Object Oriented Programming with Java (OOPJ)

Session 1: Basics of Java

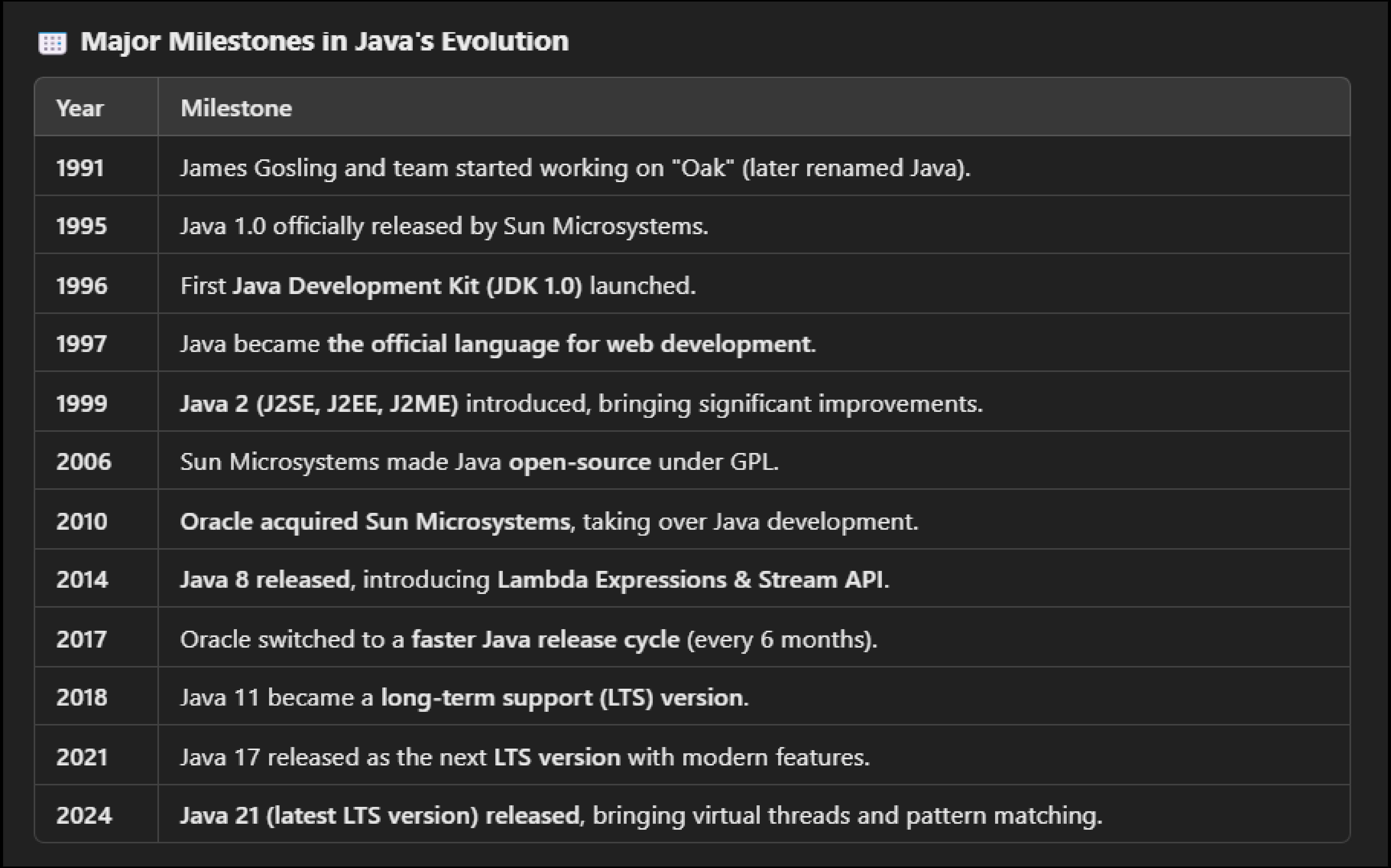
Kiran Waghmare

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).

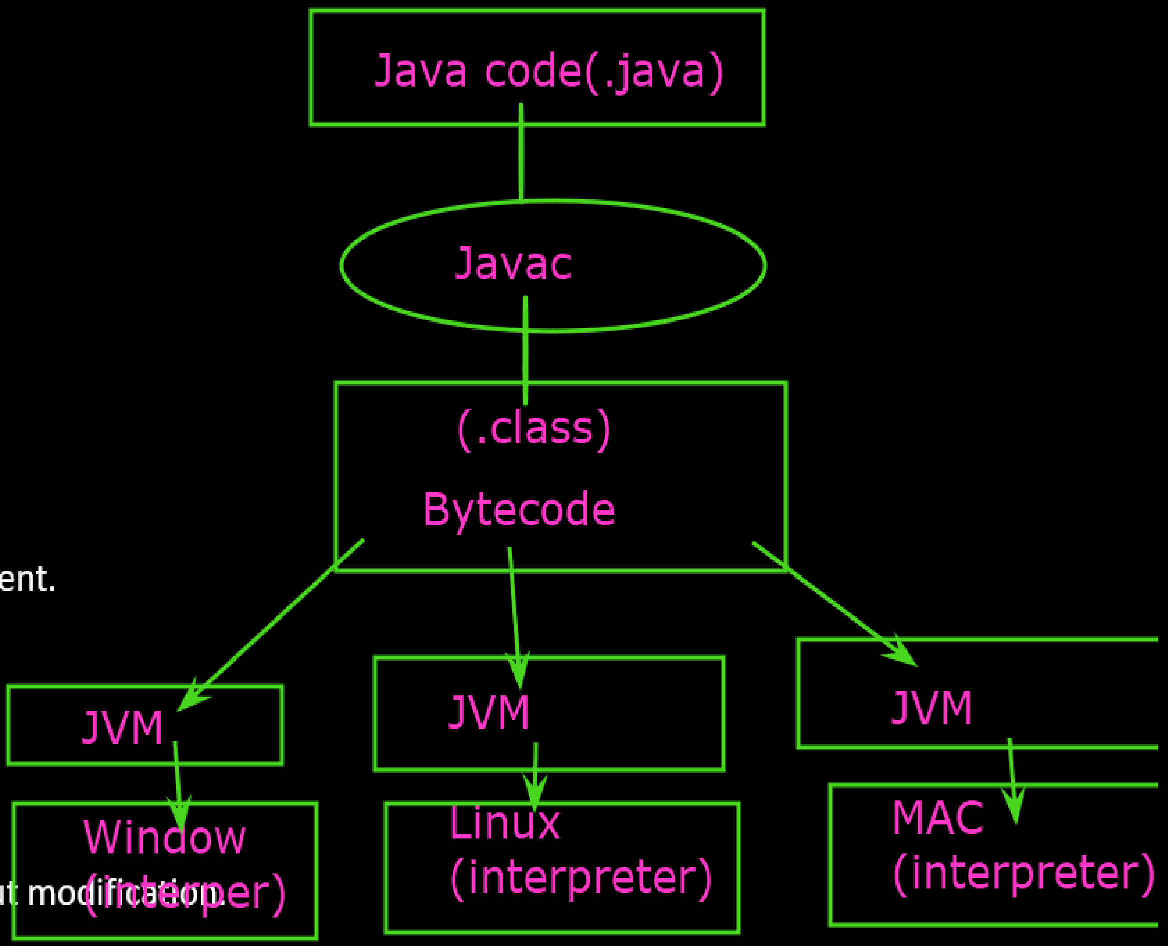


📅 Major Milestones in Java's Evolution

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.

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.



