S① :

Language:

JAVA

Kannada

Communicate

Kannada

Kannada

Chaitu

Jack

S② :-

Instructing

english

english

dinga

dingi english

S③ :

add of 4+5

4+5

Interact/

Communicate

english

?

LSO

programmer

Programming

Language

Computer / System/

Machine
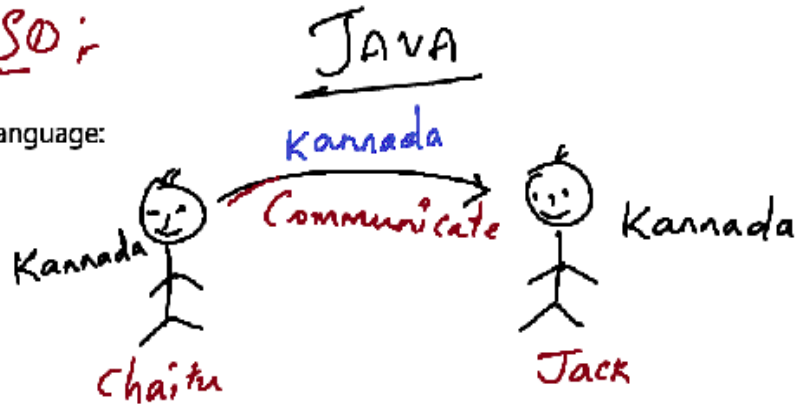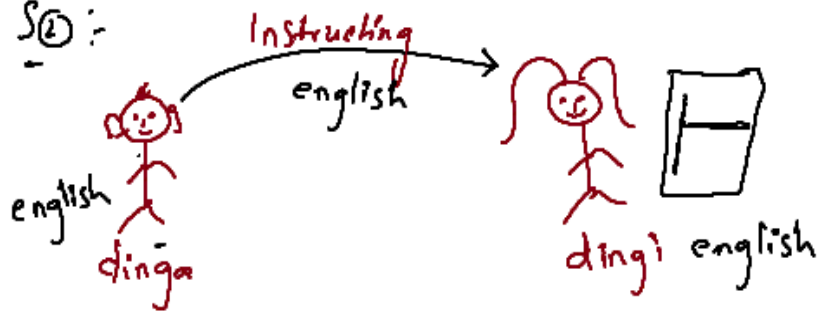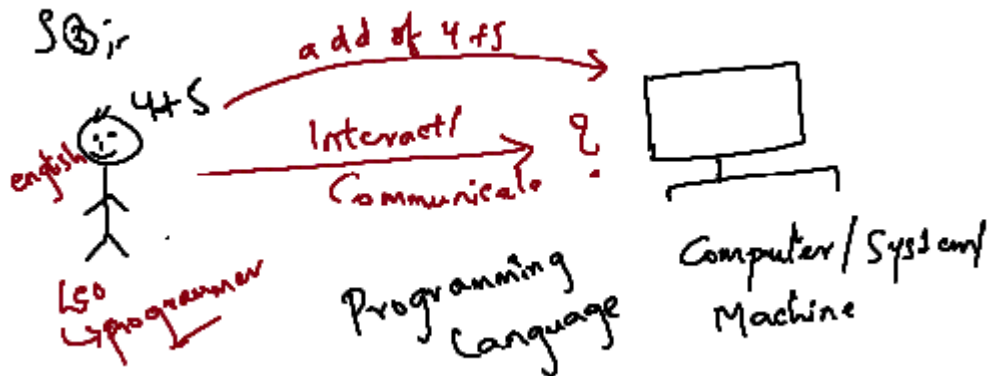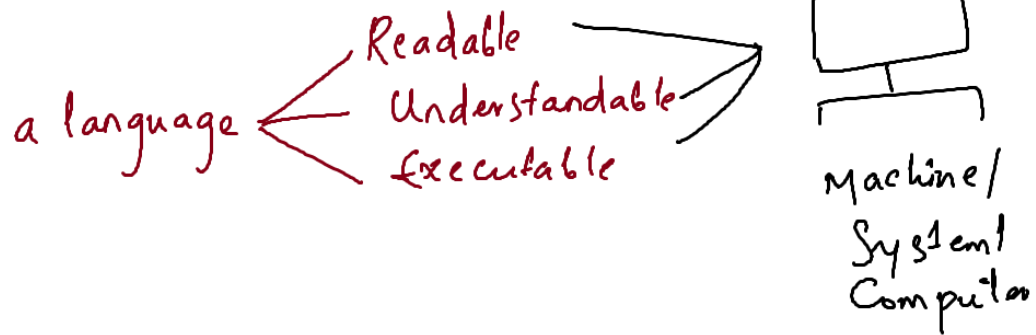
Programing Language:

A Language which is used to instruct the computer/system/machine to perform a specific task.

