1. components of JDK:

Javac:

The java compiler is a key component of JDK that transforms java src code into bytecode.

JVM:

JVM is the runtime engine that execute java bytecode.

It provide an abstraction ~~betⁿ~~ layer betⁿ java applⁿ & os.

JRE:

JRE is a subset of JDK that include JVM and essential class libraries.

API libraries:

Provide vast collection of pre built classes and methods that simplify common programming task.

e.g. GUI, database connectivity, input/output

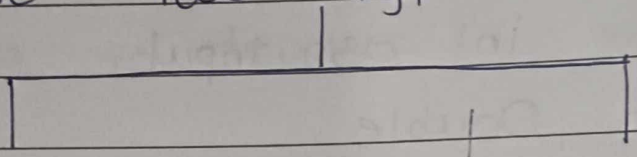
jdb:

java debugger is powerful tool for debugging java applⁿ.

Java modifiers:

Keyword that change the properties of classes, methods or variables.
defines the scope of ~~variable, in the class or of~~ field, class, method, constructor.

There are two typese of Java modifier



Access modifier
Private - within a class
Public - everywhere
Protected - inheritance
Default - within package

Non-access modifier
Static
final
abstract
volatile
strictfp
native
synchronised
transient
~~interface~~

2) Java Buzzwords:
Simple - easy to learn.
object oriented - based on concept of object containing data & code.

Portable - can run on any platform with a JVM.

Platform independent -
"write once, run anywhere."

Distributed : Designed for distributed environment.

High performance : Supports native code and compilation to native code.

Secure, Distributed, Multithreaded, Dynamic, Robust, Portsimple.

4&8. Meaning of main method

Public static void main (String Args[])

Public - it must be accessible to the JVM.

Static - allows the method to be called without creating object.

Void - This is return type that means the method does not return any value when the program ends

Main : name of the method. This method name is searched by JVM as starting point.

String args[] - it is an array of strings of java string class.

4) Java virtual machine threads:

When java appl^n starts, several threads are automatically created and started by the JVM.

Main thread: execute the main method.
Finalizer thread: responsible for running finalize() method of objects that are eligible for finalization.
Garbage collector: To perform garbage collection

Print	Println	Printf
Print is non static overloaded method of io.Printstream class	Println is non Static overloaded method of io.print Stream class	It is non static overloaded method of java.io.printstream class.
return type is void	return type is void	return type is printstream which is class.
write data to dest but keep cursor on same line	write data to dest but move the cursor on next line	write formatted data to dest^.

7. Data types:

Data types specify the different sizes and values that can be stored in the variable.

Data type describing following properties:

- Memory allocation -
  - how much memory is required to store data.

- Type of data -
  defines what kind of data can be store.

- range of data:
  It defines range of value that can be store in variable.

9. Classification of Data type

- Primitive
  - boolean
  - byte
  - char
  - Short
  - Long
  - INT
  - Float
  - double
- Non Primitive
  - Interface
  - Class
  - Enum
  - Array.

Wrapper class java.lang.*

10.

Type	Description	Default	Size	Range
boolean	true or false	false	1 bit	true false
byte	2'complement int	0	1 byte	-128 to 127
char	Unicode character	\u0000	2 byte	ASCII (0-255)
short	int	0	2 byte	-32768 to 32767
int	int	0	4 byte	-2147483648 to 2147483647
long	int	0L	8 byte	
float	IEEE 754 float	0.0f	4 byte	upto 7 decima
double	——"——	0.0d	8 byte	16 decimal

