19/1/23

Application

* Mobile App development
* Game development
* Web " etc

Java → JDK, JRE, JVM (lang. archit. and multithreaded, obj oriented.

→ Java is a Robust Lang, Case sensitive, dynamic

Founder → James Gosley, in 1992.

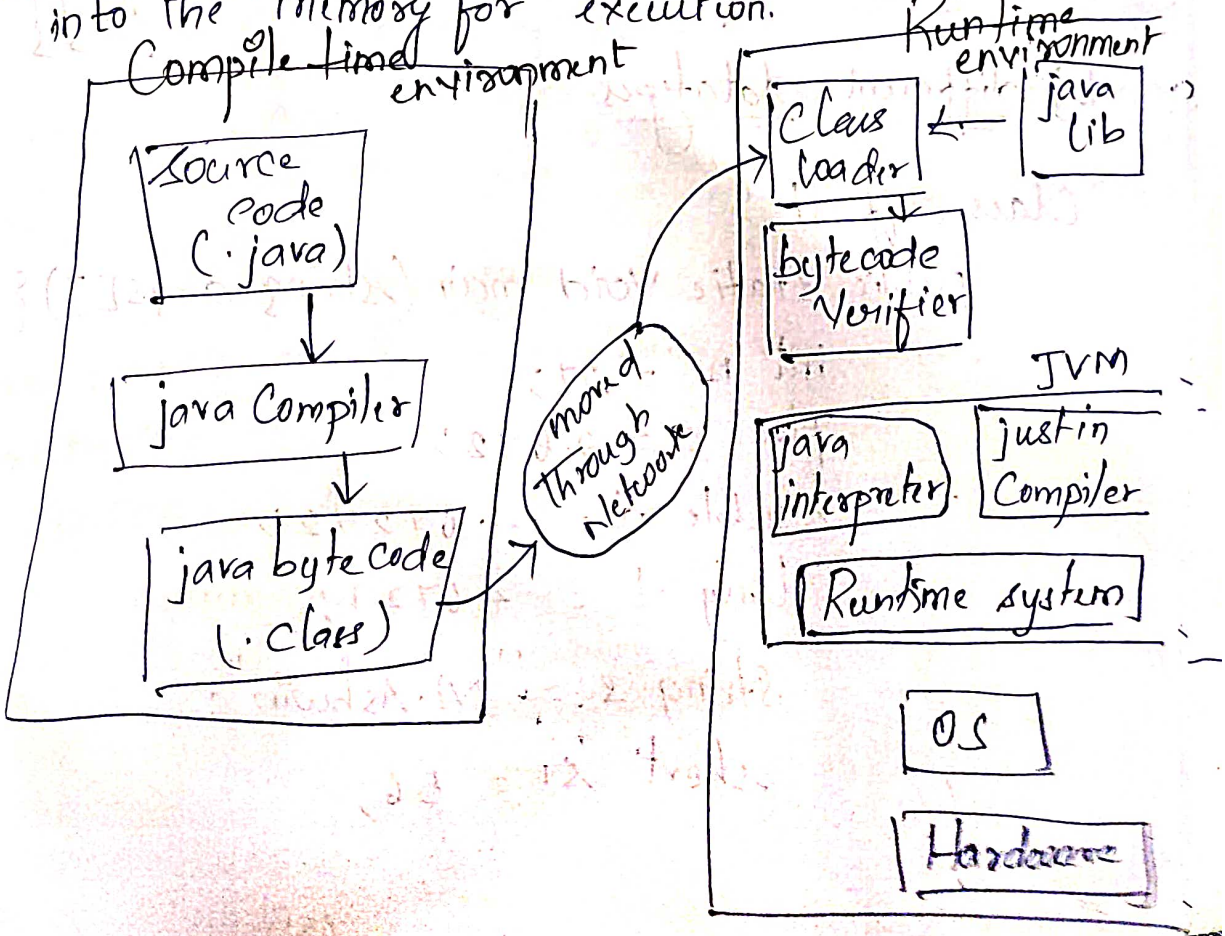
How to Compile a java program?

giving class name in Command line,
as javac (class name) to compile
and java class name to execute it

platform independent → can be implemented in any. platforms.

secure in it → classes are secure if they are in private so they are secure and safer in it

class Loader ⇒ loads the user defined and lib class into the memory for execution.



