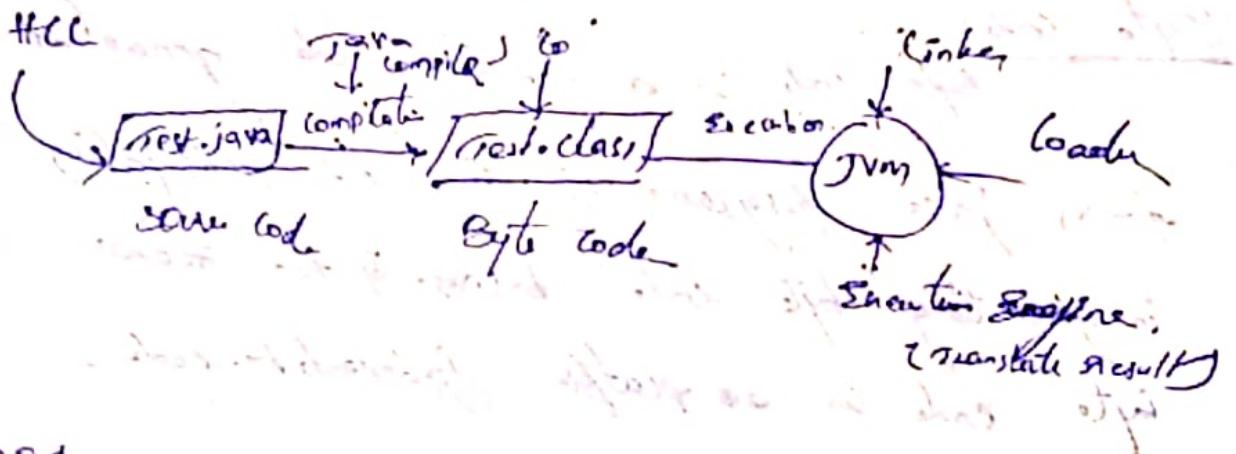
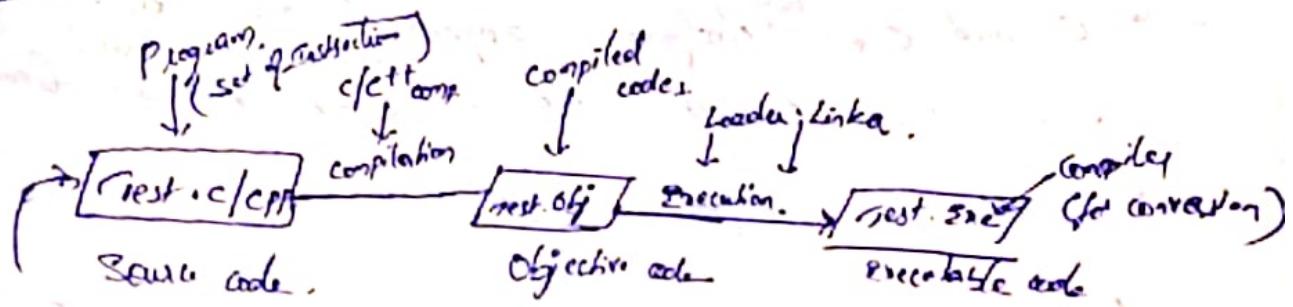


27/11/19

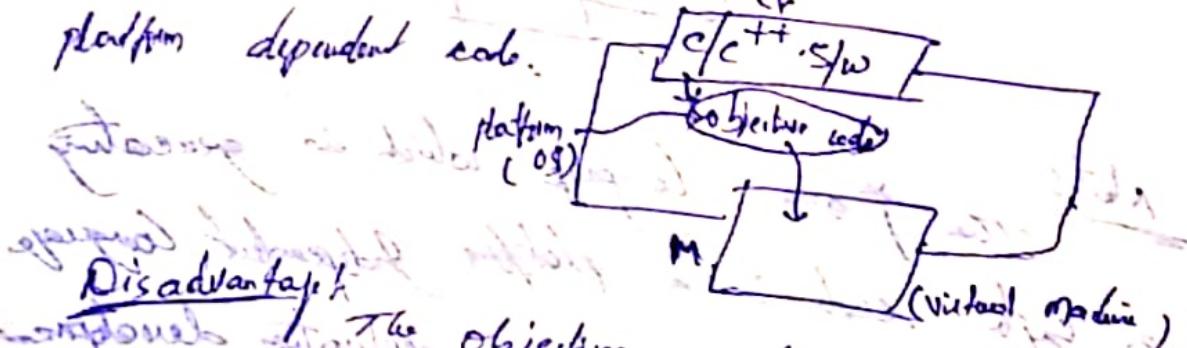


Q1. What is the difference b/w Objective code & Byte code.

