

Java - [A. James Gosling] latest version-JDK 22

1995

→ Initially it was called oak, but Oak is already famous and it is national tree in many countries, so because of trademark issues they named it as Java to change the name, so at that time James Gosling thought the name must be different and unique.

→ While having a cup of tea, he named it as Java, famous island in New Zealand, which is famous for coffee.

→ Firstly it was invented for televisions, like set-top boxes, after because of vast libraries it is used in many everywhere like e-commerce applications, bank applications, web applications, atm, robo, traffic etc.

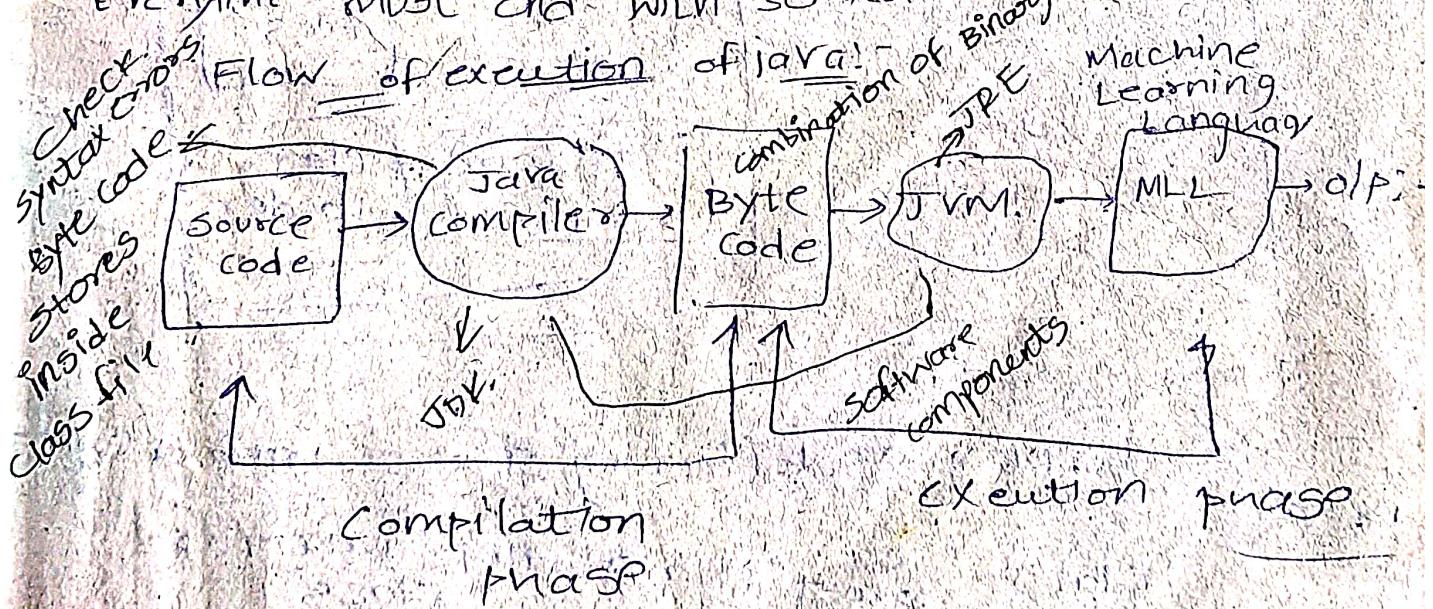
Every March & September, they will introduce their new versions.

→ Java is completely object oriented language.

→ It is platform independent, i.e. ~~wrote once and~~ read anywhere (WORA)

→ Java is an object oriented programming language so everything must be in a class.

→ Everyline must end with semicolon



→ Here firstly after Java code entered, then the compiler converts the source code into Bytecode.

→ After JVM (Java virtual machine) will convert the

→ Byte code into machine understandable language.  
→ After it will execute.  
→ Portable: - Means capable of running and executing on different platforms.

