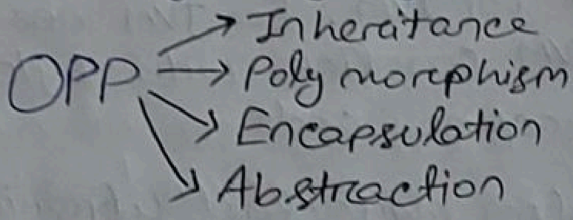
Java

Q) What is Java?

→ Java is a platform Independent Language.

→ And It is a highly popular Object oriented Programming Language.



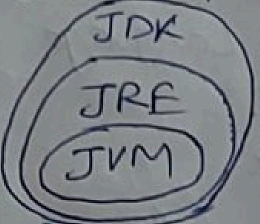
→ The major advantage of Java is it's Portability.

WORA → (Write once, Runs Every where.)

→ Suppose you write a java Program on your mobile. Now you can run that program on your laptop, Desktop.... anywhere. This is called portability.

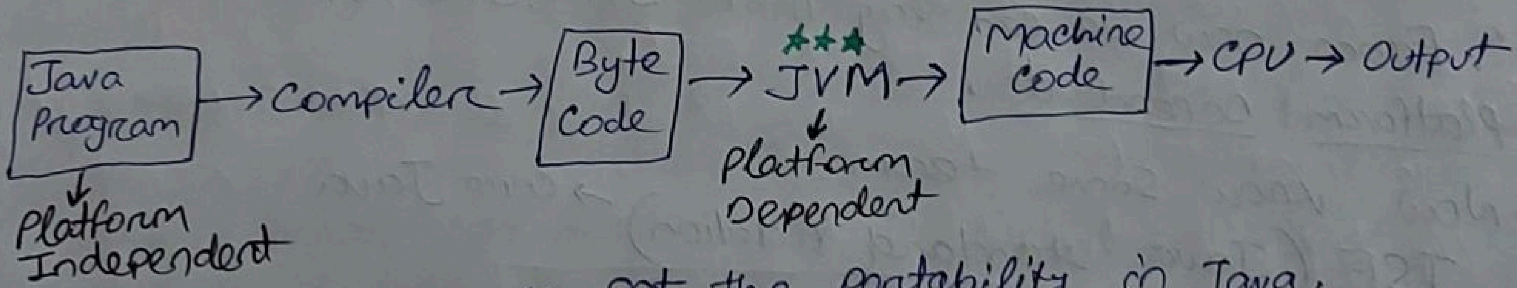
In Java there are 3 main components.

(i) JVM
(ii) JRE
(iii) JDK



JVM (Java virtual Machine)

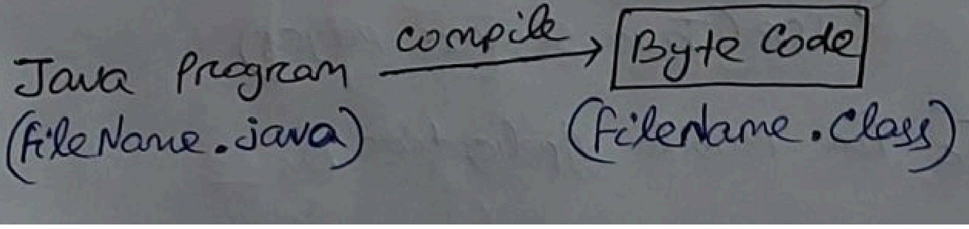
↳ It's an abstract machine.
It doesn't exist physically.



→ Because of JVM we got the portability in Java.

→ JVM has JIT (Just in time) compiler.

→ JVM takes byte code and convert it into machine code



(Now we can able to run this byte code in any platform)

JRE (Java Runtime Environment)

It two things (i) JVM
+
(ii) class Liabraries   ex:- Java . Math
Java . util

→ So when the Byte code is given by compiler, inside that some libraries are present which need to be resolve. That power is not present in JVM. So JVM nee some class libraries so that JVM can link with byte code and convert it into machine code.

→ So JRE contans JVM along with class Libraries.
So JRE can run any Java program.
JVM single handlely can not run any program.

→ But JRE can not write code.

JDK (Java Development kit)

→ As JRE can not write code, So we need JDK..
→ It has programming Language Rules.
→ It has compiler.
→ Also it has Debuger.

$$\therefore JDK = \underset{\left(\substack{JVM \\ + \\ class\ lib.}\right)}{JRE} + \substack{P.L\ Rules \\ + \\ compiler \\ + \\ Debugger.}$$

∴ From now we can say that JVM, JRE, JDK all are Platform Dependent.

→ Now know some term.
JSE (Java standard Edition) → core Java
JEE (Java Enterprise Edition)
JME (Java micro/mobile Edition)

JEE = JSE + API like → transactional API (Rollback commit ....)
→ Servelets
→ JSP (Java server pages)
↳ used for to build large Applications.

JME = It provides APIs for mobile Application.

→ • main () is the starting point of the Program.

