



# What is Java?

---

- ❑ Java is a **programming language** and a **platform**.
- ❑ **Platform:** Any hardware or software environment in which a program runs, known as a platform. Since Java has its own Runtime Environment (JRE) and API, it is called platform.



# Where it is used?

- According to Sun, 3 billion devices run java. There are many devices where java is currently used. Some of them are as follows:
  - ▣ Desktop Applications such as acrobat reader, media player, antivirus etc.
  - ▣ Web Applications.
  - ▣ Enterprise Applications such as banking applications.
  - ▣ Mobile
  - ▣ Embedded System
  - ▣ Smart Card
  - ▣ Robotics
  - ▣ Games etc.

# Types of Java Applications

- ❑ There are mainly 4 type of applications that can be created using java:
- ❑ **1) Standalone Application**
  - ❑ It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.
- ❑ **2) Web Application**
  - ❑ An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.
- ❑ **3) Enterprise Application**
  - ❑ An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.
- ❑ **4) Mobile Application**
  - ❑ An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.



# Java History

- **Java history** is interesting to know. Java team members (also known as **Green Team**), initiated a revolutionary task to develop a language for digital devices such as set-top boxes, televisions etc.
- For the green team members, it was an advance concept at that time. But, it was suited for internet programming. Later, Java technology as incorporated by Netscape.
- **James Gosling**
  - [http://en.wikipedia.org/wiki/James\\_Gosling](http://en.wikipedia.org/wiki/James_Gosling)



# Java History

- Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describes the history of java.
- 1) **James Gosling, Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.
- 2) Originally designed for small, embedded systems in electronic appliances like set-top boxes.
- 3) Firstly, it was called "**Greentalk**" by James Gosling and file extension was .gt.
- 4) After that, it was called **Oak** and was developed as a part of the Green project.



# OAK Tree



# Why Oak name for java language?

- **Why Oak?** Oak is a symbol of strength and chosen as a national tree of many countries like U.S.A., France, Germany, Romania etc.
- In 1995, Oak was renamed as "**Java**" because it was already a trademark by Oak Technologies.



# Why Java name for java language?

- ❑ **Why they choose java name for java language?** The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA" etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell and fun to say.
- ❑ According to James Gosling "Java was one of the top choices along with **Silk**". Since java was so unique, most of the team members preferred java.
- ❑ Java is an island of Indonesia where first coffee was produced (called java coffee).
- ❑ Notice that Java is just a name not an acronym.
- ❑ Originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.
- ❑ In 1995, Time magazine called **Java one of the Ten Best Products of 1995**.
- ❑ JDK 1.0 released in (January 23, 1996).

# Java Version History

- There are many java versions that has been released.
  - ▣ JDK Alpha and Beta (1995)
  - ▣ JDK 1.0 (23rd Jan, 1996)
  - ▣ JDK 1.1 (19th Feb, 1997)
  - ▣ J2SE 1.2 (8th Dec, 1998)
  - ▣ J2SE 1.3 (8th May, 2000)
  - ▣ J2SE 1.4 (6th Feb, 2002)
  - ▣ J2SE 5.0 (30th Sep, 2004)
  - ▣ Java SE 6 (11th Dec, 2006)
  - ▣ Java SE 7 (28th July, 2011)
  - ▣ Java SE 8 (March 18, 2014)

Download:

<http://www.oracle.com/technetwork/java/javase/downloads/index.htm>

|

- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- Java DB
- Web Tier
- Java Card
- Java TV
- New to Java
- Community
- Java Magazine

## Java SE Downloads



Java Platform (JDK) 8u45



NetBeans with JDK 8

### Java Platform, Standard Edition

#### Java SE 8u45

This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release  
[Learn more](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Products](#)
- [Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme Files](#)
  - [JDK ReadMe](#)
  - [JRE ReadMe](#)

#### JDK

[DOWNLOAD](#)

#### Server JRE

[DOWNLOAD](#)

#### JRE

[DOWNLOAD](#)



Junho 23–25, 2015  
São Paulo, Brasil  
#javaonebr


[Inscreva-se](#)



ORACLE



# Features of Java

- There is given many features of java. They are also known as java buzzwords.
    - ▣ Simple
    - ▣ Object-Oriented
    - ▣ Platform independent
    - ▣ Secured
    - ▣ Robust
    - ▣ Architecture neutral
    - ▣ Portable
    - ▣ Dynamic
    - ▣ Interpreted
    - ▣ High Performance
    - ▣ Multithreaded
    - ▣ Distributed
- 



# Simple

- According to Sun, Java language is simple because: syntax is based on C++ (so easier for programmers to learn it after C++). removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc. No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

# Object-oriented

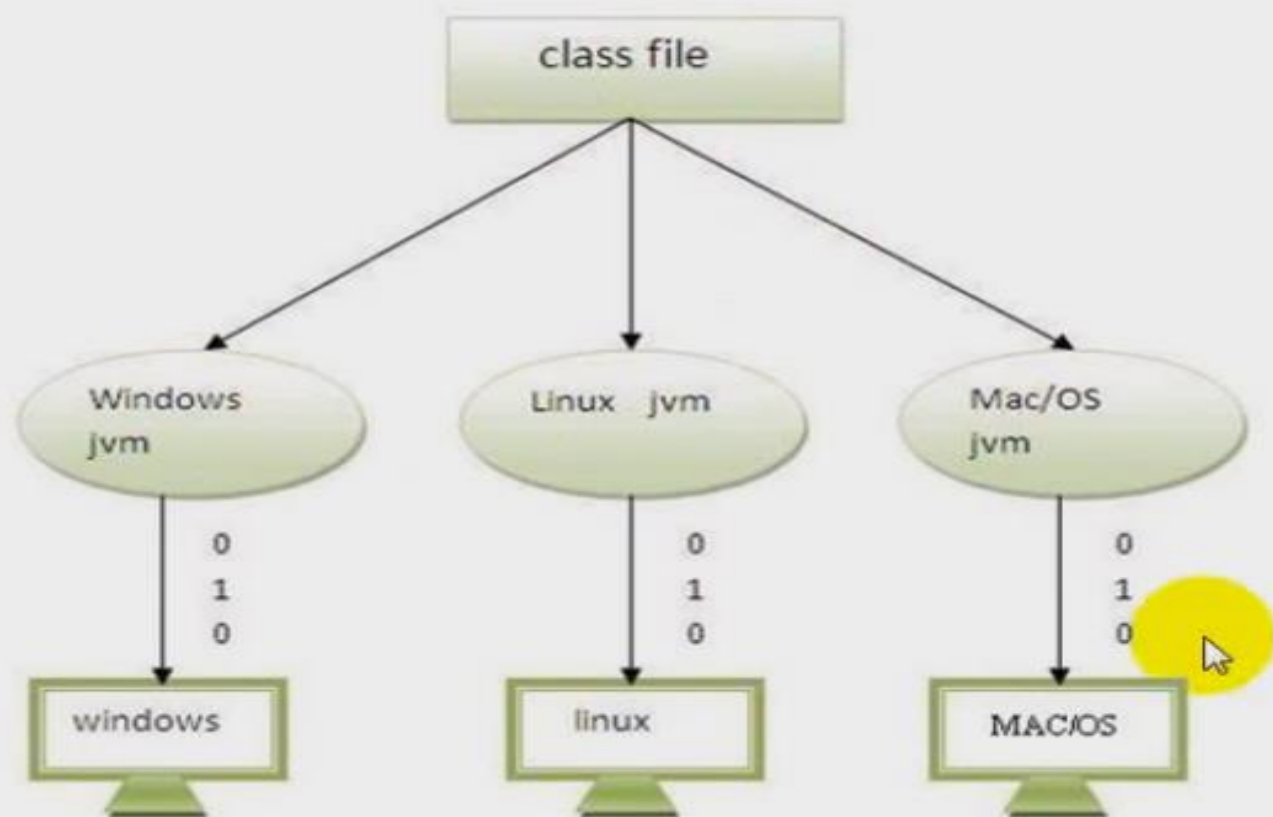
- Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour. Object-oriented programming(OOPs) is a methodology that simplify software development and maintenance by providing some rules. Basic concepts of OOPs are: Object
  - ▣ Class
  - ▣ Inheritance
  - ▣ Polymorphism
  - ▣ Abstraction
  - ▣ Encapsulation



# Platform Independent

- A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform. The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms. It has two components:
  - ▣ Runtime Environment
  - ▣ API(Application Programming Interface)
- Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

# Platform Independent

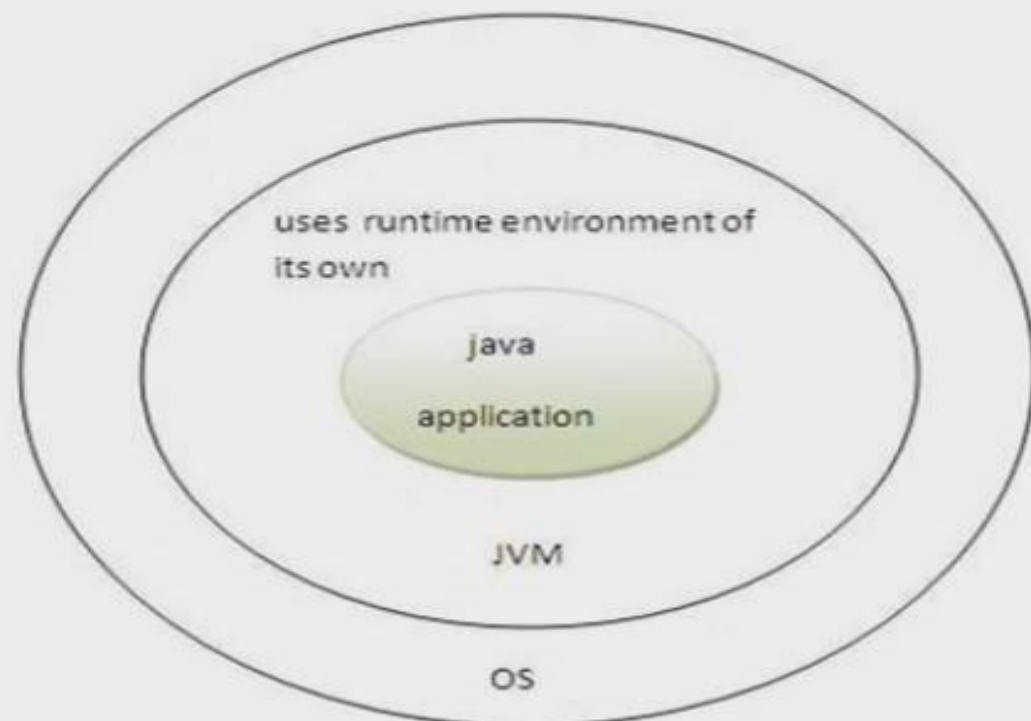
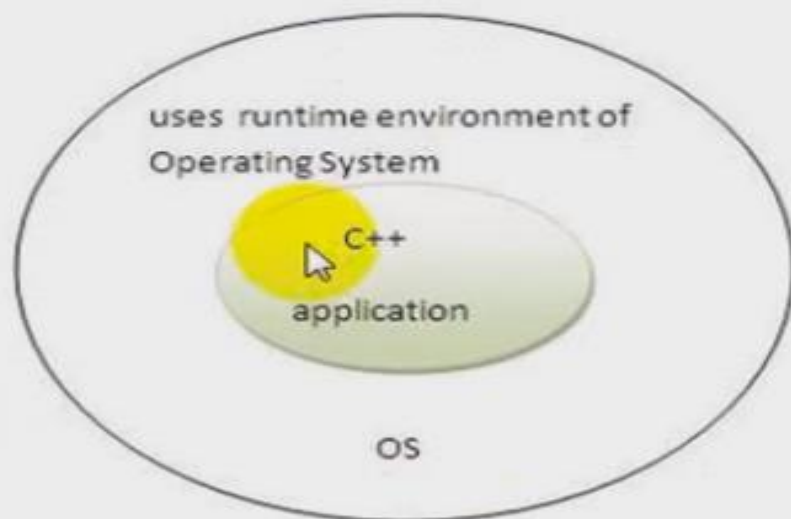