### Representation of

### Diagrammatic Representation:

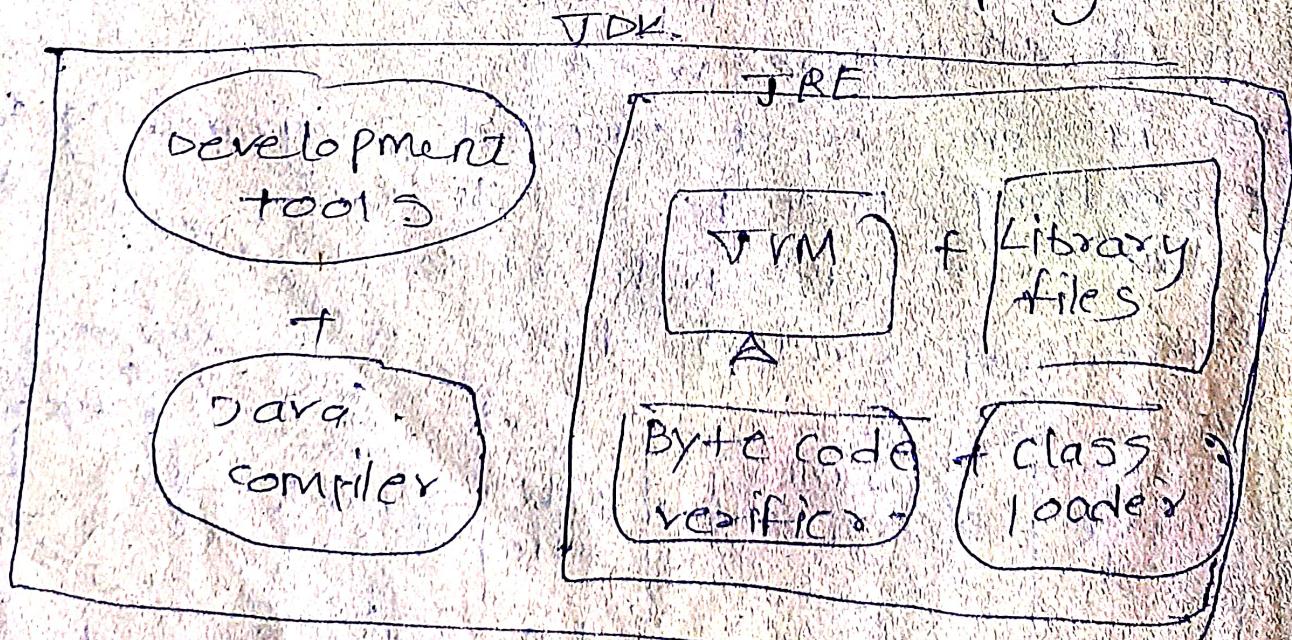
→ Here mainly 3 things:

- JDK [Java Development Kit]
- JRE [Java Runtime Environment]
- JVM [Java Virtual Machine].

JDK: - It is a abbreviation for Java development kit, it is a group of software which is required to compile and develop Java programs.

JRE: - Major responsibility for creating runtime environment for the execution of code.

JVM: - Java virtual machine, it is another type of software used to execute the programs.

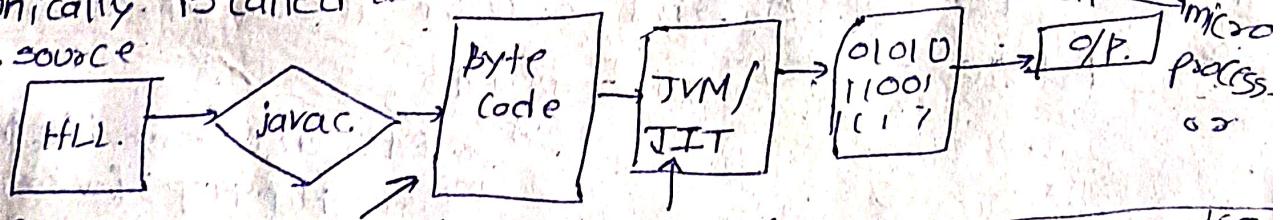


- We cannot access because it is a class defined in other package.
- Before 1940 we don't have any computer, because it was for making computer we need semiconductors which was not introduced before 1940 only semiconductors and insulators are there.
- By using semiconductor we introduced microprocessors & compilers. Microprocess only understand OR AND +SV[0 and 1] programming. → In 1960.
- Instructing the processor what to do is called as programming.
- Execution of java program: - RAM is volatile [Until power is there it will be there, if power is not there, then it will gone].  
ctrl+s → save[i.e. RAM → harddisk].
- Harddisk! - whatever we are (image, video) all are in file format.



or Contents

- The process of bringing instructions from hard-disk to RAM for the purpose of execution is called loading.
- If a file contains instructions in high level language technically is called as source file.



