



# Overview of Java Platform



# INTRODUCTION

## *Java Programming Language*



## LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Describe the evolution of Java language
- List different versions of Java
- Understand the Java development environment
- Create and execute a Simple Java program
- Use command line arguments
- Use Scanner Class





## CONCEPT

# *Brief history of Java*



## HISTORY OF JAVA

- Was originally developed by James Gosling at Sun Microsystems, Inc. in 1991
- Intended to let application developers "write once, run anywhere" (WORA)
- Derives its syntax from C and C++
- Originally designed for small, embedded systems in electronic appliances like set-top boxes
- Initially called as **GreenTalk**, later named **Oak** and finally renamed to **"Java"** in 1995



## EVOLUTION OF JAVA

Version	Year	Important New Features
1.0	1996	Language
1.1	1997	JDBC, Inner Classes, RMI
1.2	1998	Just In Time (JIT) compiler, Collections framework, Java Foundation Classes
1.3	2000	Enhancements
1.4	2004	Assertions
5.0	2004	Generics, “for each” Loop, varargs, autoboxing, metadata, enumerations, static import, Annotations
6	2006	JDBC 4.0, Console, Navigable Sets and Navigable Maps
7	2011	Strings in switch, Multiple Exception Handling, Try with Resources, underscore in literals, Diamond Syntax
8	2014	Lambda Expressions, Functional Interfaces, Date/Time changes, Stream Collection Types



## PROGRAMMING PLATFORM

- Java is not just a Language but a Programming Platform
- Java Platform has following components
  - Language with nice Features and pleasant syntax
  - High Quality Execution Environment that provides services like
    - Portability across operating systems
    - Automatic Garbage Collection
- Vast Library known as Java API containing lots of reusable code



## JAVA BUZZWORDS

- Simple : Partially modelled on C++, but greatly simplified considering the flaws of earlier languages
- Object Oriented:
  - Helps visualize the program using real-life objects
  - Provides modularity and reusability
- Distributed : Strong Networking capabilities integrated in java
- Robust : Easy Memory Management and early error checking and Exception handling





## JAVA BUZZWORDS

- Secure : Designed to prevent stack overrun, Memory corruption, unauthorized file access etc.
- Architecture Neutral/Portable: Java application developed in one platform can be used in other platforms with ease.
- High Performance
  - Uses Technology known as Just-in-time compilation
  - Java Hot Spot performance Engine includes compiler for optimizing frequently used code
- Multithreaded : Perform several tasks simultaneously within a program

