Chapter 10

Interview questions – Part 2
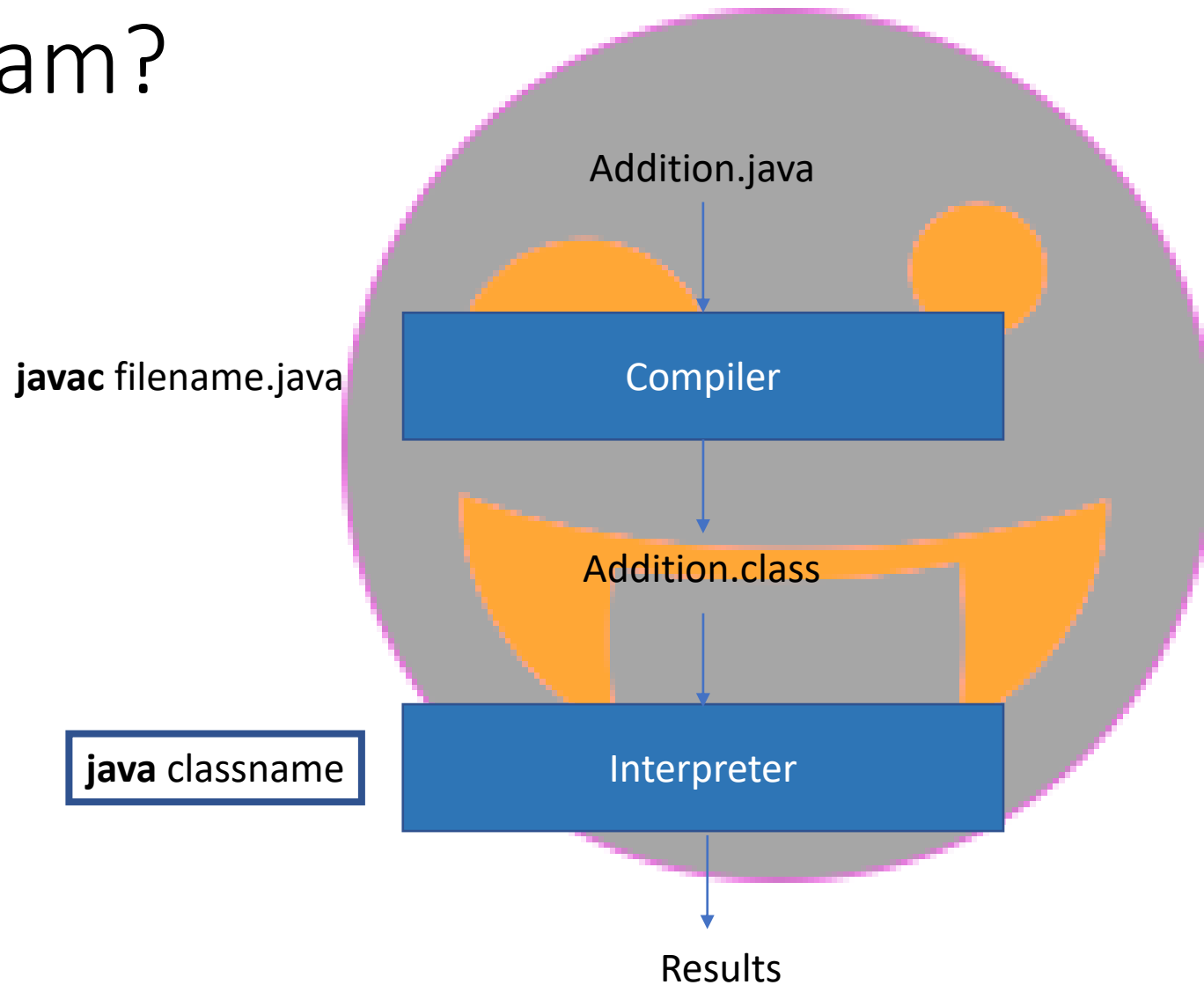
Other courses in our channel

👉 C 18 Hours Full course: https://youtu.be/3JF7ndGauZk

👉 Python 11 hours complete course: https://youtu.be/hXN0JBWIya8

👉 20 Programs for interview: https://youtu.be/16MFbFib7v8

👉 What is programming: https://youtu.be/UGfuscUWi-E

👉 Java in 10 minutes: https://youtu.be/cM82qnE_TPc

👉 Git Telugu course: https://youtu.be/LIhE7L__E6M

👉 Git English course: https://youtu.be/aysYDoEH-54

👉 HTML Full course Telugu: https://youtu.be/6P6yillxZY4

What does a compiler do in java?

