Unit 1 : Introduction of Java

Java

- Java is a **programming language** and a platform.
- Java is a high level, robust, object-oriented and secure programming language.
- Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995.
- ***James Gosling*** is known as the father of Java.
- Before Java, its name was ***Oak***. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Types of Java Applications

1) Standalone Application
2) Web Application
3) Enterprise Application
4) Mobile Application

History of Java :

- The team initiated this project to develop a language for digital devices such as set-top boxes, television, etc.
- Originally C++ was considered to be used in the project, but the idea was rejected for several reasons(For instance C++ required more memory).
- The history of Java starts with the Green Team.
- **James Gosling, Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.
- Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called **"Greentalk"** by James Gosling, and the file extension was .gt.
- After that, it was called **Oak** and was developed as a part of the Green project. Oak is a symbol of strength and chosen as a national tree of many countries like the 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 Programming named "Java"?

- 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.

- Java is an island in Indonesia where the first coffee was produced (called Java coffee). It is a kind of espresso bean. Java name was chosen by James Gosling while having a cup of coffee nearby his office.

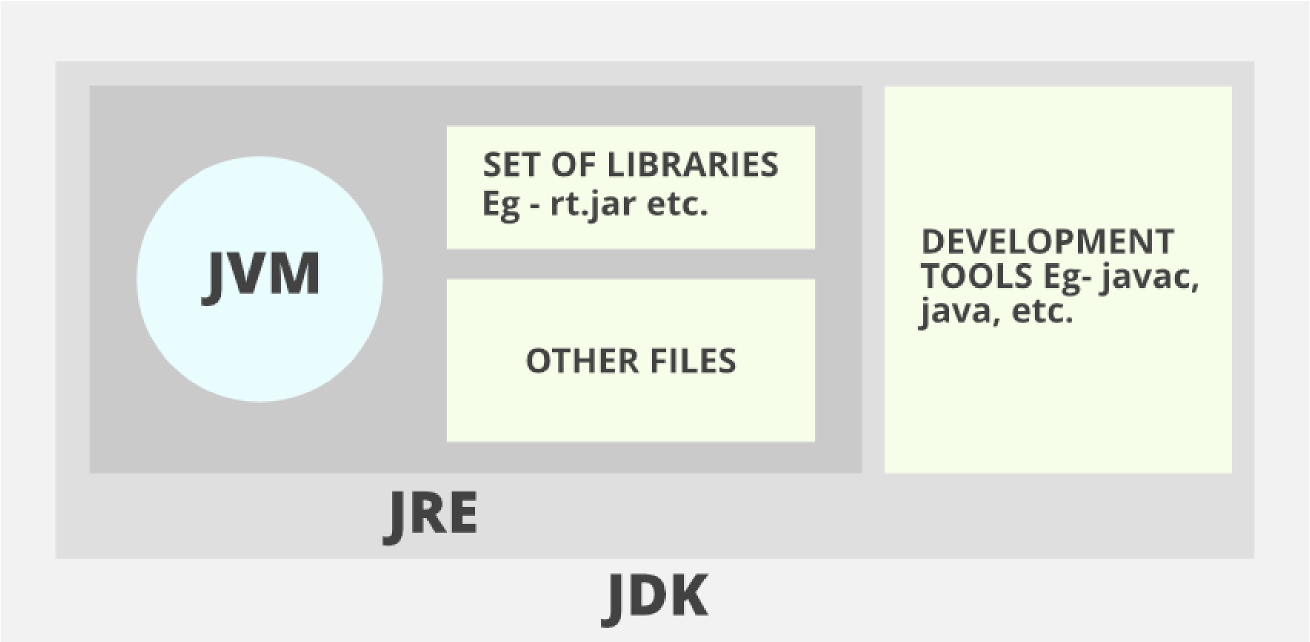
- Initially developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.

- JDK 1.0 was released on January 23, 1996. After the first release of Java, there have been many additional features added to the language. Now Java is being used in Windows applications, Web applications, enterprise applications, mobile applications, cards, etc. Each new version adds new features in Java.