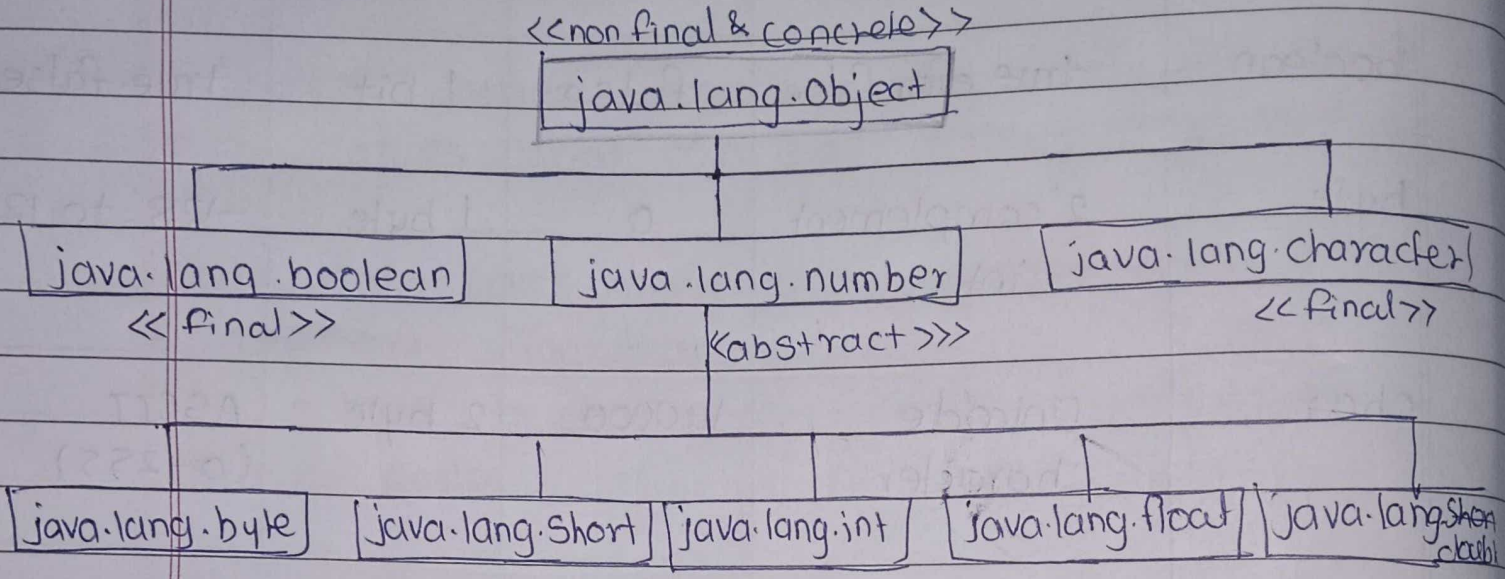
8. Comments in java:

single line comment
Syntax : //

multi line comment
Syntax /* */

documentation
Syntax /**
*/

11. Wrapper class hierarchy.



12. Overview of string.

- String is not a primitive data type in java it consider non primitive / reference type.
- It's a final class declared java.lang package.
- String is basically an object that represents sequence of character values.

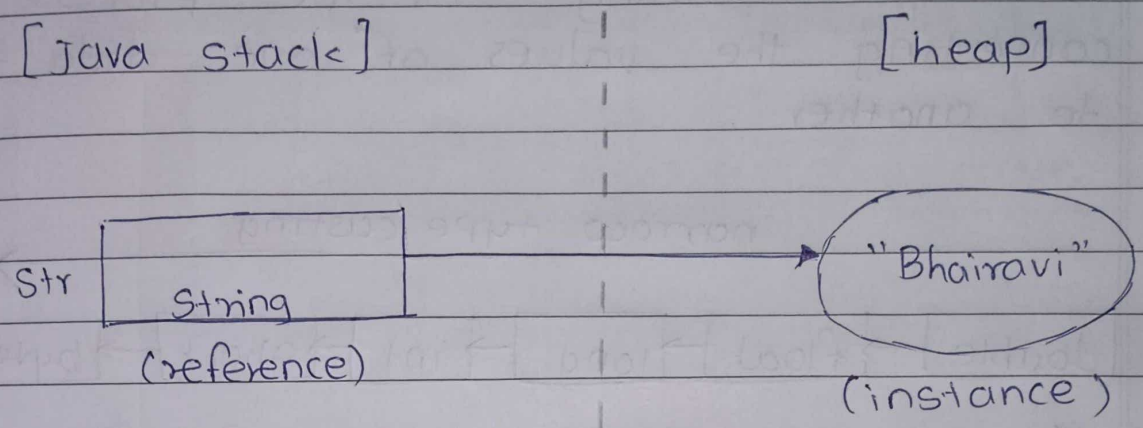
- we can create instance of string using new operator as well as without new operator.

using new operator:

String str = new String ("Bhairavi");

here str called as obj reference

13) Memory representation of string.