First.java

Intermediate language. First-class just-in time compiler. machine also because time compiled. machine also neither we can't understand don't understand automatically instant

- Java convert everything into 0's and 1's and is automatically loaded into RAM
- If we are ~~invent~~ invented a game which is very one if we want to send that to some one then how will use it?
- For example if send as source file then my friend can change the add word whatever ~~is not~~ he want. So sending source file to anyone.

→ Before Java people send as machine level language.

→ If we send them as machine level language, then while creating game, we create only belongs to windows operating system. and it only works on

Windows operating system.

- For It is not run in mac.
- Java. overcome this drawback [by JVM]
- Here in java byte code will be transferred
- Platform independent (that's why java is platform independent).
- set-up boxes (sun microsystem) invented they choose C/C++ but for inventing set-up boxes they C/C++ language is not sufficient, so overcoming those drawbacks by accidentally they invented one more language i.e. Java.

James Gosling (members) → Patrick

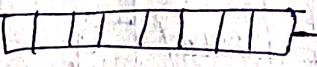
→ WORA (write once read anywhere)

→ Architectural neutral.

JRE → Java Libraries.

class second { → Java interpreter. } JVVM ↗  
    public static void main(String[] args) {  
        System.out.println("You will get your first dream job");  
    }  
}

→ Variables are the names given to memory locations in order to store and retrieve the data.

→ 1 byte  → 8-bits

8-bits are very bit made up of flip-flops [or 0 or 1].

object oriented programming:-

device which can store 0 or 1.

real world perspective [point of view]

→ the process of looking real world entity in perspective of object is called object orientation

object is existence of class

→ occupying space. Here we can able to store only one value

Data Structure

why variable is not a datastructure?

→ capable of holding the data

→ It should exhibit certain behavior

Queue → pointer example

## left shift

$0010 \rightarrow a \ll 1;$

$0010$

$\swarrow \swarrow \swarrow$

$a \ll 2:$

$0010$

$\swarrow \swarrow \swarrow \swarrow$

$0100 \rightarrow 4$

$\swarrow \swarrow \swarrow \swarrow$

$\rightarrow 2^2 = 2 \times 2$

$00100 = 4 \times 2$

$\swarrow \swarrow \swarrow \swarrow$

$1000 \rightarrow (8)$

multiplying

$a \gg 1:$

$1000$

$\swarrow \swarrow \swarrow$

$0100 \quad 0$

$a \gg 2:$

$1000$

$0100$

$0010$

$y \rightarrow \text{dividing}$

→ Bitwise operators will act on every bit on a data, so that's why it is called bitwise operators.

→ data type: Convert to real world data into binary

Type inference:

Var  $a = 10$  → integer, we are looking at integers.

byte  $a = 10$

→ Here type is defined as byte.

short  $a = 10$

int  $a = 10$

long  $a = 10$

unicode = 62536

16

public abstract void func()  
... as an abstract

binary values how they stored  
 For negative numbers, it  
 use 2's complements

$\begin{array}{r} 11 \\ 11 \\ 10 \\ 01 \\ 10 \\ 00 \\ 10 \end{array}$   $\rightarrow 1011 \rightarrow$  In this  
 way decimal  
 values converted &  
 slo'.

How float values stored are converted to binary.  
 by using IEEE format.

float a; → 4 bytes (6-precision) → IEEE single precision  
 format is used  
 float a; → 4 bytes (6-precision) → IEEE double precision format is used  
 after a values will be ok.

3.14156, after a values will be  
 double b; / b = 2.2... → after 15 values will be  
 allowed → IEEE double precision format  
 is used  
 ↓  
 8 bytes

How boolean get converted!  
 it doesn't take any standard to convert.  
 depending on OS or JVM bytes will be allocated.

for windows - 1 byte.

for mac/unix - 2 bytes.

1011 → octal [ 03216842 ]

representing binary 0 0 1 0 1

2 ]  
 2 ]  
 2 ]  
 2 ]  
 2 ]

program

float b = 0b110001000;  $2^6 + 2^2 + 2^0 \Rightarrow 64 + 4 + 1 = 69$ .

variables: The value stored in a variable can change during execution  
 → in value between two literals underscore is allowed.

