Day_1_OOPJ_Sanket_Shalukar

Monday, August 25, 2025 10:18 AM

Topics:

Foundation of Java
Basic Concepts
Flow Execution
JVM, JDK and JRE
Programs : Java

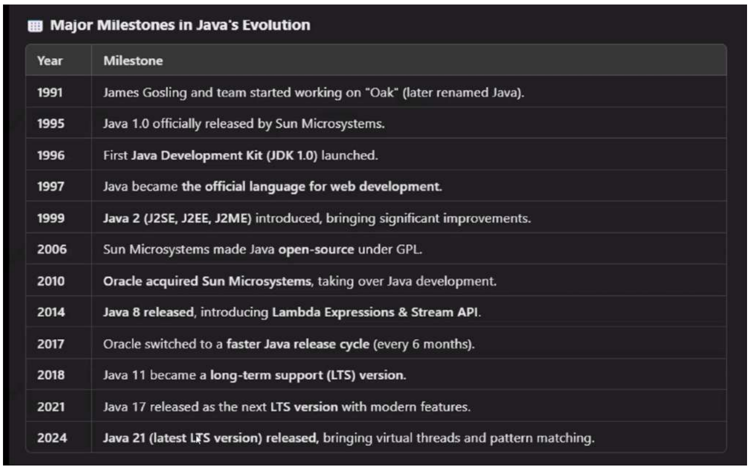
Java :

Java is high level language, Object oriented, platform independent, it is programing language.
It is developed by James Gosling
He was working for Sun Microsystem, 1995.
Java is a Technology as well as framework and it is also a programing language.

Introduction : Java

- Java is a high-level, object-oriented, and platform-independent programming language.
- Developed by James Gosling at Sun Microsystems and released in 1995.
- Java follows the principle of "Write Once, Run Anywhere" (WORA), meaning programs can run on any platform with a Java Virtual Machine (JVM).

Important for Interview :



Major Milestones in Java's Evolution

Year	Milestone
1991	James Gosling and team started working on "Oak" (later renamed Java).
1995	Java 1.0 officially released by Sun Microsystems.
1996	First **Java Development Kit (JDK 1.0)** launched.
1997	Java became **the official language for web development.**
1999	**Java 2 (J2SE, J2EE, J2ME)** introduced, bringing significant improvements.
2006	Sun Microsystems made Java **open-source** under GPL.
2010	**Oracle acquired Sun Microsystems**, taking over Java development.
2014	**Java 8 released**, introducing **Lambda Expressions & Stream API**.
2017	Oracle switched to a **faster Java release cycle** (every 6 months).
2018	Java 11 became a **long-term support (LTS) version.**
2021	Java 17 released as the next **LTS version** with modern features.
2024	**Java 21 (latest LTS version) released,** bringing virtual threads and pattern matching.

Programing Language :

- Support Classes, Object, Inheritance encapsulation, abstraction, and polymorphism.
- Platform independent: WORA=> Write once and run anywhere
- Strongly type
- Rich standard library
- Robust and secure
- Multithreaded programing.

Technology :

- A Broder concept involving tools and platform and methodologies

Framework :

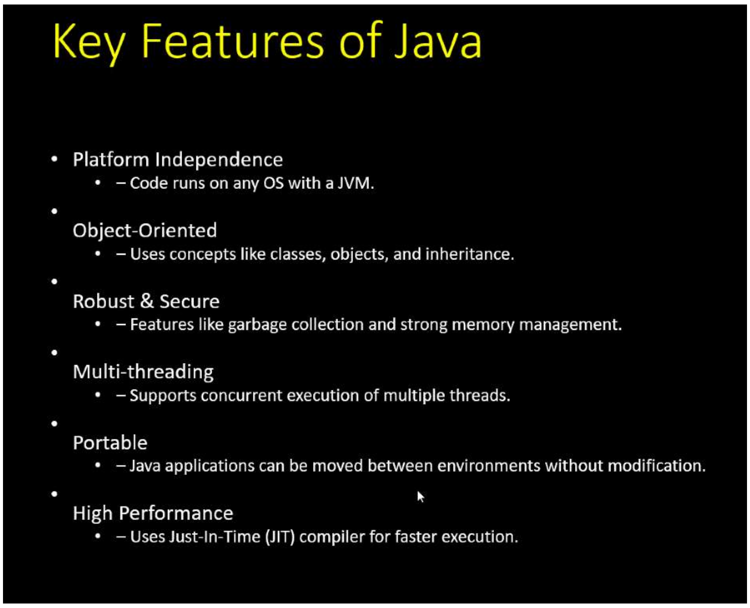
- A structured collection of code that simplifies development

- Java EE / Jakarta EE
- Spring framework
- Hibernate ORM

lang
net
util

Platform :