Source code

javac   **javac Addition.java**

YES

```java
import java.util.Scanner;
public class Addition{
public static void main(String args[]){
    Scanner scanner=new Scanner(System.in);
    System.out.println("Enter first number");
    int a = scanner.nextInt();
    System.out.println("Enter second number");
    int b = scanner.nextInt();
    int c = a+b;
    System.out.println("Addition is: "+c);

}
}
```

Errors in the program?

Addition.java

NO

Byte code

Converts source code into byte code

Addition.class

Which command do you use to run a java program?

Addition.java

javac filename.java → Compiler

Addition.class

java classname → Interpreter
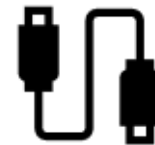
Results

What is a platform?

Platform

Hardware:
Silicon + plastic + iron + aluminium

Software:
ios – iphone operating system

A platform is the hardware or software environment
in which a program runs

What is platform Independence?

- WORA(**W**rite **O**nce **R**un **A**nywhere)

- You can **write your code once** and then **run it anywhere**, on **any platform** that provides the environment to run it **(WORA)**

- The JVM is different for each platform. So, **the platform dependent JVM's will achieve platform independence**.

- Explain **W**rite **O**nce **R**un **A**nywhere
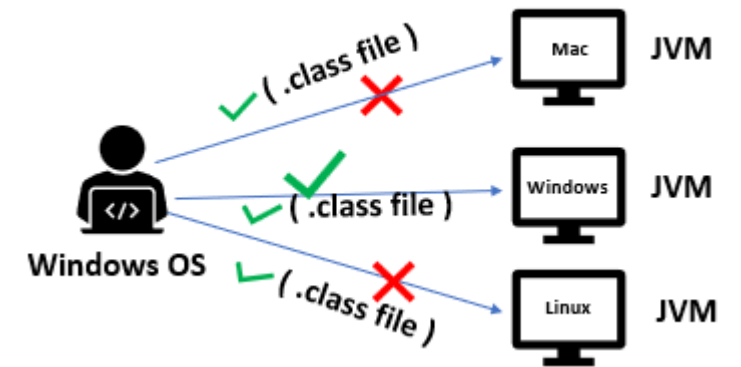
Is Java compiled or interpreted language?

- Java is **both compiled and interpreted language**
- **javac** is the **compiler** used to compile the source code
- **java** is the **interpreter** used to execute the byte code generated by the compiler
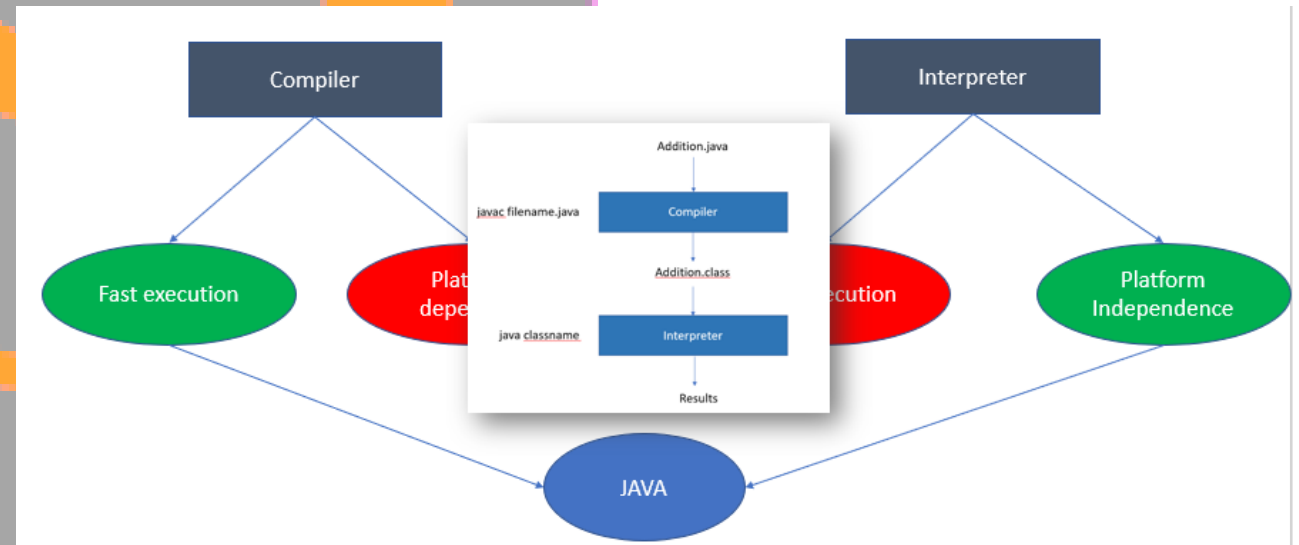
What is the value of first 4 bytes of a class file structure

