Chapter 1: Introduction

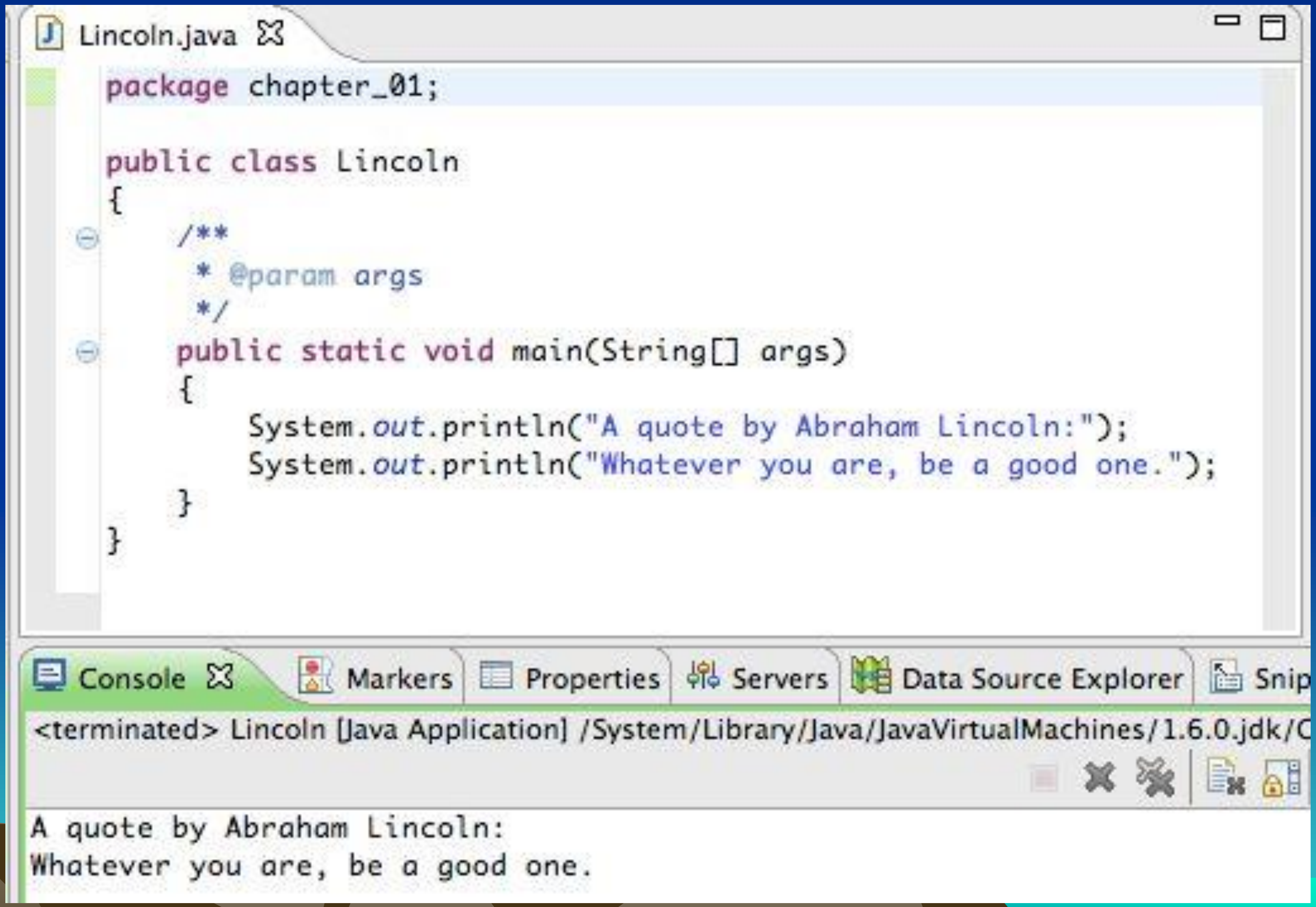
- The Java Programming Language
- Object-Oriented Programming
- Program Development

The Java Programming Language

- The Java programming language was created by Sun Microsystems, Inc.
- It was introduced in 1995 and it's popularity has grown quickly since
- Java has gone through several iterations.
- Some aspects of the original Java is now ***deprecated***.
- Java is an Object-Oriented language (More on this later)

Java Program Structure

- In the Java programming language:
  - A program is made up of one or more classes
  - A class contains one or more methods
  - A method contains program statements
- These terms will be explored in detail throughout the course
- A Java application always contains a method called main
- See Lincoln.java

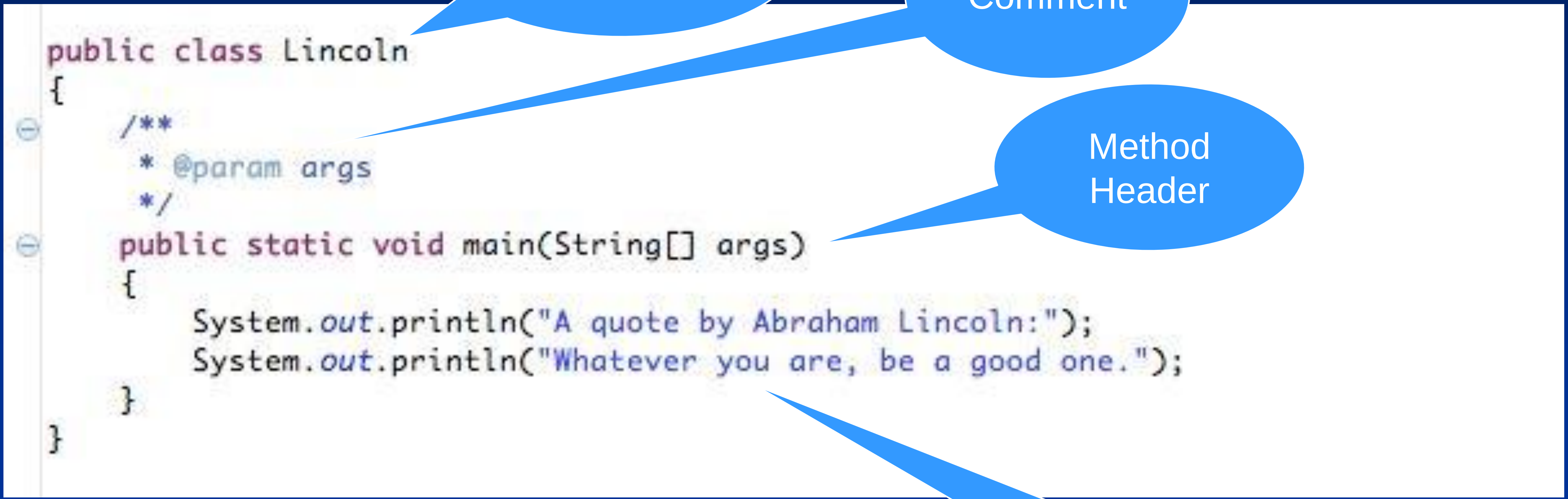


```
package chapter_01;

public class Lincoln
{
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        System.out.println("A quote by Abraham Lincoln:");
        System.out.println("Whatever you are, be a good one.");
    }
}
```

```
<terminated> Lincoln [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/C
A quote by Abraham Lincoln:
Whatever you are, be a good one.
```

