Lesson 1:
Introduction to Java And the Eclipse Development Environment

Pulling the Arrow Back to Hit the Target

Wholeness of the Lesson

- Java is an object-oriented highly portable programming language that arose as an easy alternative to the once dominant, but error-prone, C++ language. Eclipse is one of many open source, powerful but easy-to-use integrated development environments for use with Java and related technologies. Transcendental meditation is a highly portable and powerful technique working from deeper levels of intelligence allows one to accomplish more with less effort.

Outline of Topics

- Introduction to Java
- History of Java
- Java Features
- Working with JDK
- Java First Program
- Working with IDE
- Viewing Java API Source Code

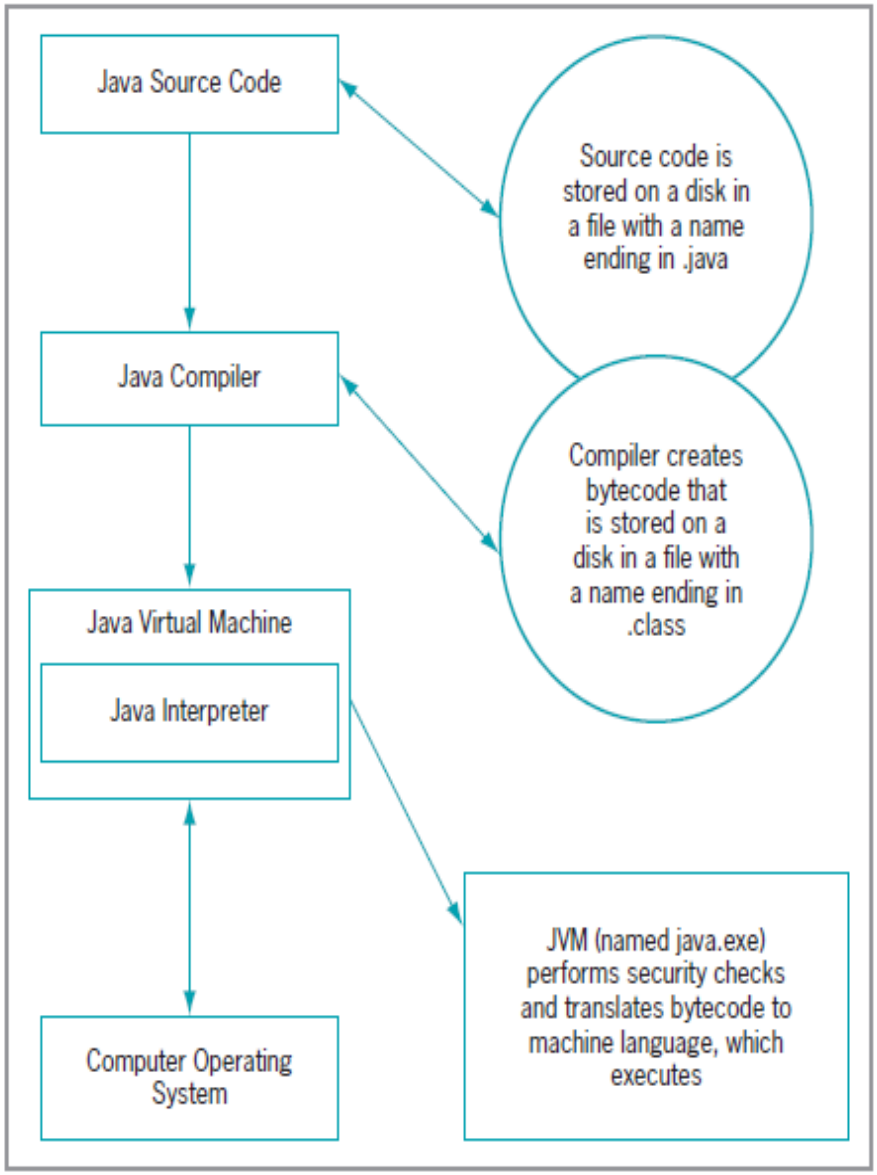
What is Java?