Java program

1. datatype declaration

a] same ones

```
class Samedatatype {
    public static void main (String args[]){
        int age = 15;
        long int rollno = 20210023;
        long int mobile = 95010;
        System.out.println ("age =" + age);
        System.out.println(" rollno = " + rollno ); } }
```

b] different datatypes

```
class diffdt. {
    public static void main (string args[]) {
        int n = 27;
        float a = 2.672;
        double d = 4.672721
        long l = 46721;
        String s = "M.Ashwin";
        short s1 = 56;
```

```
system.out.println("integer = " + a);
system.out.println("float = " + a);
system.out.println("double = " + d);
system.out.println("long = " + l);
system.out.println("string = " + s);
system.out.println("short = " + s1); }}
```

Token's → Collection of Keyword, datatype, operator, identifier and Punctuation symbols. (54)

Keywords are Reserved ones → Can't be used in naming

datatype → primary = int, char, double, float, short, void

↘ secondary = Array, Collections, meathods. [function in C]

operators → Arithmetic, Logical, Relational, Bitwise, Unary, Binary, Ternary, misalaneous

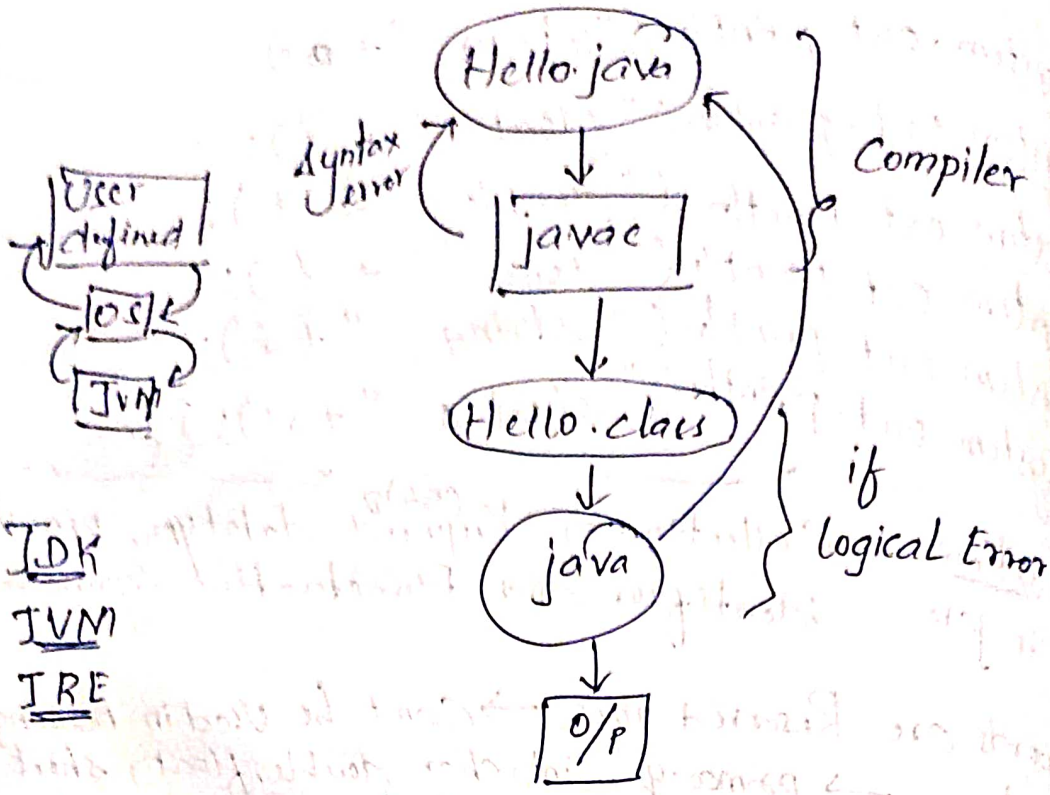
Expressions => Collution of Variables, operands, operator

Variable => Quantity that can change during Execution

Const => " that never change.