# Secured

- ❑ Java is secured because: No explicit pointer
- ❑ Programs run inside virtual machine sandbox.
- ❑ **ClassLoader**- adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- ❑ **Bytecode Verifier**- checks the code fragments for illegal code that can violate access right to objects.
- ❑ **Security Manager**- determines what resources a class can access such as reading and writing to the local disk.
- ❑ These security are provided by java language. Some security can also be provided by application developer through SSL, JAAS, cryptography etc.

# Secured



# Robust

- ❑ Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.

# Architecture-neutral, Portable

- There is no implementation dependent features e.g. size of primitive types is set.
- Portable means we may carry the java bytecode to any platform.



# High-performance

- Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

# Distributed

---

- We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.



# Multi-threaded

- A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it shares the same memory. Threads are important for multi-media, Web applications etc.

# Difference between JDK, JRE and JVM

- Understanding the difference between JDK, JRE and JVM is important in Java. We are having brief overview of JVM here.
- If you want to get the detailed knowledge of Java Virtual Machine, move to the next page. Firstly, let's see the basic differences between the JDK, JRE and JVM

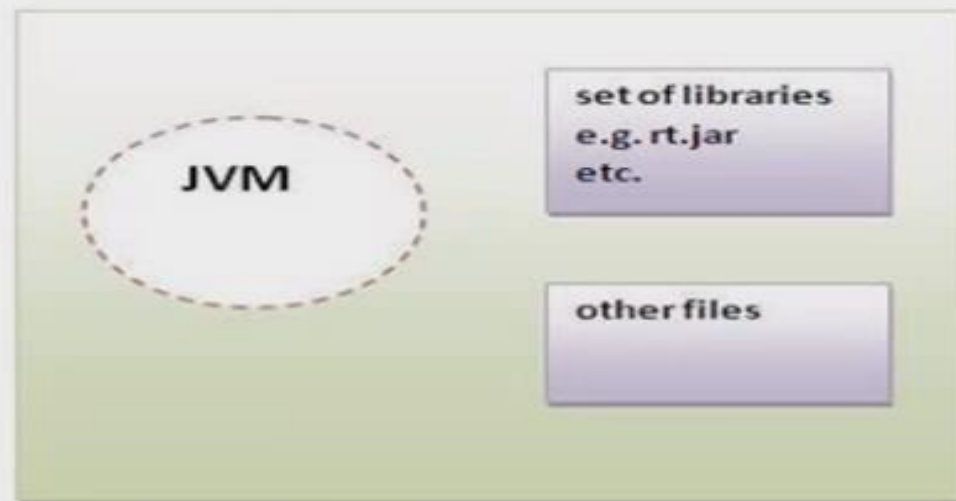


# JVM

- ❑ JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed. JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent. The JVM performs following main tasks:
  - ❑ Loads code
  - ❑ Verifies code
  - ❑ Executes code
  - ❑ Provides runtime environment

# JRE

- JRE is an acronym for Java Runtime Environment. It is used to provide runtime environment. It is the implementation of JVM. It physically exists. It contains set of libraries + other files that JVM uses at runtime. Implementation of JVMs are also actively released by other companies besides Sun Micro Systems.

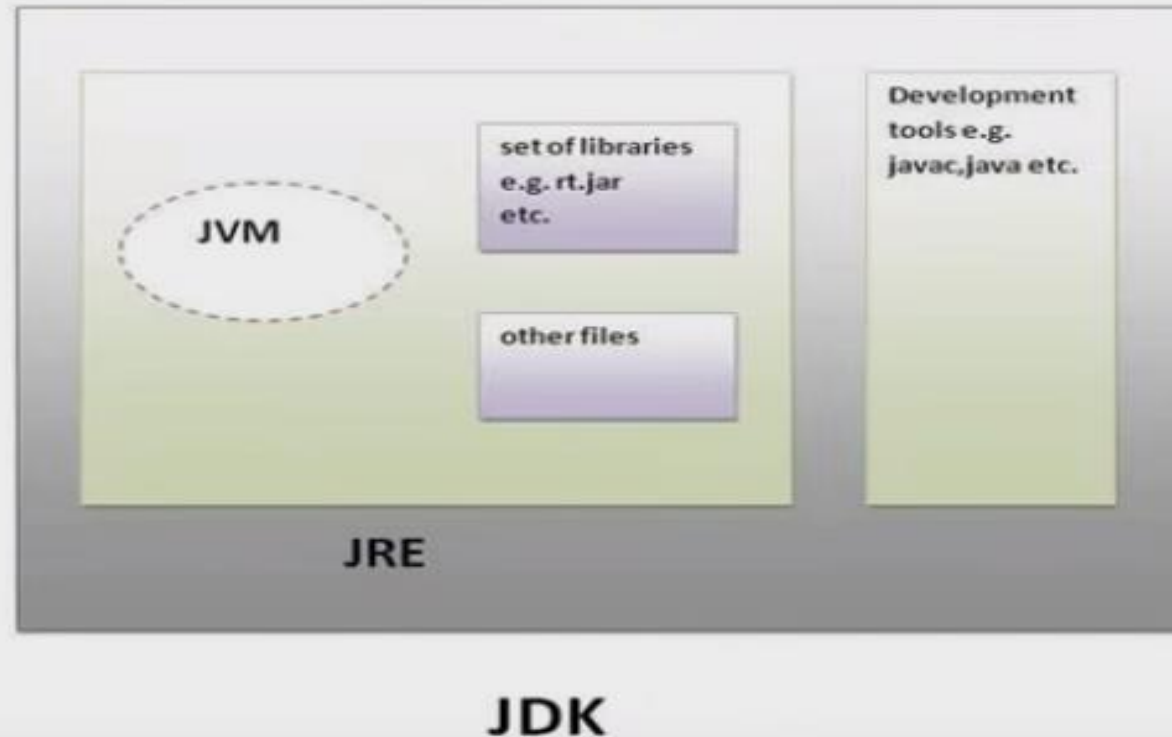


- JRE

**JRE**

# JDK

- JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.



# JVM (Java Virtual Machine) -details

- JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.
- JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).