It is Classified into 3 Types:

1)Machine Level / Low Level Programing Language
2)Assembly Level / Mid Level Programing Language
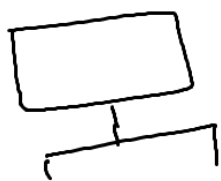3)High Level Programing Language
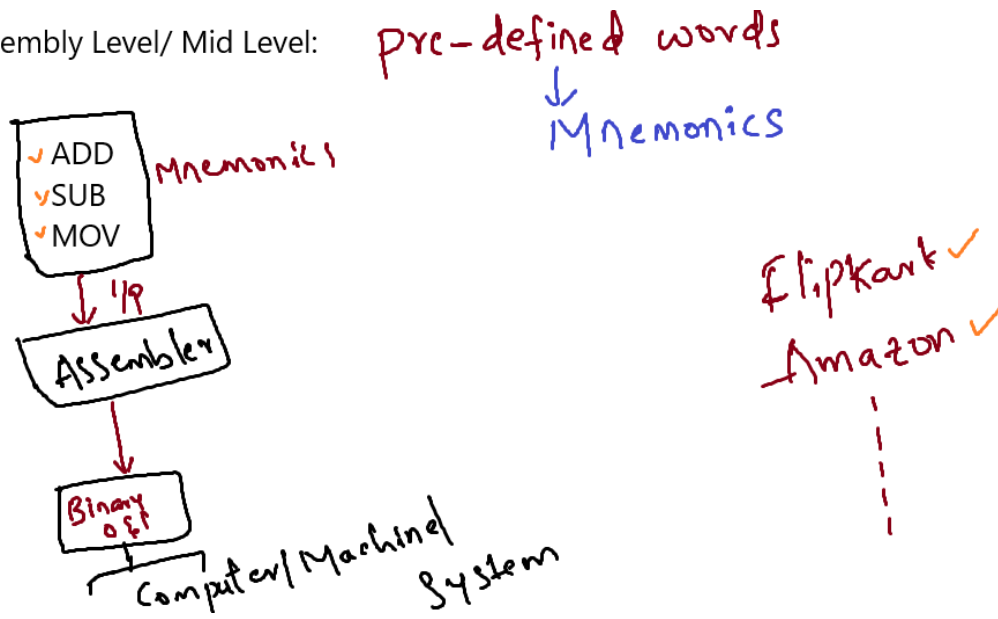
Machine Level / Low Level :

a language $\Bigg\langle$ Readable
Understandable
Executable

Machine/
System/
Computer

Ex : Binary Language

0   1

9897 6543

$\rightarrow$

2 $\lfloor$ 5 ✓
2 (2 – 1
1 – 0

101

Assembly Level/ Mid Level: pre-defined words
                                              ↓
                                         Mnemonics

✓ ADD   Mnemonics
✓ SUB
✓ MOV
        ↓ I/P
    Assembler   Flipkart ✓
                                                 Amazon ✓
    Binary
    0 & 1
         Computer/ Machine/
                    System

High Level Programing Language:

a language ⟵ Readable
              Understandable
              Instructable   Programmer

