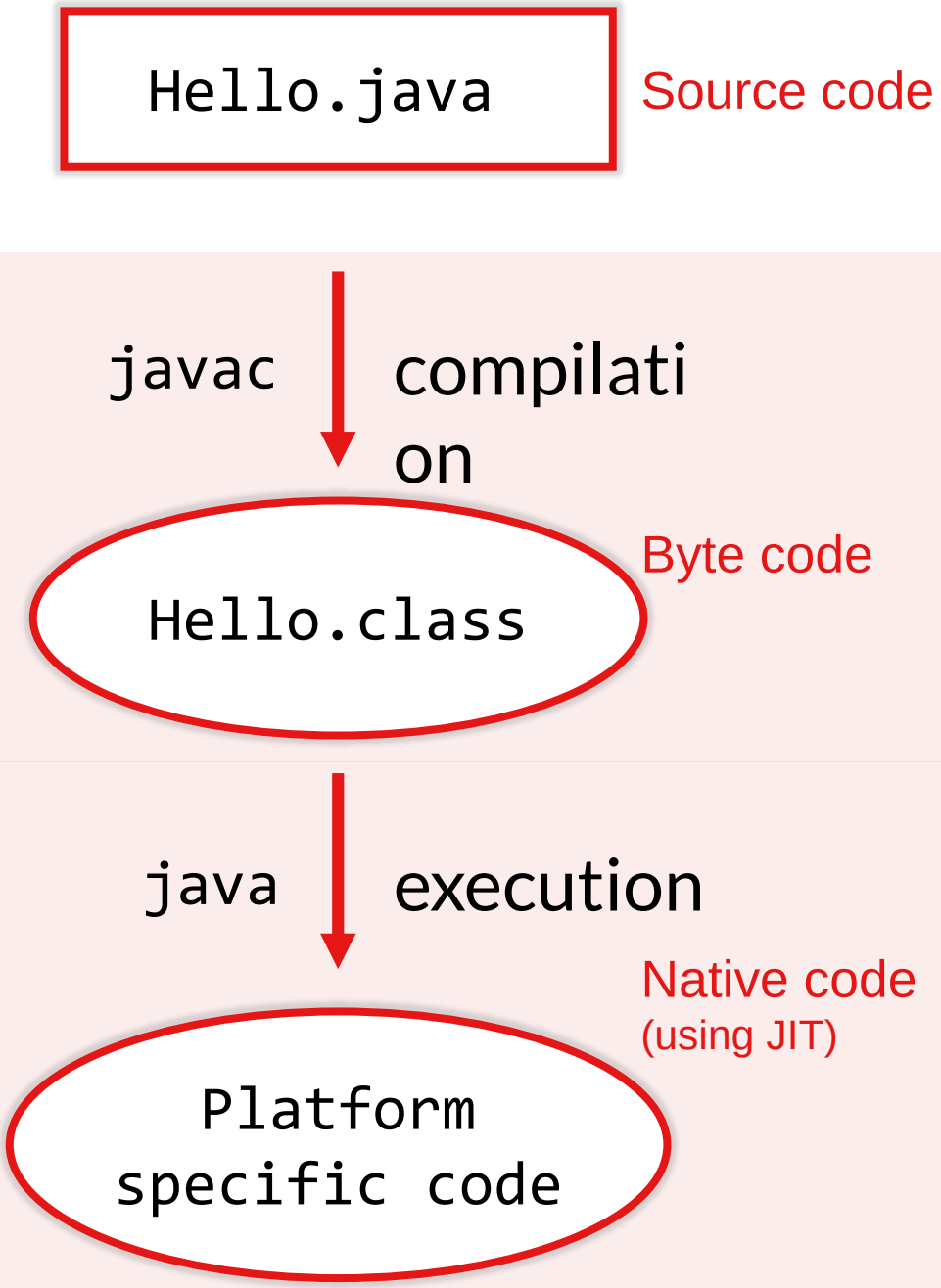
Compilation & Execution
from command prompt