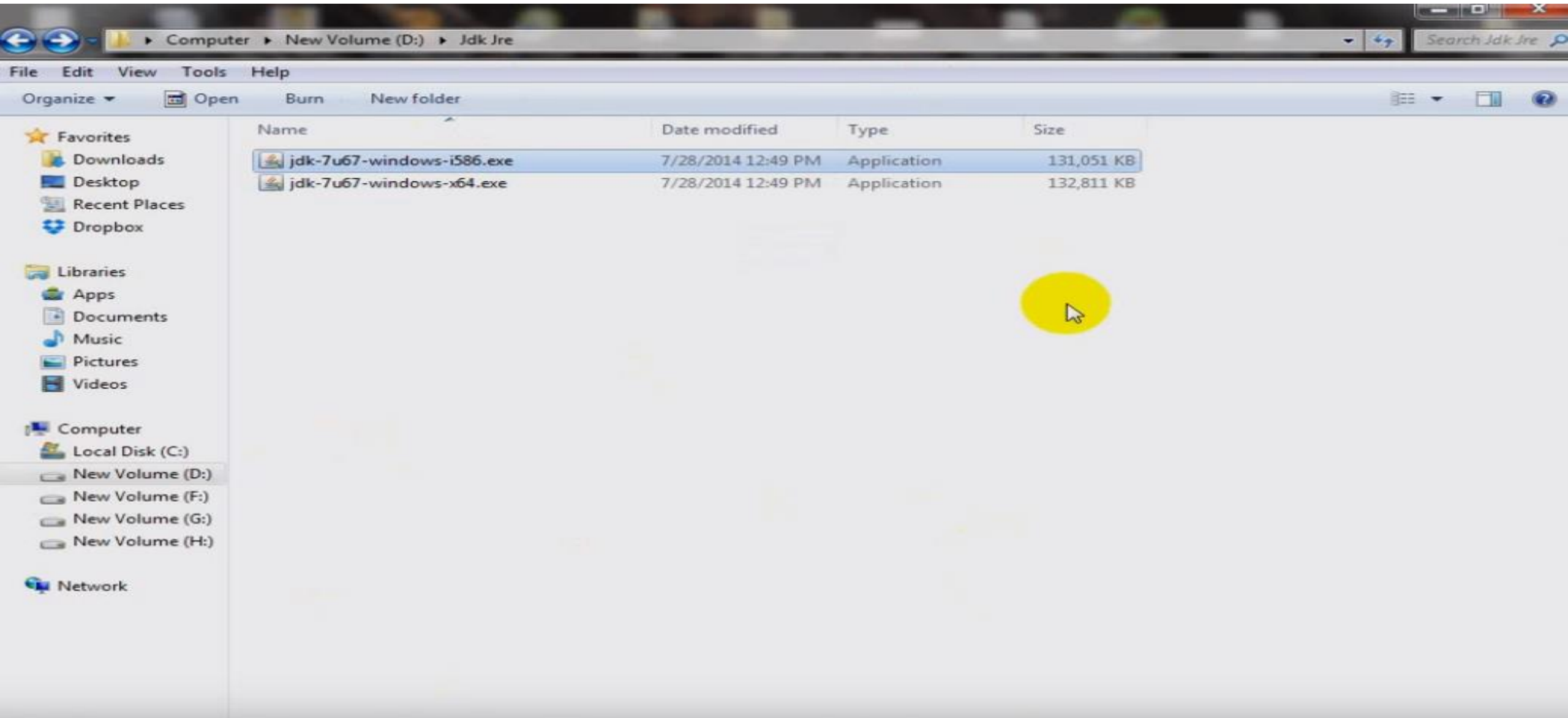


# What it does?

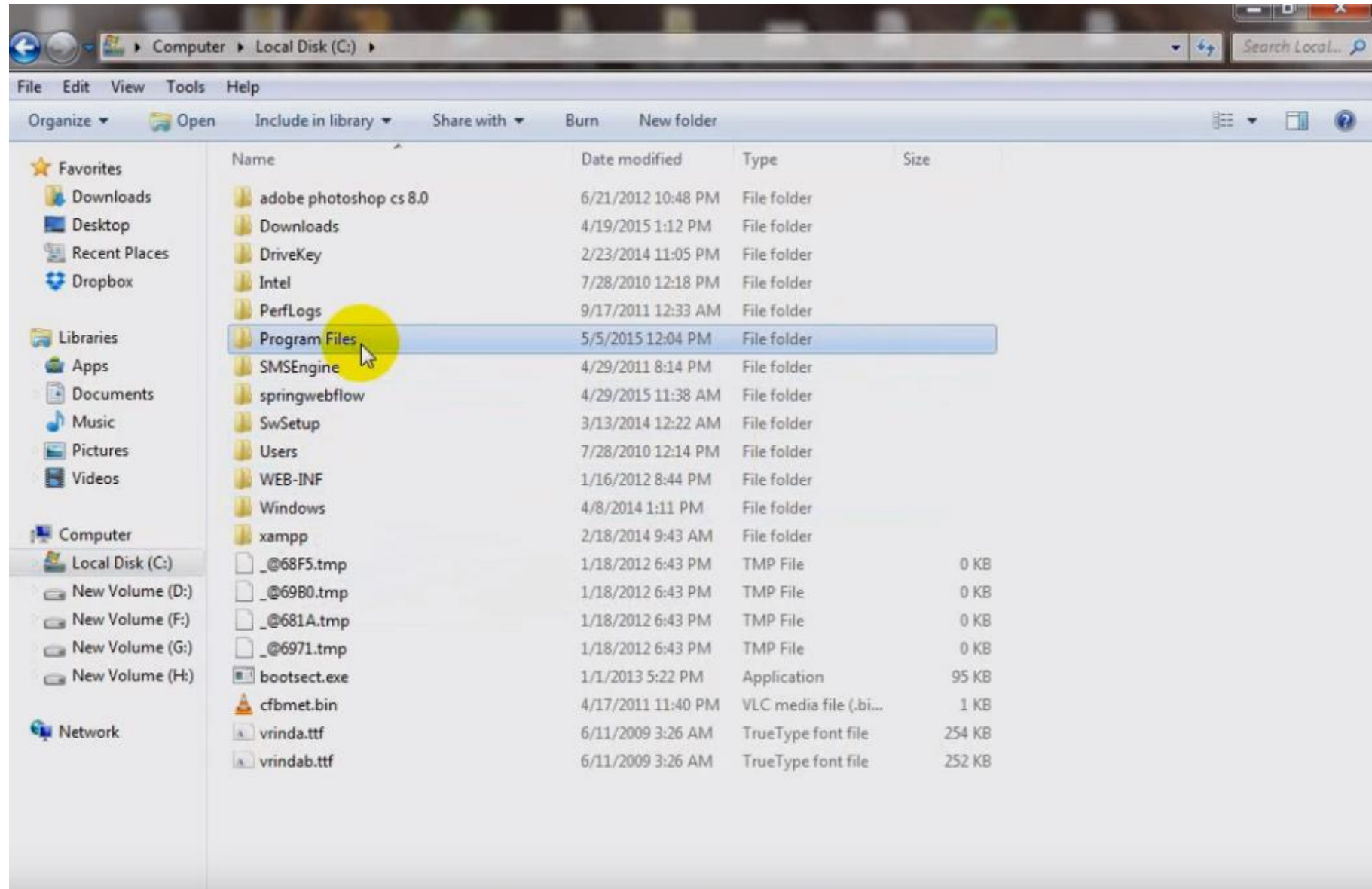
- The JVM performs following operation:
  - ▣ Loads code
  - ▣ Verifies code
  - ▣ Executes code
  - ▣ Provides runtime environment
- JVM provides definitions for the:
  - ▣ Memory area
  - ▣ Class file format
  - ▣ Register set
  - ▣ Garbage-collected heap
  - ▣ Fatal error reporting etc.