Java versions : below is the list of java versions have been released till now:

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan 1996)
3. JDK 1.1 (19th Feb 1997)
4. J2SE 1.2 (8th Dec 1998)
5. J2SE 1.3 (8th May 2000)
6. J2SE 1.4 (6th Feb 2002)
7. J2SE 5.0 (30th Sep 2004)
8. Java SE 6 (11th Dec 2006)
9. Java SE 7 (28th July 2011)
10. Java SE 8 (18th Mar 2014)
11. Java SE 9 (21st Sep 2017)
12. Java SE 10 (20th Mar 2018)
13. Java SE 11 (September 2018)
14. Java SE 12 (March 2019)
15. Java SE 13 (September 2019)
16. Java SE 14 (Mar 2020)
17. Java SE 15 (September 2020)
18. Java SE 16 (Mar 2021)
19. Java SE 17 (September 2021)
20. Java SE 18 (to be released by March 2022)

Java Application	Java Applet
Applications are just like a Java program that can be executed independently without using the web browser.	Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution.
The application program requires a main() method for its execution.	The applet does not require the main() method for its execution instead init() method is required.
The “javac” command is used to compile application programs, which are then executed using the “java” command.	Applet programs are compiled with the “javac” command and run using either the “appletviewer” command or the web browser.
Java application programs have full access to the local file system and network.	Applets don’t have local disk and network access.
Applications can execute the programs from the local system.	Applets cannot execute programs from the local machine.
It supports the reading and writing of files on the local computer.	It does not support the reading and writing of files on the local computer.



JVM

SET OF LIBRARIES
Eg - rt.jar etc.

OTHER FILES

DEVELOPMENT TOOLS Eg- javac, java, etc.

JRE

JDK

JDK	JRE	JVM
The full form of JDK is Java Development Kit.	The full form of JRE is Java Runtime Environment.	The full form of JVM is Java Virtual Machine.
JDK is a software development kit to develop applications in Java.	It is a software bundle which provides Java class libraries with necessary components to run Java code.	JVM executes Java byte code and provides an environment for executing it.
JDK is platform dependent.	JRE is also platform dependent.	JVM is highly platform dependent.
It contains tools for developing, debugging, and monitoring java code.	It contains class libraries and other supporting files that JVM requires to execute the program.	Software development tools are not included in JVM.
It is the superset of JRE	It is the subset of JDK.	JVM is a subset of JRE.
The JDK enables developers to create Java programs that can be executed and run by the JRE and JVM.	The JRE is the part of Java that creates the JVM.	It is the Java platform component that executes source code.
JDK comes with the installer.	JRE only contain environment to execute source code.	JVM bundled in both software JDK and JRE.

Procedural Oriented Programming	Object-Oriented Programming
In procedural programming, the program is divided into small parts called ***functions***.	In object-oriented programming, the program is divided into small parts called ***objects***.
Procedural programming follows a ***top-down approach***.	Object-oriented programming follows a ***bottom-up approach***.
There is no access specifier in procedural programming.	Object-oriented programming has access specifiers like private, public, protected, etc.
Adding new data and functions is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way of hiding data so it is ***less secure***.	Object-oriented programming provides data hiding so it is ***more secure***.
In procedural programming, overloading is not possible.	Overloading is possible in object-oriented programming.
In procedural programming, there is no concept of data hiding and inheritance.	In object-oriented programming, the concept of data hiding and inheritance is used.
Procedural programming is used for designing medium-sized programs.	Object-oriented programming is used for designing large and complex programs.
Code reusability absent in procedural programming,	Code reusability present in object-oriented programming.
Examples: C, FORTRAN, Pascal, Basic, etc.	**Examples:** C++, Java, Python, C#, etc.

Compiling:

- Compiling a Java program means taking the programmer-readable text in your program file (also called source code) and converting it to bytecodes, which are platform-independent instructions for the Java VM.