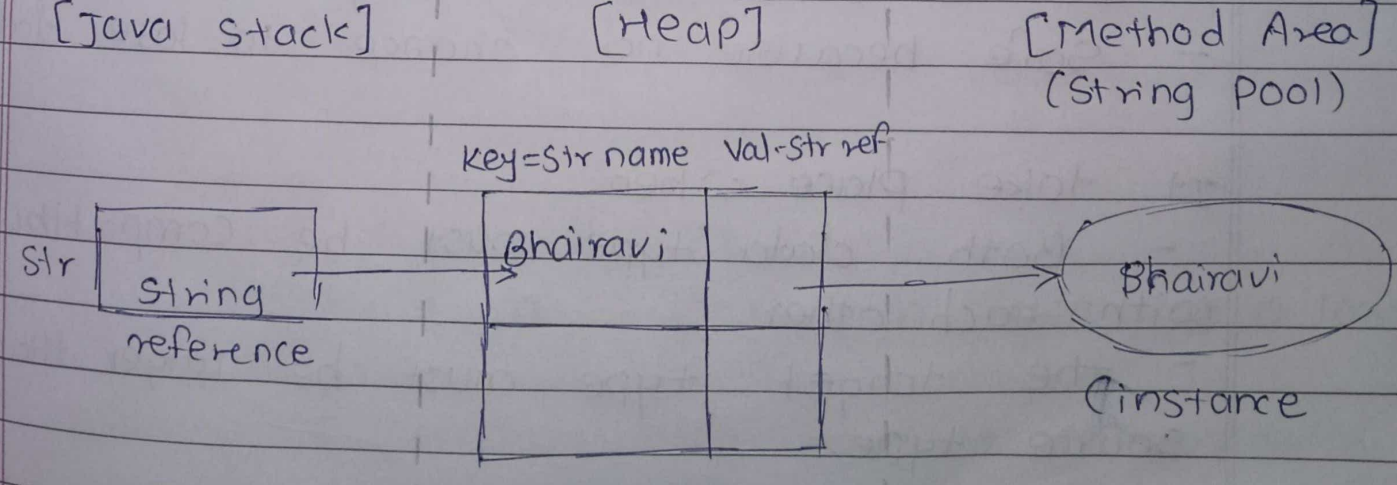


String instance without new operator:

String Str = "Bhairavi"

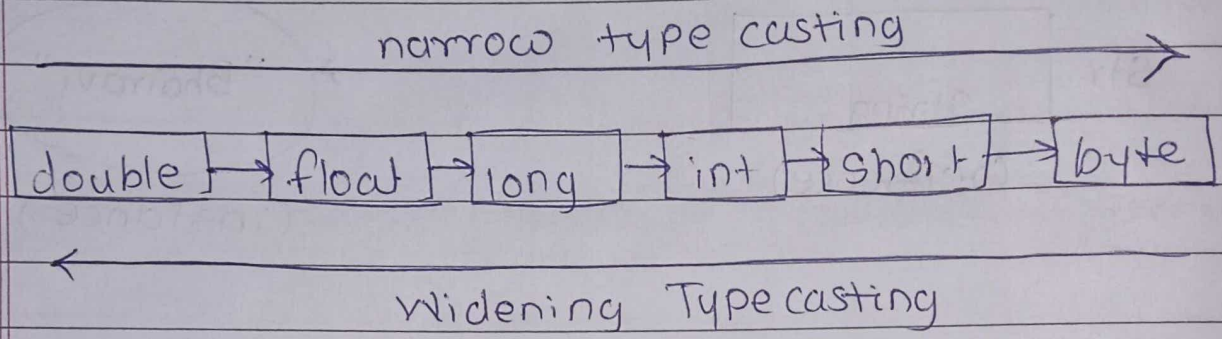
here Str is called as reference.
"Bhairavi" is String content / literal.

Memory representation.



14) Type casting:

Type casting is the process of converting the values of one data type to another.