Program Structure



```
public class Lincoln
{
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        System.out.println("A quote by Abraham Lincoln:");
        System.out.println("Whatever you are, be a good one.");
    }
}
```

Class Header

Comment

Method Header

Method body between {}

Comments Styles in Java:

// this comment runs to the end of the line

/*   this comment runs to the terminating
     symbol, even across line breaks */

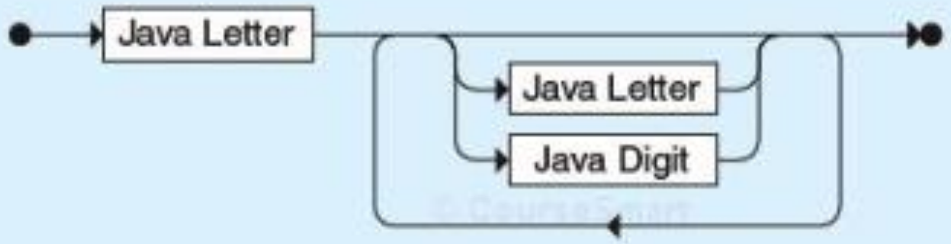
/** this is a *javadoc* comment   */

Identifiers

- Identifiers are the "words" in a program
  - A Java identifier can be made up of letters, digits, the underscore character (_), and the dollar sign
  - Identifiers cannot begin with a digit
- Java is case sensitive: Total, total, and TOTAL are different identifiers
- By convention, programmers use different case styles for different types of identifiers, such as
  - title case for class names - Lincoln
  - upper case for constants - MAXIMUM
- Reserved word is a special identifier that already have a predefined meaning in the language.
- A reserved word cannot be used in any other way

Identifiers and Reserved Words

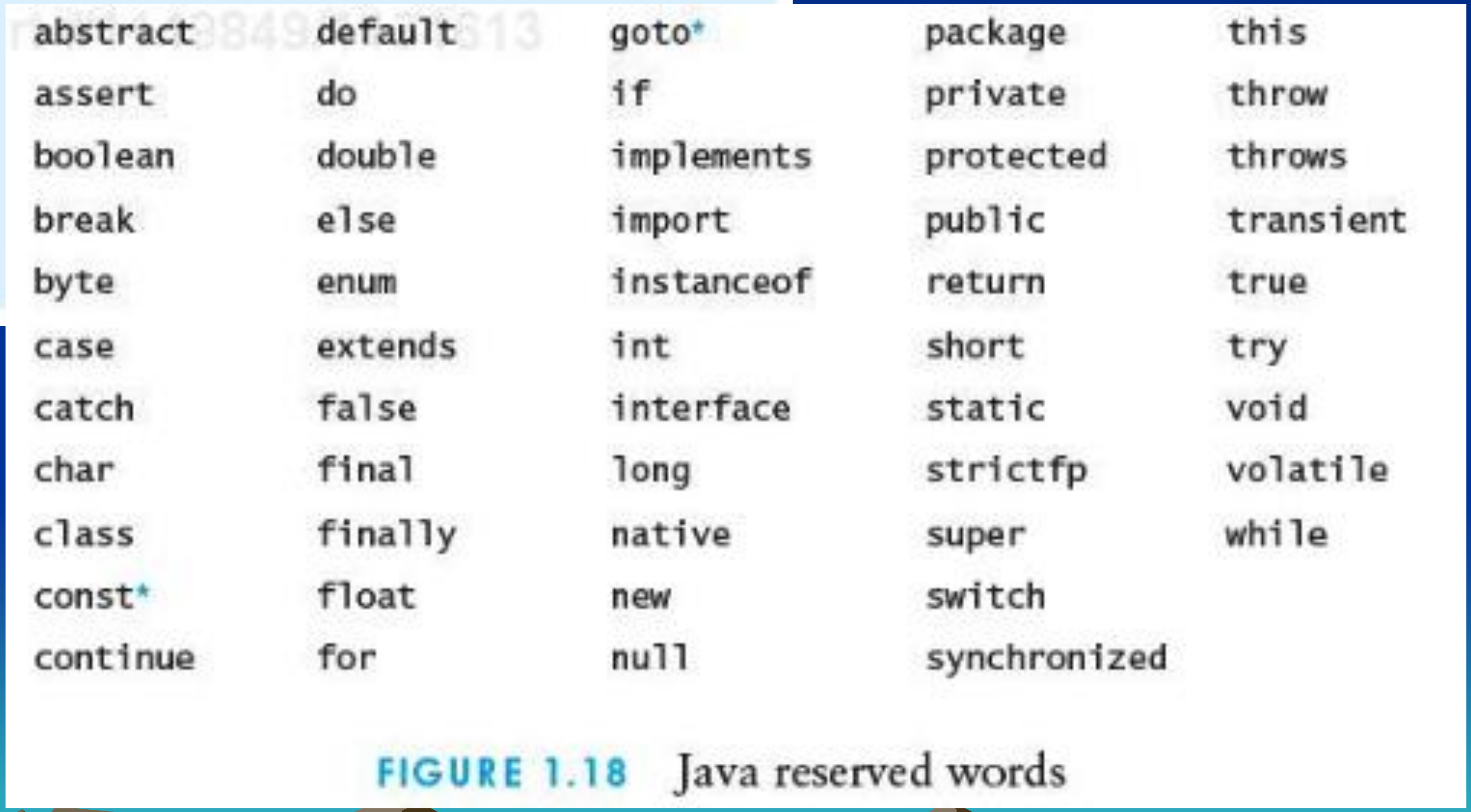
Identifier



An identifier is a letter followed by zero or more letters and digits. A Java Letter includes the 26 English alphabetic characters in both uppercase and lowercase, the $ and _ (underscore) characters, as well as alphabetic characters from other languages. A Java Digit includes the digits 0 through 9.