Ex:- C, C++, C#, Java, python, ruby, Php ----

High Level Programing Language

Procedure Oriented   Object Oriented
        C   Java

High Level
Instructions

Input

Machine Understandable

Compiler / Interpreter

Binary
0 & 1

instructions in java

Computer/
System/
Machine
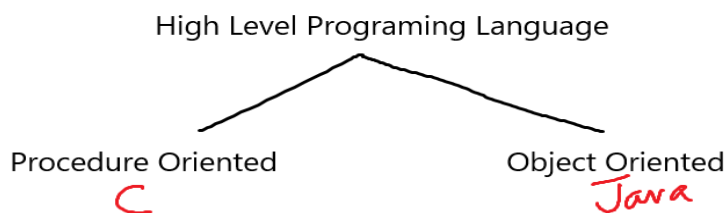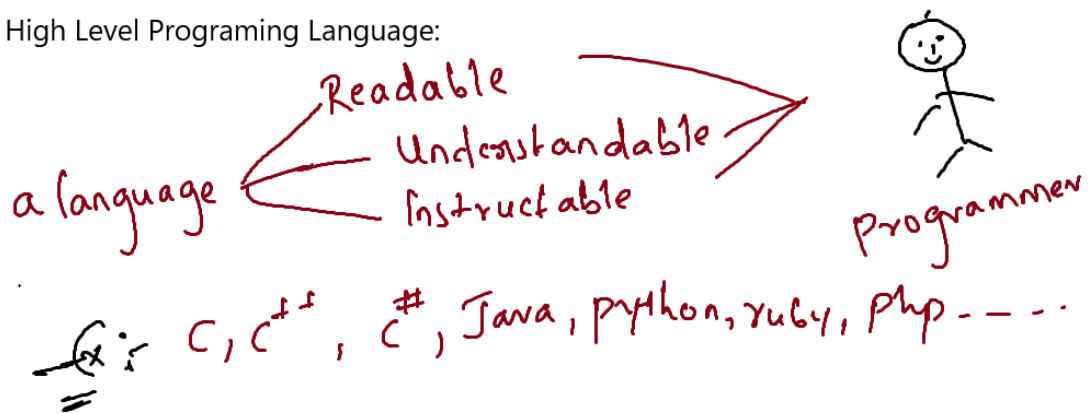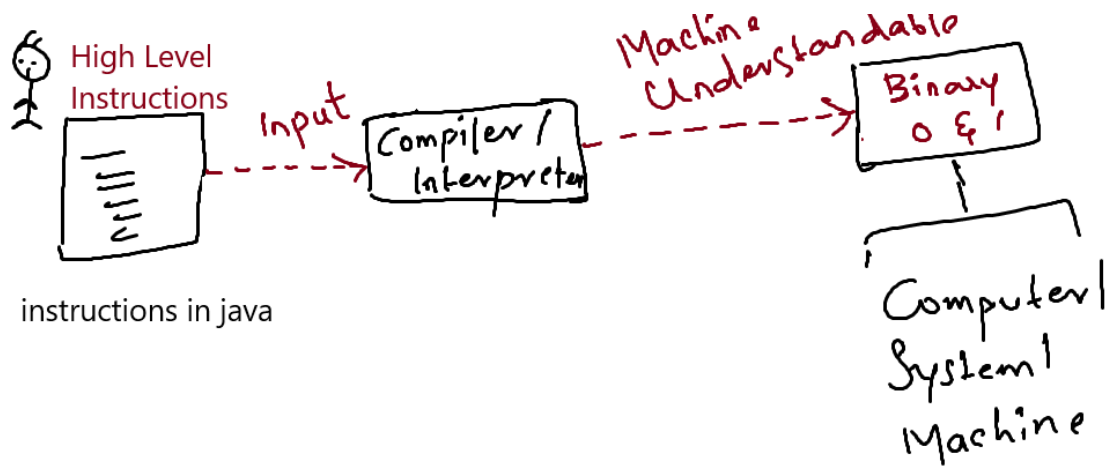
STRUCTURE OF JAVA PROGRAM:

Demo.java

class Demo
{

}

SourceFileName.java

.pdf
. jpg
.img

ClassName -)
Source Fil

Source File / Java Source File

Components
of
Java

class
interface ✔
enum ✔

Source file /
Java Source file

class'r Keyword
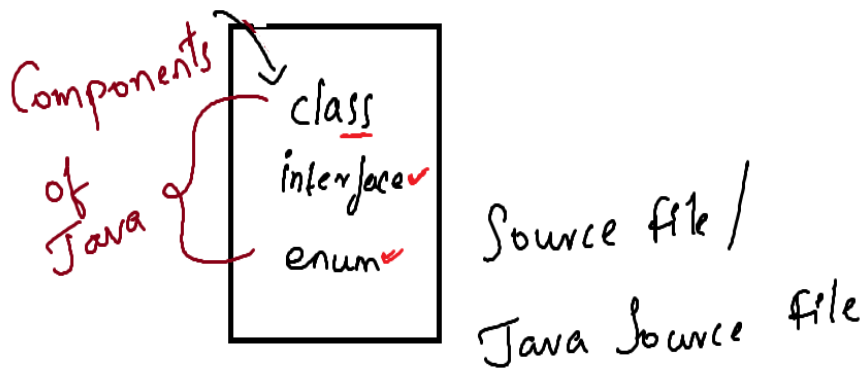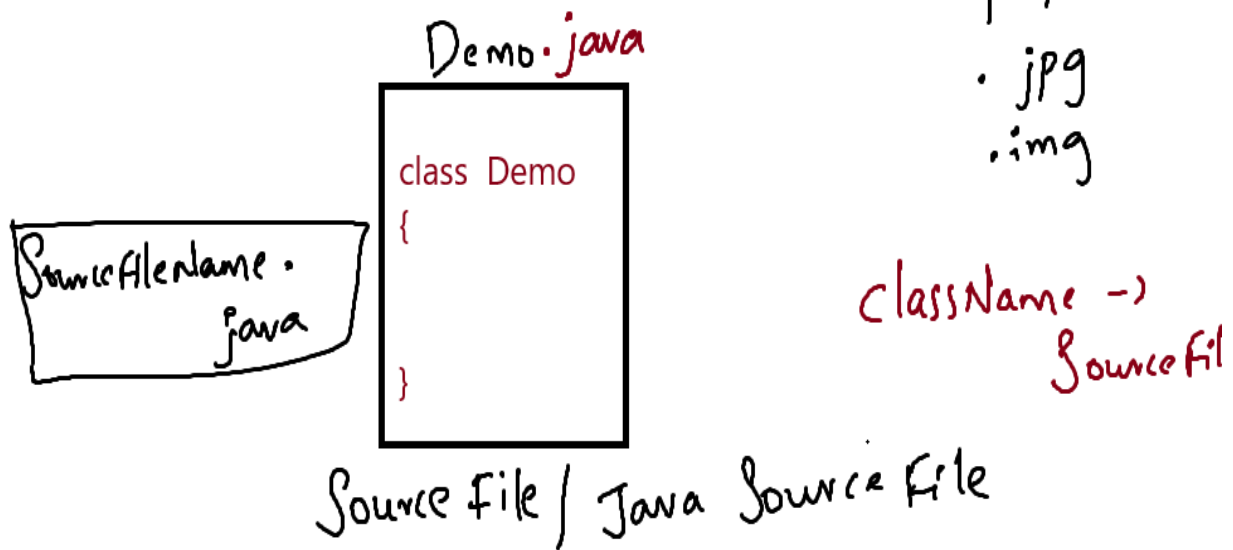   ↓.
      Pre-defined
                    → 😊 Compiler

😊 run ✓
  study ✓
ding?

       —→ Lower Case

                              Hello
                        →   Demo

SYNTAX :   class   ClassName
              {
                → class Block
              }

2 types of Instructions

```
class Demo
{

    ?

}
```

Declaration Statements

→ Variables ✔
→ methods ✔
→ class / enum

Initializers

→ static Initializers
→ non-static Initializers
→ constructors

Textbox

Store

Home Page

Instagram

| user name |

| password |

[Login]

[forgot password]

[create new account]

→ data
→ data } Variables

→ Task/ Action } method

PLATFORM:

✓	✓	✓
32-BIT WINDOWS OS	64-BIT LINUX OS	64-BIT MAC OS

PLATFORM DEPENDENT:

CALCULATOR APPLICATION

32-BIT WINDOWS OS

dinga

Calculator.c

i/p   C

compiler

executable

Binary found

Calculator.exe

✗

✓

✗

32-bit Linux os

32-bit windows os

64-bit mac os
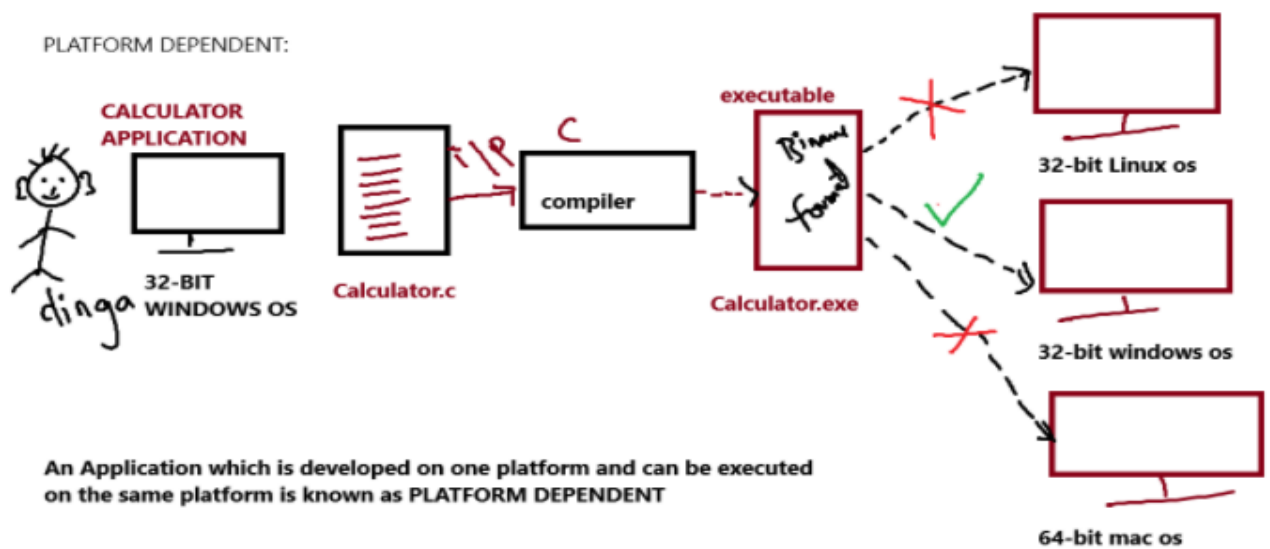
An Application which is developed on one platform and can be executed on the same platform is known as PLATFORM DEPENDENT
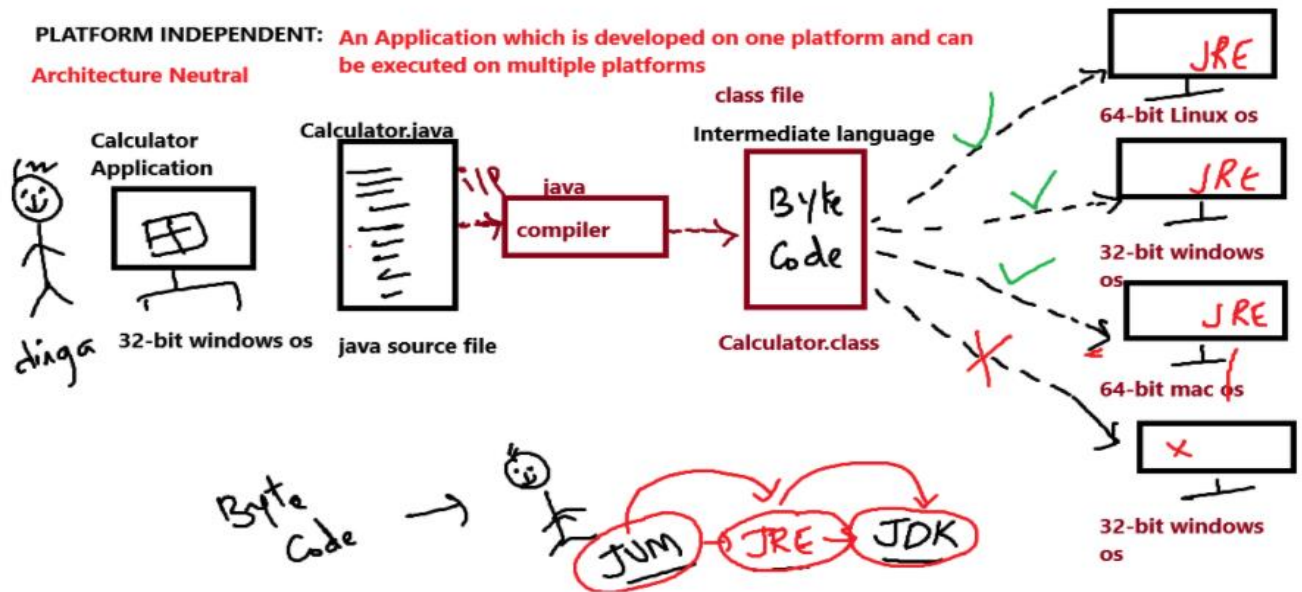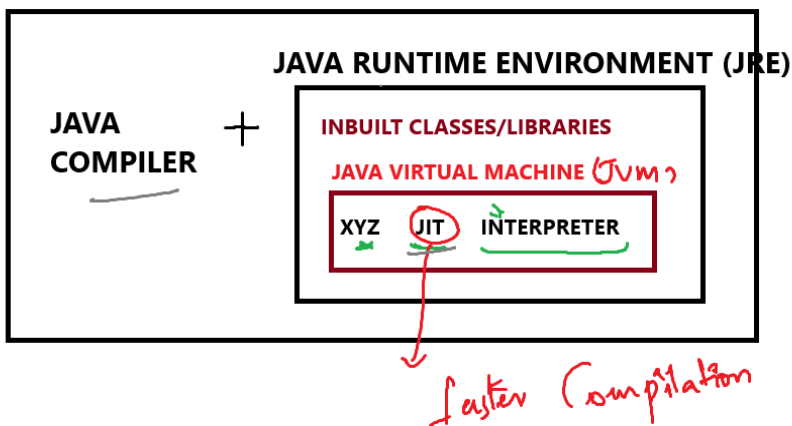
PLATFORM INDEPENDENT: An Application which is developed on one platform and can
Architecture Neutral be executed on multiple platforms

