Process $\boxed{\text{C}} \xrightarrow{1+2} \boxed{\text{machine}} \rightarrow \boxed{\text{JVM}} \rightarrow \boxed{\text{o/t}}$

Java programming

* Easy to read & understand.
* used for Software developmnt
* platform independent.
* uses Java virtual Machine (JVM).

* Java is a object oriented programming lang that is used in a distributed envnmnt on the intrnt.
* It is an high-level lang, easy to read & ~~[illegible]~~ undrstand.
* used to develop Software for devices.
* It is notonly used in Comp bt also in mob & electronic devices.
* Java files are convrted to ~~[illegible]~~ bit code format using a compiler & then executed by a Java interpretr.
* Java code is ran on JVM, which provides a runtime envrnmnt.

→ Oops concept (object oriented progming)

1) class (blue print of object)
(grp-ആയി എടുത്താൽ പറയുന്നതിനെ class വിളിക്കും)

2) object
3) Abstraction
4) Encapsulation
5) Inheritence
6) polymorphism.

1) class :

eg → human → men, women.

$\underbrace{\text{vehicle}}_{\text{cls}} \rightarrow \underbrace{\text{bike, car}}_{\text{obj}}$

2) object :

class-ന്റെ ഓരോന്നിനെയും obj വിളിക്കും.

2 types → state & behaviour.

state ↓ identity

eg → human's identity (name, age)

behaviour ↓ actions of state.

eg → mobile communication :
we call (state) → its objects →
call mute, recording (behaviour)

1) class :

* A class is a abstract description of a set of object.
* A class contains the description of all the behaviours of the obj that it represnts.
* Individual behaviours of a cls → its methods

2) objects :

* obj in oops concept is similar in the obj in real world.
* real world obj shares 2 charactristic
  → state
  → Behaviour.

* State
- State of a obj consist of a set of data field with their current values.
- A obj state is defined by the attributes of the obj & by the values.

* Behaviour
Behaviour is hw an obj acts & react in terms of its state changes and msg passing

3) Abstraction :

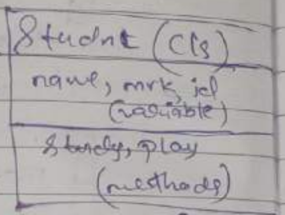
ആവശ്യം ഉള്ളതിനെ കാണിക്കും, അനാവശ്യത്തിനെ hide ചെയ്യും

eg → App installing on mob.,
mob commun from A to B.

* • It is the process of hiding the details & exposing only the essential features of a particular concept/obj

4) Encapsulation :
(Binding variable & ~~m~~ methods in a single entity)

* It is a way to restrict the direct access to some components of an obj, so users cannot access state values for all variables of a particular obj.



* It can be used to hide both data members & data ()s / methods associated with an cls / obj.

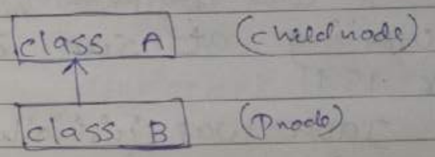
5) Inheritance : (I)
(Acquiring (pro) frm parnt cls to child cls)

* In obj oriented prgming, Inheritance is a way to form new clses using clses that ~~are~~ hve already ~~being~~ been defined.

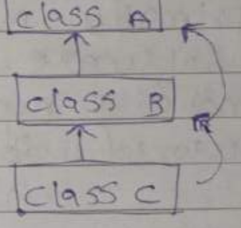
* It is process of by which the obj of 1 cls ~~aquaire~~ accure ~~acqure~~ the proprty of objects of anthr cls.

* 2 types :

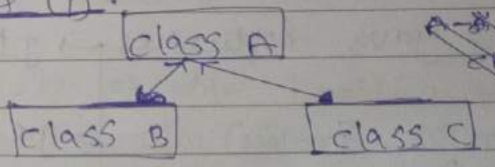
1) Single (I) :



2) multi level (I) :



3) Hierarchical (I) :
(1 of p node in 2 child node)



~~A-B & C~~
~~child node~~
~~[illegible]~~

6) polymorphism :
* It is the ability to take more than 1 forms (types).
* It takes an imp role in allowing obj hving diffrnt internal str to

Share the Same external intrface.

→ History of Java:

It was founded by ~~James~~ James Gosling at Sun microsystm in the yr 1991

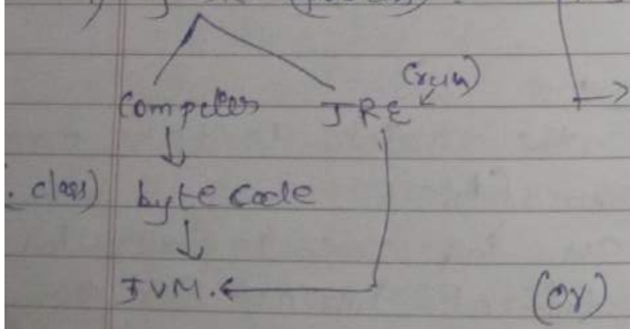
Java was initially called as Oak. & ~~being~~ was renamed as Java in the yr 1995.