Compile:

```
C:\..\Projects>javac Hello.java
```

Execute:

```
C:\..\Projects>java Hello
```



JIT – Just in Time Compiler

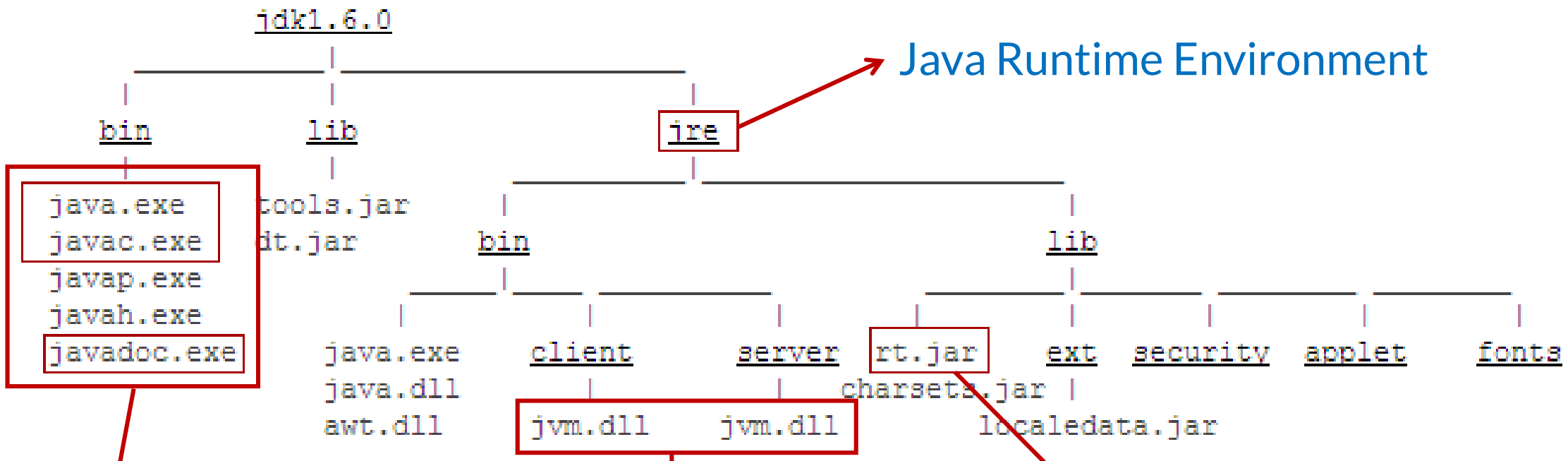
- Java Bytecodes were originally designed to be interpreted by JVM meaning bytecode are translated to machine code without it being stored anywhere.
- Since bytecode verifier (which is part of JVM) performs runtime checks, line by line execution was important.
- Since speed became an issue, Just-in-Time Compilation (JIT) came into being. JIT converts chunks of code, stores it temporarily in memory and then executes the converted code.
- JIT compilers are typically bundled with or are a part of a virtual machine and do the conversion to native code at runtime, on demand.
- The compiler also does automatic register allocation and some optimization when it produces the bytecodes.

Setup Java Developer Kit (JDK)

http://java.com/en/download/index.jsp or find appropriate link in http://www.oracle.com/technetwork/indexes/downloads

- Download Java based on the type of OS
  - Windows
  - Linux
  - Mac OS
  - Solaris
- Install JDK

JDK Installation Directory



Java Runtime Environment

Java Executables

Java Virtual Machine

Java Predefined Classes
(String, Date, Math etc)

IDE: Integrated Development Environment

Integrated development environment is an GUI interface that allows programmers to build and test their application.

- An editor where code can be written.
- Compiles as code is written. In Eclipse, red underline is used to indicate compilation errors and yellow underline is used to indicate warnings.
- Run and Debug features
- Tools to create language specific components
- Context sensitive help (In Eclipse, Ctrl+spacebar)
- Tools for auto-build (packaging a JEE application in eclipse)

Flavors of Java

JSE

- Java Standard Edition formerly known as J2SE.
- This forms the core part of Java language.

Focus of our course

JEE

- Java Enterprise Edition formerly known as J2EE.
- These are the set of packages that are used to develop distributed enterprise-scale applications.
- These applications are deployed on JEE application servers.

JME

- Java Micro Edition formerly known as J2ME.
- These are the set of packages is used to develop application for mobile devices and embedded systems.