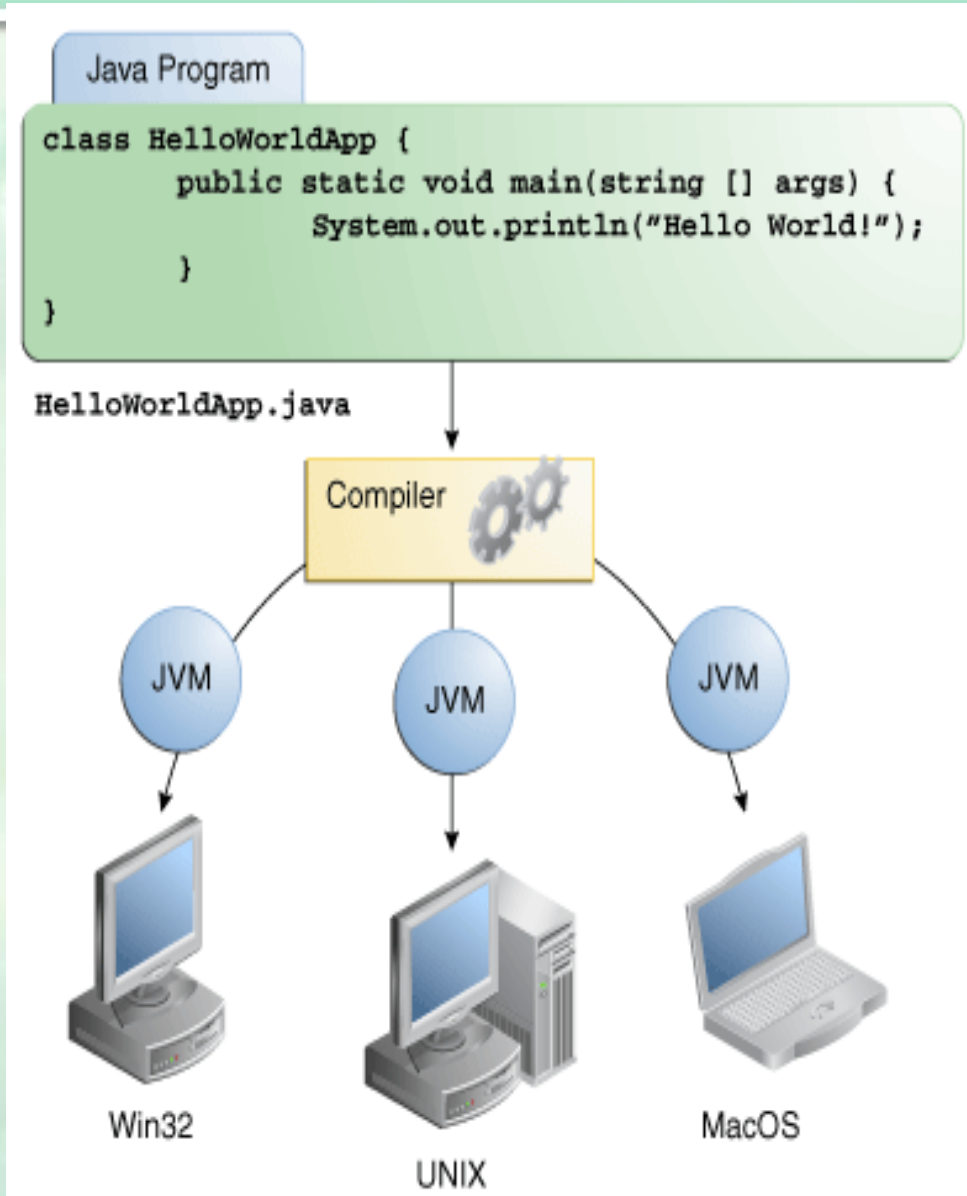
- A general-purpose object-oriented language
- Developed by Sun Microsystems
- Based on C/C++
- Designed for easy Web/Internet applications
- Widespread acceptance

Why Java Became Famous ?

Platform Independent(WORA – Write Once Run Anywhere)

Java Environment





Brief History

- The Java language began as a language for programming consumer electronic appliances.
- Its creator was James Gosling of Sun Microsystems.
- The language was first named as Oak.
- The language was developed privately starting in 1991, and was made publicly available in 1994.
- In 2009, Oracle bought the rights to Java from Sun Microsystems.

The Java Buzzwords

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Architecture-neutral
- Interpreted
- High performance
- Distributed
- Dynamic

Java Features (1)

- **Simple**
  - Java has a concise, cohesive set of features that makes it easy to learn and use.
  - Its syntax looks very similar to C/C++
  - no pointers
  - automatic garbage collection
  - rich pre-defined class library : https://docs.oracle.com/javase/8/docs/api/. It can be easily reused by application developers.
- **Object oriented**
  - Java implements the object-oriented programming paradigm by grouping data and operations into classes and/or objects.
  - Java embodies the modern, object-oriented programming philosophy. (Encapsulation, Polymorphism, Inheritance)
  - almost all data types are objects (files, strings, etc.) except primitives
  - potentially better code organization and reuse

Java Features (2)

- **Interpreted and high performance**
  - Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java byte code.
  - Byte code can be executed on any system that implements the Java Virtual Machine.
  - Java byte code was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using a Just-In-Time compiler.

Java Features (3)

Portable

- By using a hybrid compilation/interpretation approach, Java programs can be executed in a networked environment with different hardware platform and architectures.
- Moreover, there are no platform-specific features in the Java language. This also makes Java applications extremely portable, effectively realizing the “write once, run everywhere” philosophy.
- the sizes of the primitive data types are always the same
- the libraries define portable interfaces
- Standard Unicode format
- The Java environment is portable to new hardware and operating systems.
- In fact, the Java compiler itself is written in Java.

Java Features (4)

- **Dynamic**
  - Java is designed to adapt to an evolving environment
  - Java allows code to be added to libraries dynamically and then can determine which code should run at execution time. This makes it possible to dynamically link code in a safe and expedient manner. (Databases objects, GUI objects)
- **Multithreaded**
  - Java delivers the power of advanced multi-threaded capabilities to the developer in an environment without complexity.
  - More specifically, Java code can be run concurrently as multiple threads in a process, in order to improve its execution performance.
  - Multithreading is a necessity in Multimedia and network programming.