→ 3.1415 → not allowed    float a = 3.1415f; → allowed

3.14\_5 → allowed

float hepa = 3.1415f; → allowed

class void?

public static void main(String[] args) {

    float a = 3.147.89f;

3147.89f

    System.out.println(a);

↓

    float b = 3.14789E+3f;

3.14789E+3 scientific

notation

    System.out.println(b);

3.147.89e-3

    float c = 3.147.89f; 3f;

3.14789

System.out.println(c);

→ firstName [camelcase]

→ The IEEE 754 standard is widely used for representing floating-point numbers in computing because it provides a consistent and efficient way to handle a broad range of values, including very small and very large numbers.

→ Various formats:-

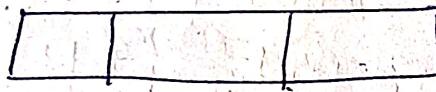
Single precision 32-bit

Double precision 64-bit

Extended precision

→ IEEE → Institute of electrical electronics Engineers.

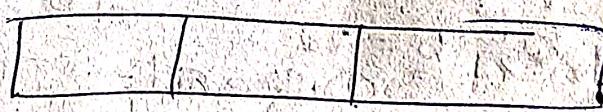
Single precision-



1. 8 23

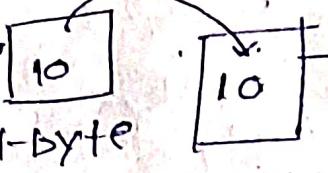
Sign Exponent Mantissa

Double precision-



1 11 sign exponent mantissa

Type Casting - The process of converting the data from one datatype to another datatype.

byte a=10; →  10 get converted short to short.

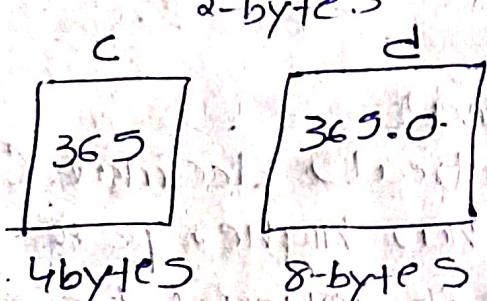
short b;

b=a;

int c=365;

double d;

d=c;

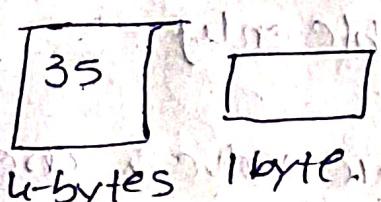


Here 365 of int get converted into double by using double precision. No. of bytes

int e=35;

byte f;

f=e; error.



whenever we are converting larger data type to shorter, it will give errors!

Explicitly - Here we are saying i know i loss the data, but it's okay convert.

f=(byte) e;

→ No error.

double g=36.45466;

int h;

n=g; error.



36.45466

8bytes

Implicit widening-type promotion

Explicit narrowing

h=(int) g;

4-bytes

In java by default all real numbers are taken in double.

float a=3.14f →

double → here what happens means double [higher data type] need to convert lower data type, float

float a=(float) 3.14f

we will get a error if data loss

↓

we have to mention this is not double. float

will occur.

OR

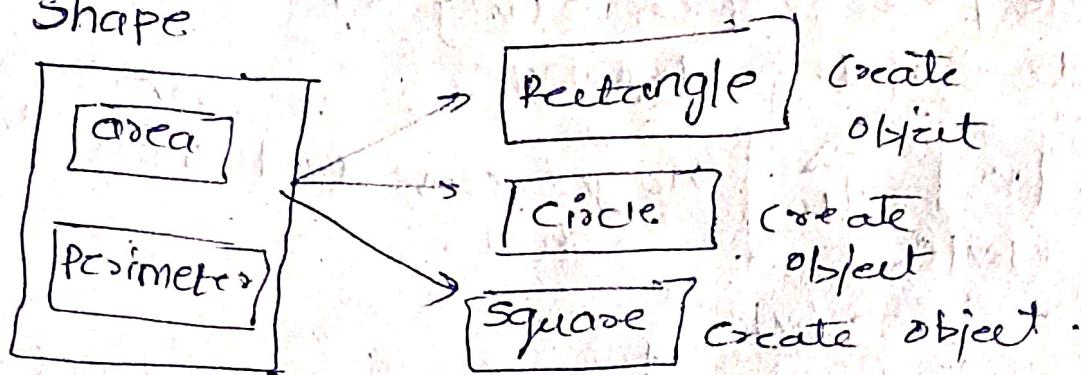
float a=3.14f

→ whenever we are going out of range of integers then we have to give L

long b=2147483649L;

→ JVM is platform dependent.