Java Program $\xrightarrow{\text{Compiler}}$ $\boxed{\text{Byte Code}}$ → JVM calls •main ()

Public static void main ( )

Call from Anywhere

Return type

when we made a method/function static then we can call that function using className. (Not like we have to create Object then we can access)

ex:-

```
class A {
    Static method1 (){
        :
    }
    int method 2 ( ) {
        :
    }
```

```
A obj = new ~~A~~(); 
obj. method 2 ();
A. method 1 (); // Here we call the method1 using
                   className. Bcz of static.
A. method 2 (); X  It is not a static method, so we need
                   an object to invoke this.
```

Q) Why in Java, a single file can have only 1 public class?

① Main method should be inside public class.

BCZ JVM invoke main method using className (Bcz of static method JVM don't need to create an object) And the class should be public bcz JVM is outside the package of that class. By making it public, we make the class to be accessible from anywhere.

② Public class name should be same as file Name.

Employee.java

```
Public class Employee
{
  :
}
```

Employee.java

```
Public class manager
{
  :
}
```

✗

Why ?

Let's suppose Employee.java file hass 100 of public class. So to run a java Program JVM needs to find .main() method. So If there are 100s of public class then how did JVM know that in which class the .main() is present?

So that's why in a single file only 1 public class is need to be present and It should have same name as file name.

→ So both the 2 points are related to each other.

<u>How float and Double values stored in memory</u>?

```
float a= .7f;
system.out.println((double)a);
```

So when print the value of a it gives 0.699999799 like this but not exact 0.7.

So we need to know How the float is stored in memory?

float : 32 bit IEEE 754

1 Bit	8 bits	23 bits
Sign	Exponent	mantissa/significant

ex:-1 4.125 f

Step 1 convert it to Binary.

4 → 100

.125 → .001

4.125f = 100.001

```
2 | 4   0  ↑
2 | 2   0  |
2 | 1   1
```

0.125 × 2 = 0.25

0.25 × 2 = 0.5

0.5 × 2 = 1.00

Step-2: we need to make it in the form of

$$\boxed{(1.xxx) \times 2^{\text{exponent}}}$$

$$(100.001)_2 \Rightarrow 1.\underbrace{00001}_{\text{mantissa}} \times 2^{2 \rightarrow \text{exponent}}$$

Step-3: Add bias to the exponent

→ $\boxed{\text{For float the bias} = 127}$

$$\underset{\text{bias}}{127} + \underset{\text{exponent}}{2} = 129 \text{ (exponent)}$$

Q) why we need a bias here? So if need to convert 0.00101 to the form of $1.xxx \times 2^{\text{en}}$

$$0.00101 = 1.01 \times 2^{-3}$$

So in IEEE they don't have the rule to store -ve in exponent, So they add bias (127)

So here 127 − 3 = 124

Step-4

		$2^{-1}\,2^{-2}\,2^{-3}\,2^{-4}\,2^{-5}$
0	$\underbrace{1000001}$	$\underbrace{00001}$ 00000....
sign	exponent	mantissa

This is how float is stored in memory of IEEE, format.

So reverse binary back to float.

$$\boxed{(-1)^{\text{sign}} \times (1 + \text{mantissa}) \times 2^{e - (127) \rightarrow \text{bias}}}$$

$$(-1)^0 \times \left(1 + \frac{1}{2^5}\right) \times 2^{129-127}$$

$$= 1 \times (1 + 0.03125) \times 2^2$$

= 4.125 (Same as previous)

(When we fetch a binary and convert it to float it is same as earlier)

Now take another example of 0.7f.

Step-1: convert to binary.

0.1 0110 0110 0110 0110......

$$\begin{array}{ll} 0.7\times2=1.4 & 1 \\ 0.4\times2=0.8 & 0 \\ 0.8\times2=1.6 & 1 \\ 0.6\times2=1.2 & 1 \\ 0.2\times2=0.4 & 0 \\ 0.4\times2=0.8 & 0 \\ 0.8\times2=1.6 & 1 \\ 0.6\times2=1.2 & 1 \\ 0.2\times2=0.4 & 0 \\ \vdots & \end{array}$$

Step-2 $(1.xxx)\times 2^{\text{exponent}}$

$(1.0110\ 0110\ 0110 \text{----})\times 2^{-1}$

Step-3 Add bias to the exponent

$127+(-1)=126$

Step-4 Store to memory.

-2 -3   -6 -7   -10 -11   -14 -15   -18 -19   -22 -23

0	01111110	0110 0110 0110 0110 0110 011
sign	exponent	mantissa

Reverse Back:

$(-1)^{sign}\times(1+\text{mantissa})\times 2^{e-127}$

$= (-1)^0 \times \left(1+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^6}+\frac{1}{2^7}+\frac{1}{2^{10}}+\frac{1}{2^{11}}+\frac{1}{2^{14}}+\frac{1}{2^{15}}+\frac{1}{2^{18}}+\frac{1}{2^{19}}+\frac{1}{2^{22}}+\frac{1}{2^{23}}\right)\times 2^{-1}$

$= 1\times(1+0.399414062)\times 2^{-1}$

$= 0.699707031$ (not equal to exact 0.7f)

So when we store float on memory as binary and when reverse back the binary to float we don't get exact same value as before.

★ So we try to avoid any value to store as float & Double. We prefer to store as BigInteger.

BCZ double : 64 bit IEEE

1	11 bit	52 bits
sign	exponent	mantissa

$\text{bias} = 2^{10} - 1 = \boxed{1023}$