Java Features (5)

- **Robust**
  - To gain reliability, Java perform extensive compile time and runtime error checking
  - No pointers. Memory corruptions or unauthorized memory accesses are impossible. Automatic garbage collection tracks objects usage over time
- **Secure**
  - Java has many facilities to guarantee security in a networked environment. It imposes various types of access restrictions to (networked) resources and carefully supervises memory allocation.
  - It allows code to be downloaded over a network and executed safely in the confined spaces of memory.

The Java Language Specification, API, JRE, JDK, and IDE

- *Java syntax is defined in the Java language specification.*
- The *Application Program Interface (API)*, also known as *library*, contains predefined classes and interfaces for developing Java programs.
  - Oracle provides online documentation of all the Java library classes. For jdk1.8, the link is **https://docs.oracle.com/javase/8/docs/api/**
- *JRE- Java Runtime Environment.* It contains set of libraries + other files that JVM uses at runtime.
- *The JDK* - Java Development Kit. *It is the software for developing and running Java programs. [JRE + development tools]*
- *An IDE is an Integrated Development Environment for rapidly developing programs.*

Java Editions

- Java is a full-fledged and powerful language that can be used in many ways. It comes in three editions:
  - Java *Standard Edition (Java SE)* to develop client-side standalone applications or applets.
  - *Java Enterprise Edition (Java EE)* to develop server-side applications, such as Java servlets, Java Server Pages (JSP), and Java Server Faces (JSF).
  - *Java Micro Edition (Java ME)* to develop applications for mobile devices
- You can get the latest information about Java SE and JDK by using this link
  - http://www.oracle.com/technetwork/java/index.html

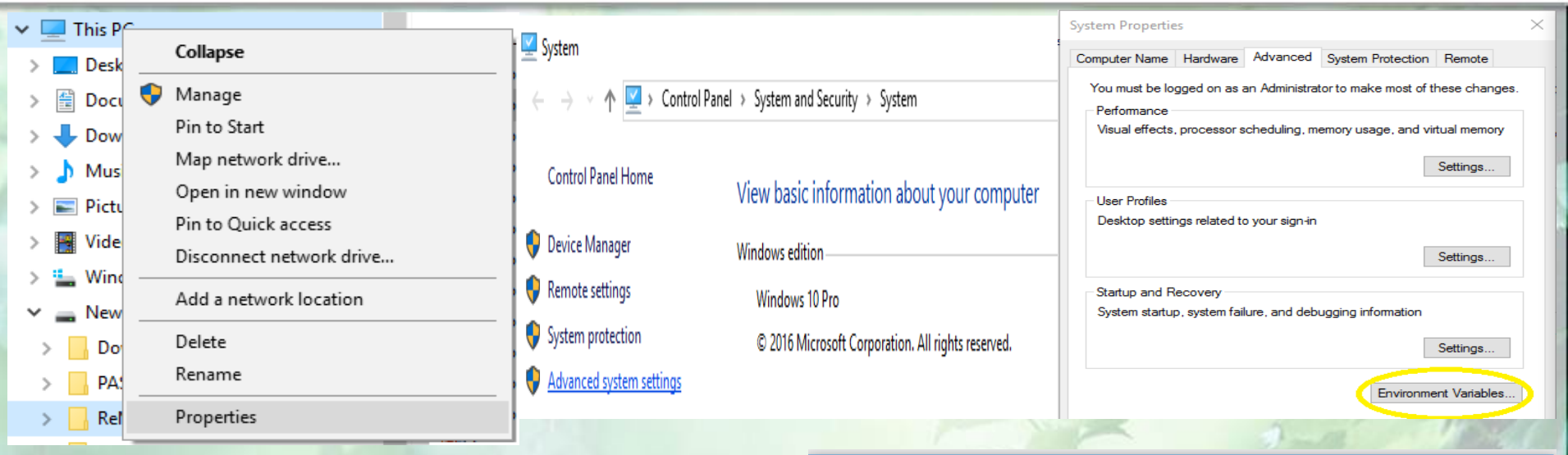
Working with the JDK

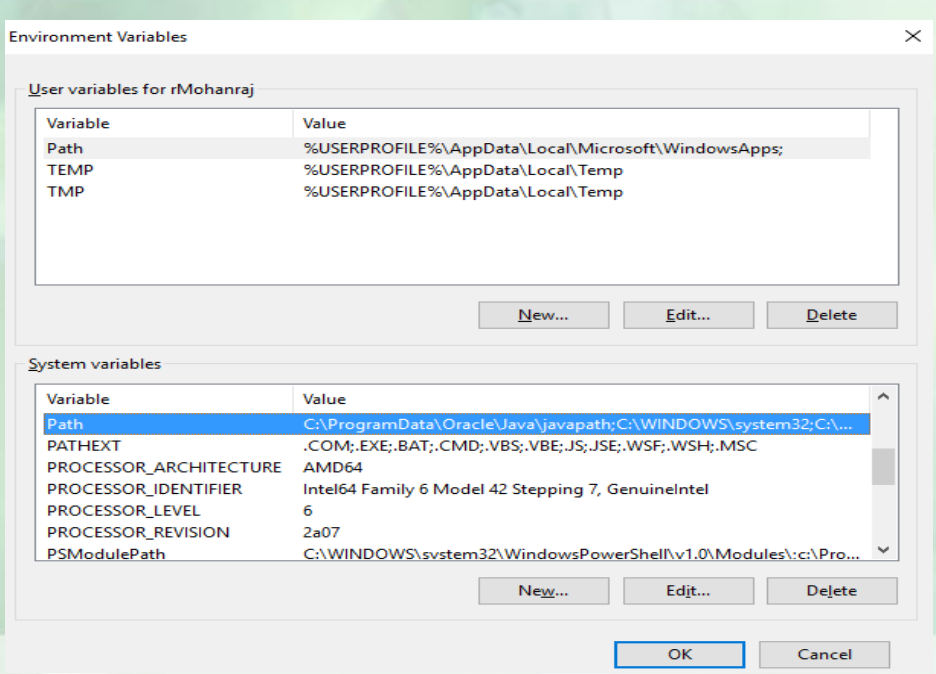
TIP: If, to your PATH environment variable, you add the path to the `javac` compiler and to `java.exe`, compiling and running java programs from the console is much easier. Typical path to these executable (the first is for the 64 bit distribution, the second for the 32 bit distribution):