Objective code:

Objective code is a compiled code generated from C and C++ program.

The Operating System (platform) is participated in generating Objective code. Because of this reason Objective code is platform dependent code.



Disadvantages:

Interoperability: The objective code from one platform cannot be executed on other platforms.

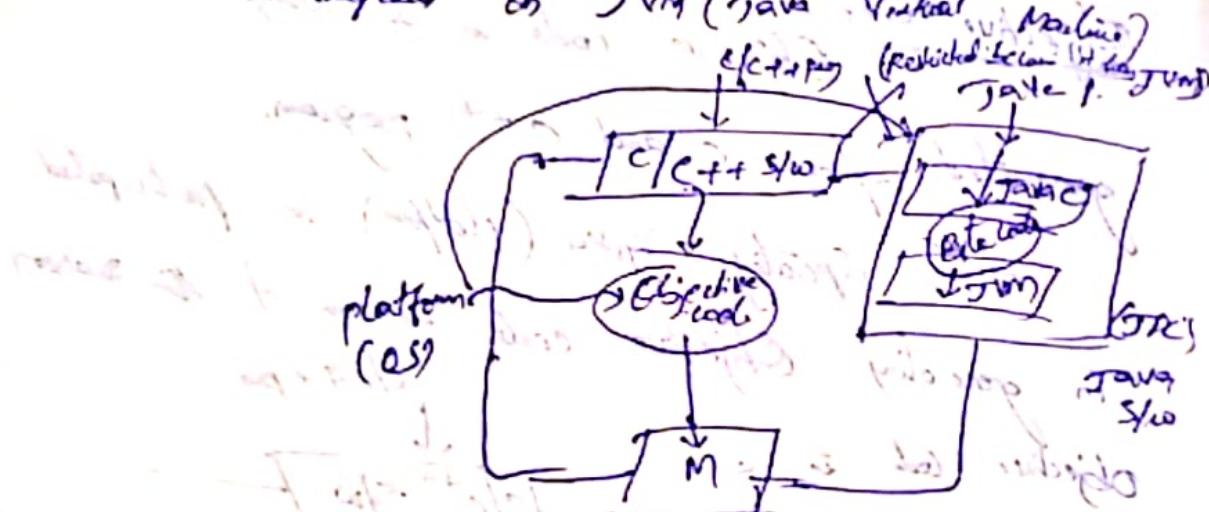
→ Platform dependent. ^{Java} can't
into internet world. ^(because of objective code) i.e., internet application developer
→ Java is platform-independent

Byte Code

Byte code is a recompiled code generated from java programs. It is not participating in generating byte code because of the reason byte code is a platform-independent code.

Advantages:

The byte code generated from one platform can be executed on all the platforms because it depends on JVM (Java Virtual Machine).



Notes:

The Java language, which is generating a byte code in a platform-independent language and preferable for internet application development.