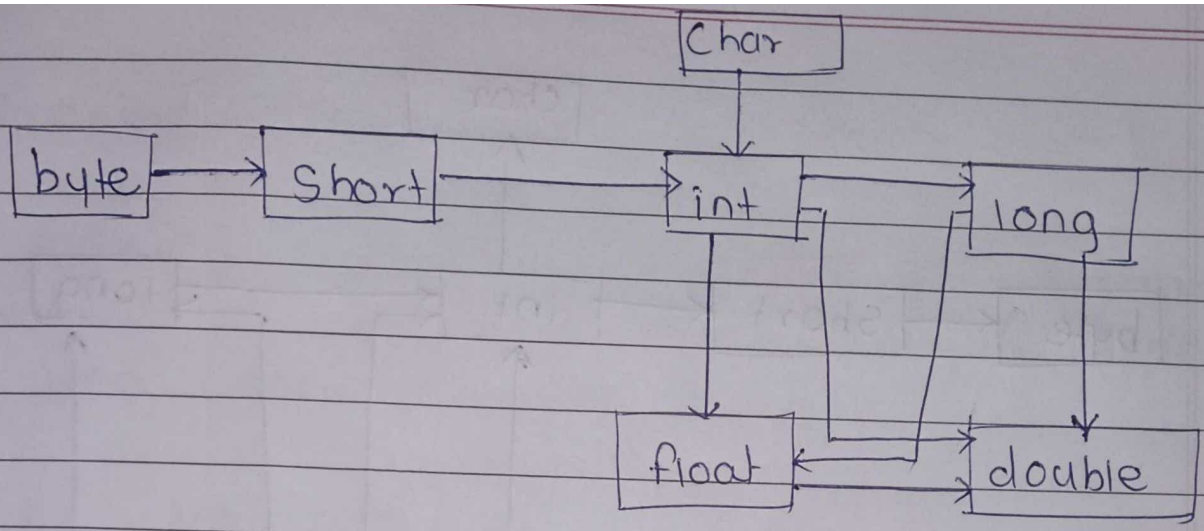


Types of type casting.

15) 1) widening conversion:-

- converting lower data type into a higher one is called coidening type casting.
- It is also known as <u>implicit conversion or casting down</u>.
- It is done automatically.
- Safe because no chance to lose data.

It take place when:
- Both data type must be compatible with each other.
- the target type must be larger than source type.



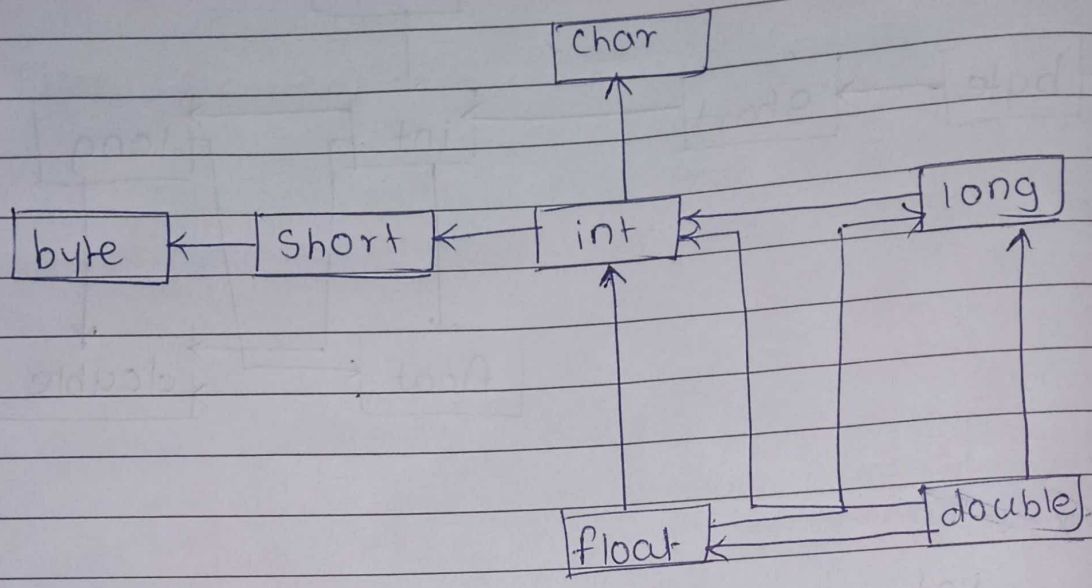
ex :

```
int x = 97

long y = x;

float z = y;

Sop (" Before conversion int:" +x);
Sop (" converting into long " +y);
Sop (" conversion into float " +z);
```

op: 97 // int value
97 // long value
97.0 // float value

(6) Narrowing Type conversion :