Calculator
Application

Calculator.java

java
compiler

class file
Intermediate language

Byte
Code

JRE
64-bit Linux os

JRE
32-bit windows os

JRE
64-bit mac os

32-bit windows os

dinga   32-bit windows os   java source file   Calculator.class

Byte Code → JVM → JRE → JDK

JAVA DEVELOPMENT KIT (JDK)

JAVA COMPILER	+	JAVA RUNTIME ENVIRONMENT (JRE)
		INBUILT CLASSES/LIBRARIES
		JAVA VIRTUAL MACHINE (JVM)
		XYZ JIT INTERPRETER

faster Compilation
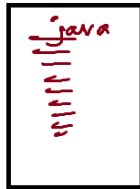
STEPS TO COMPILE AND EXECUTE A JAVA PROGRAM:

① Create a Source file

② Compile the Source File / Compilation

③ Execution

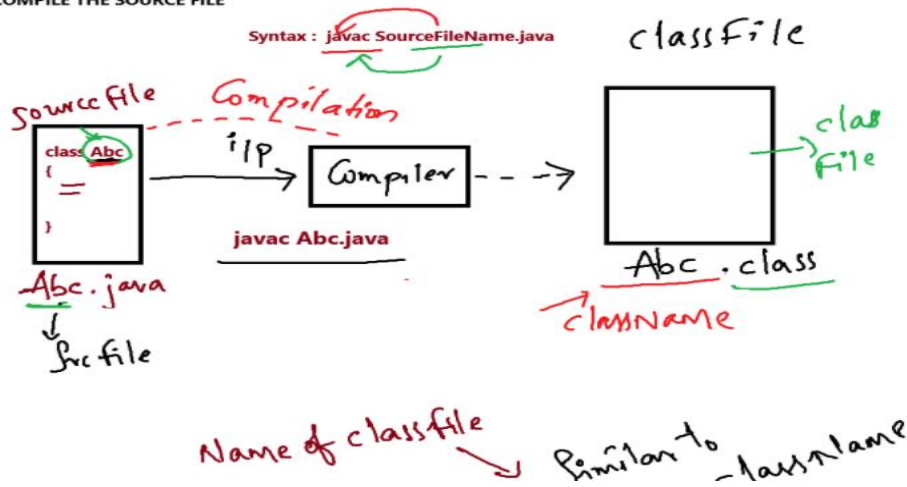**STEP 1 : CREATE SOURCE FILE / JAVA SOURCE FILE**
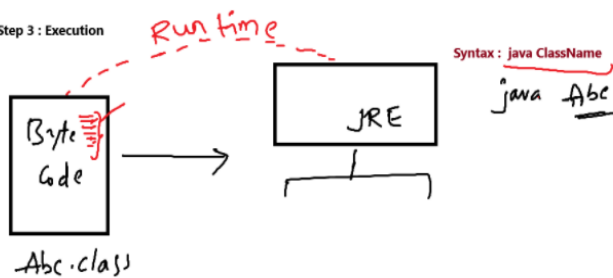
Notepad ✓
Notepad++
VsCode
eclipse

.java

SourceFileName.java

STEP 2 : COMPILE THE SOURCE FILE

Syntax : javac SourceFileName.java

classFile

Source File

Compilation

class Abc
{
=

}

i/p → Compiler --→

class
File

javac Abc.java

Abc.java
↓
Src file

Abc.class

ClassName

Name of classFile → Similar to ClassName

Step 3 : Execution

Run time

Byte Code

→ JRE

Syntax : java ClassName

java Abc

Abc.class

We can Compile the Empty Class ✓

But We Canot Execute the empty class X Error

class Abc
{
}

STATEMENTS TO PRINT : Printing Statements in java

√1) **System.out.print(data);**

 2) **System.out.println(data);**

1) System.out.print(data);

