JAVA

Installation:

Java download search → oracle Java downloads → JDK 21 → windows → 64x bit Installer

SET PATH:

program files → Java → JDK 21 → bin → (path copy)

(windows search bar)

env → environment variables → New →

name: Java_home
path: path copy } → ok → down of PATH edit → New → (path copy) ok

CMD:

Java -version

IDE for Java:

eclipse download → eclipse IDE for Java Developers → create a new Java Project

FLOW OF JAVA:

Project → package → class.

New Project → Name → J21 → uncheck module → Finish

Package:

src → New → Package → name → uncheck → Finish

class:

basics → New → class → Newpgm → public → Finish

Project and class → First name should start with Capital

Packages → not mandatory

SYNTAX :

```
packages basics;

public class Newprogram {
    public static void main (String[] args) {
// Executable program
        System.out.println ("manik shibu")
    }
}
```

Low Level PL → machine lang → Binary information

Assembly Language

High Level PL → Java, C#, Python

Assembly Language → Human readable

convert

AL → ML → machine readable

Assembler

H.P.L → high level Pegm lanuge (English)

↓ convert to machine Language

JAVA → Compiler / Interpreter → machine language

↳ compile code

Line by line code check

Founder of Java → James osling

First name → oak

new name → Java

Latest version → 23.

Application → mobile, web

What is Java:

- OOP (object)
- class base (class we can write pgms)
- Concurrent progeraming language (multiple Execu

Java Features:

(1) simple:

- platform oriented
- c++ complex topic we have avoided here

✱ (2) Platform Independ:

- write once run anywhere (OS)

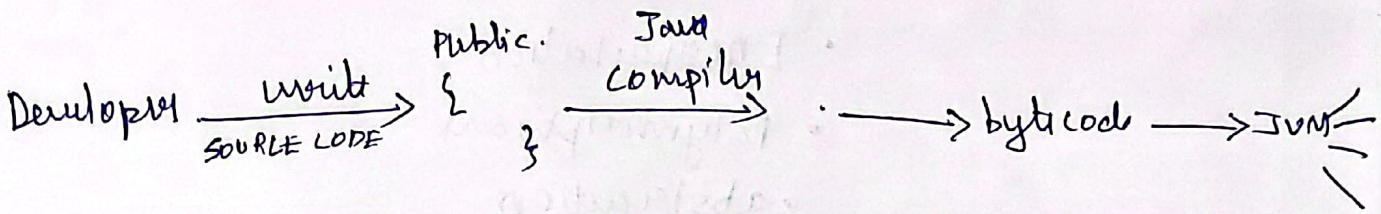
→ why? → JRE → Runtime Envis
JDK → Developt kit
JVM → virtual machine

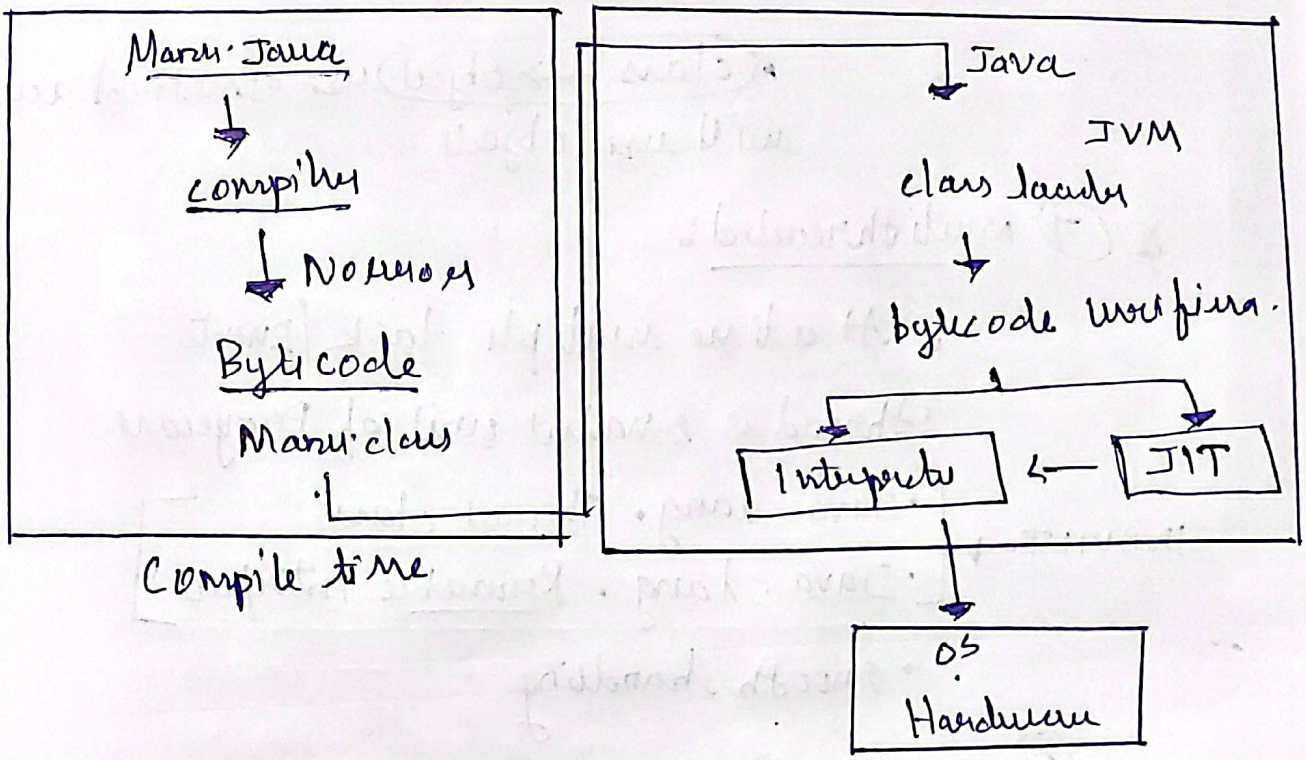
JDK → must be installed 22/21/20.
Compiler installed