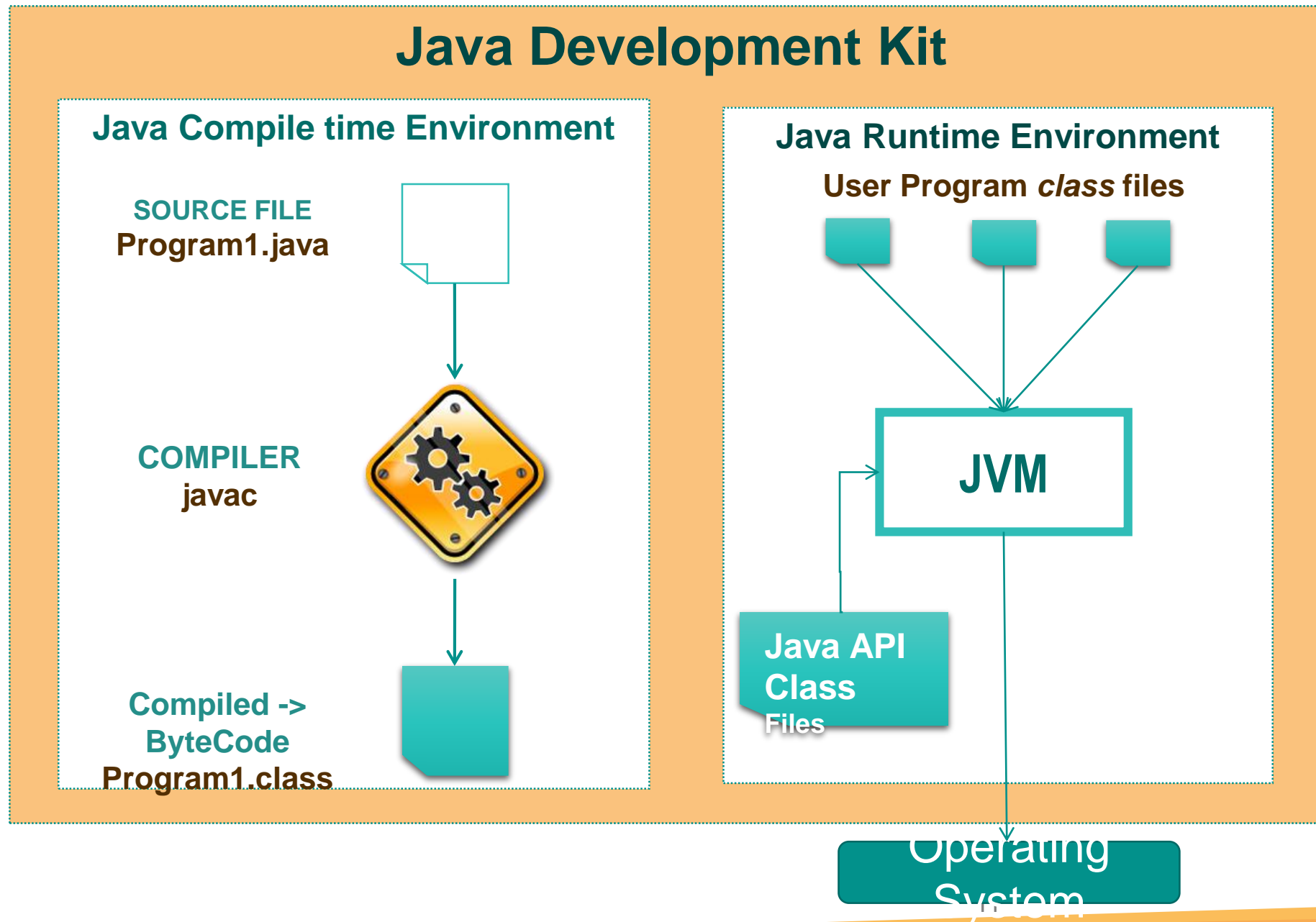


## CONCEPT

# *Java Development Environment*



## JAVA PLATFORM

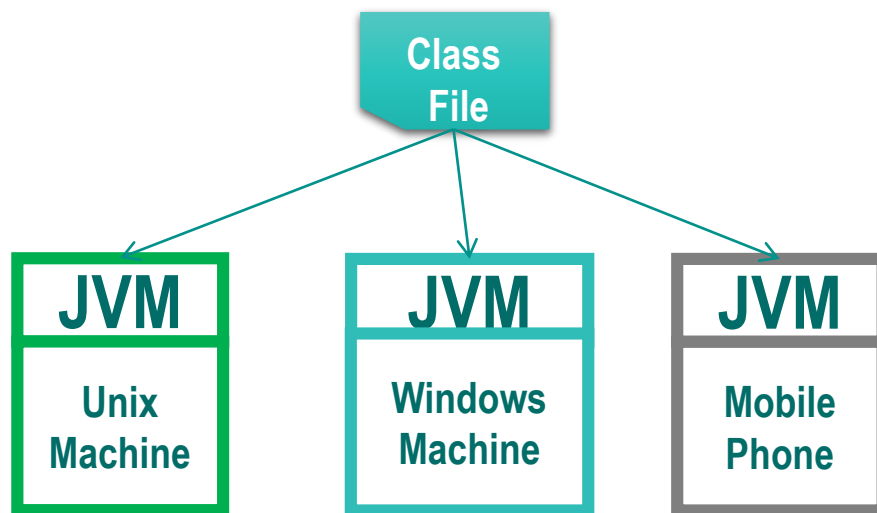


- Java compiler compiles the source code into a byte code
- The byte code will be in a file with extension .class
- JVM interprets the byte code into specific machine language of the underlying platform