Java is Simple :-

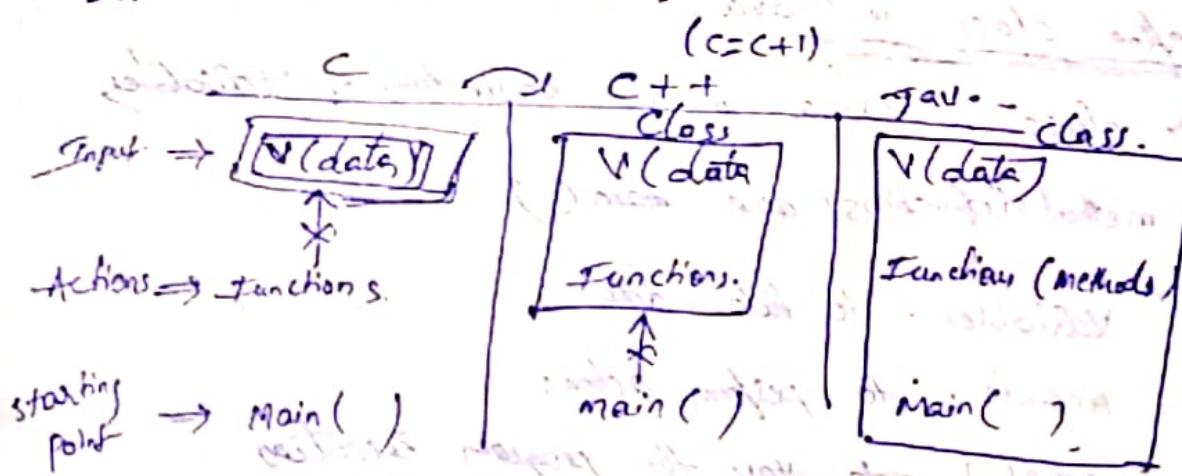
① Java is simple because there is no concept of External pointers in the hand of developers.

(External pointers are under the control of JVM)

② Java is simple because there is no concept of Memory Management in the hand of developers.

(Memory Management automatically done by the JVM)

③ Java is simple because all the components are binded into a single unit "class".



28/11/19

Defined Array: An array is a table

Array is an sequenced collection of elements of same data type.

Disadvantages:

Array can't hold disimilar data elements.

Define Structure & Unions:-

The Structures & Unions are collections of elements of different datatypes. (i.e. dissimilar data elements).

Disadvantages:-

Structures & Unions cannot hold functions.

Define "Class" in C++:-

class in C++ is a collection of variables & functions (data & fns). (A.M.)

Disadvantages:-

→ class in C++ cannot hold Main() static func.

Define class in Java:-

→ class in Java is a collection of Variables, methods(functions) and main().

Variables → to hold data.

Methods → to perform actions

main() → to start the program execution.

F.A.Q-2:-

What is the diff b/w function, member function & Method? (Ans. In a point)

- i) Functions: the part of program which is Repeatedly used in main program as well as in member functions.

ii) Member functions:

The functions which are declared within the class in C++ language are known as member functions.

Note:-

These member functions can also be declared outside the class with class reference.

(iii) Methods:-

The functions which are declared only inside the class in Java language are known as methods.

HISTORY OF JAVA LANG:-

- ① The author of JavaLang is "James Gosling".
- ② " " joined Sun Microsystems in 1991 as code writer (programmer).
- ③ The author joined the project where controlling electronic setting boxes using remote.
- ④ The author is:
④ In the process of development the author identified the existing lang. as platform dependent language, to overcome the disadvantages of platform dependent languages he started writing a new language having the feature WORA (Write Once Run Anywhere).

and to find the name "JAVA".
OAI in his first idea but it is already →

1995 - Java Alpha Beta.

1996 - JDK 1.0

1997 - JDK 1.1

1998 - JDK 1.2

2000 - JDK 1.3

2002 - JDK 1.4

2004 - Java 5 (Tiger)

⇒ JDK 1.5

⇒ JRE 1.5

2006 - Java 6.

⇒ JDK 1.6

⇒ JRE 1.6

2011 - Java 7.

⇒ JDK 1.7

⇒ JRE 1.7

2014 - Java 8

⇒ JDK 1.8

JRE 1.8

2017 - Java 9.

⇒ JDK 1.9.

⇒ JRE 1.9

2018 & 2019 - Java 10, Java 11

Java 12, Java 13

Java 14

Java 15

Java 16

Java 17

Java 18

Java 19

Java 20

Java 21

Java 22

Java 23

java, X64-Land recover.

Step-1: Download Jdk from Oracle website

Note: While downloading the Java Jdk (JDK) in platform dependent manner.

Step-2: Install the Java Jdk (JDK)

Note:

After Installation process we can find a folder with name "java" in Programs after.