4.1. The ClassFile Structure

A class file consists of a single ClassFile structure:

```
ClassFile {
    u4              magic;
    u2              minor_version;
    u2              major_version;
    u2              constant_pool_count;
    cp_info         constant_pool[constant_pool_count-1];
    u2              access_flags;
    u2              this_class;
    u2              super_class;
    u2              interfaces_count;
    u2              interfaces[interfaces_count];
    u2              fields_count;
    field_info      fields[fields_count];
    u2              methods_count;
    method_info     methods[methods_count];
    u2              attributes_count;
    attribute_info attributes[attributes_count];
}
```

0x**CA**, 0x**FE**, 0x**BA**, 0x**BE**

The items in the ClassFile structure are as follows:

magic

The magic item supplies the magic number identifying the class file format; it has the value 0xCAFEBABE.
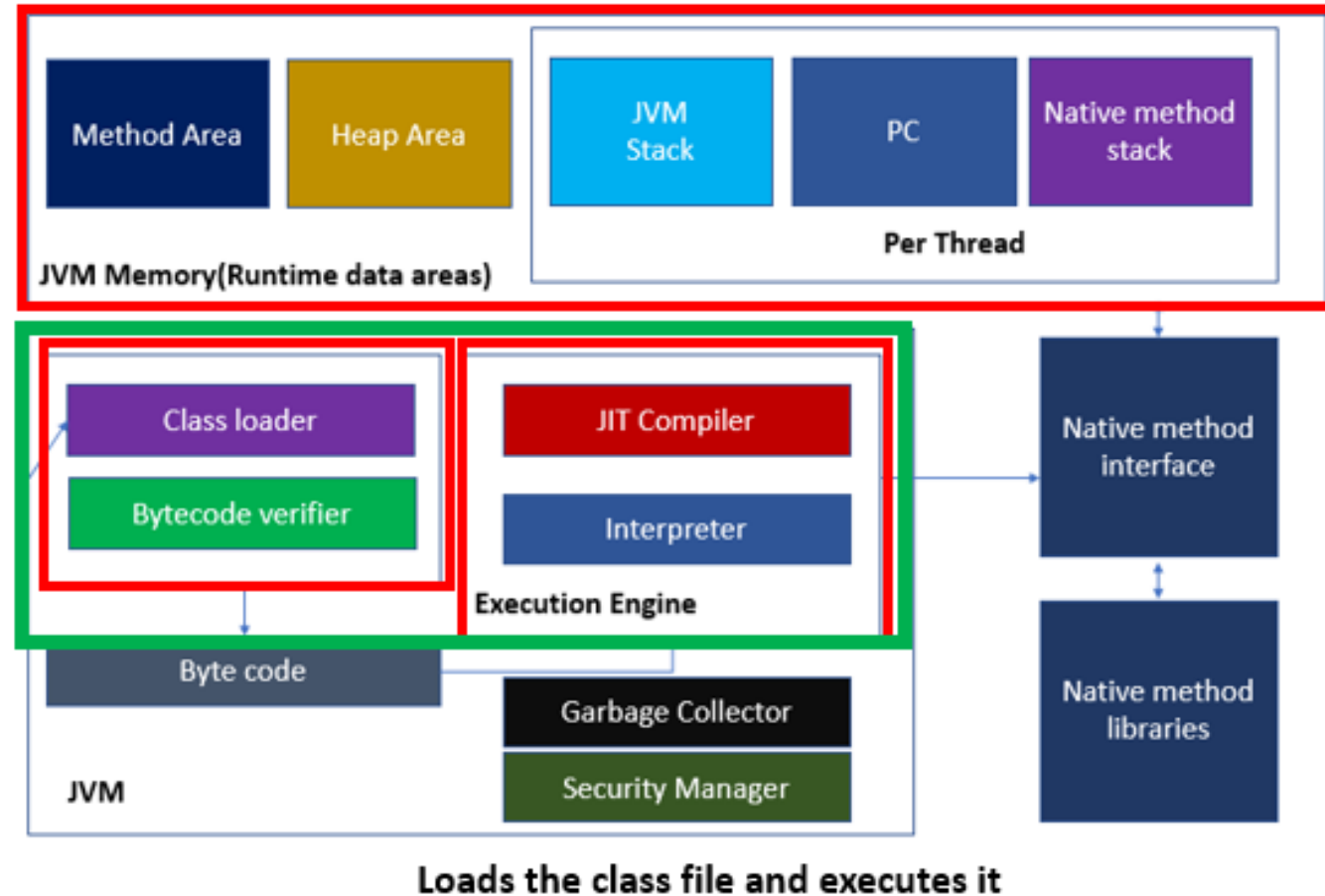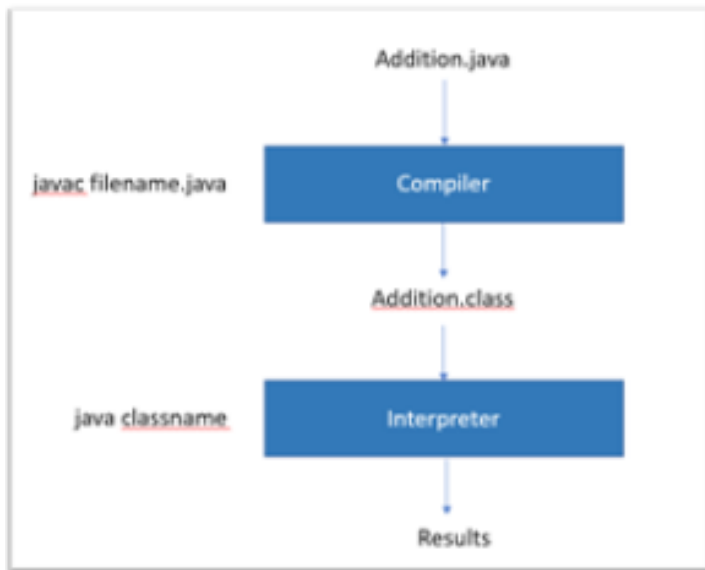
Difference between compiler and interpreter?

Compiler	Interpreter
Compiler checks the entire program and translates into the machine code at once	Interpreter checks the program one statement at a time and translates it into machine code
Compiler takes more time to analyse the source code, but takes less time to execute the program	Interpreter takes less time to analyse the source code, but takes more time(little slow than compiler) to execute the program
Ex: C, C++, Erlang, Go	Ex: Python, JavaScript, PHP, Ruby

Explain JVM Architecture

1. **Class loader subsystem**
2. **Runtime data areas**
3. **Execution Engine**

Addition.java

javac filename.java → Compiler

Addition.class

java classname → Interpreter

Results

JVM Memory(Runtime data areas)

| Method Area | Heap Area | | JVM Stack | PC | Native method stack |

Per Thread

Class loader
Bytecode verifier

JIT Compiler
Interpreter

Execution Engine

Byte code

Garbage Collector

Security Manager

JVM

Native method interface

Native method libraries

Loads the class file and executes it

What is JRE?