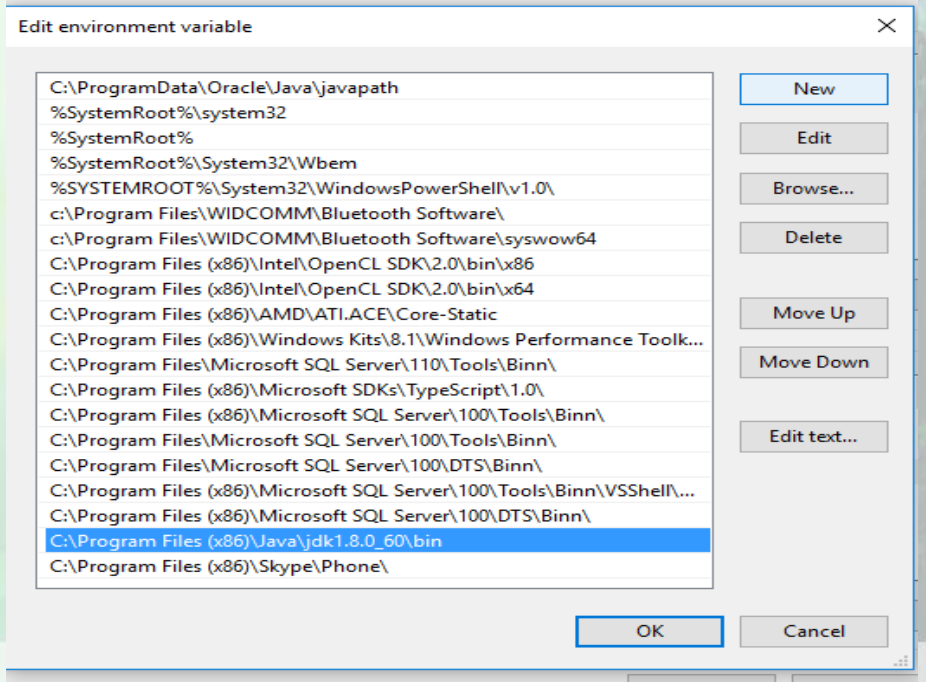
```
C:\Program Files\Java\jdk1.8.0_45\bin
C:\Program Files (x86)\Java\jdk1.8.0_45\bin
```

- To **change** the system environment variables, follow these steps: In the left panel of windows explorer, right-click My Computer and click Properties. In the System Properties **window**, click Advanced System Settings > Environment Variables.
- Now, alter the 'Path' variable by adding the above mentioned paths so that it also contains the path to the Java executable.

Steps to set the Path Variable







Working with JDK

(1) Create the source file:

- open a text editor, type in the code which defines a class (*HelloWorldApp*) and then save it in a file (*HelloWorldApp.java*)
- file and class name are case sensitive and must be matched exactly (except the `.java` part)

Example Code: HelloWorldApp.java

```
public class HelloWorldApp {
  public static void main(String[] args) {
  // Display "Hello World!"
  System.out.println("Hello World!");
  }
}
```

★ Java is CASE SENSITIVE!

Contd....

(2) Compile the program:

- compile HelloWorldApp.java by using the following command:

```
javac HelloWorldApp.java
```

it generates a file named HelloWorldApp.class

(3) Run the program:

- run the code through:

```
java HelloWorldApp
```

Introducing Java Things to understand:

- public
- class
- static
- void
- main
- String[]
- System, System.out, System.out.println (vs System.out.print)
- delimiters: ;, }, { (“blocks”)
- capitalization conventions

Quiz

public static void main(String[] a) { }. This statement is _____

a. True b. False

Byte code is Platform dependent.

a. True b. False

Main Point 1

- Java is an object-oriented language that is easier to use, similar to C++, but simplified to eliminate language features that cause common programming errors.
- *Science of Consciousness* :Transcendental Meditation is an easy and effortless technique to make a mind clear and more alert so that we can achieve anything in our life without error.

Integrated Development Environments

- A good IDE supports compiling, running, and debugging code with tools that are integrated and typically easy to use. For a large Java project, an IDE is indispensable.