- The Java compiler is invoked at the command line on Unix and DOS shell operating systems as follows:

```
javac Simple.java
```

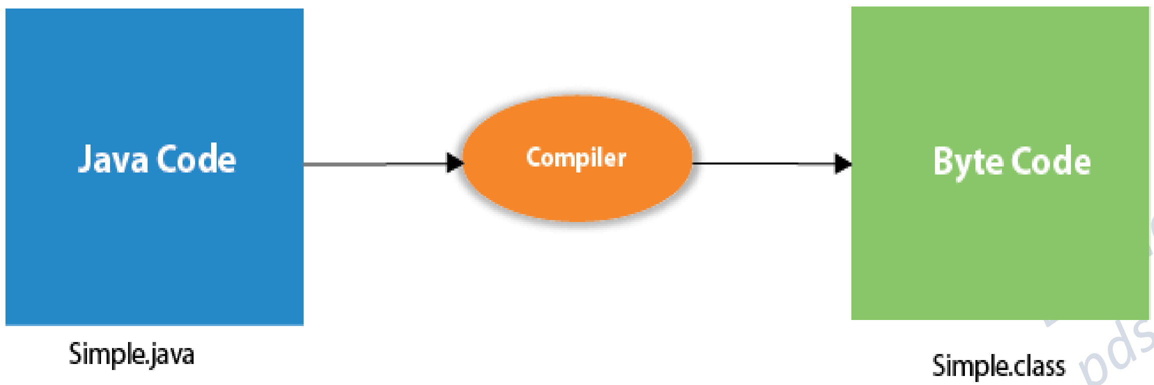


Fig. Compile Time

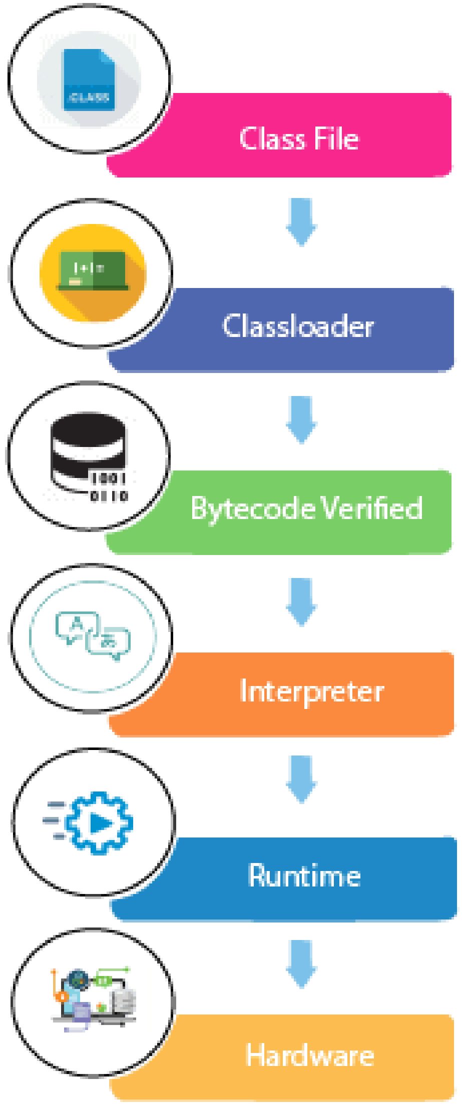
Run a Java program :

❖ The class files generated by the compiler are independent of the machine or the OS, which allows them to be run on any system.

- **Classloader:** It 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 rights to objects.
- **Interpreter:** Read bytecode stream then execute the instructions.