## Shape



why fin float values for java?

- Because by default java treats decimal (floating-point) literals as double, which has higher ~~precision~~ precision (64-b)
- unlike float here ~~d~~ is optional, because pi here everything will treat as double only.

why l for long in java?

In java, by default numeric values are treated as integer. so it is better to use L ~~or~~, mostly preferred to use L - because L will take as l.

- Because all whole numbers mostly in integer so default will taken.

Why Java is not completely oops:-

1- primitive datatypes:- Java supports primitive data types which are not objects. This means that operations on these types do not go through object model. In pure language everything is an object and all the data types derived from a base class.

2- static methods and variables:- Java allows static methods and variables, it will belong to class rather than an instance of the class which means we can call without creating an object.

3 Non object oriented features-

operator overloading:- Java does not support operator Overloading which is very important in oops

Main method:- which is static here it will call no need to create object.

→ Lack of multiple inheritance.

→ The system is ~~already~~ having many  
information ~~out~~ class.

members such as out which is shortat for output  
name in println → printline.

→ All these System.out.println("") stuff together  
to print stuff in screen.

→ Here print() & println() both functions are there.  
but print means it does not insert a  
new line.

while println() insert a new line.

System.out.print

→ Comments are same as C.

Final:-

Here Final means Constant which means unchanged  
and read-only.

Variables:-

Here everything is same but as well as underscore  
here we use dollar sign also.

Data types:-

→ 2 types

→ Fixed size data type

→ varying size data type [non primitive.]

Fixed:- primitive,

char [2 bytes]

byte [1 byte]

boolean [1 bit] [memory  
occupied by  
boolean is 1].

short [2 bytes]

float [4 bytes]

int [4 bytes]

STM

long [8 bytes]

double [8 bytes]. dependent].

Varying size

String

Array

Class

Object

Interface

enumeration

} no-limit

python doesn't support interfaces & enumeration

String y="palle";

→ Here JVM allocates 10bytes.

→ Since in string each char treated as char and hence 2-bytes.  
10 bytes.

datatype size and capacity-

byte- 1byte → 8bits

$$\text{Min} - (2^{n-1}) = -(2^8-1) \Rightarrow (2^7) \Rightarrow -128$$

$$\text{Max} - [2^{(n-1)}] = [2^{(8-1)}-1] \Rightarrow 2^7-1 \Rightarrow 128-1$$

short- 2bytes → 16bits  
⇒ 127.

$$\text{Min} - (2^{(n-1)}) \Rightarrow (2^{16-1}) \Rightarrow (2^{15}) \Rightarrow -32768$$

$$\text{Max} - [2^{(n-1)}] \Rightarrow [2^{(16-1)}-1] \Rightarrow 2^7-1 \Rightarrow 32768-1$$

int- 4byte (32bits) ⇒ 32767

$$\text{min} - (2^n) \Rightarrow -(2^{32}) \Rightarrow -(2^3)$$

long- You should end the value with 'L'.  
⇒ Continue

float- while declaring float variables

5.75f

double- while declaring the variable end with d ⇒ 19.99d;

If you are taking string input of any number input you must use an extra next line.

### Debugging:-

de=removing

bug=errors/problems

Set of techniques used to remove errors from project.

→ python doesn't support for-each loop.  
For-each loop: - [enhanced for loop or advanced for loop].

\* `for(int i=0; i<arr.length(); i++)`

{

`int x = arr[i];`      normal for loop  
`SOP(x);`

y

for (int x: arr)

In each iteration

the variable YAP-NAME

will hold the value of an element inside Array[i].  
Array starting from the first element.  
This will continue until last element.

→ It provides an easy way, to access/read array elements.