Same line

| 2 | 3 | 1 | 0 | 2 | Same line
|---|---|---|---|---|
| 0 | 9 | 5 | 5 | |
| | | | | |

√**System.out.print(23);**
√**System.out.print(10);**
√**System.out.print(20);**
√**System.out.print(9);**
√**System.out.print(55);**

Console

System.out.println(data);

Print & move new line   println

√System.out.println(7);
√System.out.println(10);
√System.out.println(12);
√System.out.println(11);

7		
1	0	
1	2	
1	1	

Console

class Demo
{

System.out.print(10);

}

Printing Stmt

Variables X

method Y

dinga

Starting Point

ending Point

100 meters Running Race

JVM

class Abc
{
  ≡
}
⟶ Main method ✓

≡

JRE
↓
preprogrammed

✗

keywords

name of a method

public static void main(String[] args)

Main method block
{
  ≡ = Statements
}
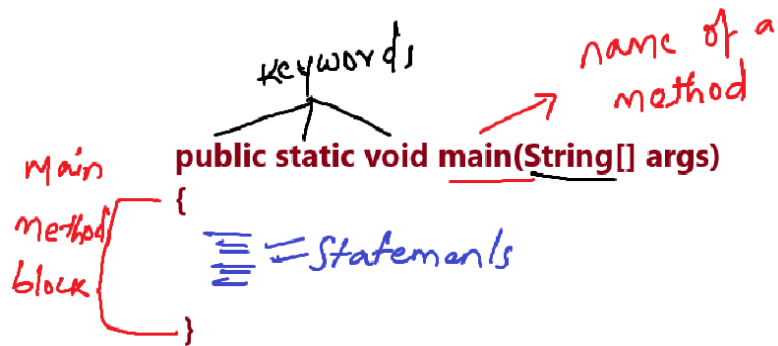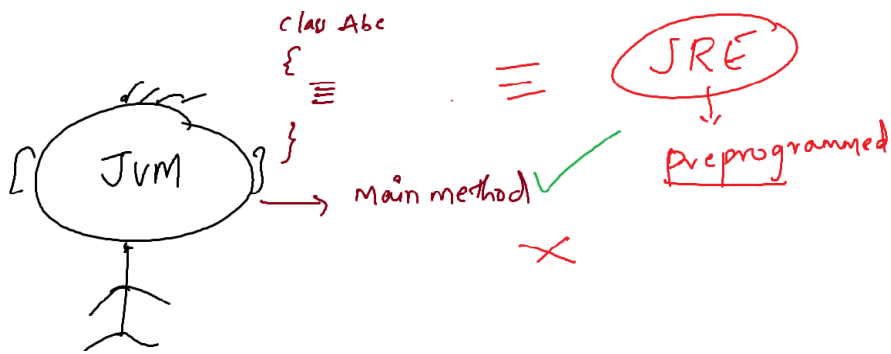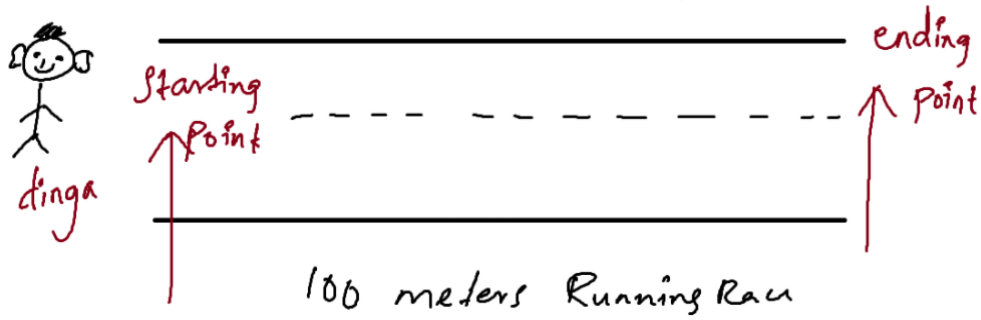
```java
class Example
{
public static void main(String[] args)
{
System.out.print(10);
}
}
```