Java code(.java)

Javac

(.class)

Bytecode

JVM

JVM

JVM

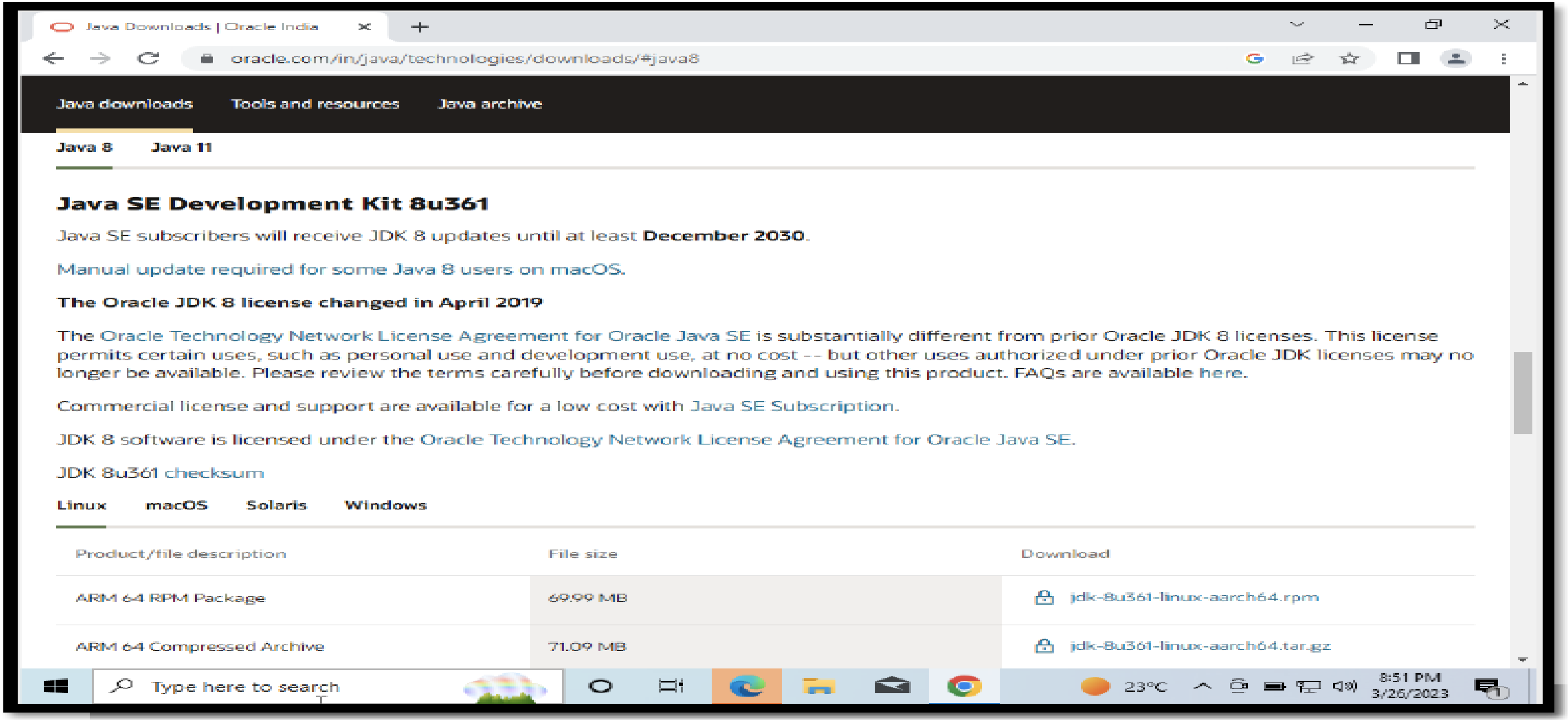
Window (interper)

Linux (interpreter)

MAC (interpreter)

JDK Software Installation Documentation

1. Visit https://www.oracle.com/in/java/technologies/downloads/ link. Scroll page and select Java 8 tab.



Java SE Development Kit 8u361

Java SE subscribers will receive JDK 8 updates until at least **December 2030**.

Manual update required for some Java 8 users on macOS.

The Oracle JDK 8 license changed in April 2019

The Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK 8 licenses. This license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. FAQs are available here.

Commercial license and support are available for a low cost with Java SE Subscription.

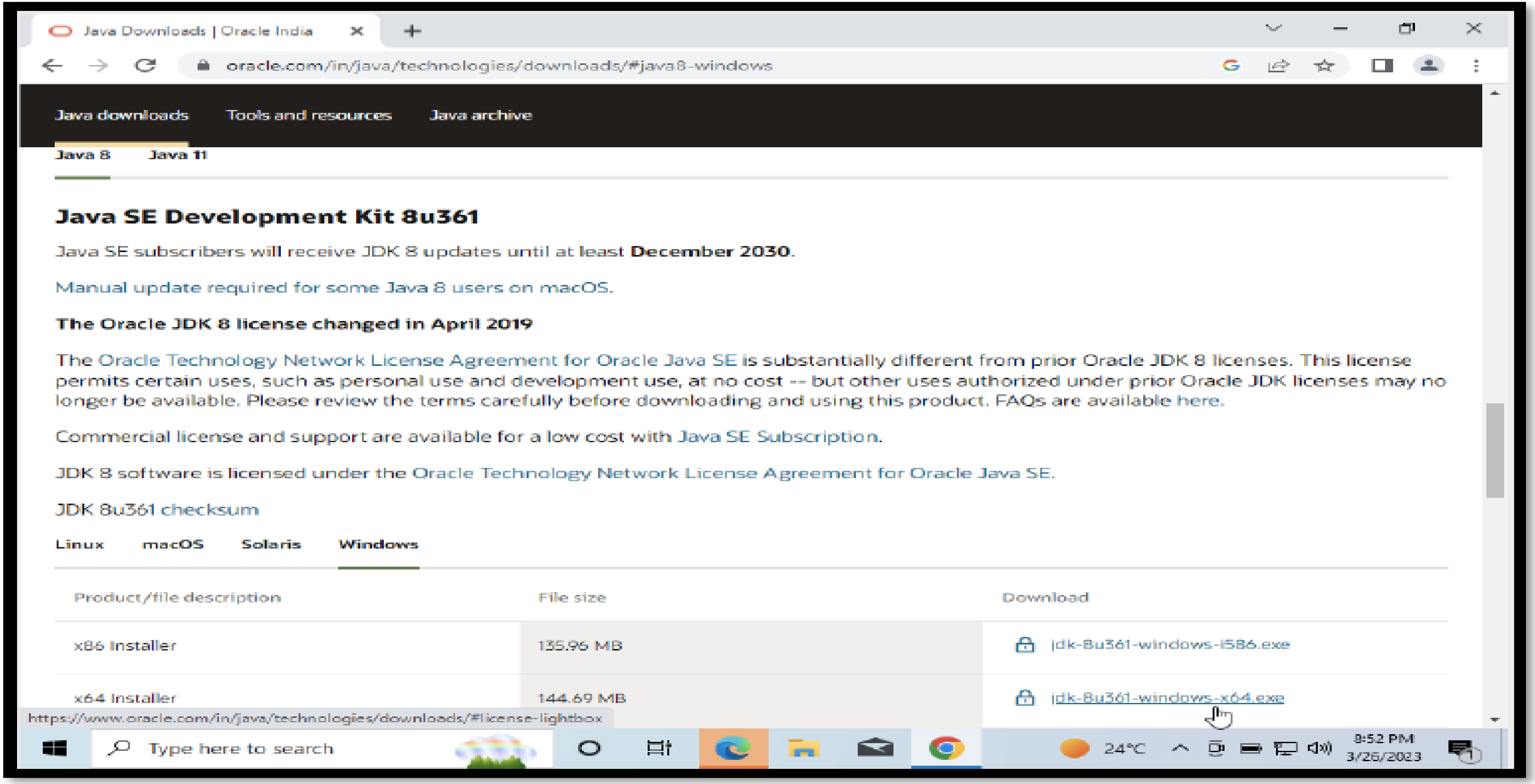
JDK 8 software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE.

JDK 8u361 checksum

Linux **macOS** **Solaris** **Windows**

Product/file description	File size	Download
ARM 64 RPM Package	69.99 MB	jdk-8u361-linux-aarch64.rpm
ARM 64 Compressed Archive	71.09 MB	jdk-8u361-linux-aarch64.tar.gz

2. Select Windows tab and click on jdk-8u361-windows-x64.exe link to download JDK.



Java Downloads | Oracle India
oracle.com/in/java/technologies/downloads/#java8-windows
Java downloads
Tools and resources
Java archive
Java 8
Java 11

Java SE Development Kit 8u361

Java SE subscribers will receive JDK 8 updates until at least **December 2030**.

Manual update required for some Java 8 users on macOS.

The Oracle JDK 8 license changed in April 2019

The Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK 8 licenses. This license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. FAQs are available here.