→ It an easy way, to access collection elements.

public class MyClass

{

    public static void main(String[] args)

{

        int[] arr = {10, 20, 30, 40, 50};

        for (int item: arr)

{

            S.O.P("item");

y  
Tz

10

20

30

40

50

~~Advantages~~

~~disadvantages~~

- For each loops are not appropriate when you want to modify the array.
- For each loop do not keep track of index.
- It only iterates forward over the array.
- For each cannot process 2-decision making statement at once.
- Here we don't have any chance to skip any element.

### unicode system:

unicode characters holds 2-bytes, so java also uses 2byte for characters

lowest value: U0000

highest value: UFFFF

more than 9-digit  
we have to use  
long data-type.

Keywords: There are 68 keywords

abstract	default	for	new	static
boolean	do	if	null	strictfp
break	double	implements	package	super
byte	else	import	private	switch
case	enum	instanceof	protected	synchronized
catch	extends	int	public	this
char	final	interface	return	throws
class	finally	native	short	transient
continue	float			
try				

### Data types:

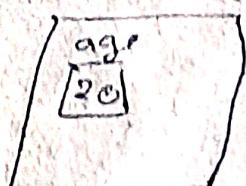
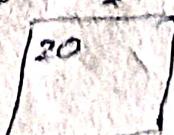
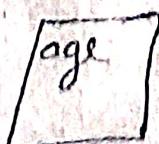
```
int age=20;
System.out.println(age);
System.out.println(20);
System.out.println("age");
```

age=20

= assignment operator

assignment operator

first goes with right side  
value after the right side, left value.



and one

## Scanner Class Methods

- nextInt() → used for integers
- nextFloat() → Float
- nextBoolean() → for Boolean
- nextLine() → used for sentence
- next() → used for string
- reset() → reset the scanner
- nextByte()
- nextDouble()
- nextShort()
- nextLong()
- hasNextLine()
- hasNextInt()

```
import java.util.Scanner;
class public class {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the rows");
        int n = sc.nextInt();
        System.out.print("Enter the columns");
        int m = sc.nextInt();
        int marks[][] = new marks[n][m];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                marks[i][j] = sc.nextInt();
            }
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                System.out.print(marks[i][j]);
            }
        }
    }
}
```

### hasNextLine()

It checks if there is another line of input available. It returns true if there is another line and false otherwise.  
This is often used when reading input from the console or from a file.

```
import java.util.Scanner;
public class Example {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter lines of text (type 'exit' to quit)");
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            if (line.equals("exit")) {
                break;
            }
            System.out.println("you entered " + line);
        }
        scanner.close();
    }
}
```

hasNextInt():- → Checks if the next input is an integer, it returns true if the next token can be interpreted as an `int` and false if not.

### public class Example {

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter an integer");
}
```

while (!scanner.hasNextInt()) {

System.out.println("That's not valid");

scanner.next();

}

int number = scanner.nextInt();

System.out.println("You entered " + number);

```
3  
Static double plusmethod(double x)  
    return x+y;
```

```
3  
public static void main(String[] args){  
    int myNum = plusmethod(8,5);  
    double myNum2 = plusmethod(10.3,6.25);  
    System.out.println("int:" + myNum);  
    System.out.println("double:" + myNum2);
```

→ multiple methods can have the same name as long as the number and/or type of parameters are different.

Java scope: - variables   
                  | local  
                  | non-static (instance variable)  
                  | static or class