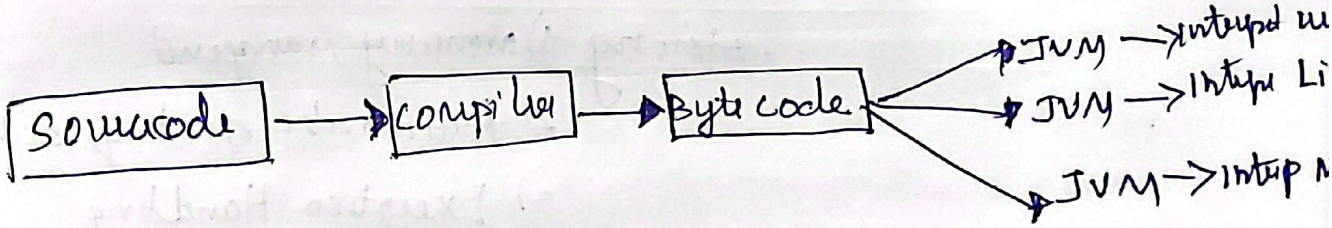
JRE → installed with JDK, only for Ap
→ independently download

JVM → installed with JDK
→ By default already installed
→ platform dependent because of
that Java platform independ

INTERAL working of source code:







③ Architectural neutral:

- Hardware change does not affect
- OS changes. (X)

④ Portable:

- Java byte code to any platform without changes.
- OS free

⑤ Secure:

- JVM Bytecodes verifier is responsible for verifying bytecode

⑥ Java is object oriented Pgm:

- Inheritance
- Encapsulation
- Polymorphism
- abstraction
- class → object use of method access we will use objects

⑦ multithreaded:

- At a time multiple task / Events
- Thread is smallest unit of Program

INBUILT → [• Java.lang.Thread class:
• Java.lang.Runnable Interface]

- Smooth handling

⑧ Java is Robust:

- strong 1) memory managent
  2) Automatic garbage collection
  3) Exception Handling

1) Memory Managent:

- Main method
- static keyword use
- one time memory space allocate
- for static Extra objects create NO.
- space free we will use static

2) Automatic garbage collection:

- int a, but not using in Pgm.
- unused variable space Remove by Automatic

3) Exception Handling:

- un Expected condition / errors
- 5th line Exception 6th line it won't Execute
- Arithmetic Exception
- Path Exception
- we need to handle that

④ Java is distributed Programming Language:

- Mob, web Applications, bank create
- RMI / EJB

⑩ High Performance:

- sourcode → bytecode

Explain code:

- System → (inbuit class)
- out → System subclass printstream
  printstream object called as 'out'
- println → inside printstream subclass
  Method in Printstream class
  out method is called as Println
- (.) → operator dot for connection purpose.

~~Cursor~~

For printing:

- println / print also we can use

println → print of new line cursor

print → still get print of cursor won't move to next line

println → new line

Progeram structure:

Acess modifier → public, private
main method →

class creat ← public class class name {

main method ← public static void main (String args[])
{

System.out.println ("hello word");

}

}

CMD:

Javac name project → compile
Java name project → Run

class name and Progeram name must be same