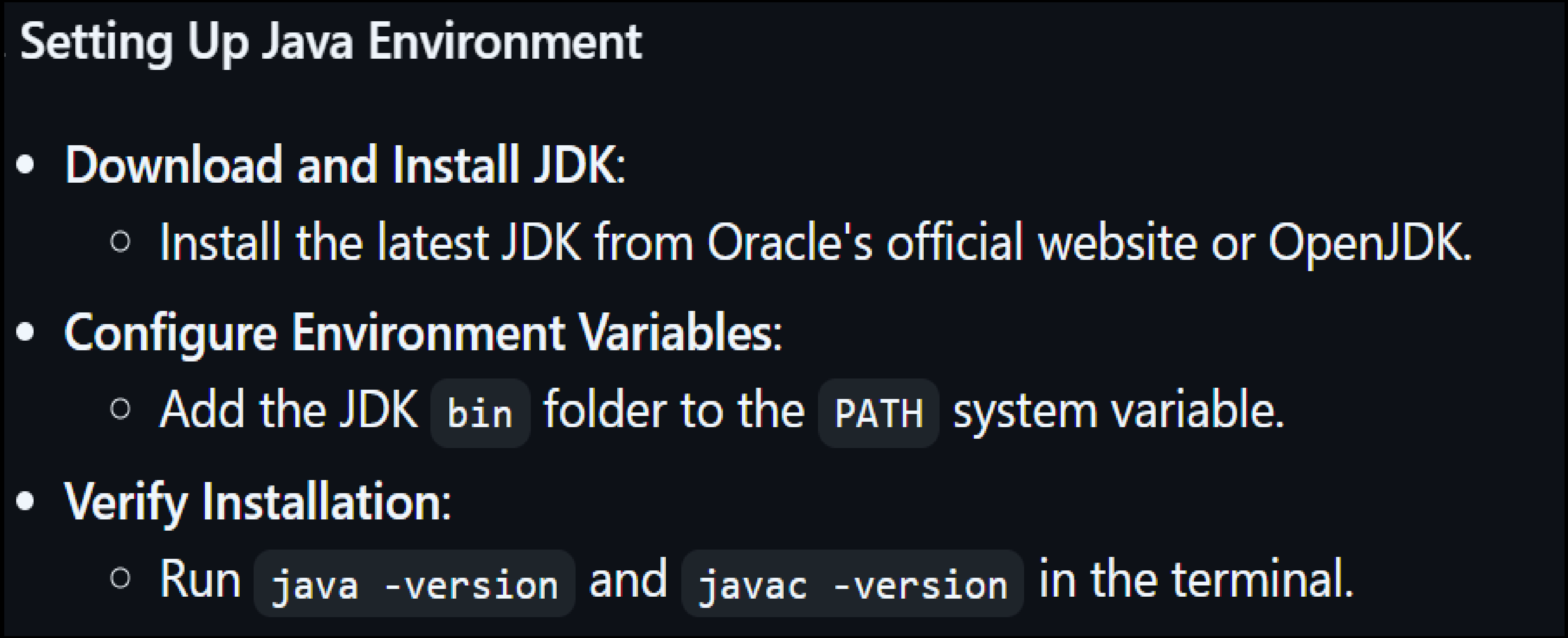
Commercial license and support are available for a low cost with Java SE Subscription.

JDK 8 software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE.

JDK 8u361 checksum

Product/file description	File size	Download
x86 Installer	135.96 MB	jdk-8u361-windows-i586.exe
x64 Installer	144.69 MB	jdk-8u361-windows-x64.exe

https://www.oracle.com/in/java/technologies/downloads/#license-lightbox
Type here to search
24°C
8:52 PM
3/26/2023



Setting Up Java Environment

- **Download and Install JDK:**
  - Install the latest JDK from Oracle's official website or OpenJDK.
- **Configure Environment Variables:**
  - Add the JDK `bin` folder to the `PATH` system variable.
- **Verify Installation:**
  - Run `java -version` and `javac -version` in the terminal.

Introduction to Java & Features

- 1. Introduction to Java
  - Java is a **high-level, object-oriented, platform-independent** programming language.
  - Developed by **James Gosling at Sun Microsystems** in 1995 (later acquired by Oracle).
  - Designed **for portability, security, and robustness**.
  - Java programs are compiled into bytecode, which runs on the Java Virtual Machine (JVM).
  - Used in desktop applications, mobile apps (Android), web applications, enterprise solutions, and cloud computing.

Features of Java

- A. Platform Independence (Write Once, Run Anywhere - WORA)
  - Java code is compiled into bytecode (.class file).
  - The JVM interprets bytecode, making Java platform-independent.
  - Can run on Windows, Linux, Mac, etc., without modification.

- B. Object-Oriented Programming (OOP)
  - Java is based on OOP principles:
    - **Encapsulation** (Data hiding through access modifiers).
    - **Abstraction** (Hiding implementation details).
    - **Inheritance** (Reusing code via class hierarchy).
    - **Polymorphism** (Method overloading & overriding).

Features of Java

- C. Simple & Familiar
    - Java is easy to learn for programmers familiar with C and C++.
    - No need to manage pointers or memory allocation manually.
    - Provides automatic garbage collection.
- D. Secure
    - No direct access to pointers, reducing memory leaks and security vulnerabilities.
    - Java has a bytecode verifier that checks for illegal operations.
    - Supports encryption and secure communication via APIs.
- E. Robust & Reliable
    - Strong memory management with automatic garbage collection.
    - Exception handling mechanism (try-catch-finally) to manage runtime errors.
    - No direct memory manipulation (e.g., pointer arithmetic is not allowed).

Features of Java

- F. Multithreading Support
  - Java supports multithreading, allowing multiple tasks to run concurrently.
  - Threads can be created using Thread class or Runnable interface.
  - Synchronization prevents data inconsistency in multi-threaded programs.
- G. High Performance
  - Uses Just-In-Time (JIT) Compiler to convert bytecode into native machine code at runtime.
  - Optimization techniques like HotSpot Compiler improve performance.
  - Slower than C/C++, but performance is optimized for real-world applications.

Features of Java

- H. Distributed Computing & Networking Support
  - Java supports network programming via built-in APIs.
  - Can develop socket programming, web applications, and RMI (Remote Method Invocation) applications.
  - Java applications can interact with databases, servers, and cloud services.
- I. Dynamic & Extensible
  - Java supports dynamic memory allocation and loading of classes at runtime.
  - Allows developers to extend existing applications using APIs, libraries, and frameworks.
  - Uses Reflection API to inspect and modify classes at runtime.
- J. Backward Compatibility
  - Java ensures that older Java programs still work on newer versions of the language.
  - New versions of Java introduce enhancements without breaking old code.

Java Editions & Their Uses

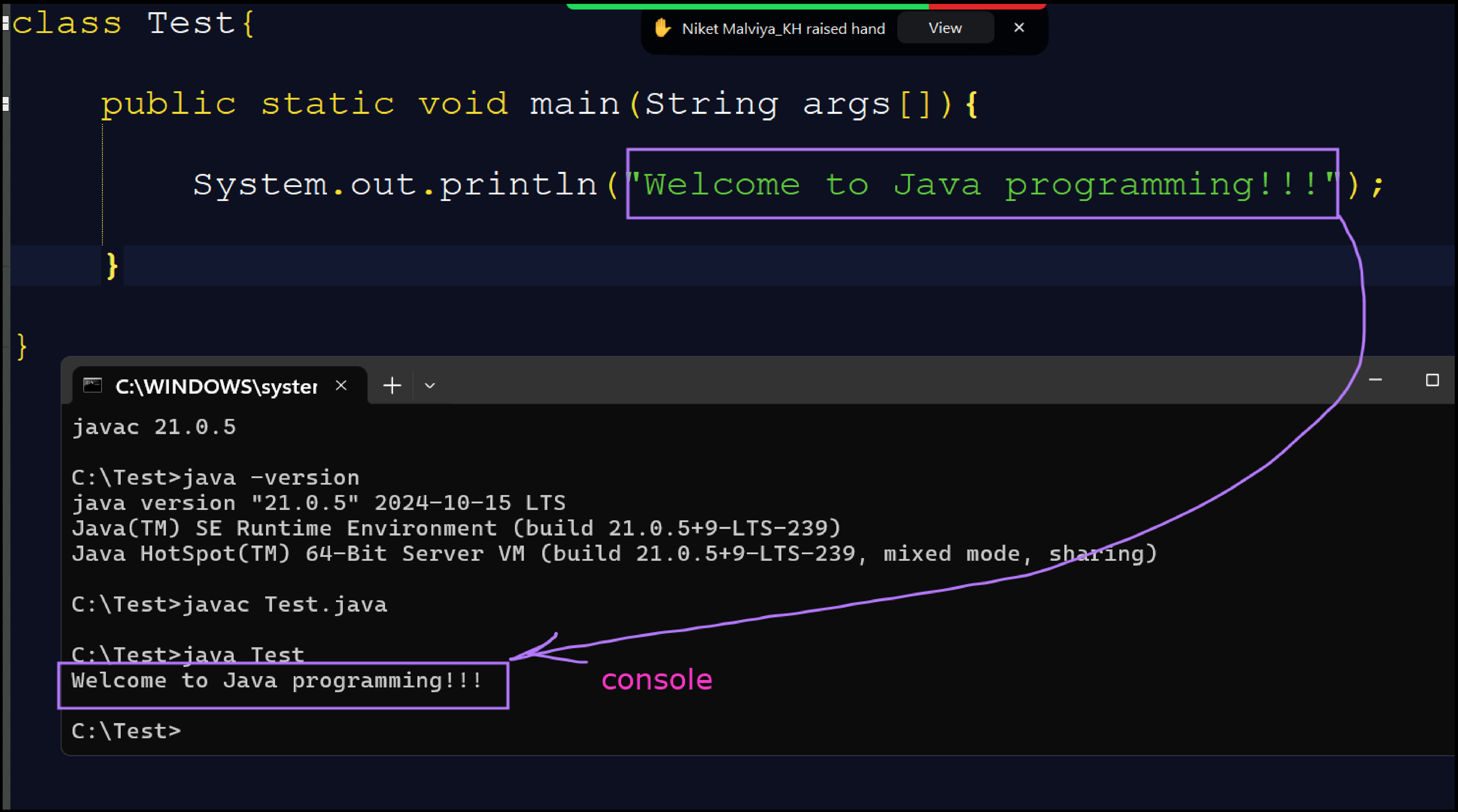
- Java SE (Standard Edition) – Core Java, desktop applications, utilities.
- Java EE (Enterprise Edition) – Web applications, enterprise solutions.
- Java ME (Micro Edition) – Embedded systems, mobile applications.
- JavaFX – GUI development.