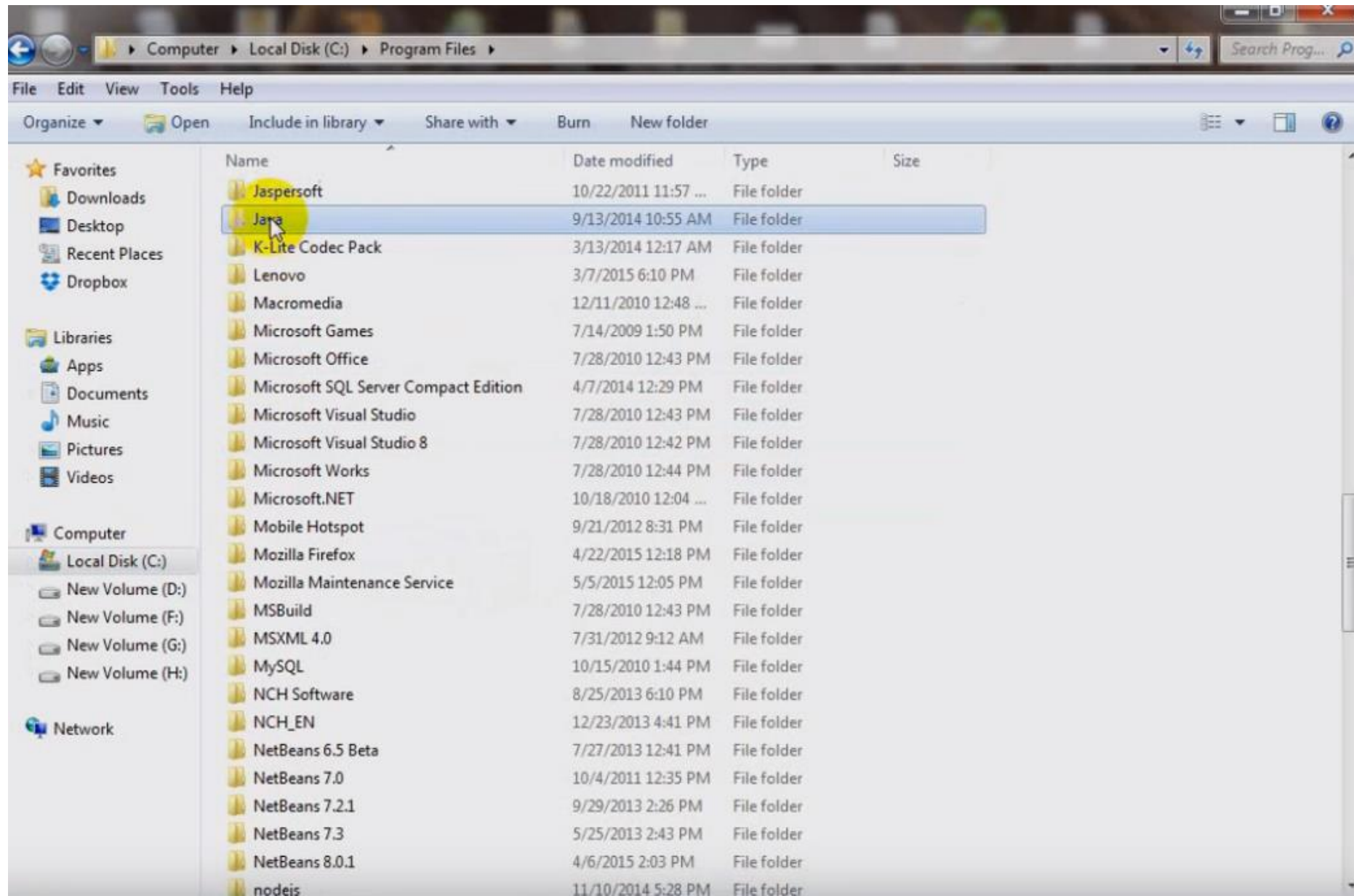
## Installations JDK



## Installations JDK

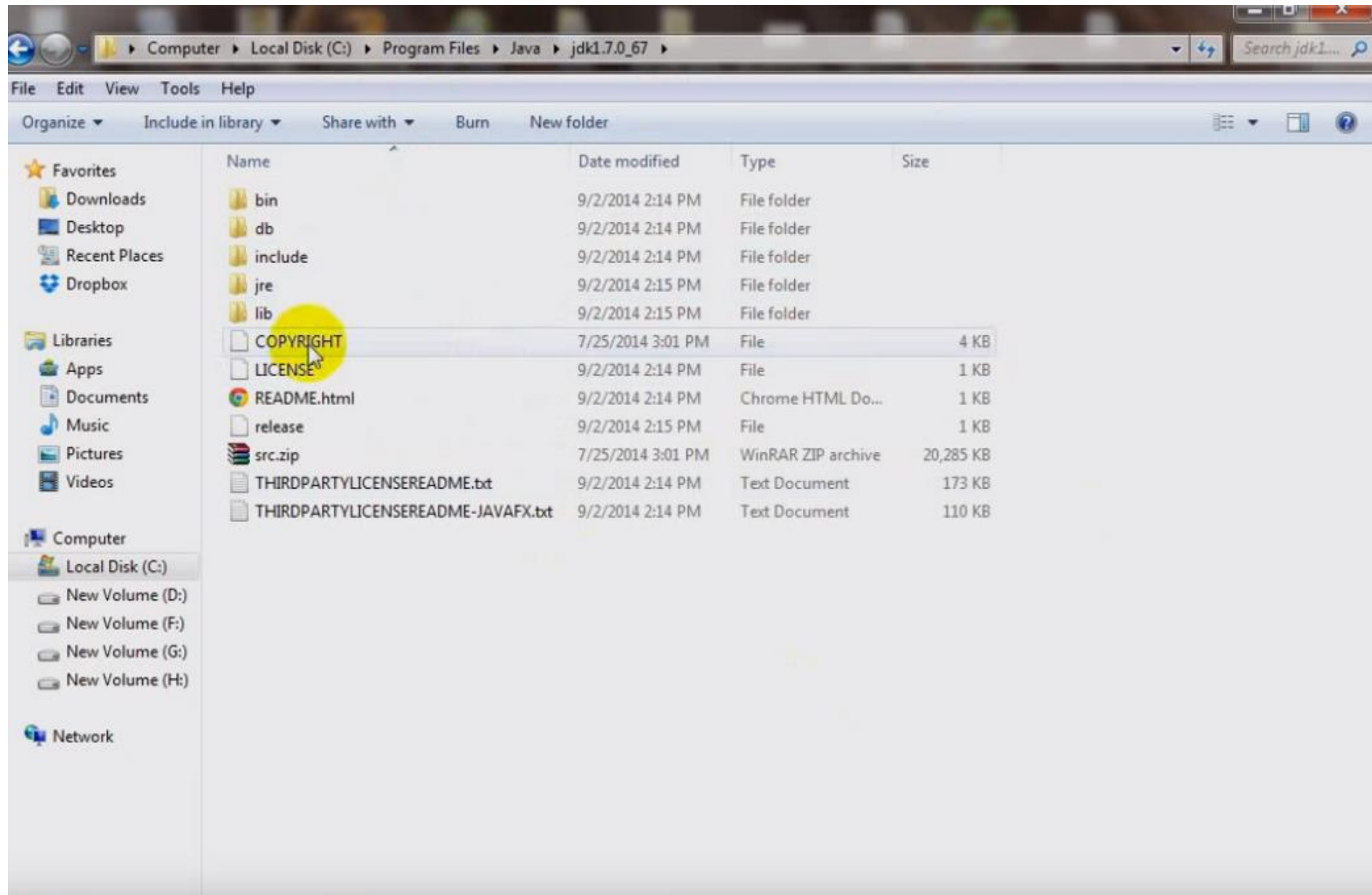


## Installations JDK

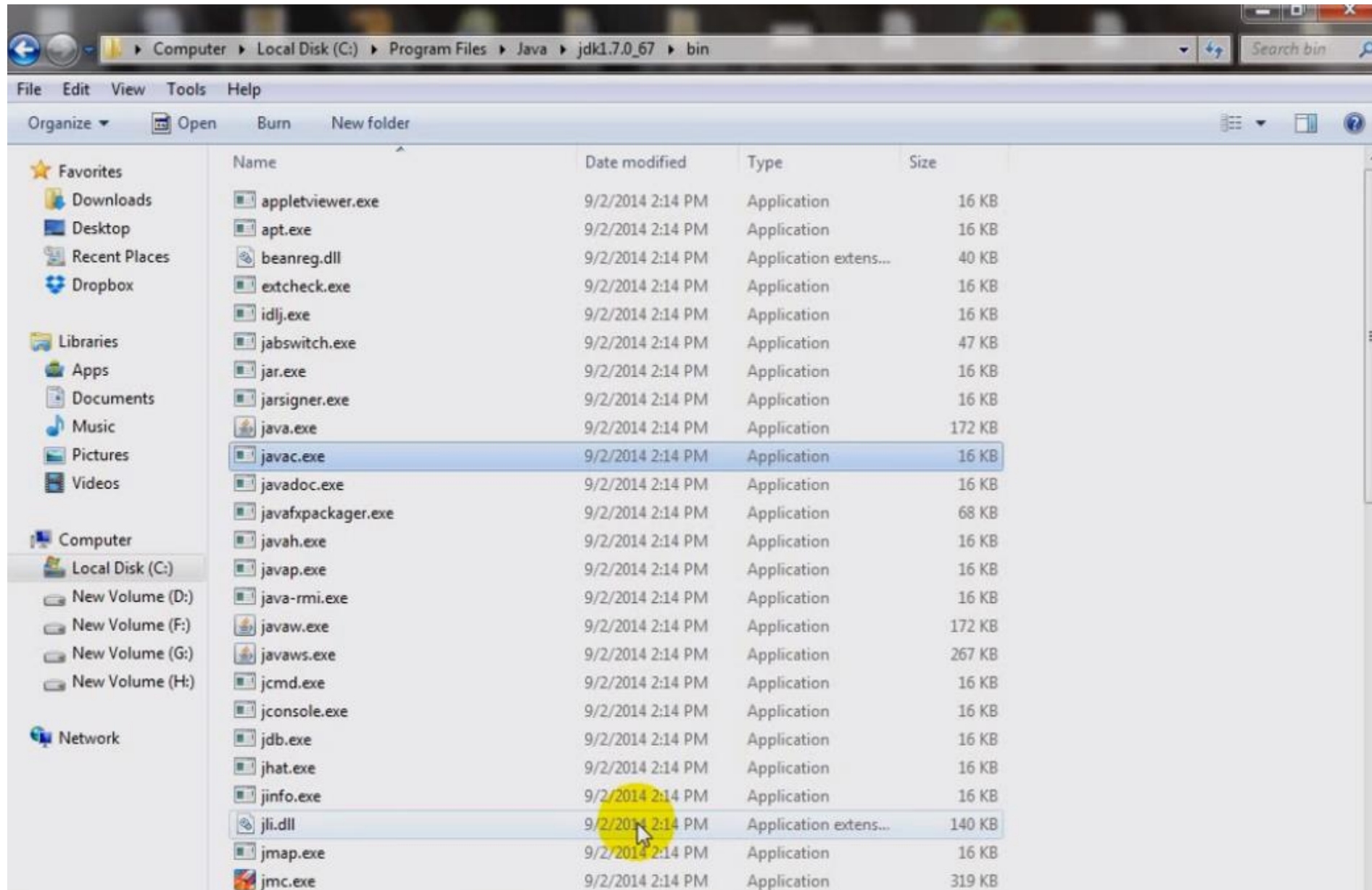




## Installations JDK

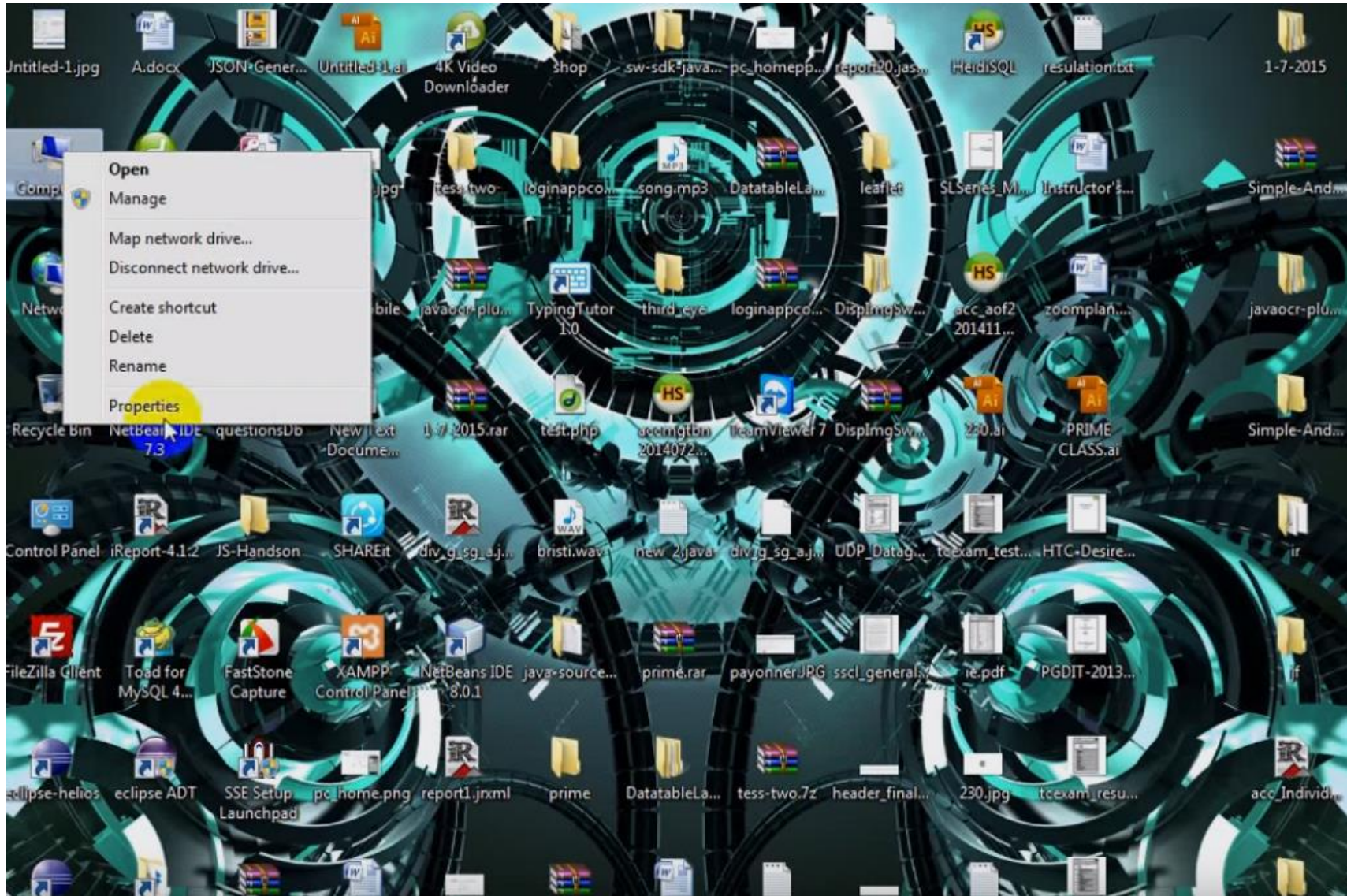


## Installations JDK

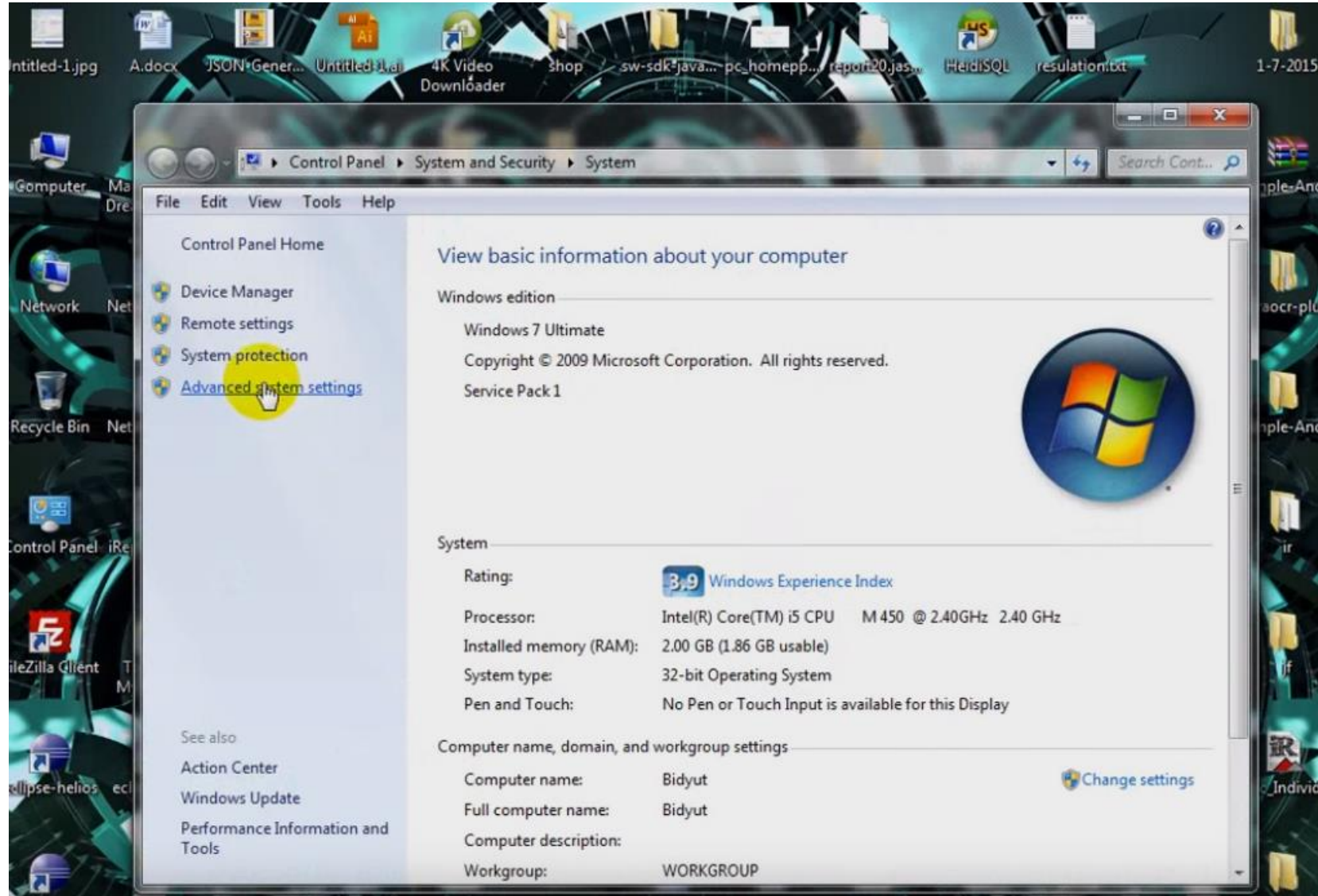




## JDK Path setup

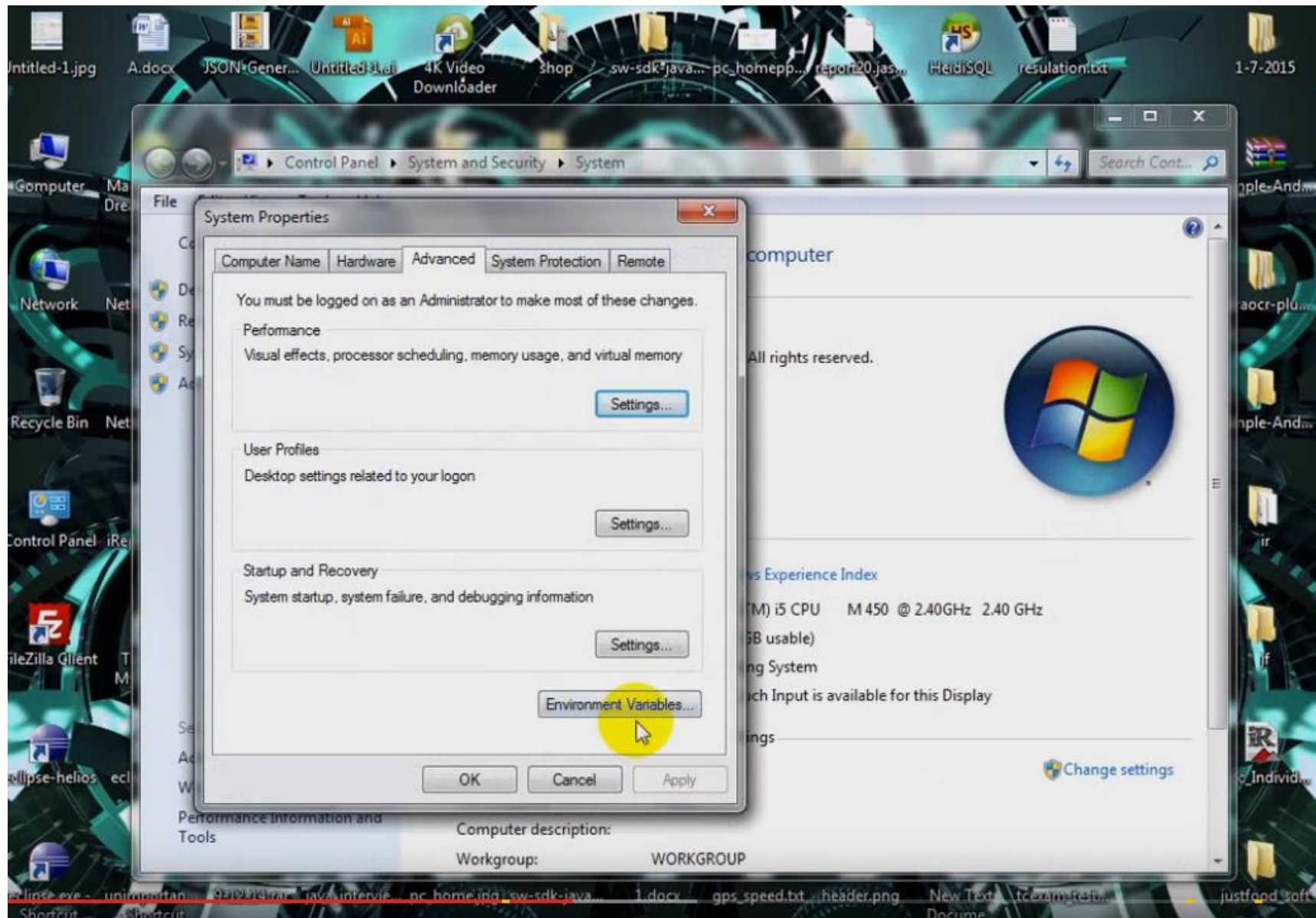


## JDK Path setup

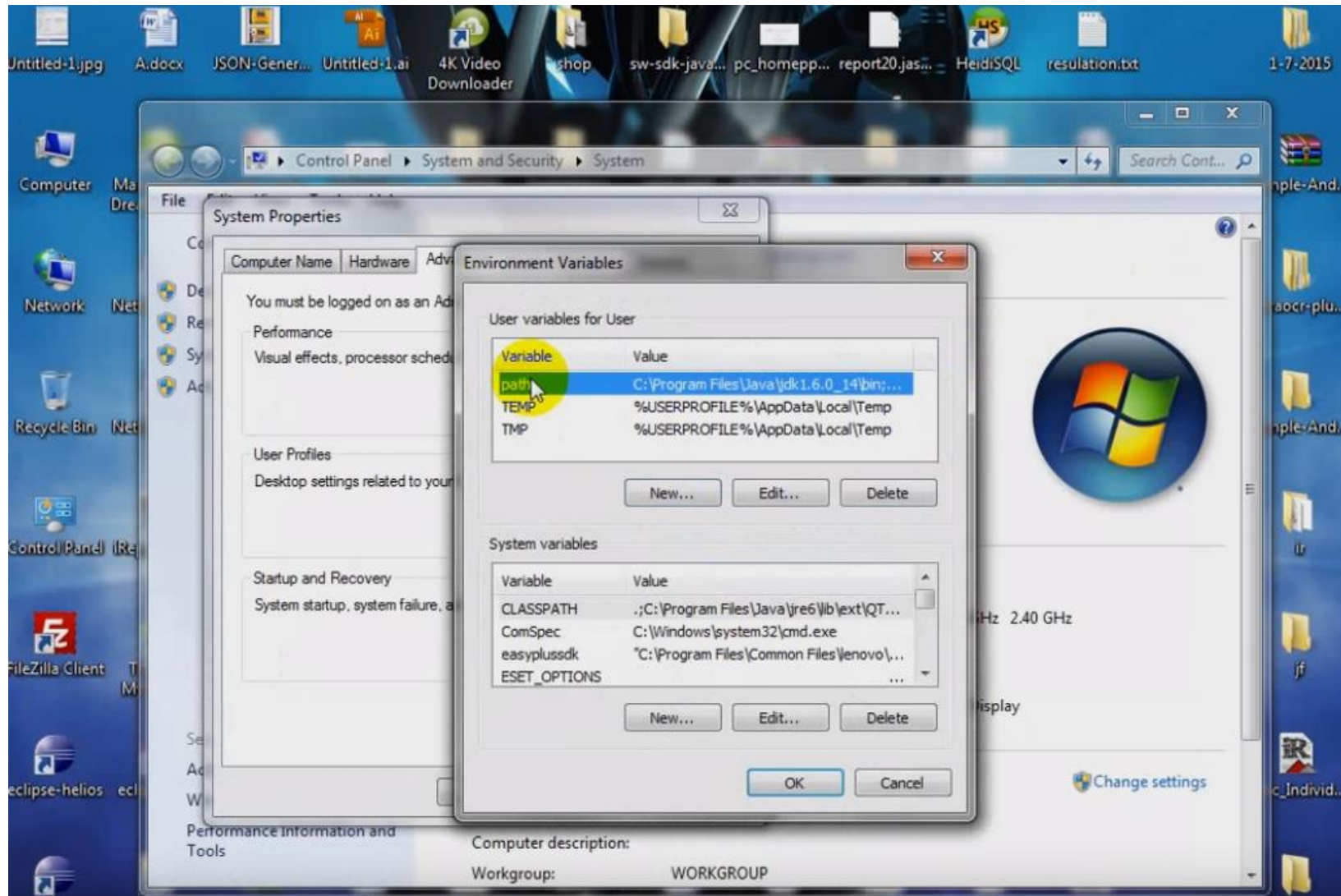