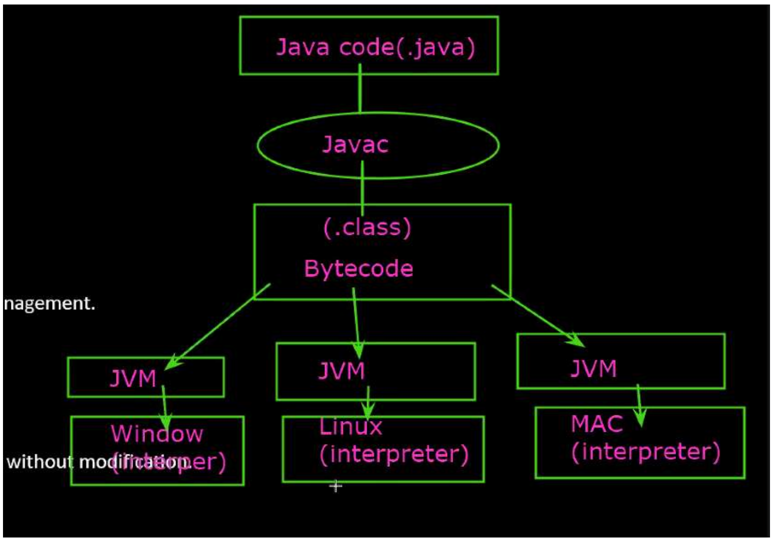
- JVM : Executes java program
- JDK: Tools for development of program
- JRE: Runtime environment



Key Features of Java

- Platform Independence
  - – Code runs on any OS with a JVM.
- Object-Oriented
  - – Uses concepts like classes, objects, and inheritance.
- Robust & Secure
  - – Features like garbage collection and strong memory management.
- Multi-threading
  - – Supports concurrent execution of multiple threads.
- Portable
  - – Java applications can be moved between environments without modification.
- High Performance
  - – Uses Just-In-Time (JIT) compiler for faster execution.

JVM
Window
(interpreter)
JVM
Linux
(interpreter)
JVM
MAC
(interpreter)



Step 1: Download Java JDK

1. Open a web browser.
2. Go to the official Oracle website:
   https://www.oracle.com/java/technologies/javase-downloads.html
3. Click on the latest **Java SE Development Kit (JDK)** download link.
4. On the JDK downloads page, find the **Windows Installer** (.exe) under the section "x64 Installer".
5. Click on the download link to start downloading the `.exe` file.

Step 2: Run the Installer

6. Once the download is complete, go to your **Downloads** folder.
7. Double-click on the downloaded `.exe` file to launch the installer.
8. If prompted by User Account Control, click **Yes** to allow the installer to run.

Step 3: Install the JDK

9. In the installer window, click **Next**.
10. Choose the installation location or leave it as default.
11. Click **Next** again to begin installation.
12. Wait for the installation process to complete.
13. When the installation finishes, click **Close**.

Step 4: Set Environment Variables

14. Open the **Start Menu** and search for **Environment Variables**.
15. Click on **"Edit the system environment variables"**.
16. In the **System Properties** window, click on the **Environment Variables** button.

Set JAVA_HOME:

17. Under **System Variables**, click **New**.
18. In **Variable Name**, type `JAVA_HOME`.
19. In **Variable Value**, enter the path to your JDK folder (e.g., `C:\Program Files\Java\jdk-21` or the version you installed).
20. Click **OK**.

Update Path Variable:

21. In the same **System Variables** section, find and select the **Path** variable.
22. Click **Edit**.
23. Click **New**, and type:

```
%JAVA_HOME%\bin
```

24. Click **OK** on all open windows to apply changes.

Step 5: Verify Installation

17. Open **Command Prompt** (search `cmd` in Start Menu).
18. Type:
    ```
    java -version
    ```
    and press Enter.
19. Then type:
    ```
    javac -version
    ```
    and press Enter.
20. If Java is installed correctly, it will display the installed version numbers.

Editions:

- Java SE: Standard Edition
- core programming and libraries
- -Desktop applications
- Java EE: Enterprise Edition
- Enterprise & Server side programming
- Web applications, APIs (Application Programming In
- Java ME: Micro Edition
- Embedded and Mobile applications
- IOT, Sensor, gamming applications
- JavaFX
- GUI and rich client app

Java ME: Micro Edition

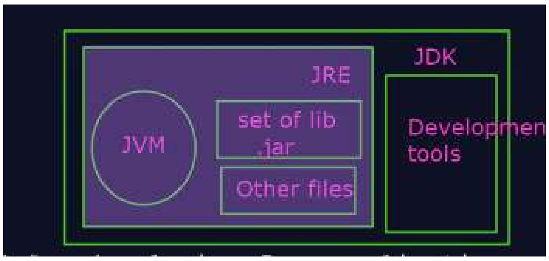
- Embedded and Mobile applications
- IOT, Sensor, gamming applications

JavaFX

- GUI and rich client applications
- Visual desktop applications, Dashboards

JDK : Java Development Kit

- It is a software development environment used for developing Java applications.
- JDK is a superset of JRE (Java Runtime Environment), which includes tools for
- development and debugging.