Java Development Kit (JDK) & Java Runtime Environment (JRE)

- JDK (Java Development Kit)
  - Includes **JVM, compiler (javac), and development tools**.
  - Required for developing Java applications.
- JRE (Java Runtime Environment)
  - Includes **JVM and standard libraries** to run Java applications.
  - No compiler, used for running Java programs **(not for development)**.
- JVM (Java Virtual Machine)
  - Converts bytecode to **machine-specific code**.
  - Provides features like **memory management, garbage collection, and security**.

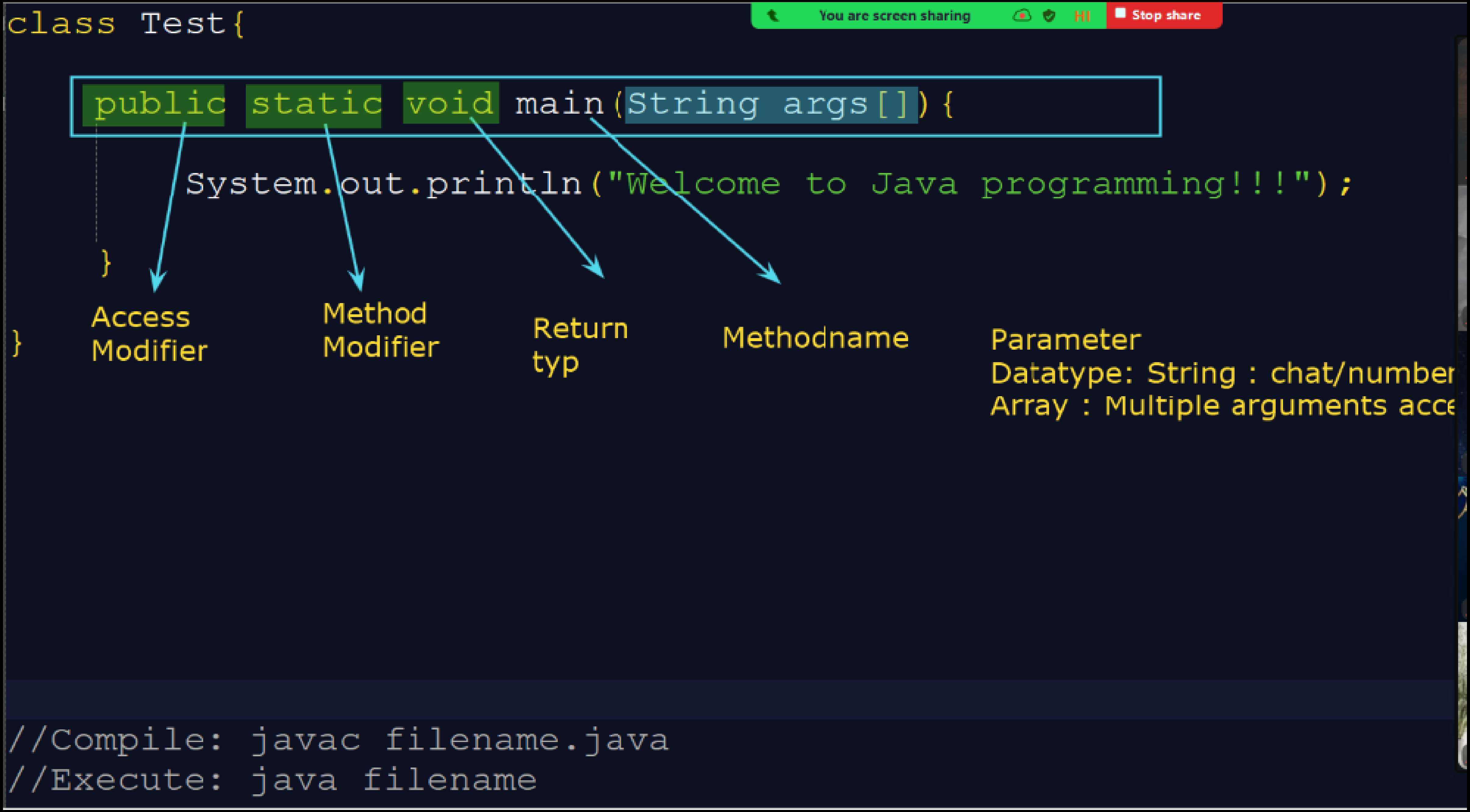
The requirement for Java Hello World Example

- For executing any java program, you need to
  - Install the JDK if you don't have installed it, download the JDK and install it.
  - Set path of the jdk/bin directory
  - Create the java program
  - Compile and run the java program

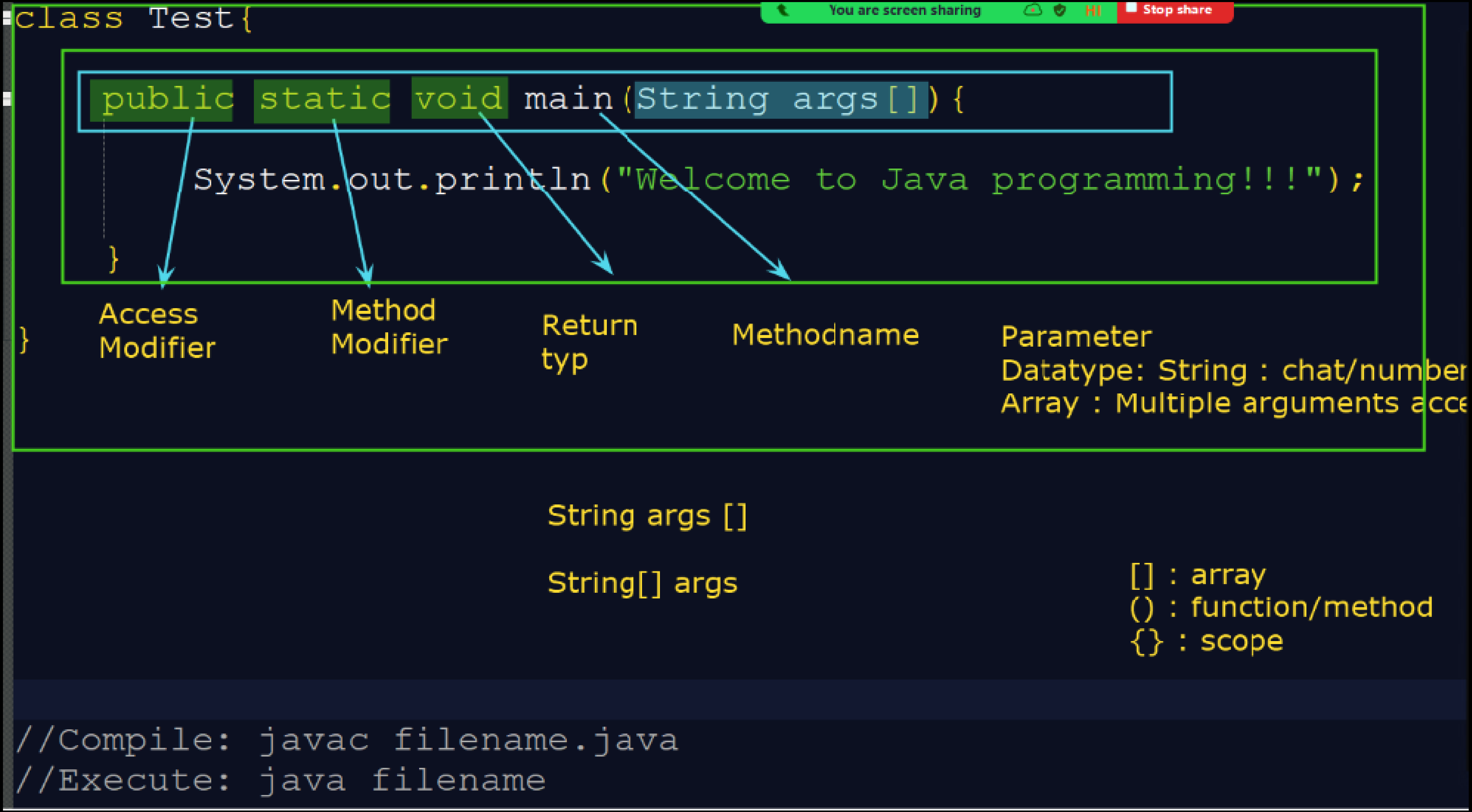


```
class Test{

    public static void main(String args[]){

        System.out.println("Welcome to Java programming!!!");

    }

}
```

```
javac 21.0.5

C:\Test>java -version
java version "21.0.5" 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Test>javac Test.java

C:\Test>java Test
Welcome to Java programming!!!

C:\Test>
```



```
class Test{

    public static void main(String args[]){

        System.out.println("Welcome to Java programming!!!");

    }

}

//Compile: javac filename.java
//Execute: java filename
```



```
class Test{
    public static void main(String args[]){
        System.out.println("Welcome to Java programming!!!");
    }
}
```