Examples:

```
total
MAX_HEIGHT
num1
Keyboard
System
```



abstract
assert
boolean
break
byte
case
catch
char
class
const*
continue
default
do
double
else
enum
extends
false
final
finally
float
for
goto*
if
implements
import
instanceof
int
interface
long
native
new
null
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
true
try
void
volatile
while

FIGURE 1.18 Java reserved words

Quick Check

`grade` Valid

`quizGrade` Valid

`NetworkConnection` Valid

`frame2` Valid

`3rdTestScore` Invalid – cannot begin with a digit

`MAXIMUM` Valid

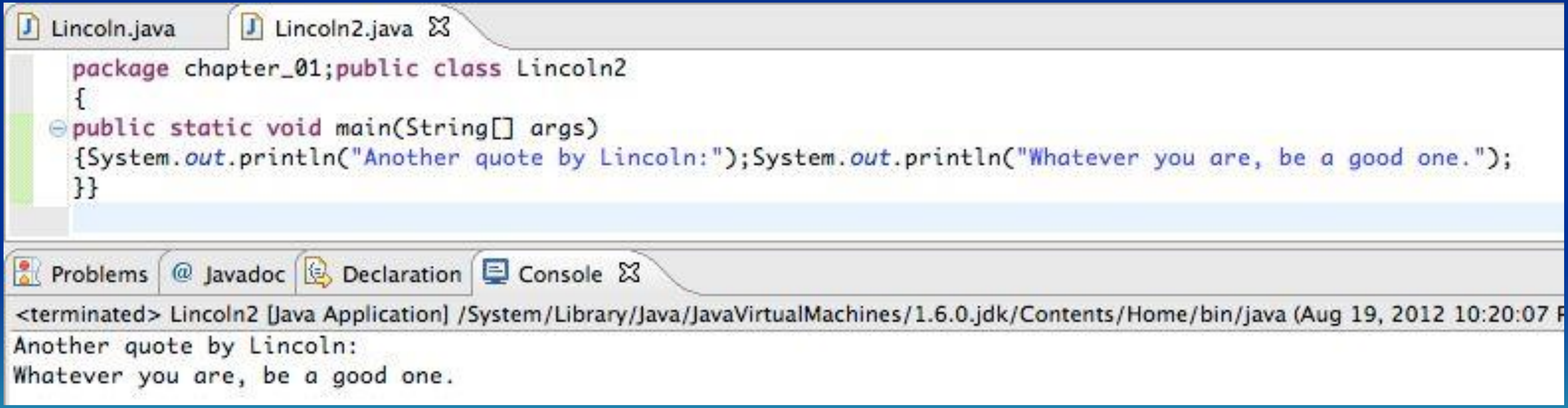
`MIN_CAPACITY` Valid

`student#` nvalid – cannot contain the '#' character

`Shelves1&2` Invalid – cannot contain the '&' character

White Space

- Spaces, blank lines, and tabs are called white space
- White space is used to separate words and symbols in a program
- Extra white space is ignored
- A valid Java program can be formatted in many ways
- Programs should be formatted to enhance readability, using consistent indentation



```
package chapter_01;public class Lincoln2
{
public static void main(String[] args)
{System.out.println("Another quote by Lincoln:");System.out.println("Whatever you are, be a good one.");
}}
```

```
<terminated> Lincoln2 [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Aug 19, 2012 10:20:07
Another quote by Lincoln:
Whatever you are, be a good one.
```

Program Development

- The mechanics of developing a program include several activities:
  - writing the program in a specific programming language (such as Java)
  - translating the program into a form that the computer can execute
  - investigating and fixing various types of errors that can occur
- Software tools can be used to help with all parts of this process

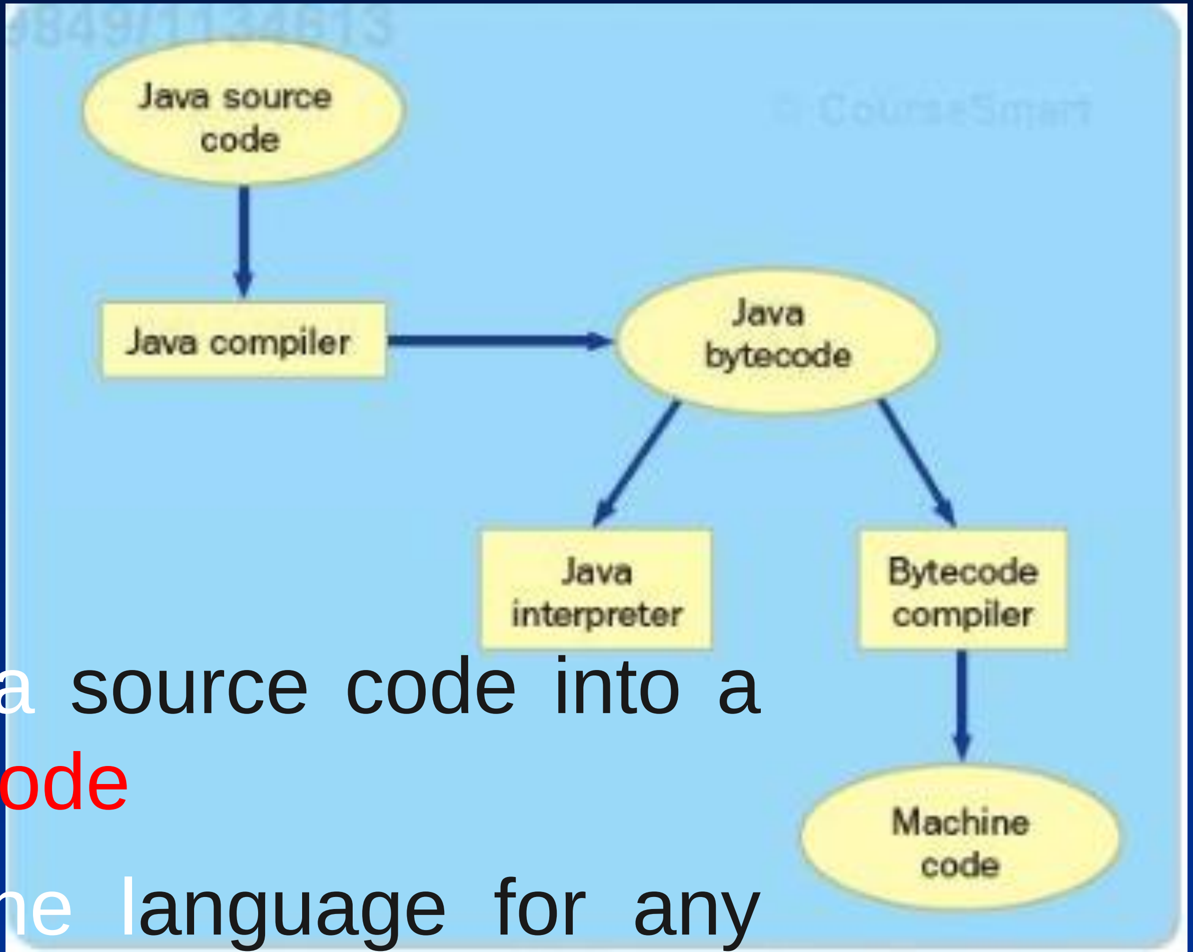
Language Levels

- There are four programming language levels:
  - machine language
  - assembly language
  - high-level language
  - fourth-generation language
- Each type of CPU has its own specific machine language
- The other levels were created to make it easier for a human being to read and write programs

Programming Languages

- A program must be translated into machine language before it can be executed
- A compiler is a software tool which translates source code into a specific target language
- Often, that target language is the machine language for a particular CPU type. Example: C
- The Java approach is somewhat different