- converting a higher data type into a lower is called narrowing.
- also known as explicit conversion or casting up.
- done manually.



e.g.

```
double d = 177.64;

long l = (long) d;

int i = (int) l;

sop ("before conversion: " +d);

sop ("After conversion into long: " +l);

Sop (" int : " +i);
```

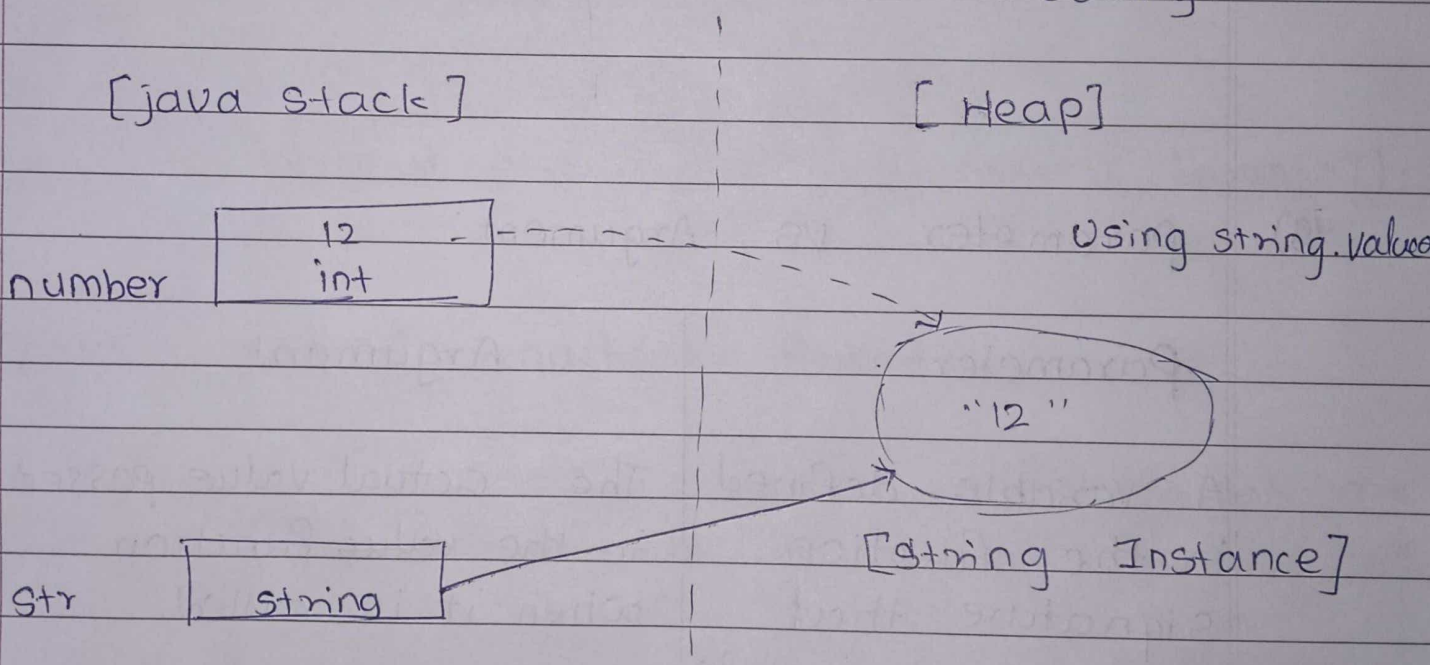
o/p:

```
before conversion : 177.64
 into long : 177
 into int : 177
```

17. Boxing conversion

Boxing is a process of converting value of variable of primitive type into non primitive type