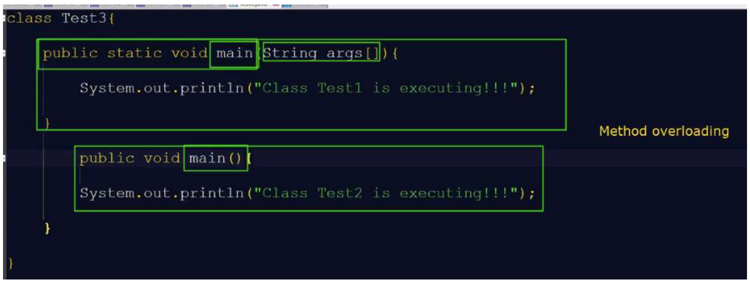


JRE: Java Runtime Environment

- JRE is a subset of JDK
- It provides the environment to run java program, but it doesnot contain any
- development tool.

Components :

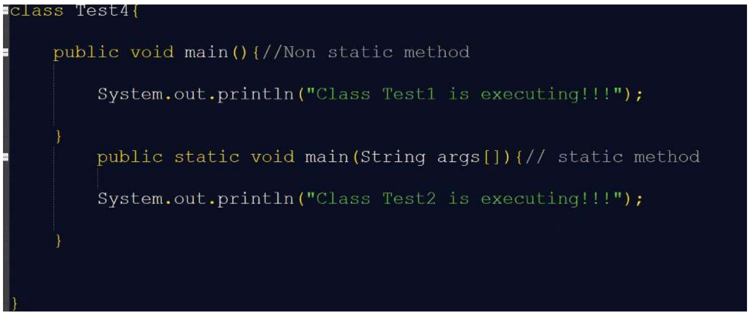
- JVM : Java virtual maachine: Execute the compiled bytecode
- core libaries: java.lang, java.net, java.sql, etc



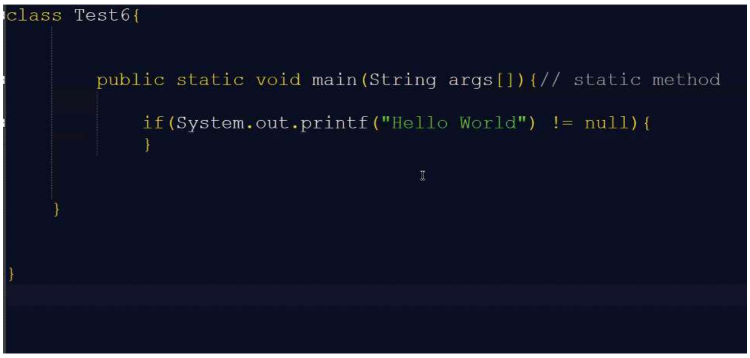
```
class Test3{
    public static void main(String args[]){
        System.out.println("Class Test1 is executing!!!");
    }
      public void main(){
      System.out.println("Class Test2 is executing!!!");
    }
}
```

Method overloading

METHOD OVERLOADING

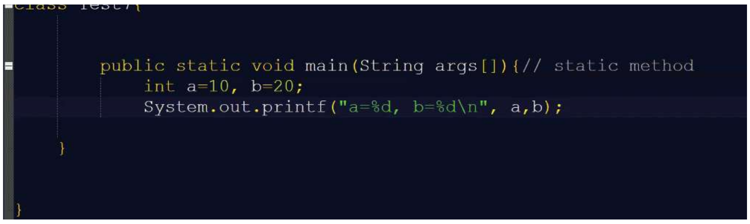


```
class Test4{
    public void main(){//Non static method
        System.out.println("Class Test1 is executing!!!");
    }
        public static void main(String args[]){// static method
        System.out.println("Class Test2 is executing!!!");
    }
}
```



```
class Test6{
        public static void main(String args[]){// static method
            if(System.out.printf("Hello World") != null){
            }
    }
}
```

```
class Test7{
```