- Good choices of IDE are NetBeans, Sun Forte, IBM Rational Application Developer (formerly WebSphere Application Developer), Borland's JBuilder, JetBrains' IntelliJ

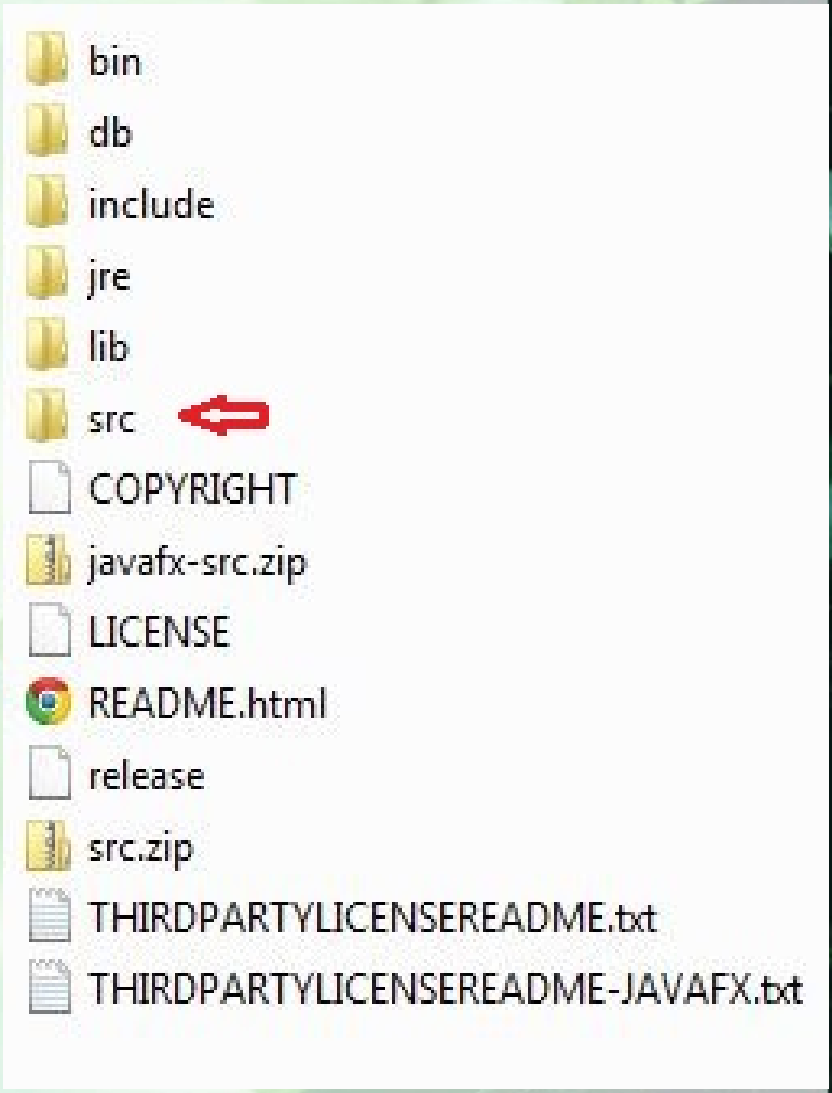
- Another excellent choice, which has become an industry standard, is the open-source IDE Eclipse, written entirely in Java. We will use Eclipse in this course.
  - Install JDK
  - Install Eclipse IDE
  - To use JSE 8, you need Eclipse Luna (or later); earlier versions of Eclipse do not support JSE 8.