Java Translation

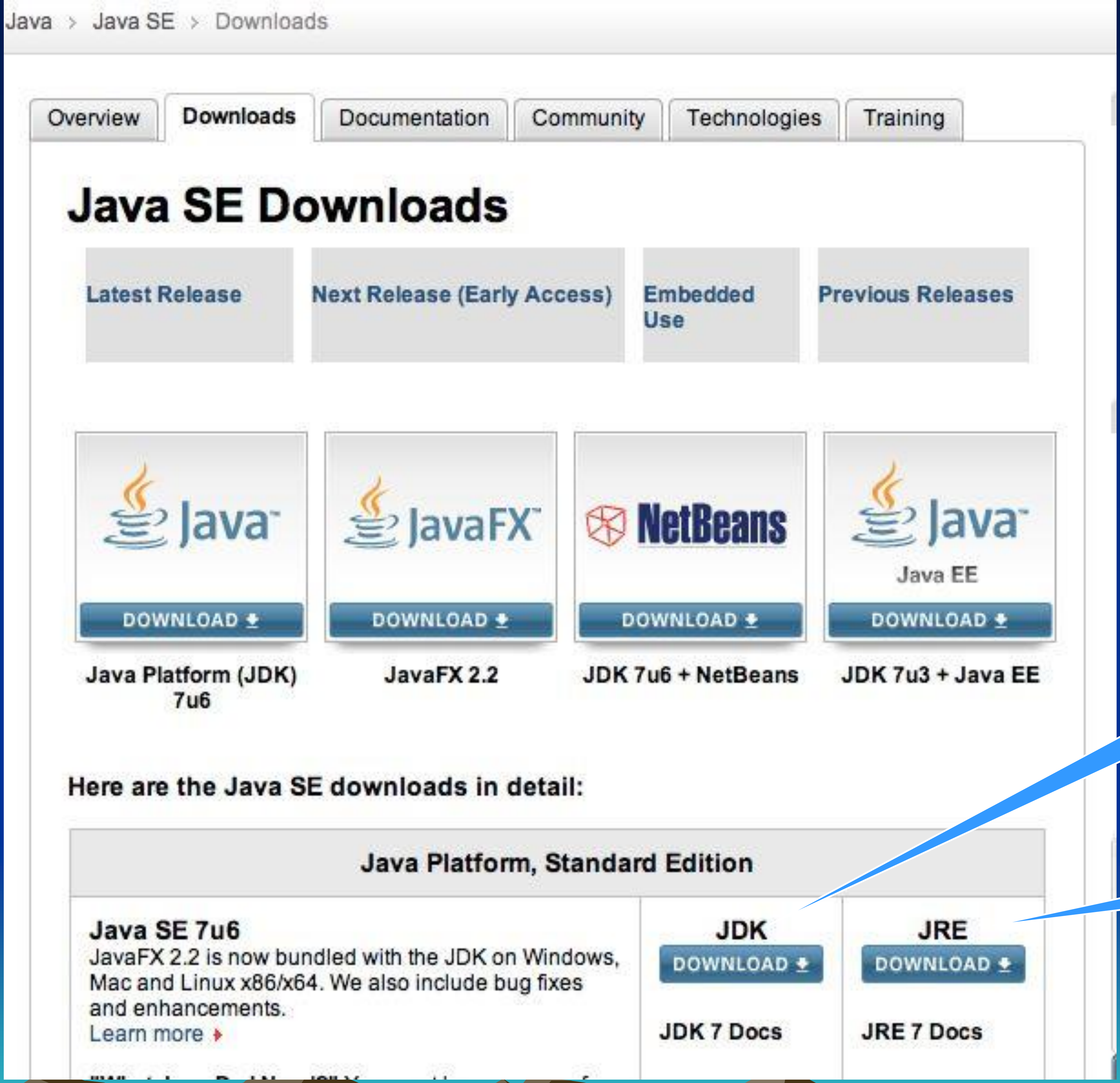


- The Java compiler translates Java source code into a special representation called bytecode
- Java bytecode is not the machine language for any traditional CPU
- Another software tool, called an interpreter, translates bytecode into machine language and executes it
- Therefore the Java compiler is not tied to any particular machine
- Java is considered to be architecture-neutral

Development Environments

- There are many programs that support the development of Java software, including:
  - Java Software Development Kit (JDK)
  - Eclipse
  - NetBeans
  - BlueJ
  - jGRASP
- The details of these environments differ.
- The basic compilation and execution process is essentially the same

http://www.oracle.com/technetwork/java/javase/downloads/index.html



Java SE Downloads

Latest Release | Next Release (Early Access) | Embedded Use | Previous Releases

Java Platform (JDK) 7u6

JavaFX 2.2

JDK 7u6 + NetBeans

JDK 7u3 + Java EE

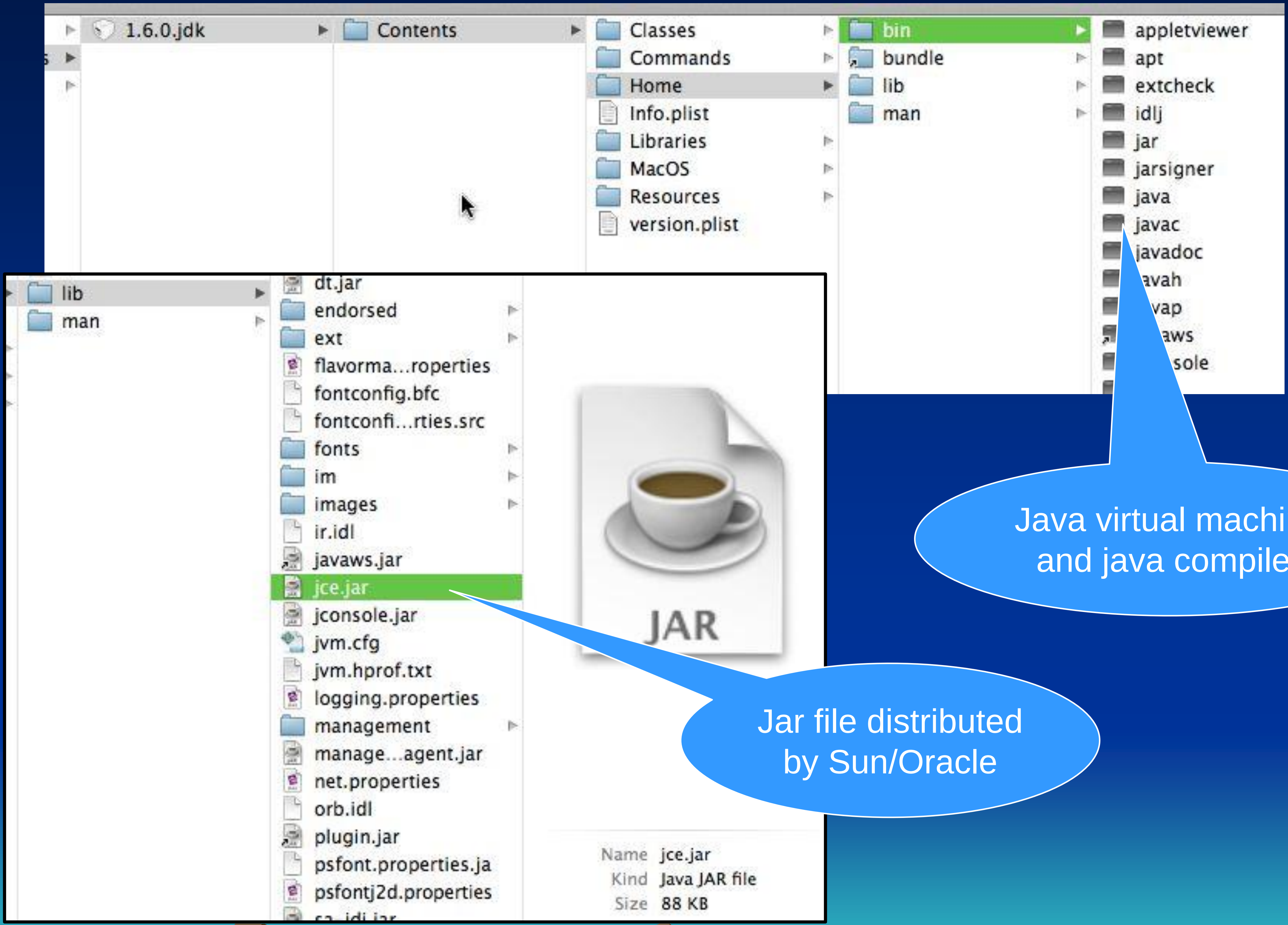
Here are the Java SE downloads in detail:

Java Platform, Standard Edition		
Java SE 7u6 JavaFX 2.2 is now bundled with the JDK on Windows, Mac and Linux x86/x64. We also include bug fixes and enhancements. Learn more ›	**JDK** DOWNLOAD JDK 7 Docs	**JRE** DOWNLOAD JRE 7 Docs

Java Development Kit

Java Runtime Environment

Java JDK and Jar Files



1.6.0.jdk

Contents

Classes
Commands
Home
Info.plist
Libraries
MacOS
Resources
version.plist

bin
bundle
lib
man

appletviewer
apt
extcheck
idlj
jar
jarsigner
java
javac
javadoc
avah
vap
aws
sole

lib
man

dt.jar
endorsed
ext
flavorma...roperties
fontconfig.bfc
fontconfi...rties.src
fonts
im
images
ir.idl
javaws.jar
jce.jar
jconsole.jar
jvm.cfg
jvm.hprof.txt
logging.properties
management
manage...agent.jar
net.properties
orb.idl
plugin.jar
psfont.properties.ja
psfontj2d.properties

JAR

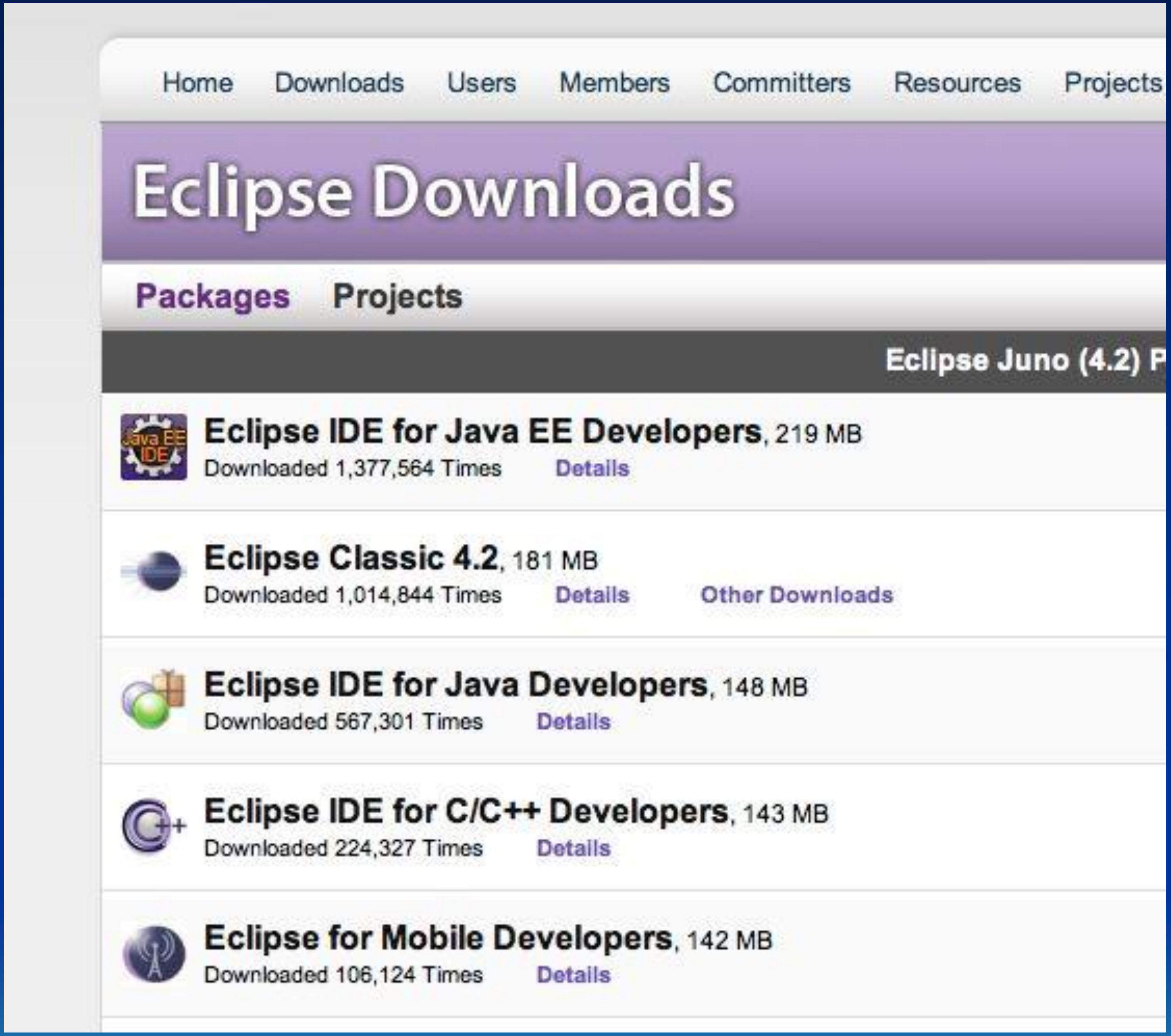
Name jce.jar
Kind Java JAR file
Size 88 KB

Java virtual machine and java compiler

Jar file distributed by Sun/Oracle

Eclipse Downlaods: http://www.eclipse.org/downloads/

- Download and expand the Eclipse subfolder then move it anywhere you desire.
- Run eclipse.exe or eclipse.app ... etc, based on your OS.