The primary motivation is to develop Software for electronic devices & secondary motive was to develop world wide web.(WWW)

→ Java working:

3 Java Softwares → JDK (Java development kit)
→ JRE (Java runtime envrnmnt)
→ JVM.

1) JDK (process).



s/w → Software.

Sample.java → compile → Sample.class
→ window (window JVM)
→ Linux (Linux JVM)
→ Unix (Unix JVM)

a) JDK : (Java development kit)

* It is a Software development envrnmnt which is used to develop Java appli- & applets
* JDK is a implimentation of Java platforms. like Standard edition Java platform, enterprice Edition J.P, micro edition J.P.

b) JRE : (Java runtime envrnmnt)

* It is a set of s/w tools which are used for developing Java appli..

c) JVM : (Java virtual machine)

* It is called a virtual machine, bcz it doesn't physically exist.
* It is a Specification that provides

a runtime envnmnt in which Java bytecode can be excecuted.

* It can also run those prgms which are written in other lang & compiled to Java byte code.

*

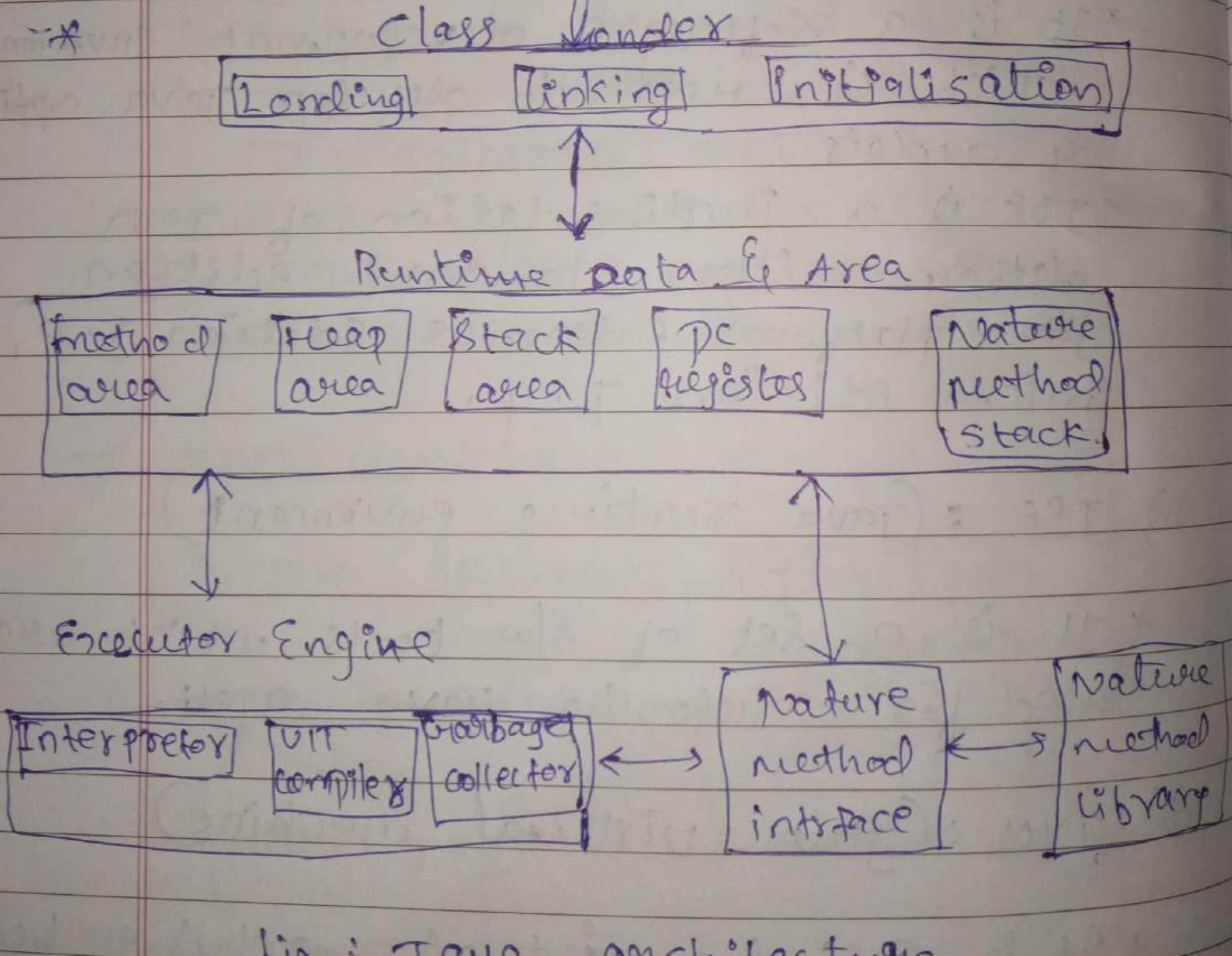


fig : Java architecture

⇒ Basic Syntax :