  www.vogella.com/articles/Eclipse/article.html

The JDK Distribution & Viewing Source Code

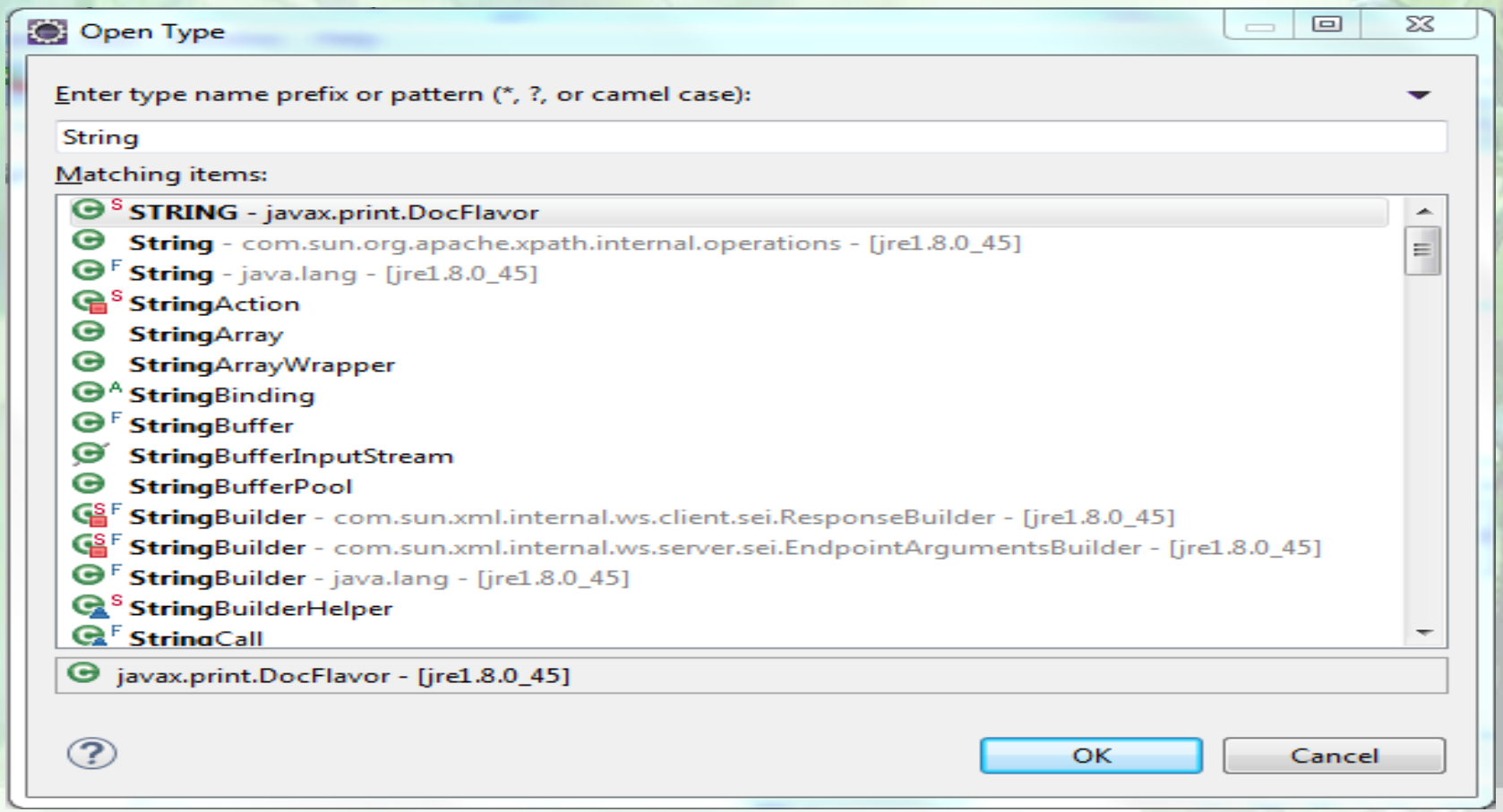
- It's a good idea to unzip src.zip and place in a source folder (called "src" in the screen capture above) – this will allow you to see how Oracle has implemented its library classes. You can also unzip javafx-src.zip to look at JavaFX source code.

- Oracle provides online documentation of all the Java library classes. For jdk1.8, the link is https://docs.oracle.com/javase/8/docs/api/