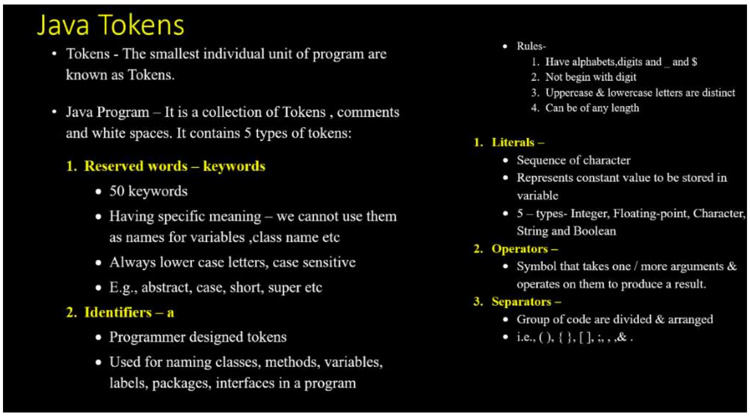


```
class Test7{

        public static void main(String args[]){// static method
            int a=10, b=20;
            System.out.printf("a=%d, b=%d\n", a,b);

    }

}
```

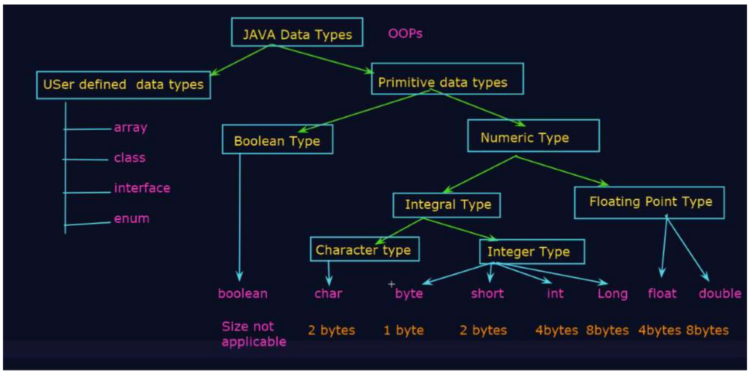


Java Tokens

- Tokens - The smallest individual unit of program are known as Tokens.
- Java Program – It is a collection of Tokens , comments and white spaces. It contains 5 types of tokens:

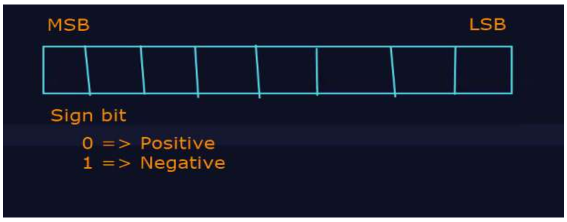
1. **Reserved words – keywords**
    - 50 keywords
    - Having specific meaning – we cannot use them as names for variables ,class name etc
    - Always lower case letters, case sensitive
    - E.g., abstract, case, short, super etc
2. **Identifiers – a**
    - Programmer designed tokens
    - Used for naming classes, methods, variables, labels, packages, interfaces in a program
    - Rules-
        1. Have alphabets,digits and _ and $
        2. Not begin with digit
        3. Uppercase & lowercase letters are distinct
        4. Can be of any length

1. **Literals –**
    - Sequence of character
    - Represents constant value to be stored in variable
    - 5 – types- Integer, Floating-point, Character, String and Boolean
2. **Operators –**
    - Symbol that takes one / more arguments & operates on them to produce a result.
3. **Separators –**
    - Group of code are divided & arranged
    - i.e., ( ), { }, [ ], ;, , ,& .



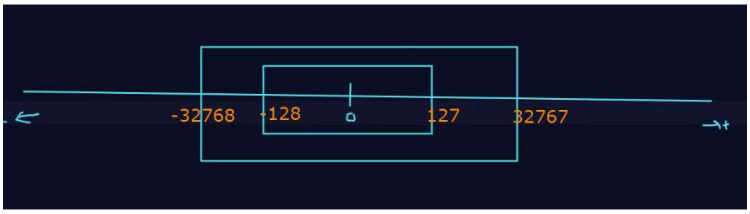
Data Types:

1. **Byte -**
    - Byte – 1 byte = 8 bit.
    - Range : -128 to 127
    - Byte b = -130;



2. **Short -**

- 2 bytes = 16bits
- Range : -32768 to 32767



3. **Int :**

- 4 bytes + 32 bits
- Range : -2^31 to 2^31 –1

- 16 bit processor : 2bytes
- 32 bit processor : 4 bytes

4. **Long :**

- 8 bytes = 64 bits
- Range : -2^63 to (2^63) -1

5. **Flot :**

- 4 bytes
- Range +/-3.4E38
- Precision: 5/6 decimal point

6. **Double :**

- 8 bytes
- Range +/- 1.7E308
- Precision 14 /15 decimal places (double precision)

7. **Boolean :**

- Size- Na
- Range: NA (true or false)
- Case sensitive : true/false
- Boolean b= 0; //ERR: Incompatible
- C/C++ = false =0 true false = 1
- Java = true / false

8. **char data type:**

- C/C++ = char = 1bytes (support ASCII characters)
- Java: Size: 2 bytes (supports unicode character) unicode = global characters.
- '\u0000' : Representation of unicode characters