Access Modifier

Method Modifier

Return typ

Methodname

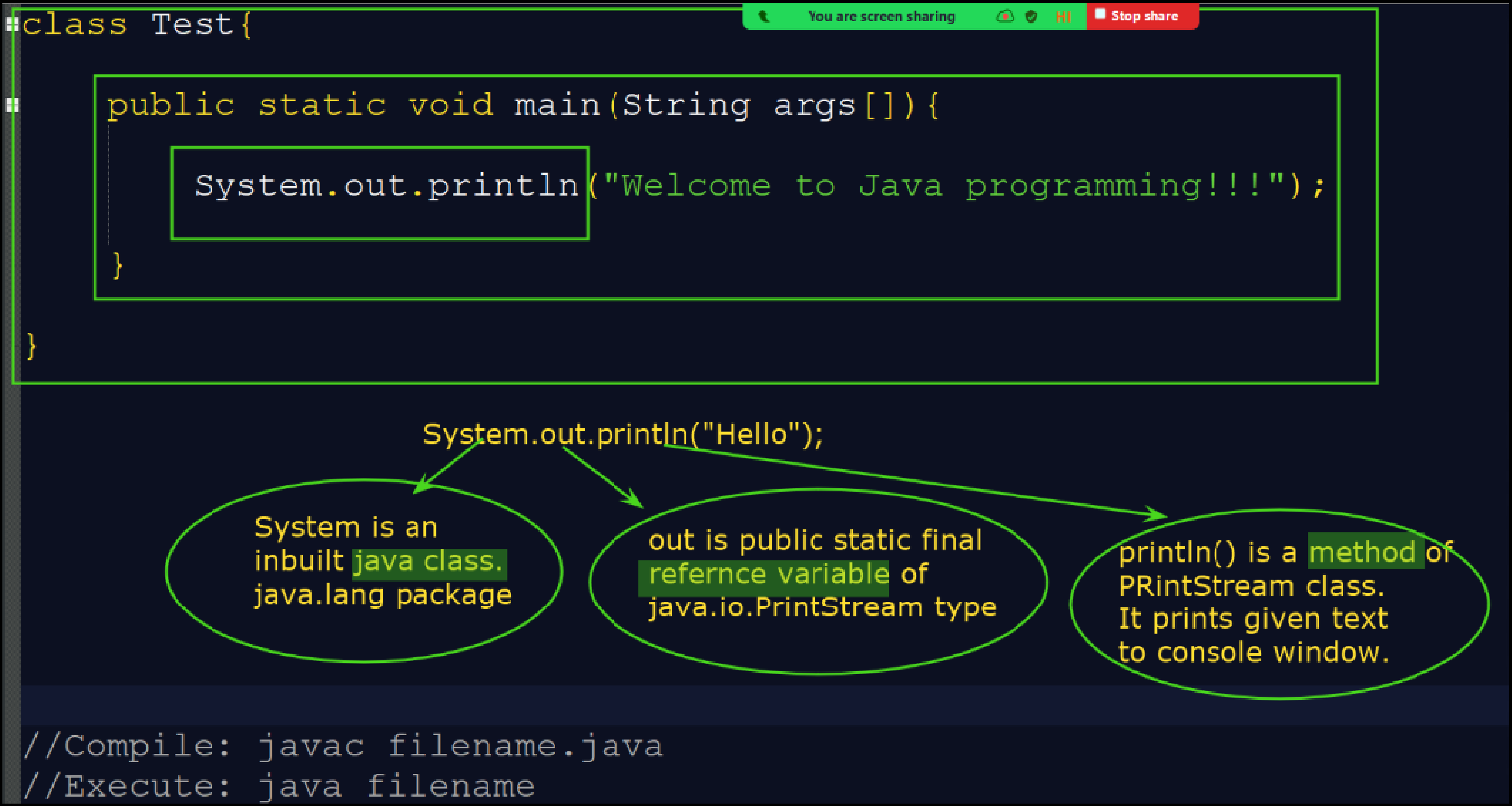
Parameter
Datatype: String : chat/number
Array : Multiple arguments acce

String args []

String[] args

[] : array
() : function/method
{} : scope

```
//Compile: javac filename.java
//Execute: java filename
```



```
class Test{

    public static void main(String args[]){

        System.out.println("Welcome to Java programming!!!");

    }

}
```

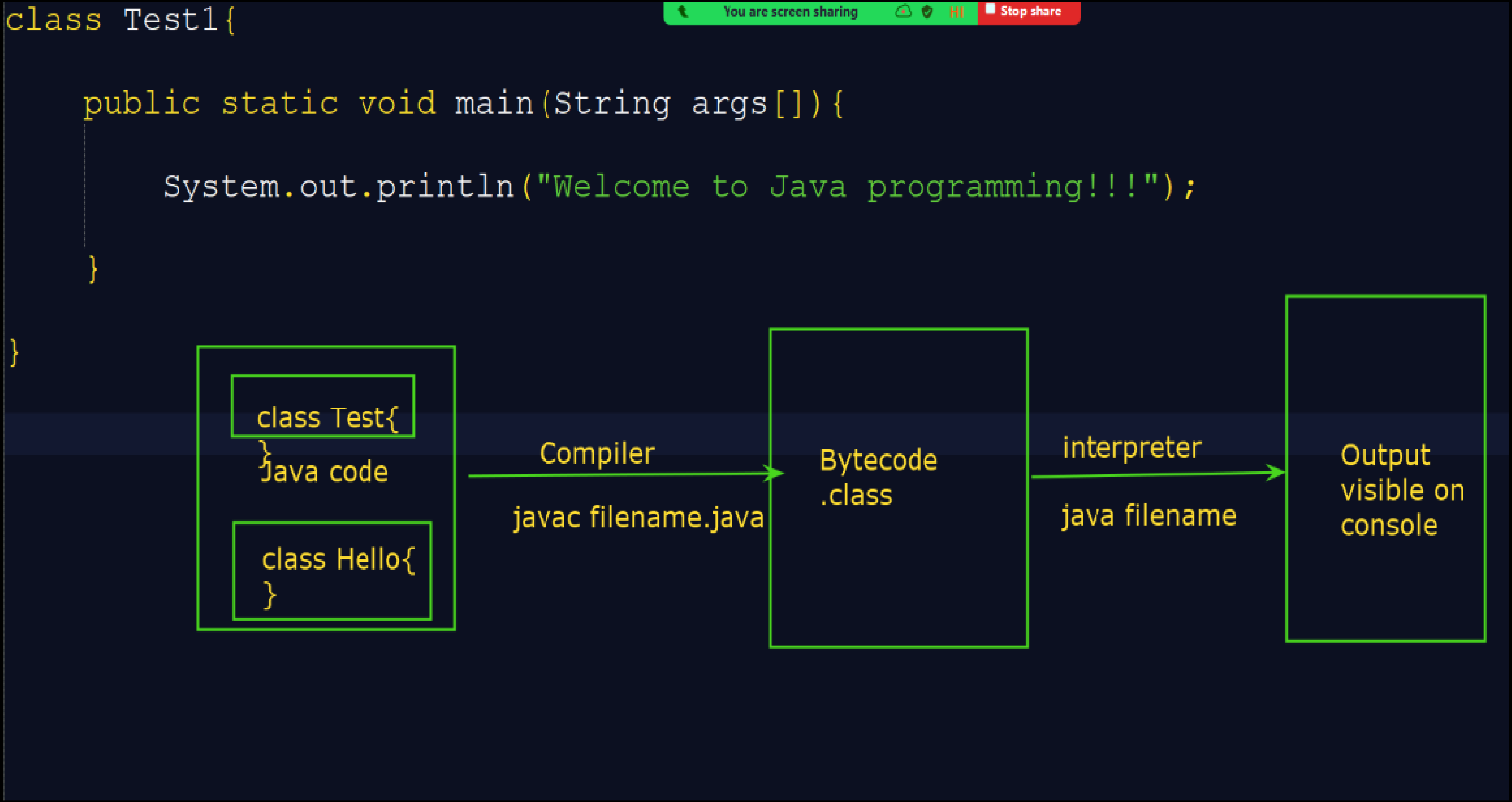
System.out.println("Hello");