## JVM – JAVA VIRTUAL MACHINE

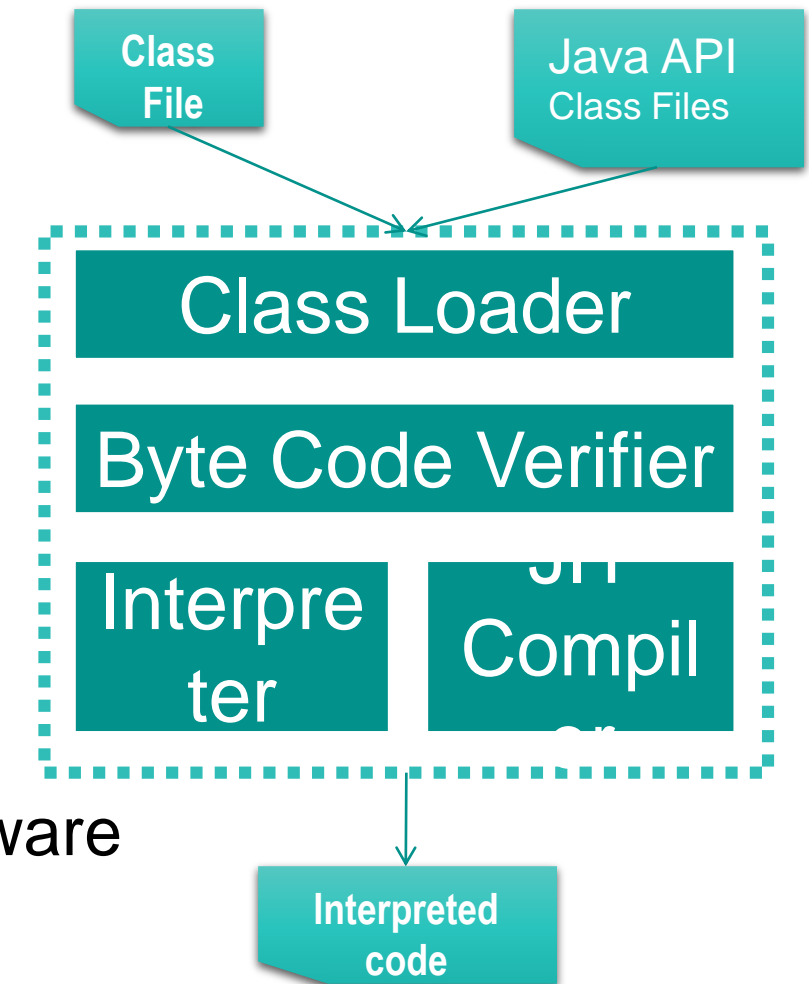
- JVM is Platform dependent
  - Different JVM's for different operating systems
- The Platform executes the JVM interpreted Machine language





## JVM INTERNALS

- Class Loader
  - Loads all the classes needed to execute a given class
- Byte Code Verifier
  - Verifies access violations, Illegal code, data conversions
- Interpreter
  - Executes bytecode and makes calls to the underlying hardware
- Just-In-Time (JIT) compiler
  - Part of JVM
  - JIT compiler is enabled by default, and is activated when a Java method is called
  - Compiles the bytecode of a method being called repeatedly into native platform code
  - Improves performance of the application





## JRE - JAVA RUNTIME ENVIRONMENT

- Intended for end users executing Java Applications
- Contains
  - JVM
  - Java API's
  - Java Plugins for Web Browsers
  - Java Web Start
- Does not contain development tools



## JDK - JAVA DEVELOPMENT KIT

- Software used by programmers to develop Java Applications
- Contains JRE and other tools like compiler, debugger
- Current version : Java SE 8 JDK
- Can be downloaded from Oracle Website

<https://www.oracle.com/downloads/index.html>



## JDK EDITIONS

- **Java Standard Edition (J2SE)**  
can be used to develop client-side standalone applications or applets.
- **Java Enterprise Edition (J2EE)**  
can be used to develop server-side applications such as Java servlets and Java Server Pages.
- **Java Micro Edition (J2ME).**  
can be used to develop applications for mobile devices such as cell phones.