Eclipse IDE for Java EE Developers, 219 MB
Downloaded 1,377,564 Times Details

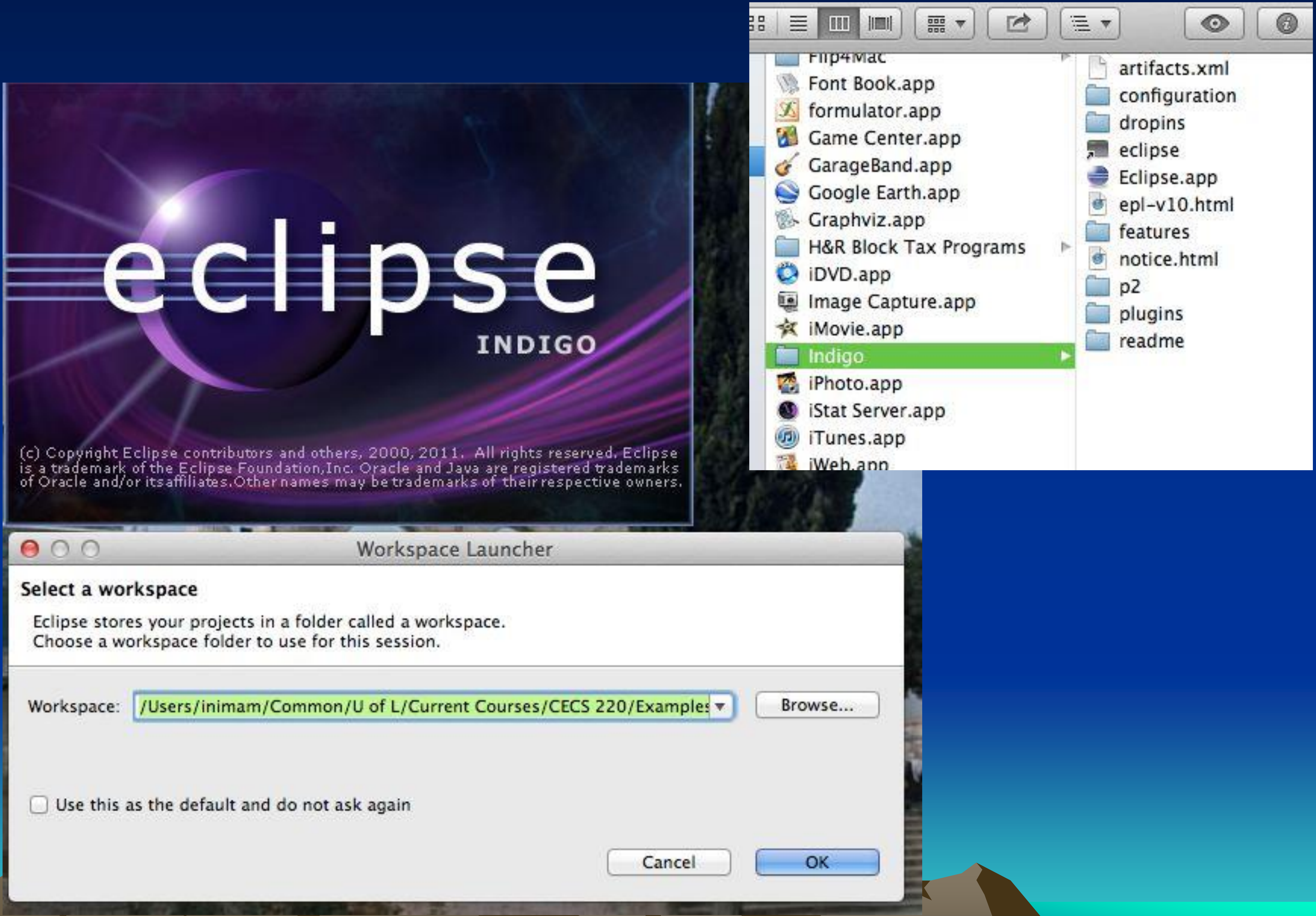
Eclipse Classic 4.2, 181 MB
Downloaded 1,014,844 Times Details Other Downloads

Eclipse IDE for Java Developers, 148 MB
Downloaded 567,301 Times Details

Eclipse IDE for C/C++ Developers, 143 MB
Downloaded 224,327 Times Details

Eclipse for Mobile Developers, 142 MB
Downloaded 106,124 Times Details

Running Eclipse



(c) Copyright Eclipse contributors and others, 2000, 2011. All rights reserved. Eclipse is a trademark of the Eclipse Foundation,Inc. Oracle and Java are registered trademarks of Oracle and/or its affiliates.Other names may be trademarks of their respective owners.

Workspace Launcher

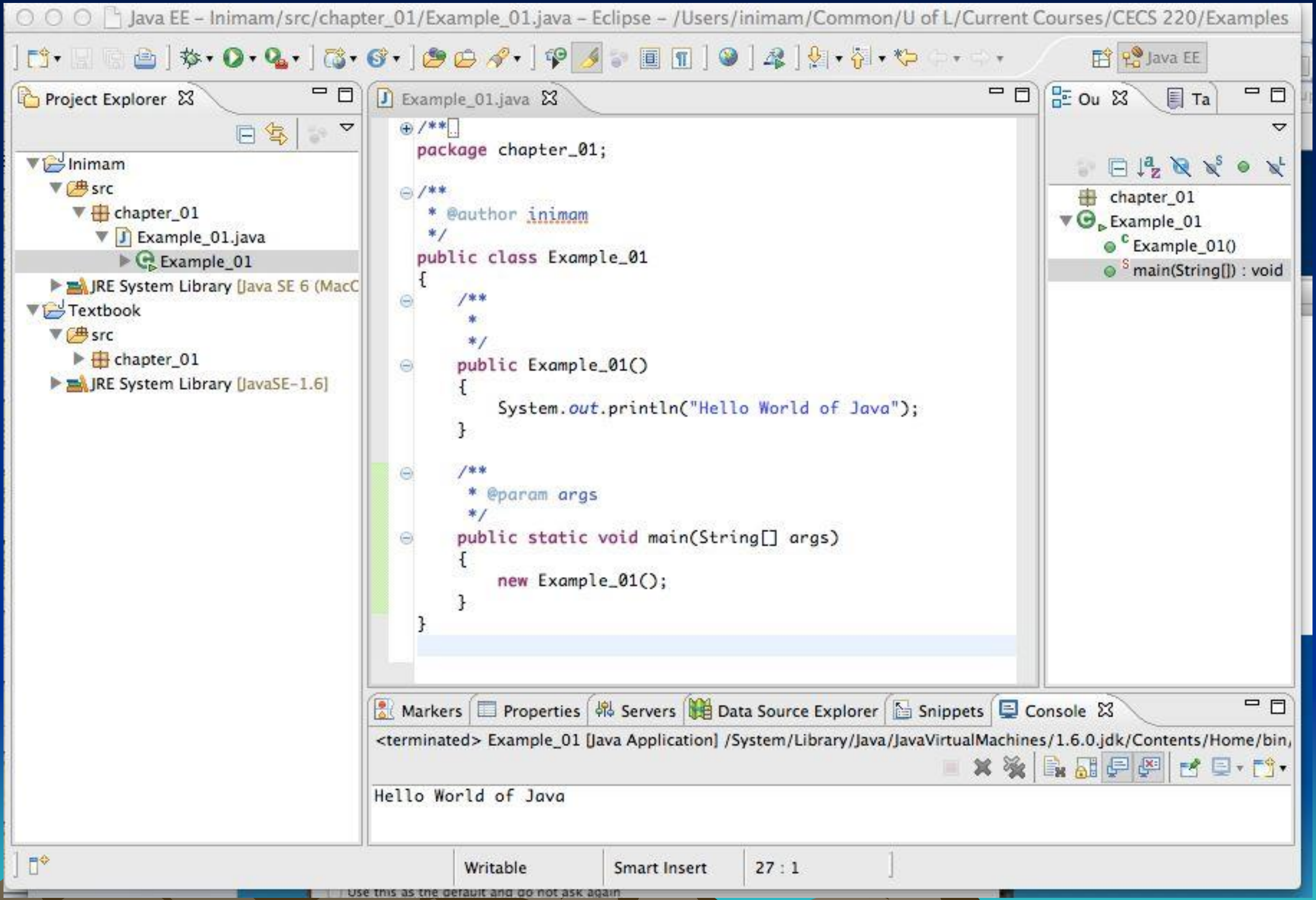
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/inimam/Common/U of L/Current Courses/CECS 220/Examples