System is an inbuilt java class. java.lang package

out is public static final refernce variable of java.io.PrintStream type

println() is a method of PRintStream class. It prints given text to console window.

```
//Compile: javac filename.java
//Execute: java filename
```



```
class Test1{

    public static void main(String args[]){

        System.out.println("Welcome to Java programming!!!");

    }

}
```

class Test{
}
Java code

class Hello{
}

Compiler

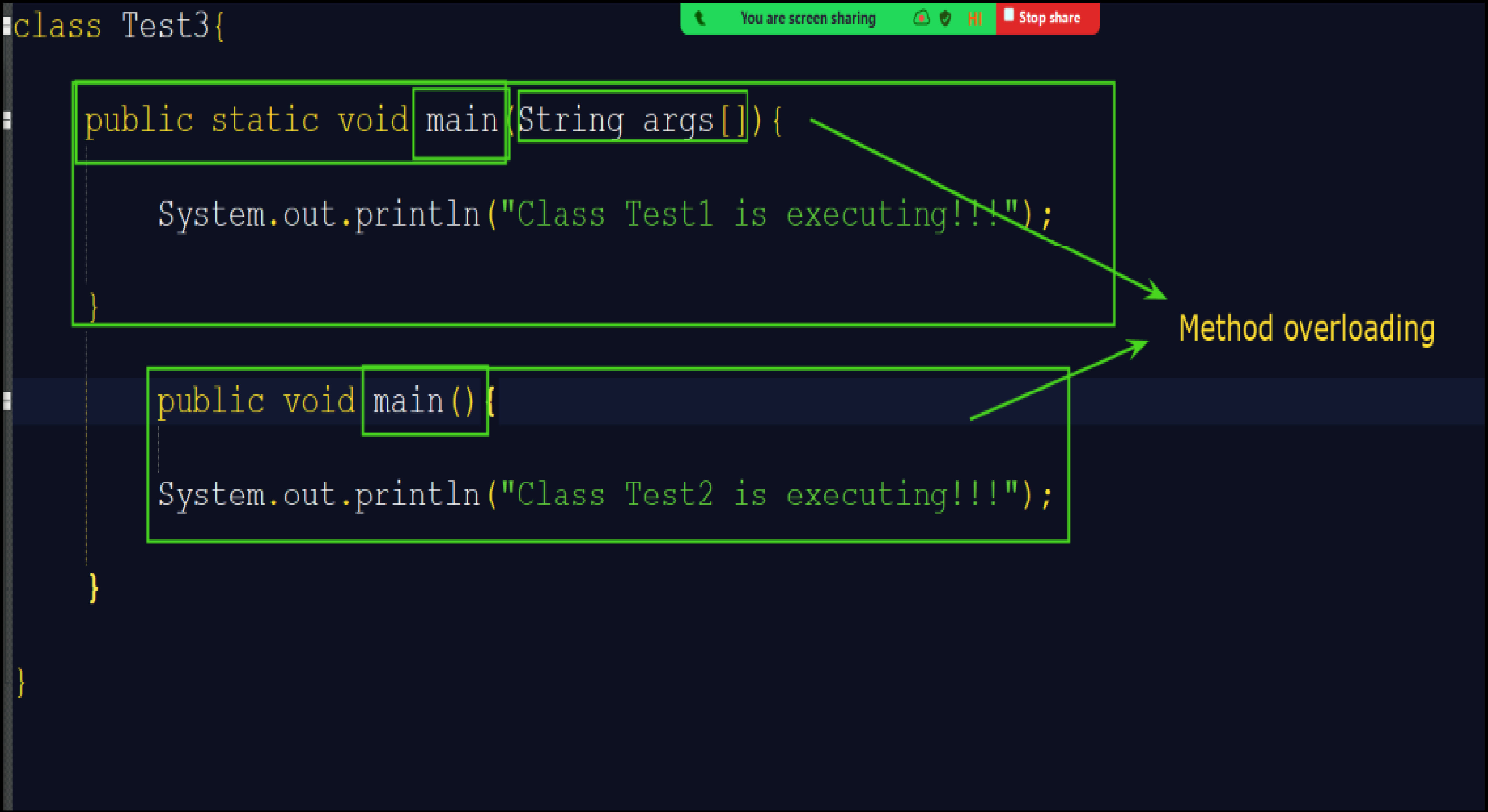
javac filename.java

Bytecode
.class

interpreter

java filename

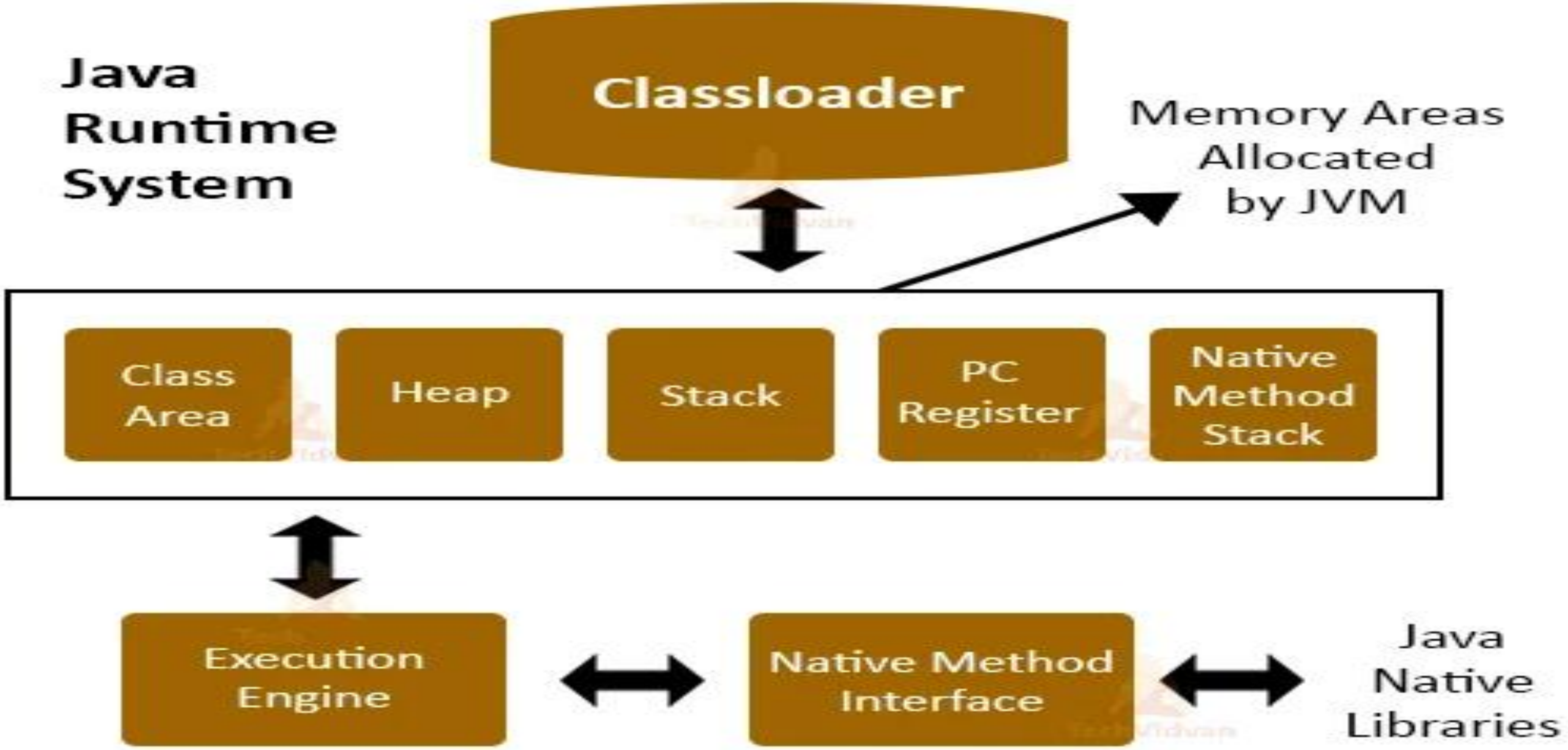
Output
visible on
console

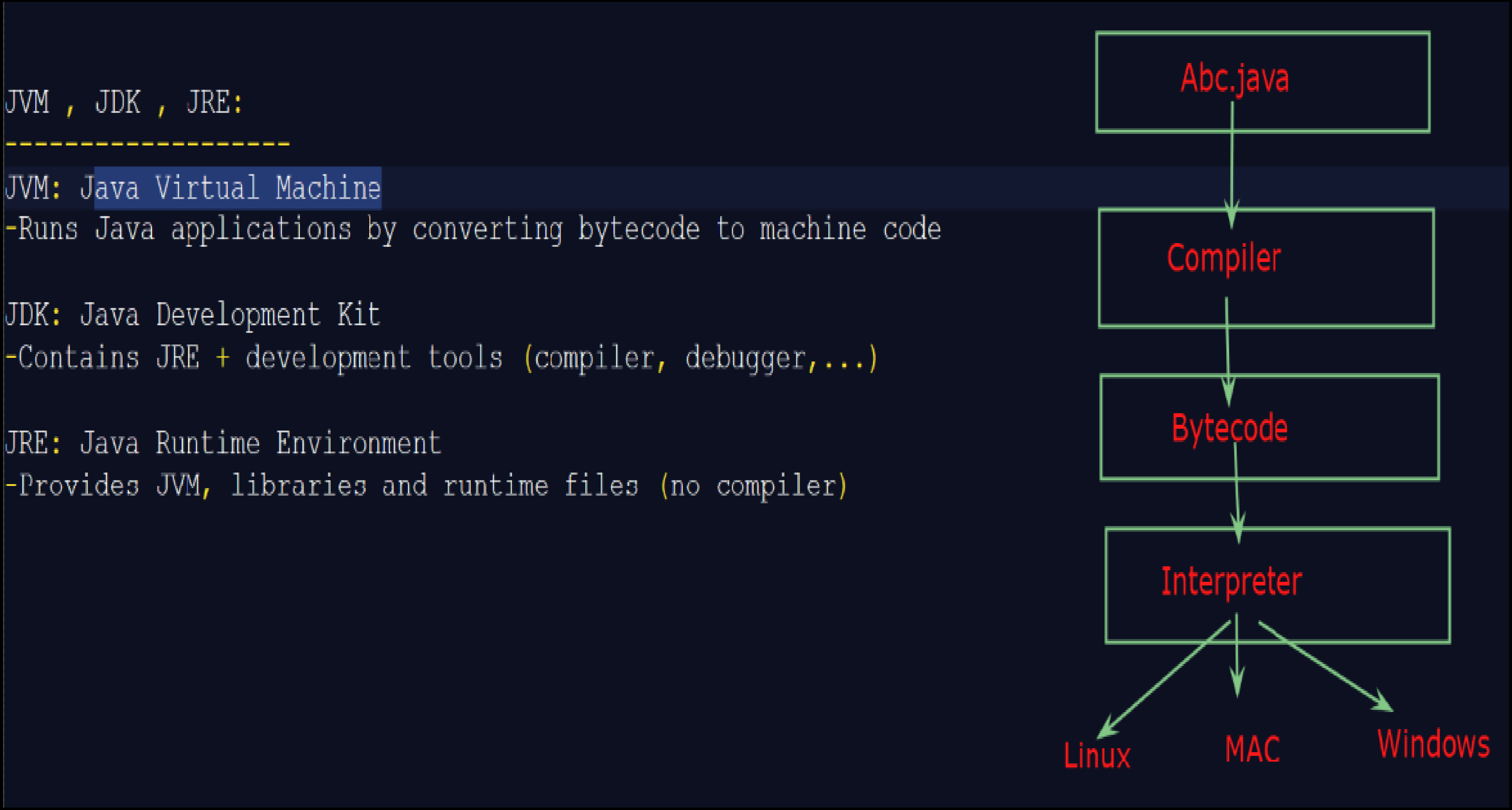


```
class Test3{

    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

JVM Architecture



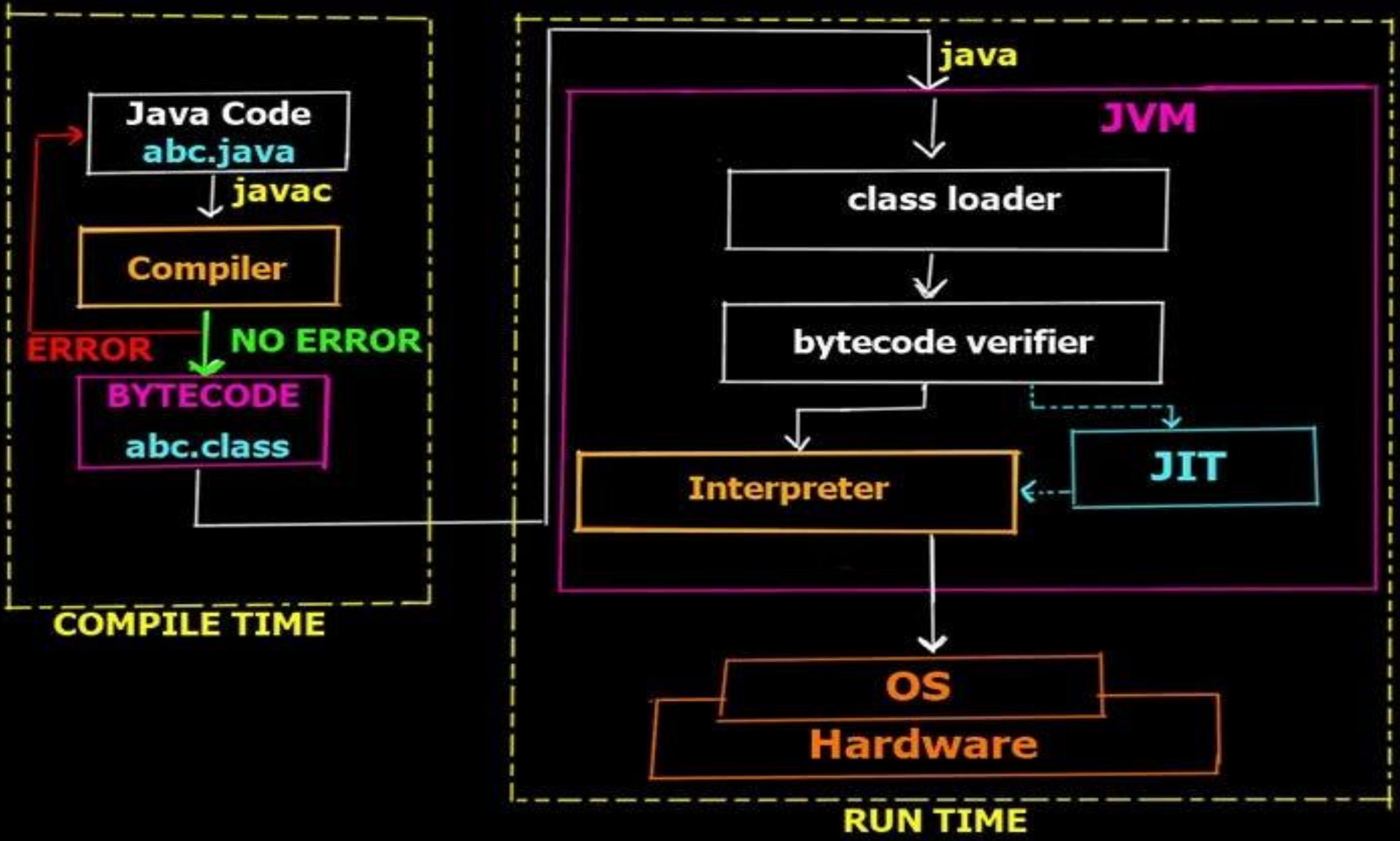


JVM , JDK , JRE:

JVM: Java Virtual Machine
-Runs Java applications by converting bytecode to machine code

JDK: Java Development Kit
-Contains JRE + development tools (compiler, debugger,...)

JRE: Java Runtime Environment
-Provides JVM, libraries and runtime files (no compiler)



Java Code
abc.java
javac
Compiler
ERROR
NO ERROR
BYTECODE
abc.class
COMPILE TIME
java
JVM
class loader
bytecode verifier
Interpreter
JIT
OS
Hardware
RUN TIME

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., ( ), { }, [ ], ;, , ,& .

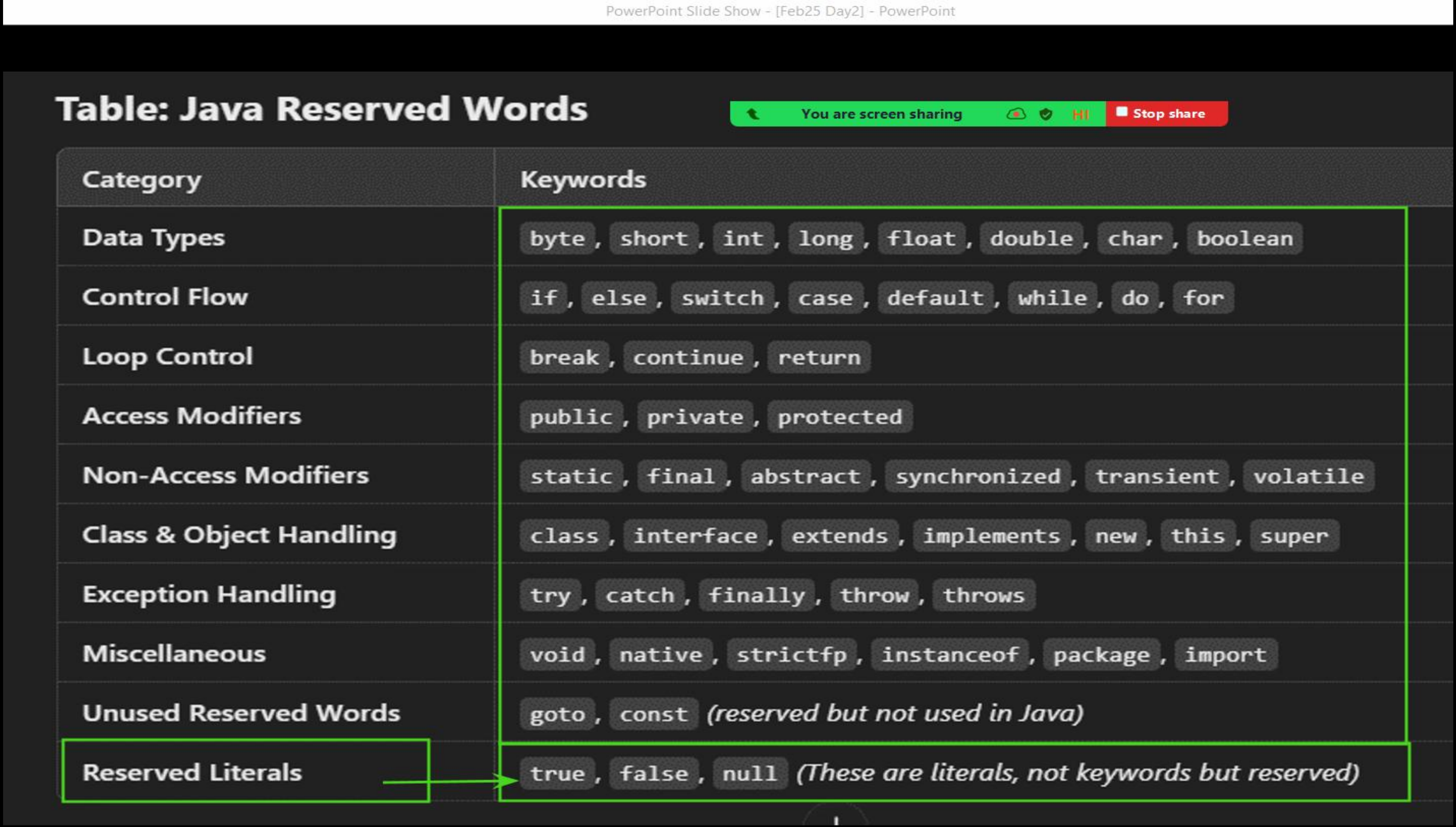
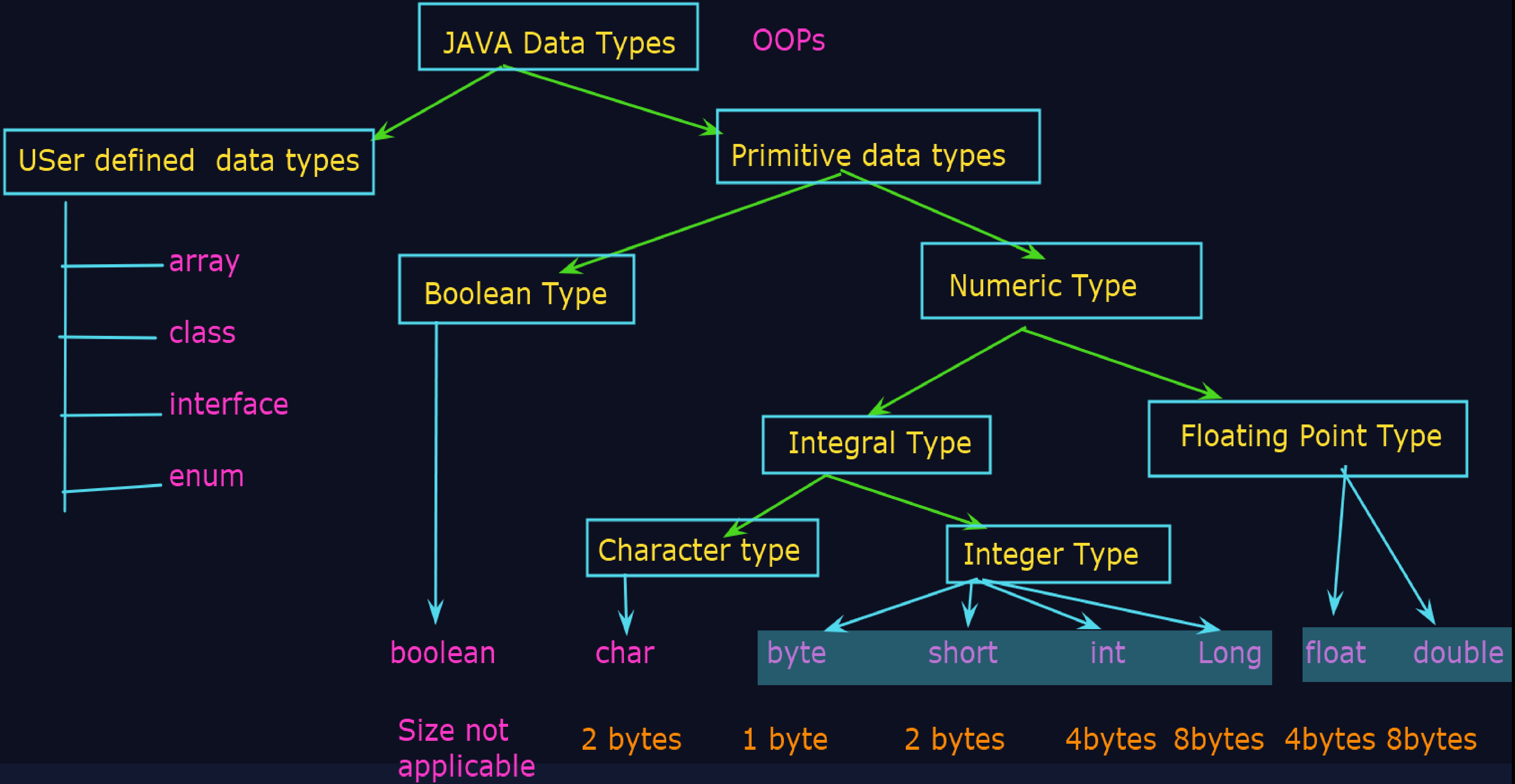


Table: Java Reserved Words

Category	Keywords
Data Types	`byte`, `short`, `int`, `long`, `float`, `double`, `char`, `boolean`
Control Flow	`if`, `else`, `switch`, `case`, `default`, `while`, `do`, `for`
Loop Control	`break`, `continue`, `return`
Access Modifiers	`public`, `private`, `protected`
Non-Access Modifiers	`static`, `final`, `abstract`, `synchronized`, `transient`, `volatile`
Class & Object Handling	`class`, `interface`, `extends`, `implements`, `new`, `this`, `super`
Exception Handling	`try`, `catch`, `finally`, `throw`, `throws`
Miscellaneous	`void`, `native`, `strictfp`, `instanceof`, `package`, `import`
Unused Reserved Words	`goto`, `const` *(reserved but not used in Java)*
Reserved Literals	`true`, `false`, `null` *(These are literals, not keywords but reserved)*



JAVA Data Types

OOPs

USer defined data types

- array
- class
- interface
- enum

Primitive data types

Boolean Type

Numeric Type

Integral Type

Floating Point Type

Character type

Integer Type

boolean char byte short int Long float double

Size not applicable 2 bytes 1 byte 2 bytes 4bytes 8bytes 4bytes 8bytes