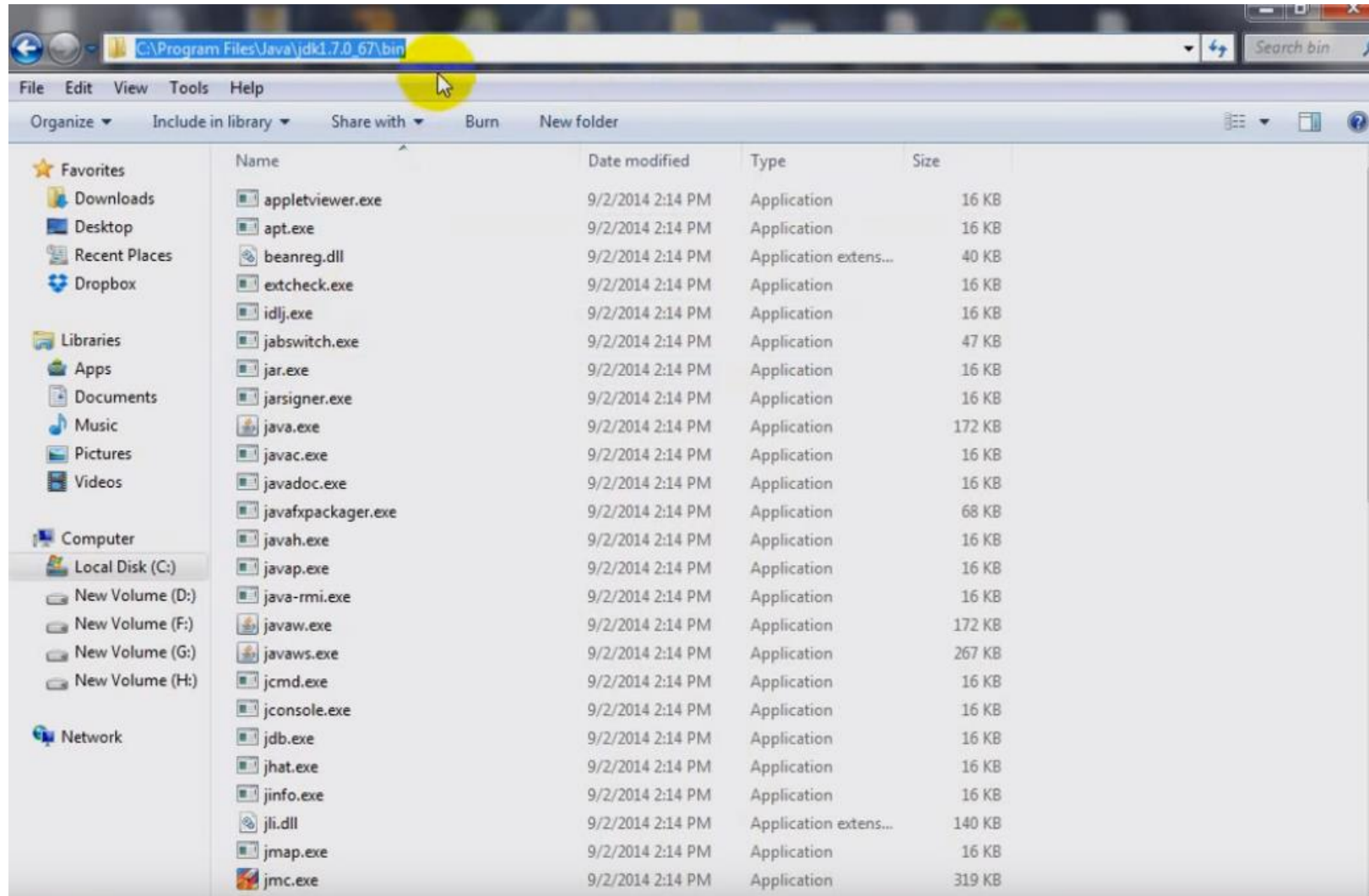
## JDK Path setup



## JDK Path setup

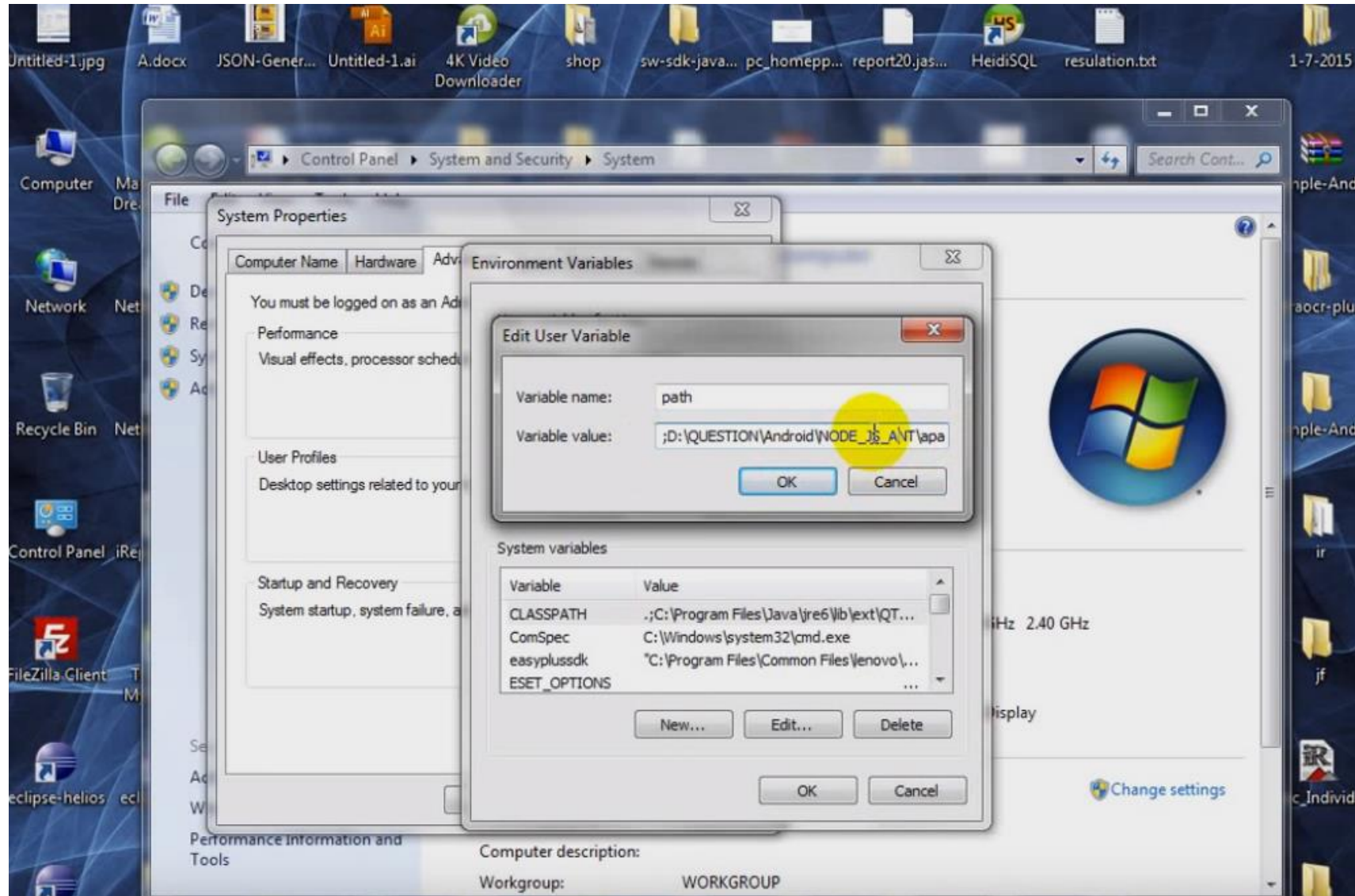


## JDK Path setup



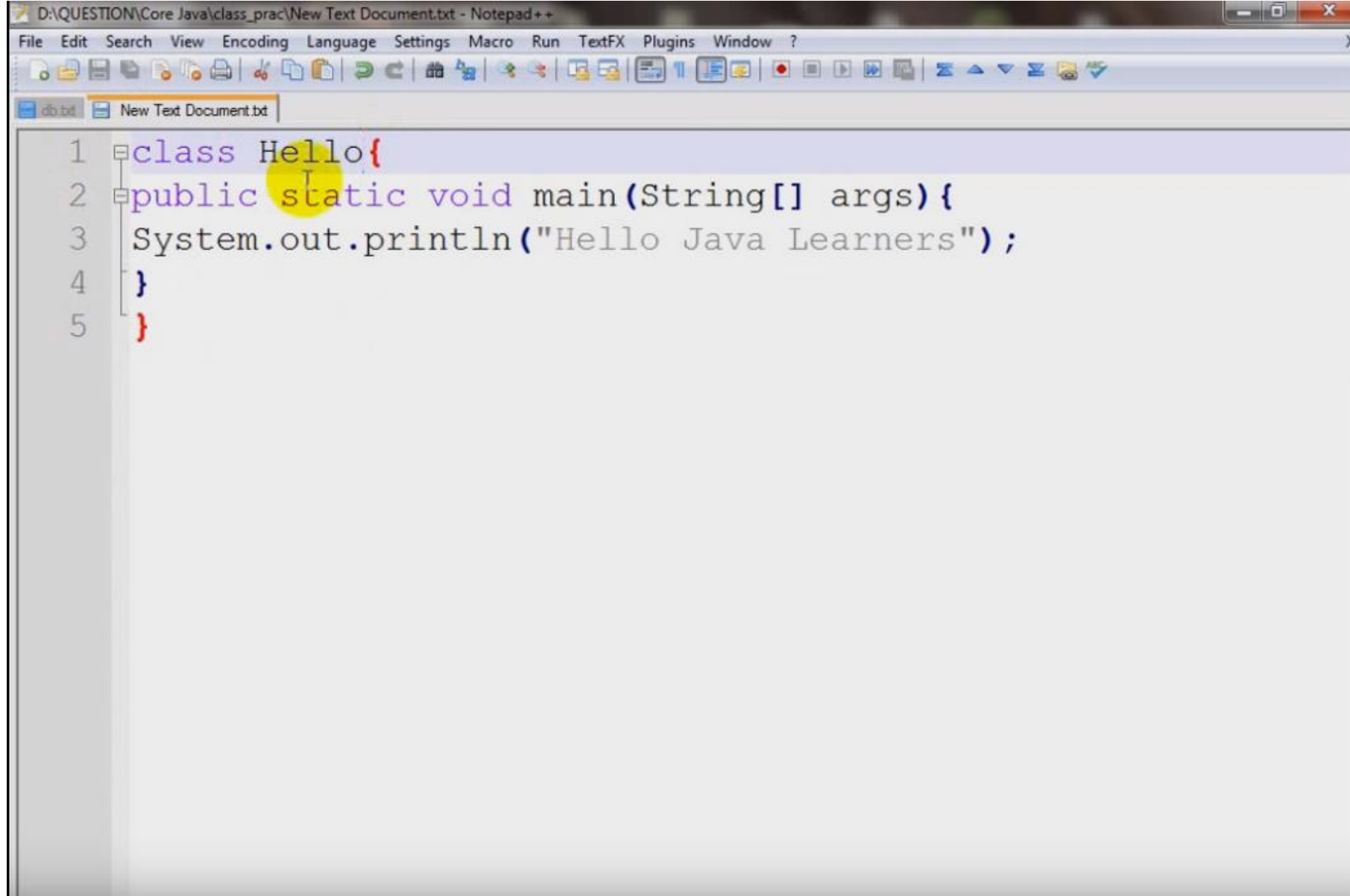


## JDK Path setup



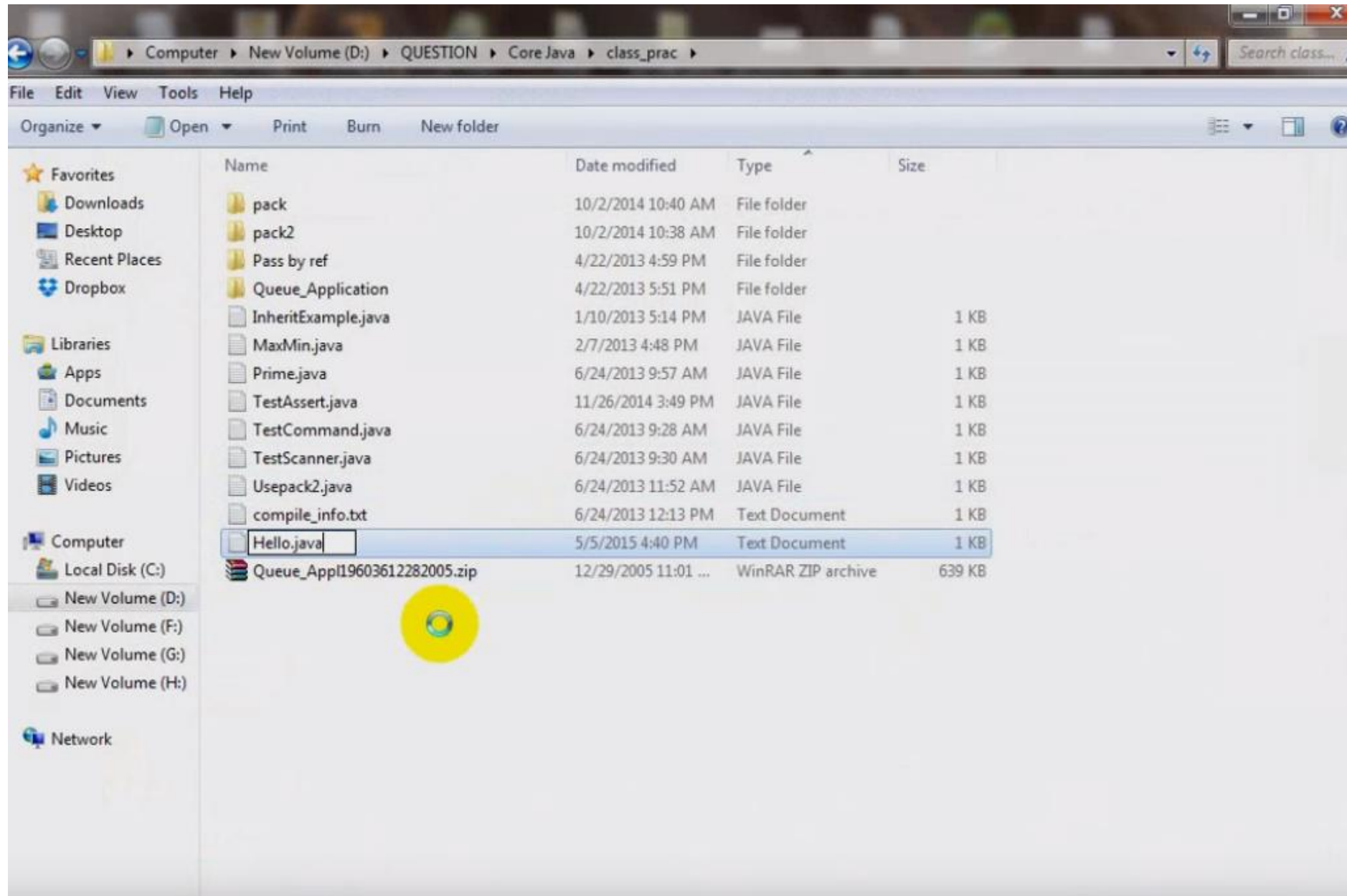


## Writing a Java Code

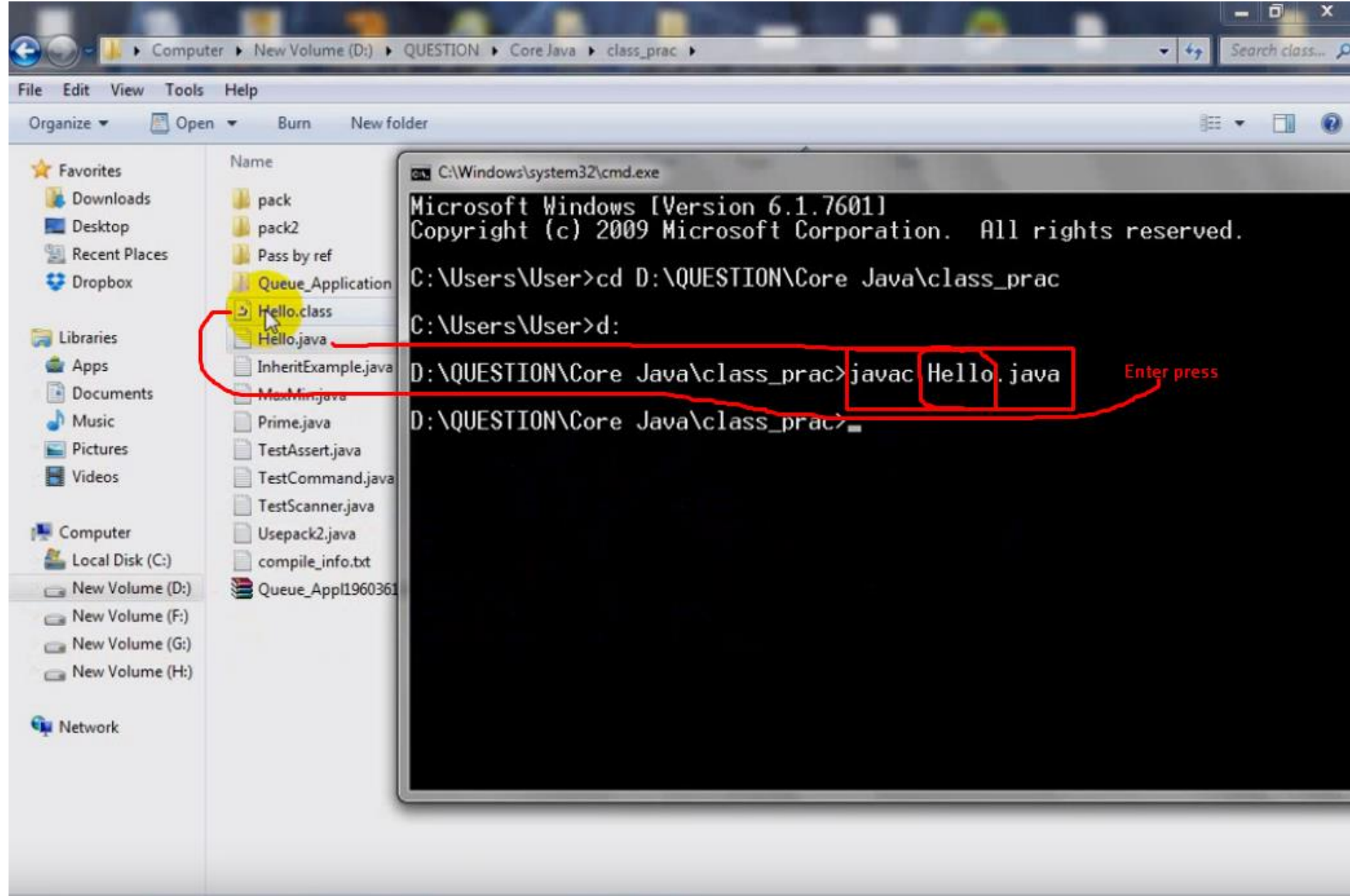
A screenshot of the Notepad++ text editor. The title bar shows the file path 'D:\QUESTION\Core Java\class\_prac\New Text Document.txt' and the application name 'Notepad++'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Encoding', 'Language', 'Settings', 'Macro', 'Run', 'TextFX', 'Plugins', 'Window', and '?'. The toolbar contains various icons for file operations and editing. The editor window shows a Java code snippet with line numbers 1 through 5 on the left. The code is: 1 class Hello{, 2 public static void main(String[] args){, 3 System.out.println("Hello Java Learners");, 4 }, 5 }. The word 'static' on line 2 is highlighted with a yellow circle. The code is color-coded: 'class' and 'public' are purple, 'static' is yellow, 'void' is blue, 'main' is black, 'String' is blue, 'args' is black, 'System.out.println' is black, and the string "Hello Java Learners" is in quotes. The closing braces '}' are red.

```
1 class Hello{
2 public static void main(String[] args){
3 System.out.println("Hello Java Learners");
4 }
5 }
```