□ Use this as the default and do not ask again

Eclipse Perspectives



Java EE – Inimam/src/chapter_01/Example_01.java – Eclipse – /Users/inimam/Common/U of L/Current Courses/CECS 220/Examples

```
/**
package chapter_01;

/**
 * @author inimam
 */
public class Example_01
{
    /**
     *
     */
    public Example_01()
    {
        System.out.println("Hello World of Java");
    }

    /**
     * @param args
     */
    public static void main(String[] args)
    {
        new Example_01();
    }
}
```

```
<terminated> Example_01 [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/
Hello World of Java
```

Syntax and Semantics

- The syntax rules of a language define how we can put together symbols, reserved words, and identifiers to make a valid program

- The semantics of a program statement define what that statement means (its purpose or role in a program)

- A program that is syntactically correct is not necessarily logically (semantically) correct

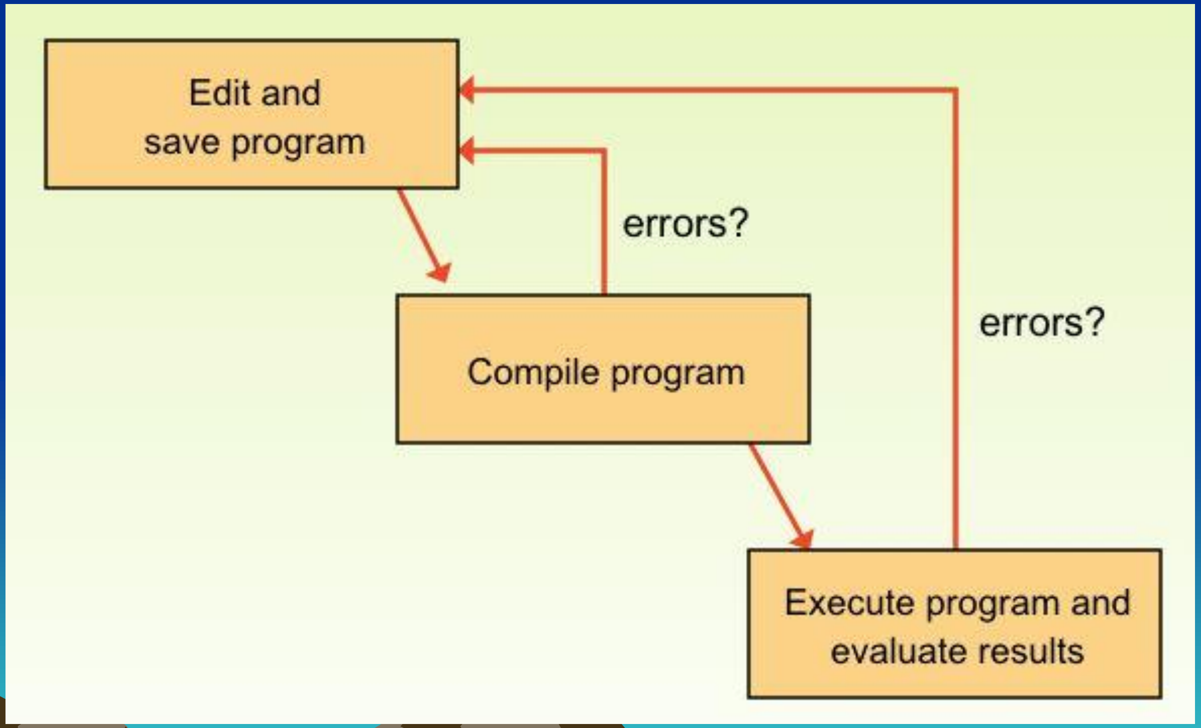
- A program will always do what we tell it to do, not what we meant to tell it to do

Errors

- A program can have three types of errors

1. Compile-time errors: The compiler will find syntax errors and other basic problems.
2. Run-time errors: A problem can occur during program execution which causes a program to terminate abnormally. Example: Division By Zero.
3. Logical errors: A program may run, but produce incorrect results. Example: Incorrect Formula.

Basic Program Development



Object-Oriented Programming

- Java is an object-oriented programming language
- As the term implies, an object is a fundamental entity in a Java program.
- Objects can be used effectively to represent real-world entities
  - For instance, an object might represent a particular employee in a company.
  - Each employee object handles the processing and data management related to that employee.