Command to run the Java program:

```
java  Simple
```



Class File

Classloader

Bytecode Verified

Interpreter

Runtime

Hardware

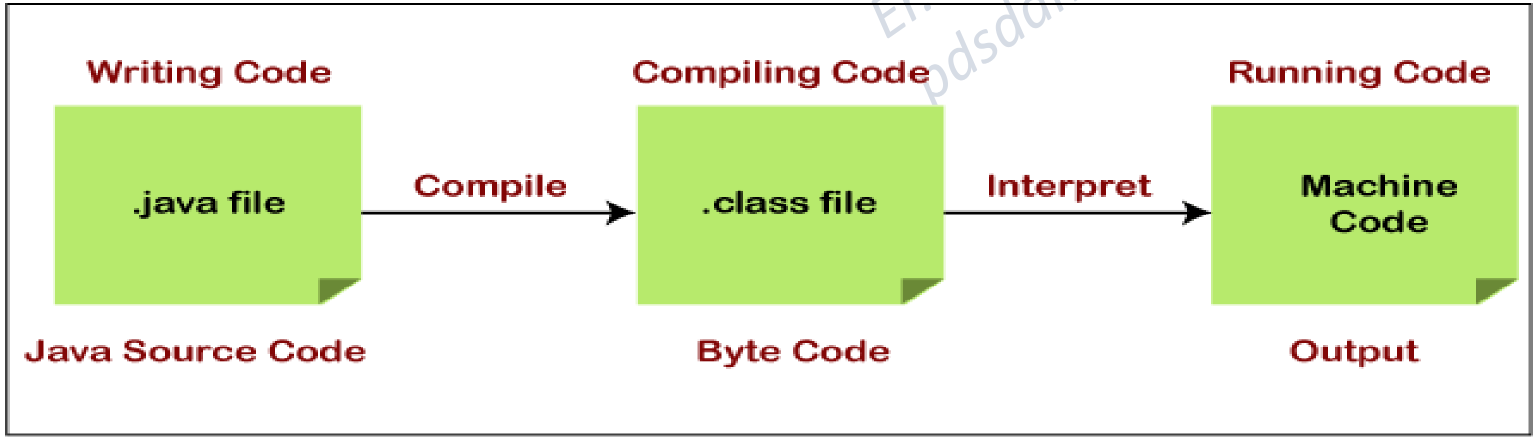
Interpreting and running Java Program

- Interpreting and running a Java program is process of **executing the Java Program**. In this process the **Java VM** is invoked and it takes the byte code and interprets it. In this process the byte code is converted to platform-dependent machine codes so that your computer can understand and run the program.

- Once your program successfully compiles into Java bytecodes, you can interpret and run your applications on any Java VM or JVM enabled web browser as applet.

How does the Java interpreter work?

- To convert the byte code into machine code, we deploy the .class file on the JVM. The JVM converts that code into machine code using the Java interpreter. The JVM uses the interpreter at runtime, after that it execute the code on the host machine.



Interpreter	Compiler
It translates the code instruction by instruction.	It translates the entire program at once.
Its execution is slower.	Its execution is faster.
Its compile time is less.	It takes more time to compile the code.
It does not generate the intermediate object code.	It generates the intermediate object code.
It compiles the program until an error is found.	All the errors show once at the end of the compilation.
Python, PHP, Ruby, and Perl use an interpreter.	Java, C++, Scala, and C uses a compiler.

Errors :

- An error or exception refers to an interruption in the execution of code due to which we cannot attain the expected outcome to the end-users.

1. Compile-time errors

- Compile-time errors are the errors that occurred when we write the wrong syntax.
- If we write the wrong syntax or semantics of any programming language, then the compile-time errors will be thrown by the compiler.
- The compiler will not allow to run the program until all the errors are removed from the program.

The compile-time errors can be:

a. Syntax errors

- When the programmer does not follow the syntax of any programming language, then the compiler will throw the syntax error.

b. Semantic errors

- The semantic errors exist when the statements are not meaningful to the compiler.

2. Runtime errors :

The runtime errors are the errors that occur during the execution and after compilation.

Compile-time	Runtime
The compile-time errors are the errors which are produced at the compile-time, and they are detected by the compiler.	The runtime errors are the errors which are not generated by the compiler and produce an unpredictable result at the execution time.
In this case, the compiler prevents the code from execution if it detects an error in the program.	In this case, the compiler does not detect the error, so it cannot prevent the code from the execution.
It contains the syntax and semantic errors such as missing semicolon at the end of the statement.	It contains the errors such as division by zero, determining the square root of a negative number.

Setting up your Computer for Java Environment:

1. User variables :
JAVA_HOME : path of jdk (C:\Program Files\Java\jdk version)

2. System variables :

- Select Path variable
- Click on edit
- Add path of both JDK and JRE

e.g. C:\Program Files\Java\jdk version\bin
C:\Program Files\Java\jre version\bin