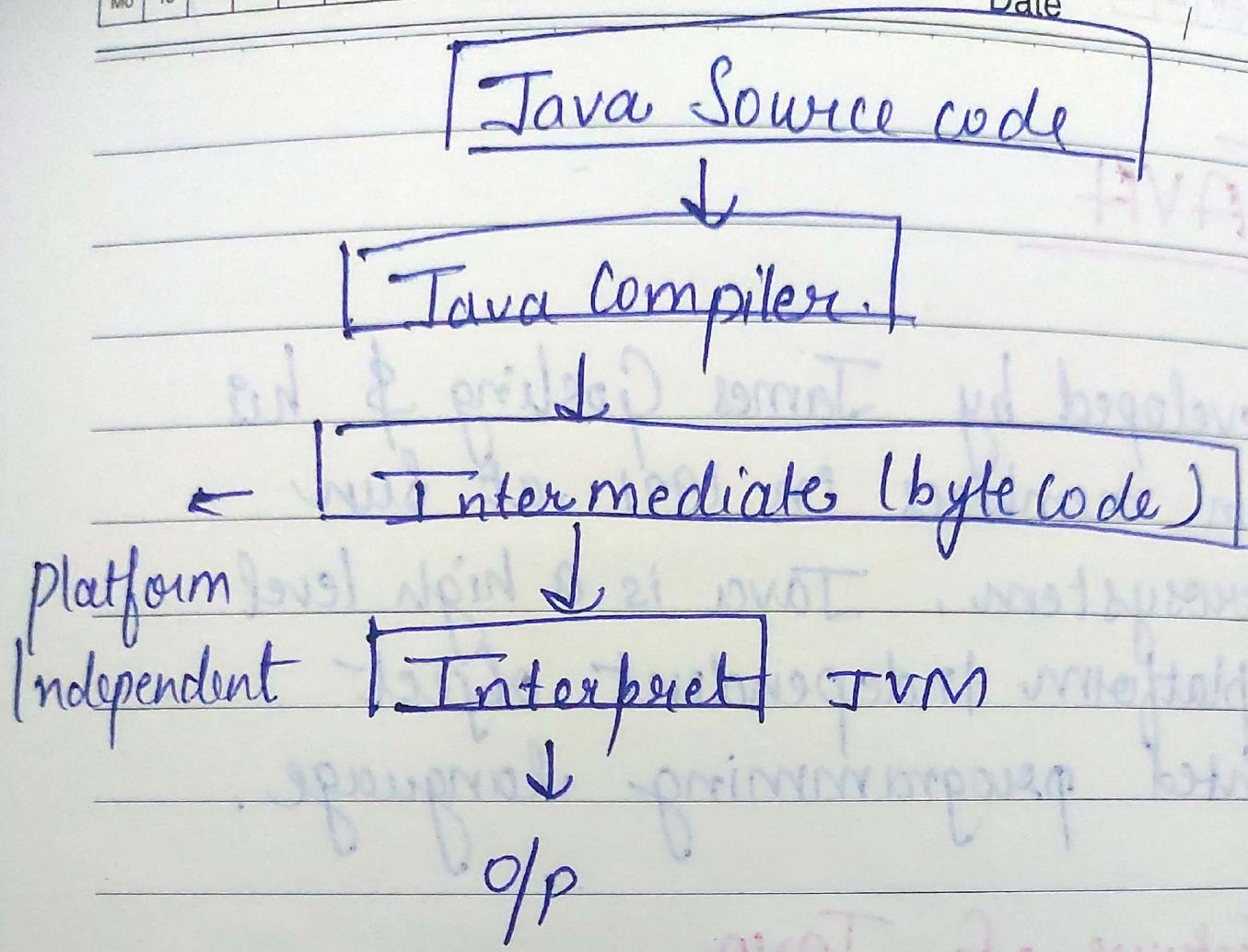


JAVA

→ developed by James Gosling & his team members in 1991 at Sun microsystem. Java is a high level & platform independent object oriented programming language.

Features of JAVA

- 1) Simple → C lang + C++ +
No pointers OOP concepts +
No operator overloading +
No multiple inheritance.
- 2) Platform Independent - (WORA)
Write once run Anywhere.
- Machine Independent as well
(we can say)



JDK (Java Development Kit)

JRE (Java Runtime Environment)

JVM (Java Virtual Machine)

JDK contains JRE and JRE contains JVM.

Versions of JAVA

- J2SE - Standard Edition
- J2EE - Enterprise Edition
- J2ME - Mobile Edition



Mo Tu We Th Fr Sa Su

Memo No.

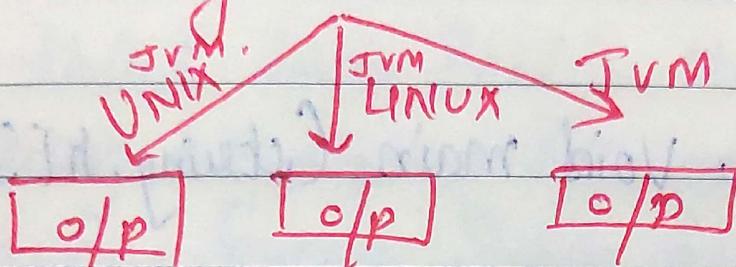
Date

/ /

JAVA

↓ (compiler JAVA)

Byte code



Java is a case-sensitive language

Pre-defined class.