Objects

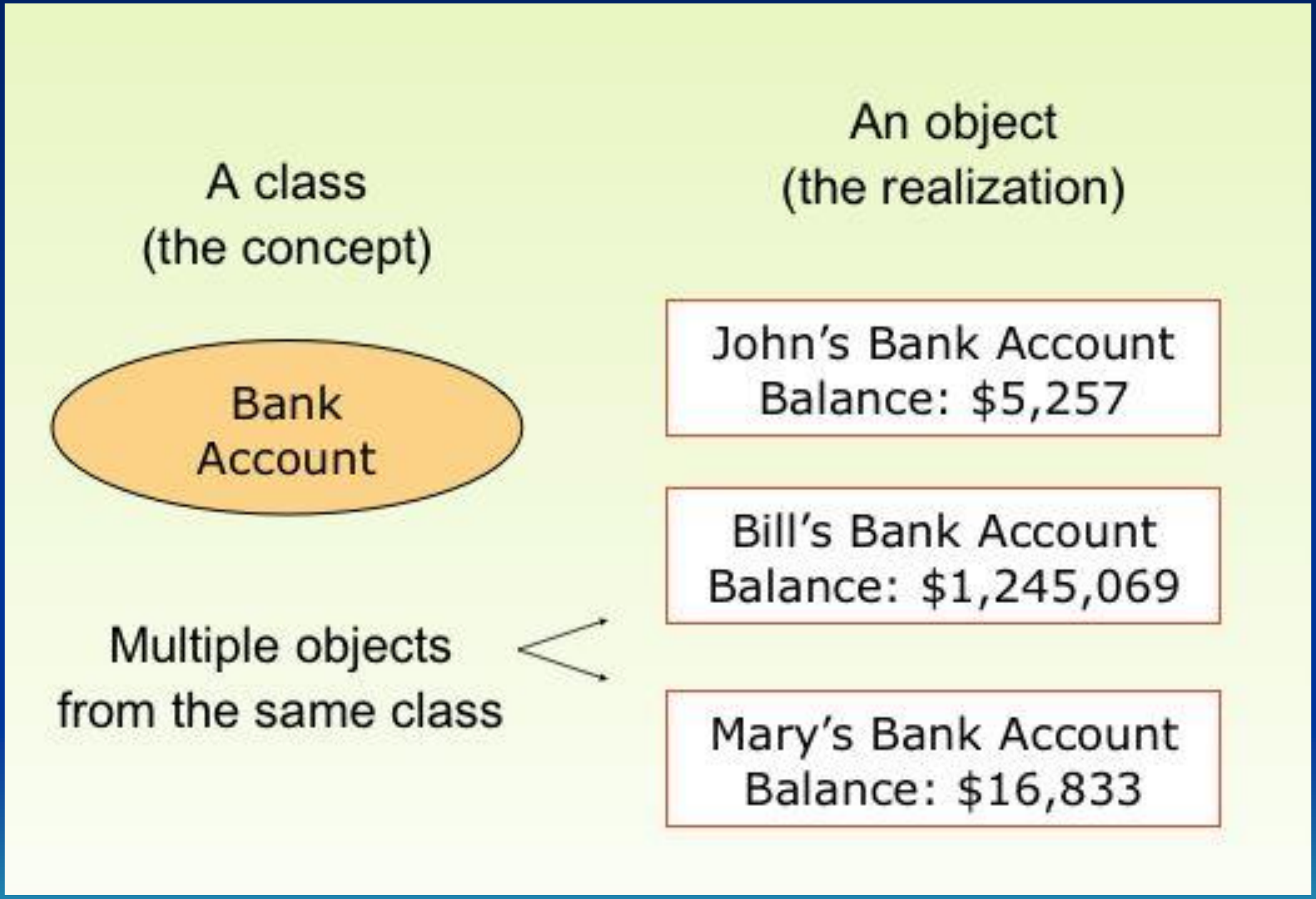
- An object has:
  - State - descriptive characteristics
  - Behaviors - what it can do (or what can be done to it)
- The state of a bank account includes its account number and its current balance
- The behaviors associated with a bank account include the ability to make deposits and withdrawals
- The behavior of an object might change its state

Classes

- An object is defined by a class
- A class is the blueprint of an object
- The class uses methods to define the behaviors of the object
- The class that contains the main method of a Java program represents the entire program
- A class represents a concept, and an object represents the embodiment of that concept
- Multiple objects can be created from the same class
- Class = Blueprint

Objects and Classes

- Class is a design concept while an object is a runtime entity



A class
(the concept)

Bank
Account

Multiple objects
from the same class

An object
(the realization)

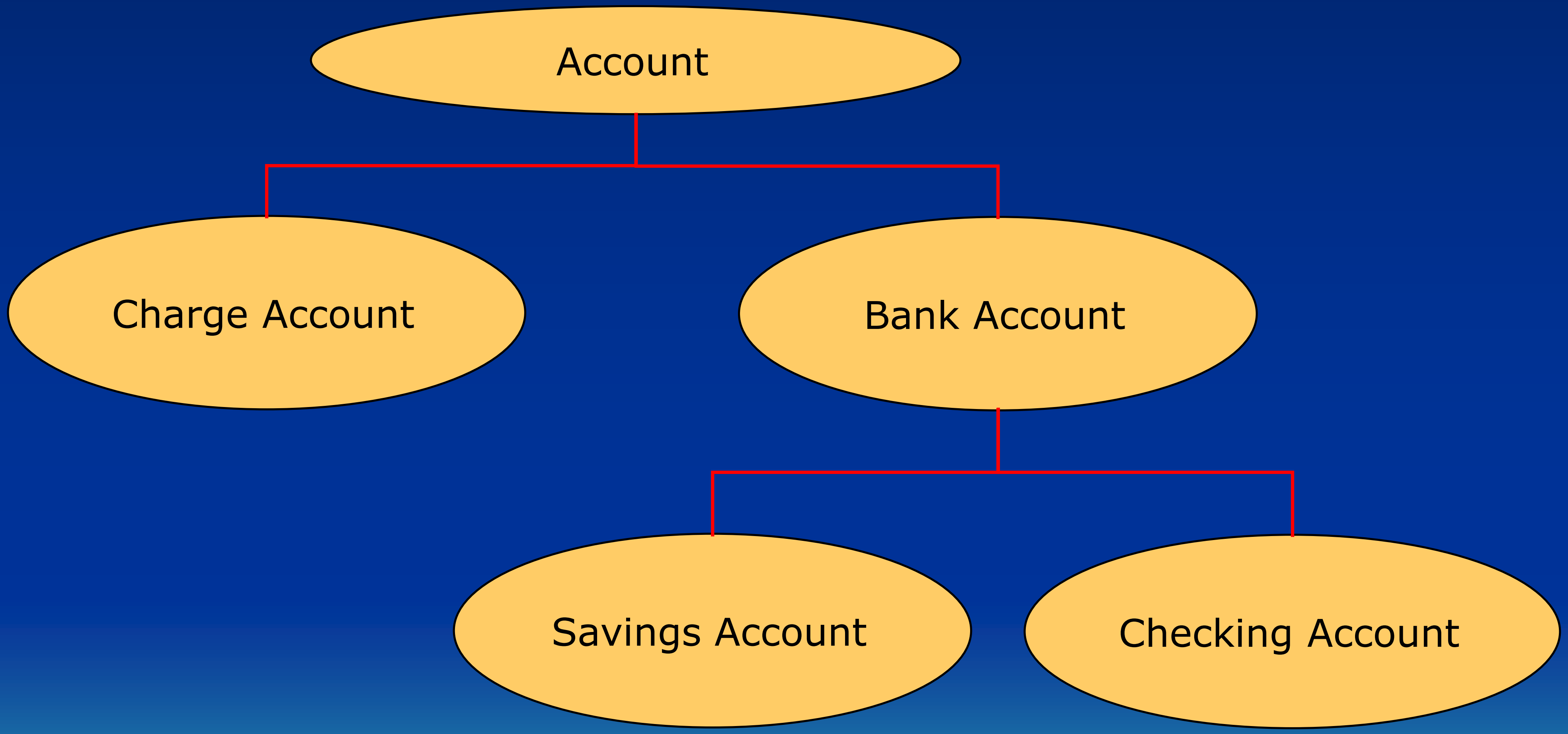
John's Bank Account
Balance: $5,257

Bill's Bank Account
Balance: $1,245,069

Mary's Bank Account
Balance: $16,833

Inheritance

- One class can be used to derive another via inheritance (More on this later)
- Classes can be organized into hierarchies


Account
Charge Account
Bank Account
Savings Account
Checking Account