C:\Program files (x86)\Java\jdk1.8.0_20\bin

Step-3: Set path in Environment Variables

Right click on my computer → Properties → Advanced

System Settings → Environment Variables, click "new" from System Variables.

Variable Name : Path

Variable Value :

C:\Program files (x86)\Java\jdk1.8.0_20\bin;

Step 4: click 'ok' for "File"

Note:

→ Open Command Prompt and check "javac"

"java" commands are working or not.

javac - for compilation process.

java - for execution process.

FAQ:

What is the diff b/w JDK & JRE?

JDK: JDK stands for Java Development Kit, which provides java compiler, JVM and JavaLib.

Java Compiler → It is the one which is used to compile the program and translate the source code into byte code.

JVM: Java Virtual Machine, which runs Java byte code and generate result.

JavaLib: JavaLib provides packages of classes and interfaces, which are used in constructing Java applications.

Note:

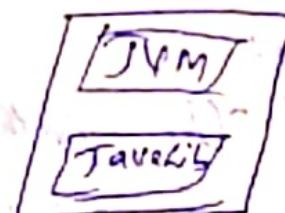
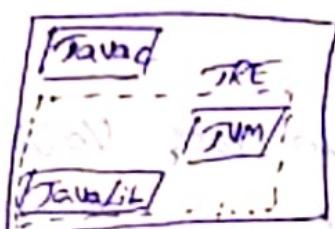
The core java Engineering system must be installed with JDK.

Path: C:\Program Files\Java\jdk1.8.0_121\bin

in JRE)

Note:

In dual-time JRE is a part of webserver (Tomcat) to execute web applications.

FAQ: Define Environment Variables.

Command Prompt (dos).



The Variables part of os, which are used to control and manage resources of computer system are known as Environment Variables.

Environment Variables

- (i) System Variables.
- (ii) User Variables.

(i) System Variables:

The path in system variables are available to total computer system and can be used by all users of computer system.

(ii) User Variables:

The path in user variables are available to only individual user.

FAQ:-

What is the diff b/w path & classpath?

The values in the path are used by OS requests.

The values in the classpath are used by the java application while executing in JVM.

FAQ:- Why do we have to set a path in Environment Variables?

⇒ When we haven't set path in Environment Variables then the java program can be compiled and executed from any location of computer system.

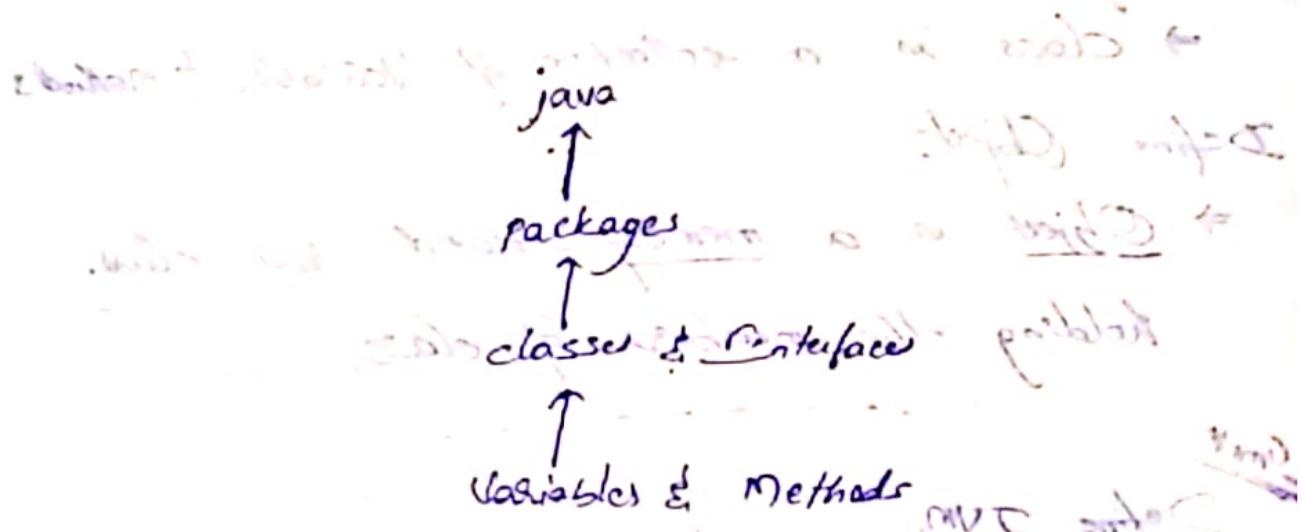
So with wrong path the java program can't be executed.