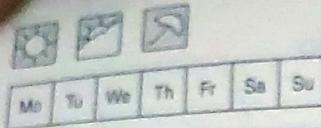
• (1st letter capital)

Java is more object Oriented than C++ because it take any function in class only.

In C++ Default Access specifier is Protected Mode

In JAVA, Default Access specifier is Default

However it use public most of the time.



Memo No.
Date

JAVA 1st Program.

Class demo

{

 Public Static Void main (String, k[])

{

 System. Out. Pecinln ("Hello");

{

The first line class demo declares
a class which is an object oriented
construct.

Java is a true object Oriented language.
Every must be inside class.

Class is a keyword and demo is a
Java identifier that specifies the
name of the class.

Every class definition begins with an opening braces and ends with closing .

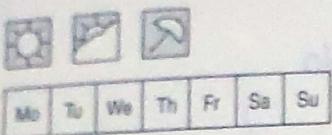
The third line Public static void main (String, k) defines the method named main . Every Java program must include main method .

The line contain three keyword Public , static and Void .

The keyword public is an access specifier that declares the main method as unprotected and therefore making it accessible outside class .

Static : The main method must always be declared as static since this method is used before any objects are created .

(i.e no need of object initialization to call main method)



Memo No.

Date

Void : The return type modifier.

void states that the main method does not return any value.

String k declares a parameter named k which contains an array of objects of the class type string.

System in as a predefined class and out as predefined object stream.

Println method is used to print string in Java. Every Java statement must end with semicolon.

① Data Types In Java.

→ Primitive [→ Numeric
 , Non Numeric]

→ Non- Primitive. [Array, class,
 • Interfaces]

Numeric Data type

* Integer → Short (2 byte)

int (4 byte)

long (8 byte)

byte → (1 byte)

* Floating No - float (4 byte)

double (8 byte)

Non numeric → char (2 byte)

Boolean (1 bit)

* Char take 1 byte in C++

Bytes may differ acc. to operating Sys.

Object Oriented Programming

Concepts : Data hiding Abstraction
Inheritance Dynamic binding
Polymorphism Message Passing



Memo No.

Date

Constructor In JAVA

They are special member functions.

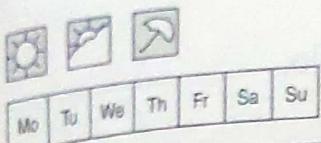
- 1) Constructor name is same as that of class name
- 2) Do not have return type (not even void)
- 3) Const. provide memory to the objects and helps in object initialization,
- 4) Const. are called implicitly at time of object
- 5) Const. can be parametrized or non-parametrized

Syntax

```
Class ABC  
{
```

ABC() // const keyword

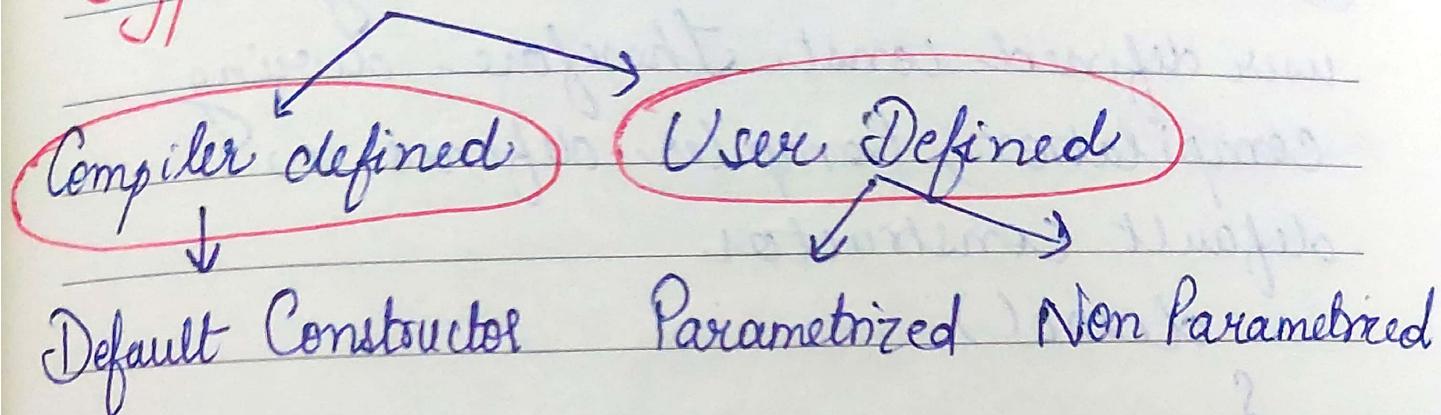
```
{  
Class Name object_name = new classConst();  
}
```



Memo No.

Date

Types of Constructors



Compiler Defined- If a developer does not create a const. for a class JAVAC Compiler provides a default constructor for that class. Such constructor do not have any arguments and have null body.

for eg → class demo

```
{  
    void show()  
}{  
    Capital always  
    System.out.println("I am from class demo");  
}
```

```
class DAV {  
    public static void main (String KI)  
    {  
        class demo d1 = new demo ();  
        d1.show();  
    }  
}
```



Mo Tu We Th Fr Sa Su

Memo No.

Date / /

Demo class do not have any user defined const. Therefore during compilation, Compiler defines a default constructor

```
demo()  
{  
}
```

User Defined Constructor

* Non-parametrized constructor

```
class demone
```

```
{ demone()  
}
```

```
{ System.out.println("I am non-parametrized") }
```

```
Void S
```