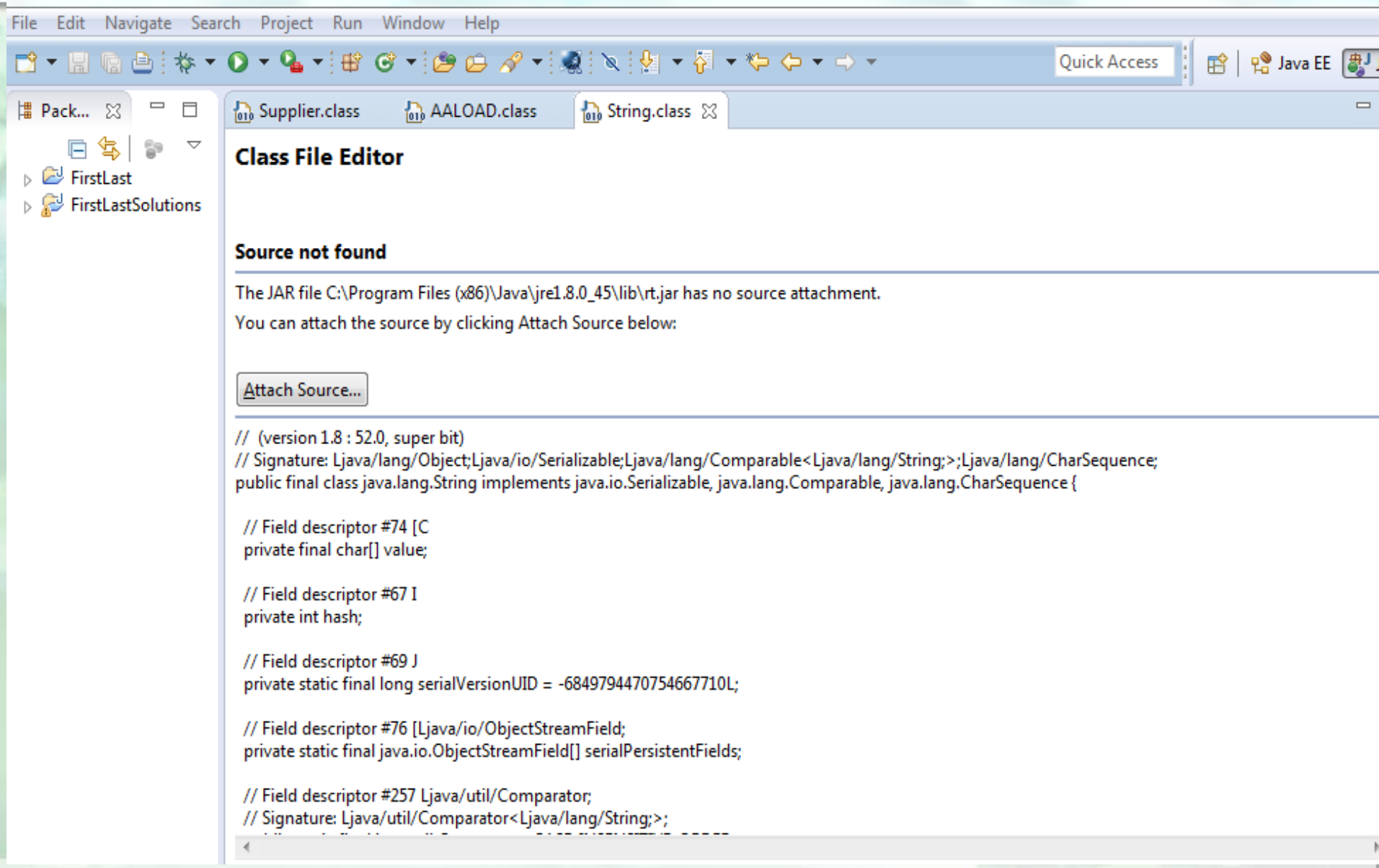


View of Attaching Java Source Code from API

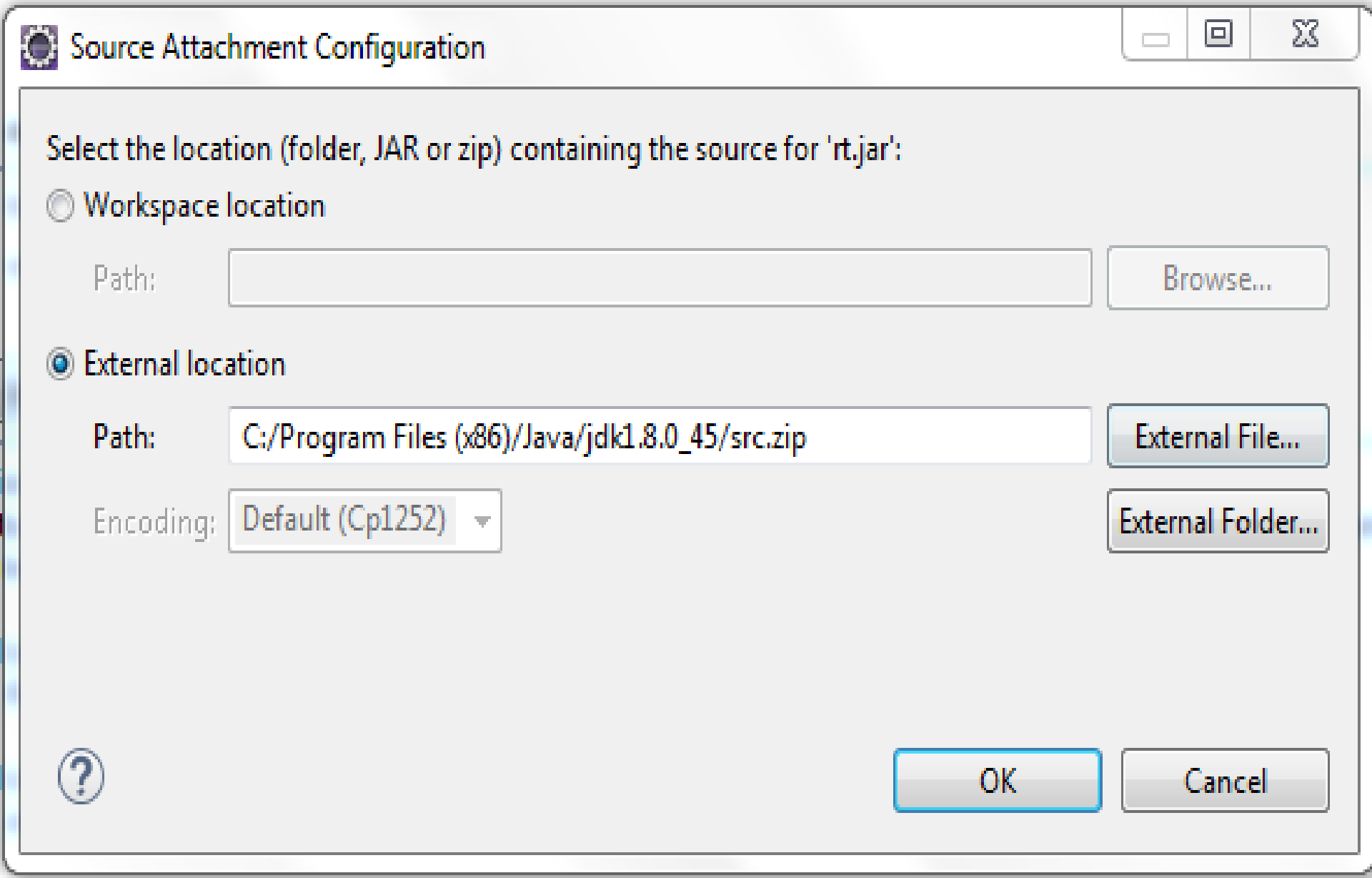
- Allows you to see how Java has implemented their library code.
- Steps:
  1. Ctrl – Shift – T
     - Brings up this screen: (Here, “String” was typed into search window”)
     - Click OK.