**CONCEPT**

*Getting Started*



## SETTING UP JAVA

- To Execute java programs, path and classpath system variables need to be set
- **path** variable is used by the operating system to locate executable files
- The executable files of JDK are present in the folder where java is installed

Windows default path: C:\Program Files\Java\JDK<version>\bin

For windows, add the above path to the PATH environment variable

- **classpath** variable is used by compiler and JVM to locate the compiled class files  
( bytecodes)



## WRITING A SIMPLE JAVA PROGRAM

- Open any Text Editor like Notepad and create a class as below

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



## COMPILING A JAVA PROGRAM

- **Compiling with javac**
  - Open a command prompt/Terminal window
  - Change to the directory where the program is saved
  - Type **`javac HelloWorld.java`**
  - If you get the prompt without any message, the compilation was successful



## EXECUTING A JAVA PROGRAM

- **Executing with java**
  - Type **java HelloWorld** at the command prompt/Terminal
  - The result should be displayed



## CONCEPT

# *Command Line Arguments*



## COMMAND LINE ARGUMENTS

- Are used to pass information into a program when it is run.
- Accomplished by passing *command-line arguments* to **main( ) method**
- Arguments immediately follow the program's name on the command line during execution
- Stored as strings in a **String** array passed to the **args** parameter of **main( )**.
- The first argument is stored at **args[0]**, the second at **args[1]**, and so on



## COMMAND LINE ARGUMENTS

- The program given below displays the use of command-line arguments

```
class CommandLineDemo {  
    public static void main(String args[]) {  
        System.out.println("First argument " + args[0]);  
        System.out.println("Second argument " + args[1]);  
    }  
}
```

To execute, give the following command with the arguments:

```
java CommandLineDemo Bangalore Hyderabad
```





**CONCEPT**

# *Scanner Class Basics*



## SCANNER CLASS

- Used to read all types of numeric values, strings, and other types of data, from keyboard, file or another source.
- Defined in java.util package
- To use the Scanner class in the application, import the package as given below

```
import java.util.Scanner;
```

- ```
Scanner sc = new Scanner(System.in);  
String s = sc.next();  
int i = sc.nextInt();
```



## Scanner Demo

- This program accepts the student details from console and displays back

```
import java.util.Scanner;

class ScannerTest{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your rollno");
        int rollno= sc.nextInt();
        System.out.println("Enter your name");
        String name=sc.next();
        System.out.println("Enter your fee");
        double fee=sc.nextDouble();
        System.out.println("Rollno:" + rollno + " name:" + name + " fee:"+fee);
        sc.close();
    }
}
```



## QUIZ

The Java compiler converts the source code into \_\_\_\_\_

\_\_\_\_\_ is the component of JRE which make java programs platform independent and portable

User of a Java application needs to install \_\_\_\_\_ in the system





## QUIZ QUESTION

JVM is platform independent?

- ☐ TRUE
- ☐ FALSE





Which is of the following is NOT TRUE for JVM ?

- ☐ JVM reads Byte Code and generates Machine Code.
- ☐ JVM is a virtual Machine that acts as a intermediary between Java Application and Host Operating System
- ☐ JVM reads Source Code and generates Byte Code





## QUIZ QUESTION

J2SE is used to develop

- ☐ Web application
- ☐ Enterprise application
- ☐ Mobile application
- ☐ Stand alone application





Scanner class is defined in \_\_\_\_\_ package.

- ☐ java.lang
- ☐ java.util
- ☐ java.io
- ☐ None of the above







## References

- Refer following demo videos on EduNxt
  - M1L1L1\_Brief\_history\_of\_Java
  - M1L1L2\_Getting\_started\_with\_Java\_Language
  - M1L1L3\_Getting\_started\_with\_Java\_Language – Demo





## SUMMARY

# *Introduction to Java Programming Language*



## SUMMARY

In this lesson, you've learned to:

- Describe brief history of Java
- Describe Java Development Environment
- Compile and Execute a simple java program
- Use Command line arguments
- Use Scanner class