- **JRE** stands for **J**ava **R**untime **E**nvironment
- It is required to execute the byte code generated by the compiler
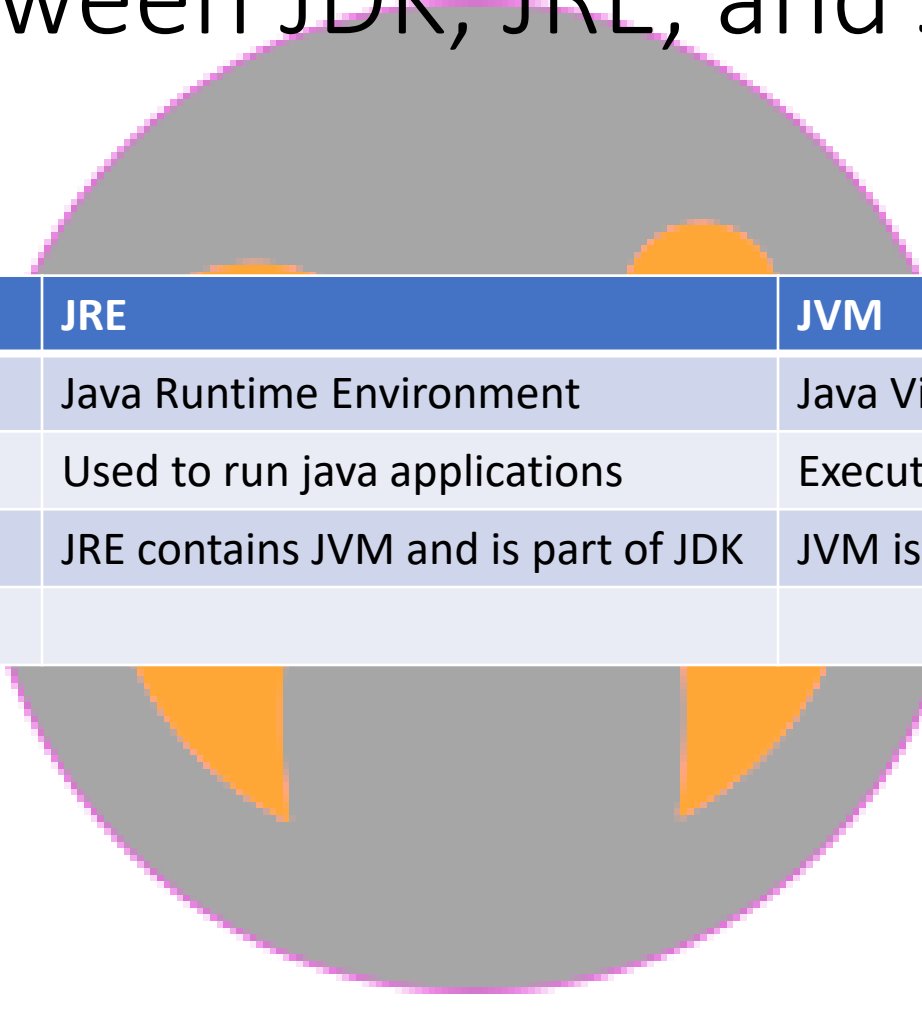
Development tools:
javac
java
jar
javadoc
javap
jdb

Java Runtime Environment(JRE)