1) case sensitive → uppercase / lowercase.
2) cls name → 1st letter capital → Welcome
3) method name → 1st letter small → welcome.
4) prgm file name → class name = file save name.
5) identifiers → cannot be keywords
6) commnts → // comments.

I case sensitive:

Java is case sensitive. (i.e) identifier with uppercase & lowercase would hue diffrnt meaning in Java.
eg → Hello and HEllo treated as 2.

II Class name:

For all clsname 1st letter should be in uppercase.
If several words are used to form a name of the cls each innerword 1st letter should be in uppercase
(eg → FindSum)

III method names:

It should start with lowercase letter.

IV pgm filename:

Name of the pgm file should exactly match the cls name.

V Java identifiers:

Names used for clses, variables & methods are called as identifiers.

A keyword cannot be used as the identifiers.

VI Comments:

They are like helping text in ur java pgm.

They are ignored by the compiler.

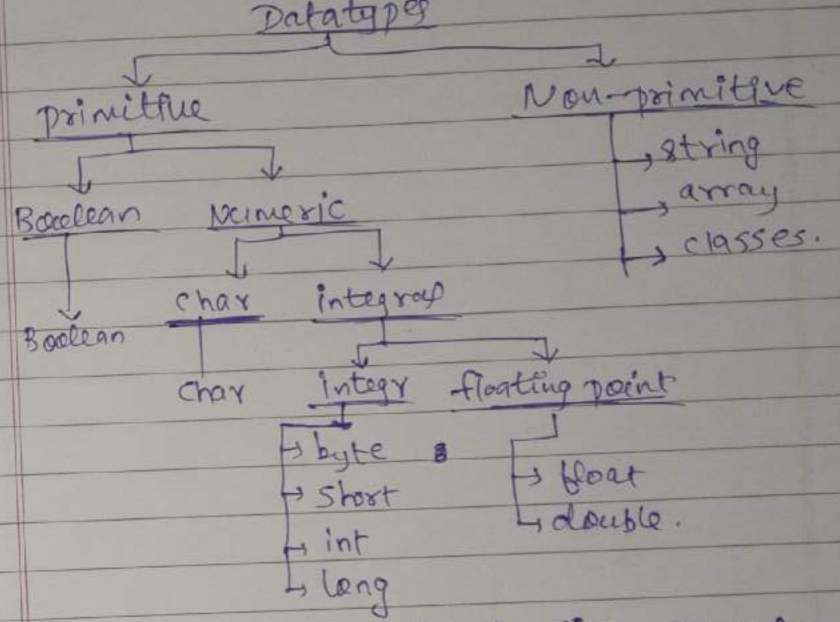
* Sample Java pgm:

```
import java.util.*;
public class Welcome {
    public static void main(String[] args){
        System.out.println("welcome");
    }
}
```

* 1st line → import the package (util).

* String[] args → array of arguments.

Datatypes



primitive → fixed size [illegible].

* Datatypes defines the storage method available for representing info along with hw the info is interpreted

* Datatypes are ~~[illegible]~~ linked tightly to the storage of variables in memory bcz the datatype of a variable determines hw the compiler interpretes the content of the memory

→ Literals =
value assigned to a variable.
eg = int i=10
here 10 → literal

5 types:

1) integr (L) → octal (0-7), hexadecimal (0-9 & A-F)
2) floating (L) → 25.07
3) character (L)
4) Str (L)
5) Boolean (L) → T/F.

literals → program elements that are used in an invarient manner.

- Also → Constnts.
- Literals can be no., char / str.
- Numerical (L) includes int, floating point & booleans.
- 5 types:

a) Integer (L) =
They ar the primary (L) used in java programming which are in different formats (octal, hexadecimal & decimal (10)).
This formats Correspond to the base of the no. system used by the (L).

b) Floating (L) =
It represnts decimal no. with fractional parts.
They can be represnt eithr in standard / scientific notations.

c) Char (L) =
It represnts single unicode char & appear within a pair of single quatation mrk

d) Str (L) =
represnt multiple char & appear within a pair of double quotation mrk.

e) Boolean (L) =
They are simple & only 2 logical values that a boolean value can hue:-that are T & F.