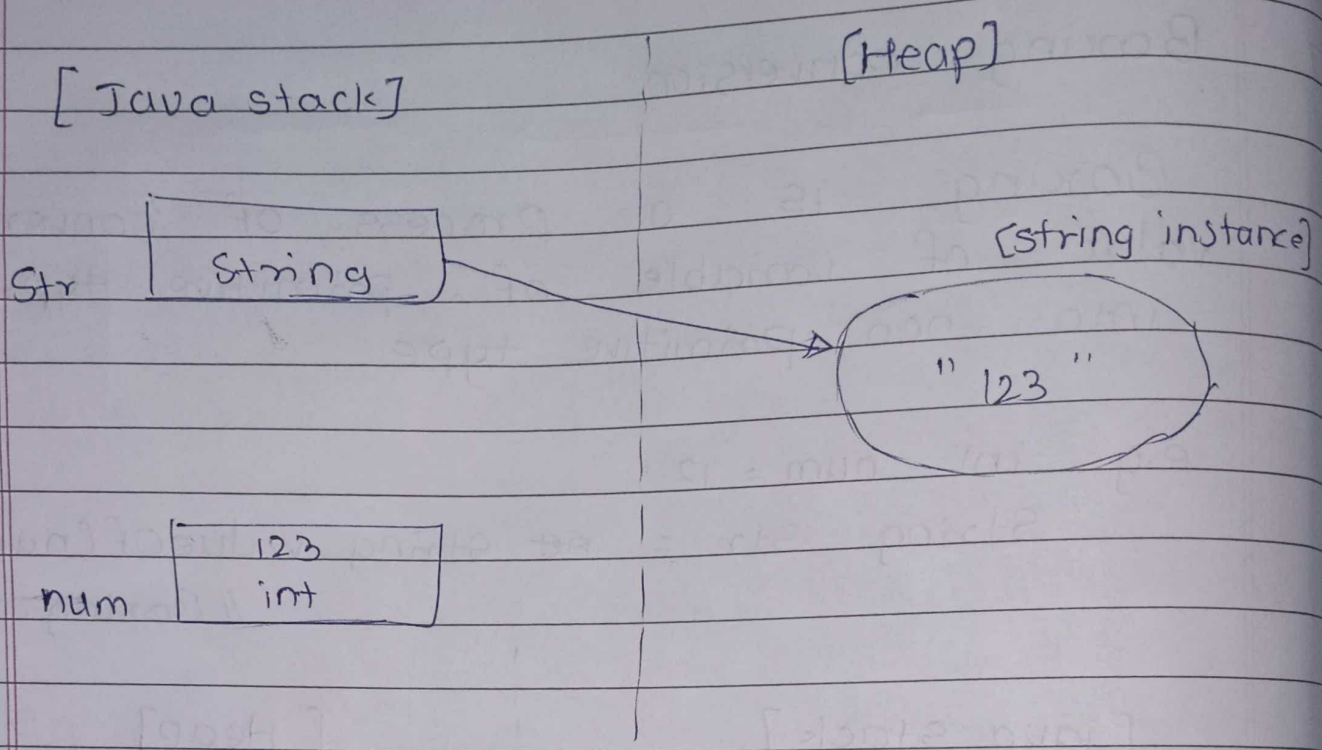
e.g int num = 12;
String str = String.valueOf(num);
// Boxing.



18 Unboxing conversion.

Unboxing is the process of converting value of non primitive type into primitive type.

String str = "12";
int num = Integer.ParseInt(str);



19) Parameter vs Argument:

Parameter	Argument
A variable defined in the function signature that acts as a template for the expected input.	The actual value passed to the ~~value~~ function when it is called.
Parameter are local to method where they are defined and can have default value.	Argument can be mutable immutable depending on their data type. can exist outside the fun'

20. Command line arguments.

java command line argument is an argument that passed at the time of running the java program.

The passed argument can receive in java program and it can be used as an i/p.

Syntax.

```
System.out.println("1st argument is :"+args[0]);
```

4) Java virtual machine threads:

When java appln starts, several threads are automatically created and started by the JVM.

Main thread: execute the main method.
Finalizer thread: responsible for running finalize() method of objects that are eligible for finalization.
Garbage collector: To perform garbage collection