Hello.java

```
import java.io.*; { // package of io
class Hello {
    public static void main(String args[]){
        system.out.println("Hello"); }}
```



1. JDK
2. JVM
3. JRE

Expression's => x = a+b
↓ ↳ operand
Variable

(1.) x++ => postincrement x = (x)+1 (1, 2)

(2.) ++x => preincrement x = (x+1) (2, 1)

(1.) int x = 100, y; y = ++x

System.out.println y = (x) = x+1 (2, 1)
(x) // 101
// (y) // 101

(2.) int x = 100, y; y = x++

System.out.println (x) // 101
(y) // 101

```
class Avg {
    public static void main (string args[]){
        int x =
```

Features in OOPS

classes, object, abstraction, binding, encapsulation, inheritence and polymorphism

- class is collection of similar obj.
- all tangable things are obj.
  - ↳ real time entity
- both class & obj
  - class name
  - attributes
  - operations.
  - methods.
- obj ← instance of a class
  - ↳ name
  - ↳ id.
  - ↳ state
- Combines data and methods.
- Var in class is instance Variable.
  - ↳ accessed by methods.
  - ↳ they are static or final Var

Minor elements

Abstraction (Hidden)

- ↳ expressed in higher & lower level.
- ↳ act of Representing essential characteristics without background detail and differs on the Viewer. (person to person)

2. Encapsulation → a method which acces a private class Var or function in it.

④ process of wrapping up of data in the real world.

3. Binding: process of associating a method by creating a object and then call that method to do the operations in it

∴ obj → calls method (public) → call's private class data

process of binding.

maybe the referance of the class

→ Code Reusability

Inheritence: When we have more than one class and procus of deriving from one class to another, class 1 = super class
class 2 = derived ,,

java support
- one — many ⇒ Hierarchical
- one — one ⇒ Single
- one — one, one, one ⇒ multi level

many to one ⇒ multiple & Hybrid } java Using interface can be support.

polymorphism: greek word [many-form]:

- ⊕ it is the ability to take many forms.
- ⊕ two types
  - ↳ static & dynamic → runtime process
    - ↳ compile time process
      - (method overloading)
      - (operator overloading)
    - ↳ (overriding) abstract class in
      - some fun with same parameter and same prototype

Java program

1] Hello World

```
public class Hello world {
    public static void main (String args[]){
        System.out.println (" Hello World");
    }}
```

2] Check Pass or Fail using (if-else)

```
import java.util.Scanner;
public class Pass or fail {
    public static void main
        (String [] args){
    Scanner sc = new Scanner (System.in);
    System.out.println (" enter ur mark : ");
    int mark = sc.nextInt ();
    if (marks >= 50){
        System.out.println(" Passed...");}
    else { System.out.println ("failed....");}}}
```

if mark >= 50 pass
else fail

3] check odd or Even Using (if-else)

```
import java.util.Scanner;
public class oddorEven {
    public static void main (String args[]){
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter ur no: ");
        int no = sc.nextInt();
        if (no % 2 == 0) {
            System.out.println ("it is even...");
        else {
            System.out.println ("it is odd...");
        }}}
```

4] Print Number in Word Using (nested-if and switch)

```
import java.util.Scanner;
public class NoinWord {
    public static void main (String args[]){
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter ur no (1-9");
        int num = sc.nextInt();
        if (num >= 1 && num <= 9){
```

```
switch (num){
    Case1:
        System.out.println ("One"); break;
    Case 2:
        System.out.println ("Two"); break;
    Case 3:
        System.out.println ("Three"); break;
    Case 4:
        System.out.println ("Four"); break;
    Case 5:
        System.out.println("Five"); break;
    Case 6:
        System.out.println("Six"); break;
    Case 7:
        System.out.println("Seven"); break;
    Case 8:
        System.out.println("Eight"); break;
    Case 9:
        System.out.println("Nine"); break;
    default:
        System.out.println("Invalid num");
            }
else {
    System.out.println("Num Out of Range (1-9)");
    }}}
```

5] Day Printing Using Numbers.