```
class A{  
    public static void main(String[] args){  
        public void m1(){  
            Inside class int a; → local variable  
            Outside class y  
        } method.  
    } class A{  
        P.S.V.M(String[] args){  
            static int a=10;  
            public void m1();  
        }  
    }  
}
```

class A{  
 P.S.V.M(String[] args){  
 static int a; → static variable  
 public void m1();  
 }  
}

Inside a  
class but y  
outside method

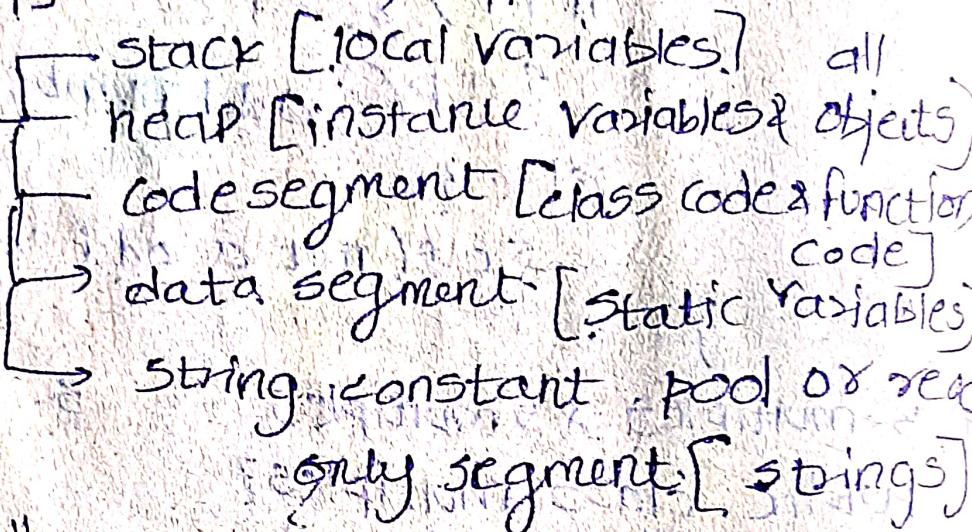
→ Java does not support global variables,  
class variables / static variables are  
placement for global variables.

else {

S.O.P ("not a prime number")  
which is divided by operating system.

Memory Java:-

22 or 23 partitions



Purpose of arrays:-

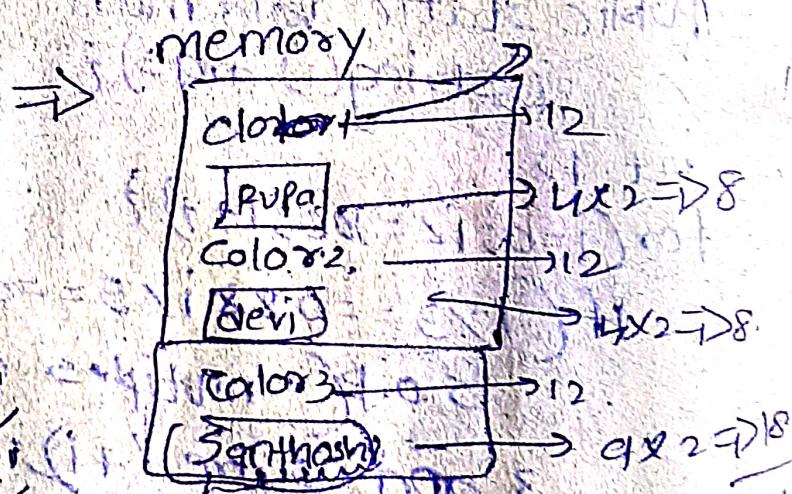
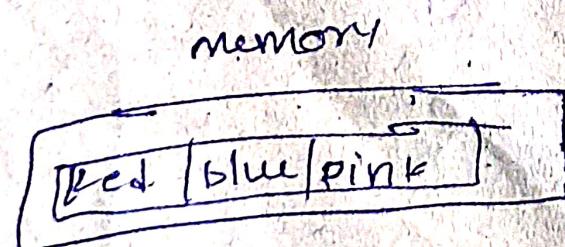
String color1 = "popa";

String color2 = "deri";

String color3 = "Santhoshi";

arrays

String[] colors = {"red", "blue", "pink"};



memory that much

memory not stored if allocated

we don't need for using arrays. the variable names also memory allocated.



## unicode:-

unicode is a universal international standard character encoding that is capable of representing most of the world's written languages.

## Why Java uses unicode system:-

ASCII (American standard code for Information interchange), for United States.

ISO 8859-1 for western European language.  
KOI-8 for Russian.

## problems:-

→ A particular code value corresponds to different letters in the various language standards.

→ The encodings for languages with large character sets have variable length i.e. some characters are encoded as single bytes, others require two or more bytes.

→ To solve this problem, a new language standard was developed i.e. unicode.

→ Unicode character needs 2-byte,  
lowest value: 100000  
highest value: FFFFFF

## bitwise AND & &

Bitwise & is used for bitwise operations.

$$\begin{array}{r} 5 \& 3 \\ 0101 \\ 0011 \\ \hline 0001 \end{array}$$

result is decimal value

Whereas, logical AND is used for composition of logical operations. results true or false.