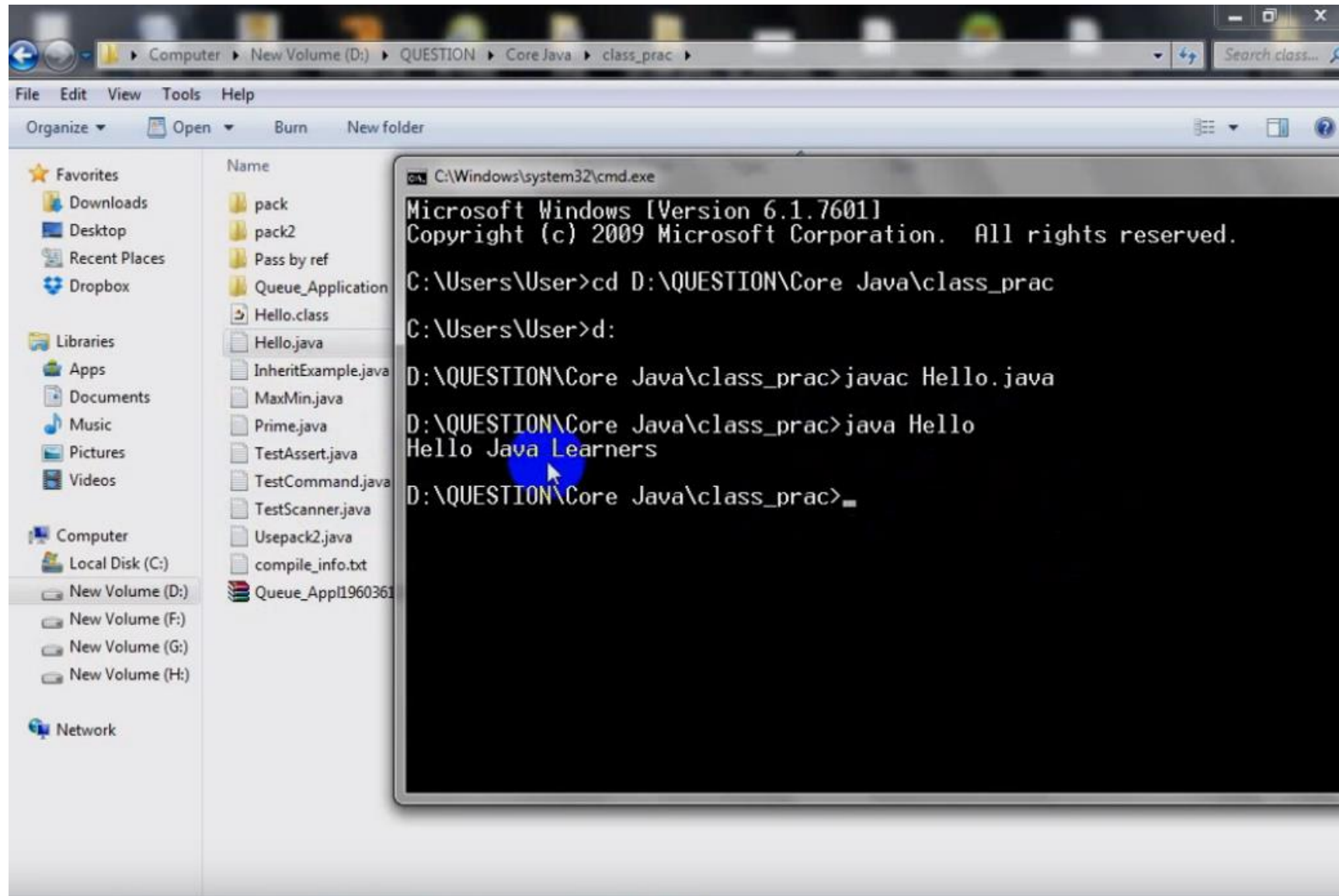
## Save a Java Code



## Compile a Java Code



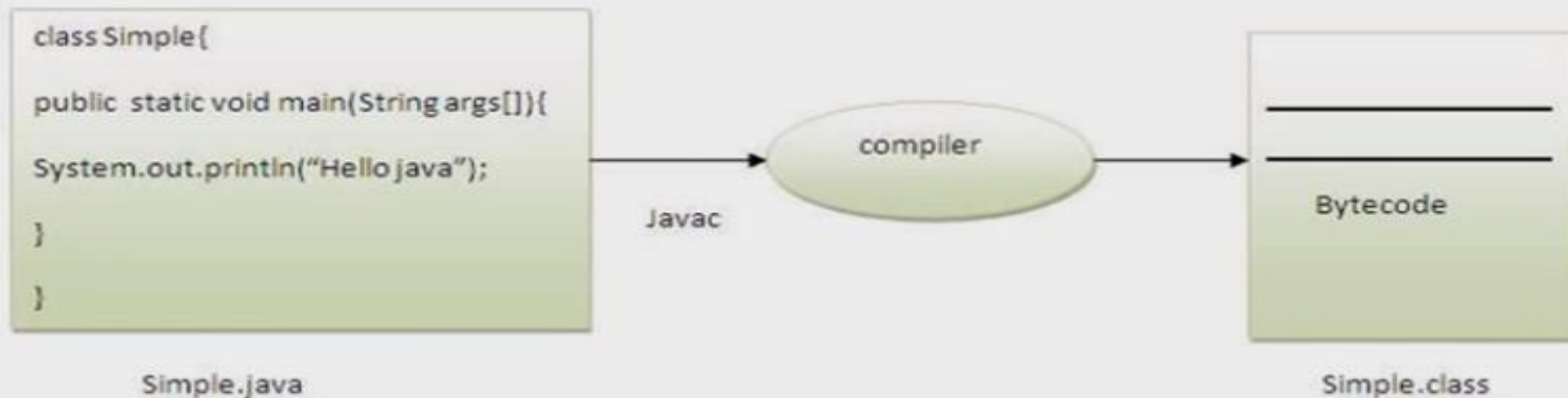
## Run a Java Code





# What happens at compile time?

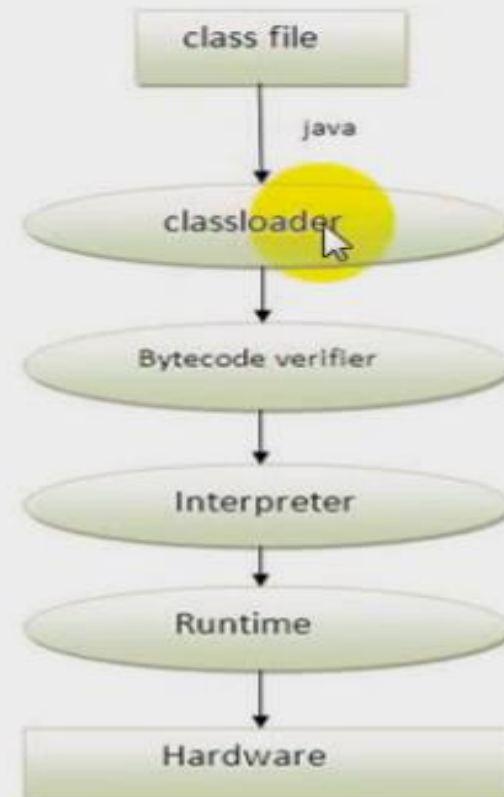
- At compile time, java file is compiled by Java Compiler (It does not interact with OS) and converts the java code into bytecode.



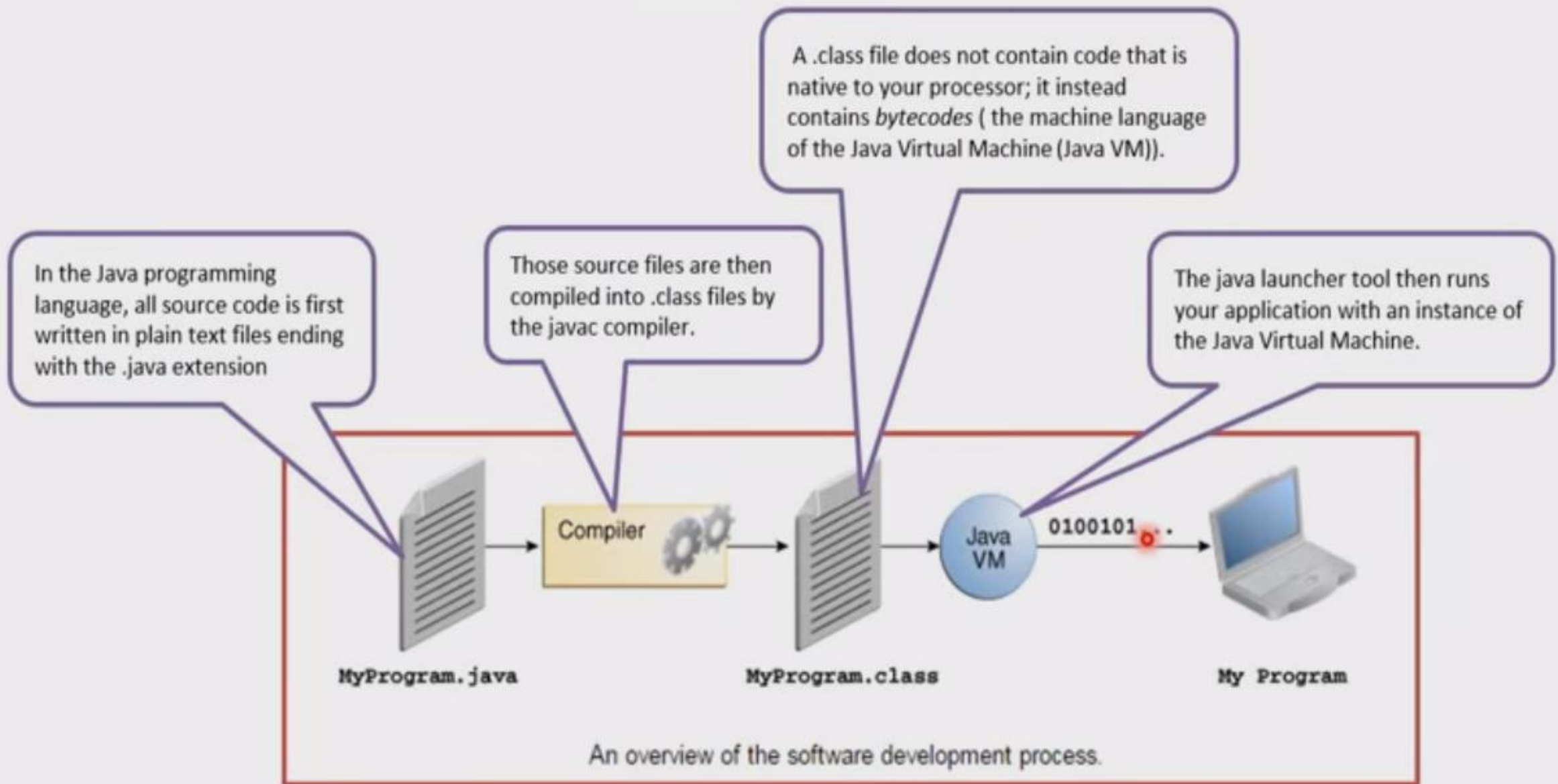
# What happens at runtime?

□ At runtime, following steps are performed:

- ▣ **Classloader:** is the subsystem of JVM that is used to load class files.
- ▣ **Bytecode Verifier:** checks the code fragments for illegal code that can violate access right to objects.
- ▣ **Interpreter:** read bytecode stream then execute the instructions.

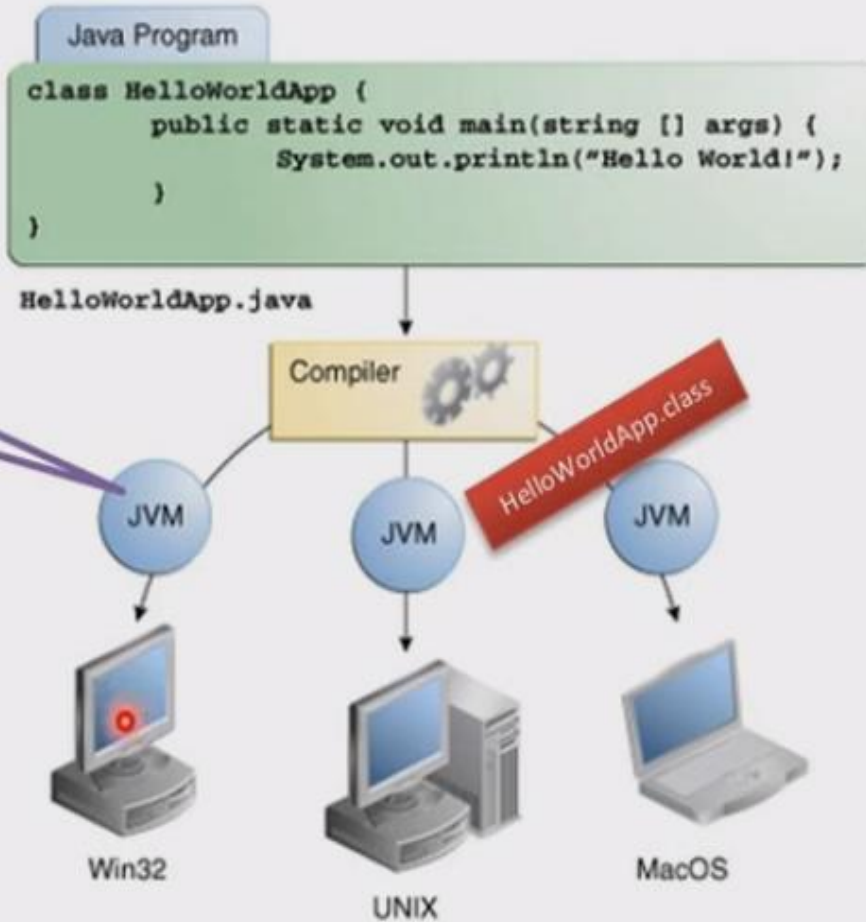


# The Java Programming Language



# The Java Programming Language

The Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS.



Through the Java VM, the same application is capable of running on multiple platforms.