JVM
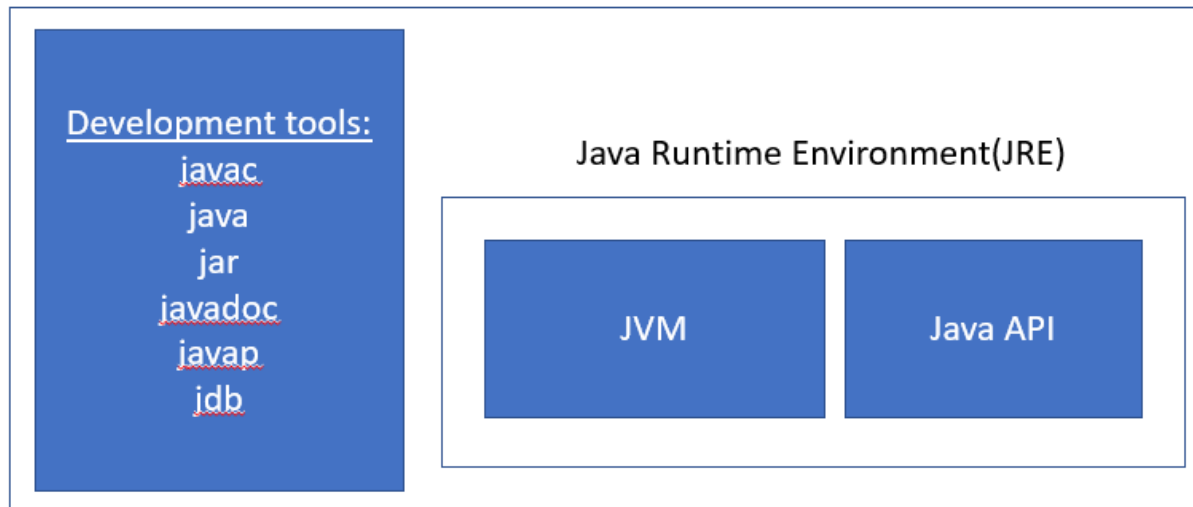
Java API

Difference between JDK, JRE, and JVM

| JDK	JRE	JVM
Java Development Kit	Java Runtime Environment	Java Virtual Machine
Used to develop java applications	Used to run java applications	Executes java byte code
JDK contains JRE	JRE contains JVM and is part of JDK	JVM is part of JRE

What does JIT compiler do?

- The **Just-In-Time (JIT) compiler** is a component of the **Java Runtime Environment** that improves the performance of Java applications at run time.

JIT Compiler

```
class JITTest{
    String name="JIT";
    static int totalSum = 0;

    void display(){
        System.out.println(name);
    }

    void calculate(){
        totalSum = totalSum + 1;
    }
}
```

Method call count > 1000 = save the machine code for future calls

Threshold value = 1000

JIT Compiler

Interpreter

Execution Engine
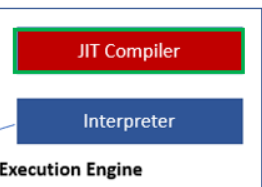
The **problem with the interpreter** is that **it interprets every time, even the same method multiple times,** which reduces the performance of the system.

Interpret = converting byte code to machine code

In order **to improve performance**, JIT compilers interact with the Java Virtual Machine (JVM) at run time and compile suitable bytecode sequences **into native machine code based on the threshold count set by JIT** in combination with the JVM method calls

Java SE stands for?

- Java **S**tandard **E**dition

- **Core Java**

Java Platforms / Editions

Let us Install JDK

- Java SE
  - Java **S**tandard **E**dition
  - J2SE
  - Stand alone applications for desktops & servers
- Java EE
  - Java **E**nterprise **E**dition
  - J2EE
  - Large scale applications
- Java ME
  - Java **M**icro **E**dition
  - J2ME
  - Applications on small devices like mobile phones
- JavaFX
  - Java Special E**ff**ects
  - Graphics and media
  - Modern look and feel

Specifications

1. Java Language Specification (JLS)
   - Rules(Syntax and semantics)
2. Java Virtual Machine Specification (JVMS)
   - How a JVM should work, byte code instruction set
3. Java API Specification
   - Java library specifications

Implementations   **JDK (Java Development Kit)**

1. Oracle **JDK**
   - **Free for learning**, Paid for commercial purpose
2. Oracle Open**JDK**
   - Free even for commercial purpose
3. Adopt Open**JDK**
   - Free
4. Red Hat's Open **JDK**
   - Free
5. IBM J9, Azul Zulu, Azul Zing

Find the mistakes(3) in this program

```java
class Welcome {
    public static void main(String[] a){
        System.out.println("Welcome to suresh techs, I am learning Java");
        System.out.println("My name is Hareesh, I will get job soon");
        system.out.println(1);
        System.out.println(2);
        System.out.println(3);
        System.out.prntln(4);
        System.out.println("'Suresh techs' is 5 star");
    }
}
```

What next?

Write your first class on your own

చిన్న బ్రేక్ చిటికలో వచ్చేస్తా

Other courses in our channel

☞ C 18 Hours Full course: https://youtu.be/3JF7ndGauZk

☞ Python 11 hours complete course: https://youtu.be/hXN0JBWIya8

☞ 20 Programs for interview: https://youtu.be/16MFbFib7v8

☞ What is programming: https://youtu.be/UGfuscUWi-E

☞ Java in 10 minutes: https://youtu.be/cM82qnE_TPc

☞ Git Telugu course: https://youtu.be/LIhE7L__E6M

☞ Git English course: https://youtu.be/aysYDoEH-54

☞ HTML Full course Telugu: https://youtu.be/6P6yillxZY4