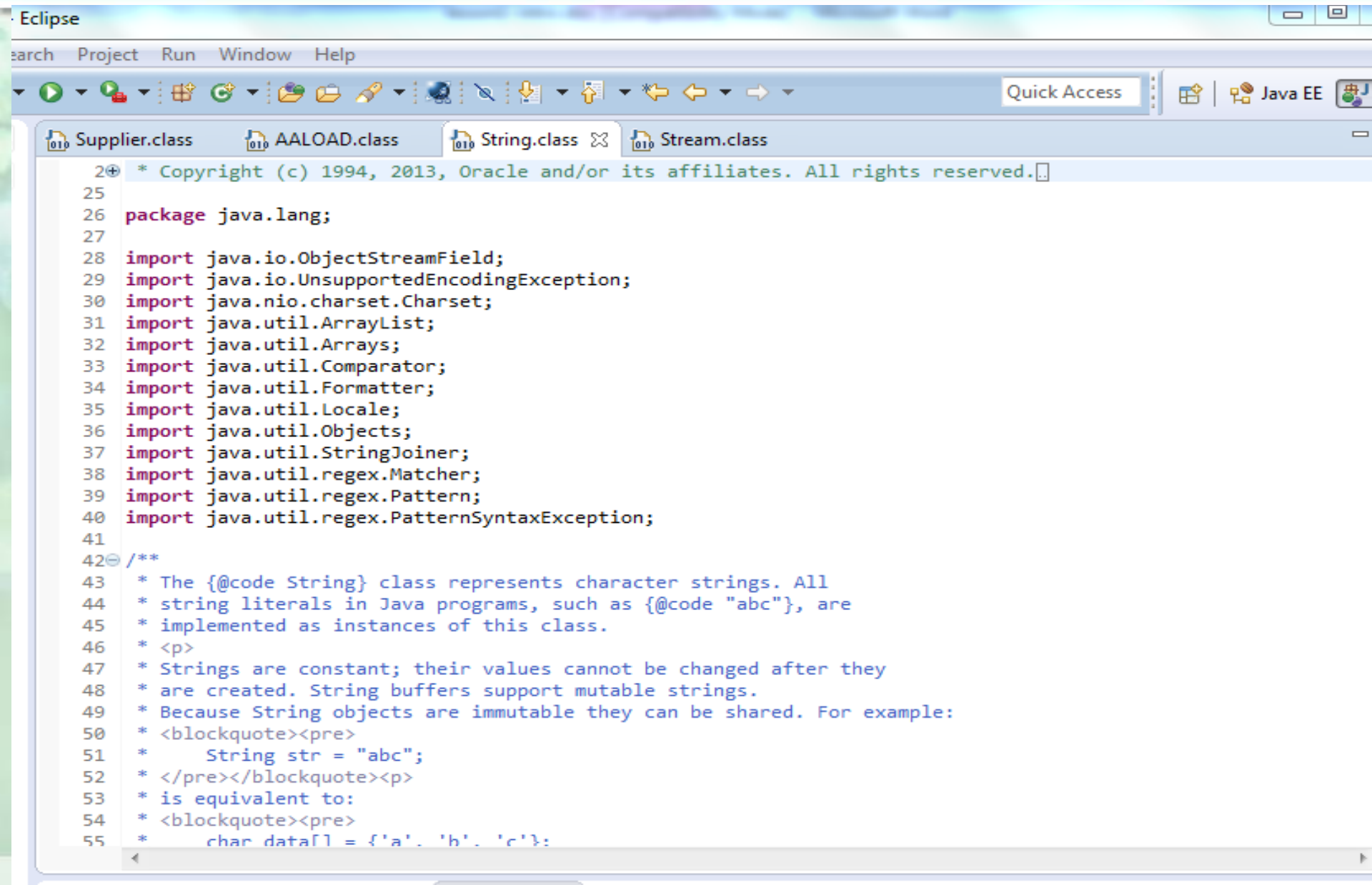
2. After Clicking OK from previous screen lets you see incomplete info of the class file (of String in this case), Click Attach Source



3. After Clicking Attach Source and navigate to src.zip in your jdk distribution (do only once – next time just do steps A and B) by clicking External File.



4. You can now see the source code for your selected type



MAIN POINT 2

- Eclipse is a leading, open-source, 100% Java, integrated development environment, which provides excellent support for editing, compiling, running, and debugging Java applications. By analogy, to create a good life, we need to handle inner life and at the same time, structure a life-supporting environment – the goal is to live 200% of life.

UNITY CHART

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

The Usefulness of Java and Eclipse IDE

- Using Java, highly functional applications can be built more quickly and easily with fewer mistakes than is typically possible using C or C++.
- To optimize the use of Java's features, IDE's such as Eclipse ease the work of the developer by handling in the background many routine tasks.

- ***Transcendental Consciousness*** is the source from the field of pure intelligence which is the basis for successful action.
- ***Impulses within the Transcendental Field :*** *At this level of experience, consciousness acting within itself to produce the desired outcome.*
- ***Wholeness moving within Itself****: In Unity Consciousness, the pure intelligence located in TC is found pervading all of creation, from gross to subtle.*