With right path the java program can be executed.

2/12/19

Define JavaLib: a library - where & what

- JavaLib is represented with the word "java".
- JavaLib is a collection of packages.
- Packages collection of classes and interfaces.
- Classes and Interfaces are the collection of Variables and methods.



The following are some important packages from JavaLib:

- java.lang - Language Package
- java.io - Stream and file package
- java.net - Networking package
- java.util - JCF package (Java Collection Framework)
- java.awt - Abstract Window Toolkit (GUI)
- javax.swing - Swing programming (GUI)
- java.applet - Applet Programming (GUI)

`java.sql` - DB connection package }

`javax.servlet` - Servlet Programming } Adv. Java

`javax.servlet.jsp` - JSP Programming }

Advantages

Define Class.

→ Class is a Structural Layout generating Objects.

→ Class is a collection of Variables & Methods.

Define Object.

→ Object is a memory related to a class holding the members of class.

Define JVM.

→ JVM stands for 'Java Virtual Machine', which is used to execute Java Byte code.

→ While executing Java Byte code JVM runs independently. (OS is not involved)

Define Virtual Machine

→ The few programs which internally having the behavior like a machine is known as

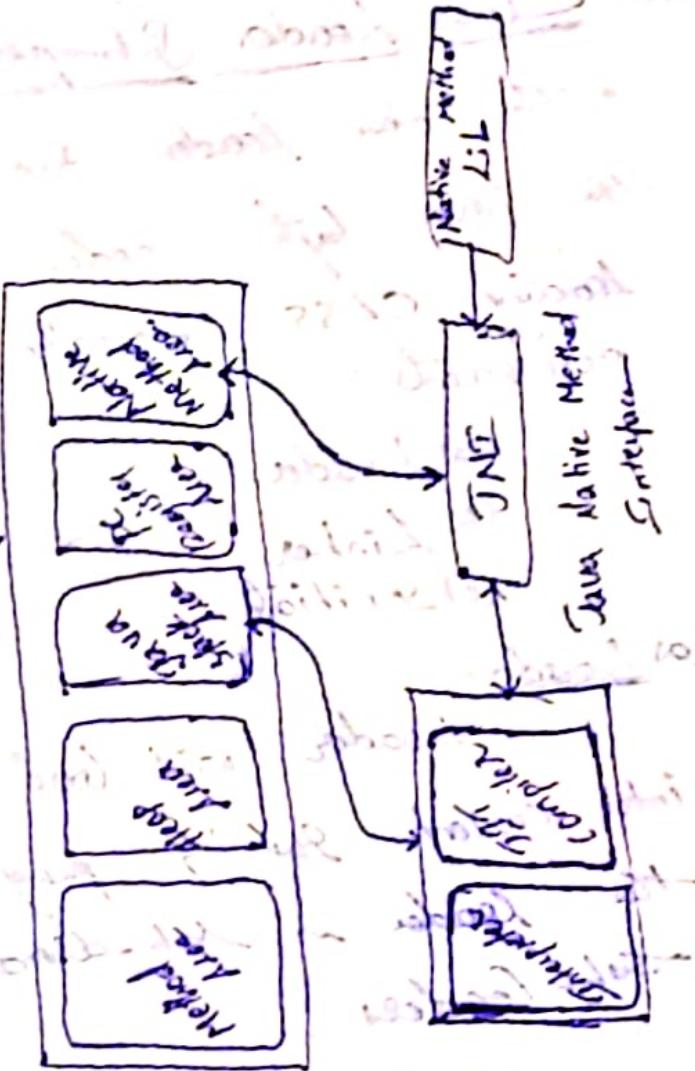
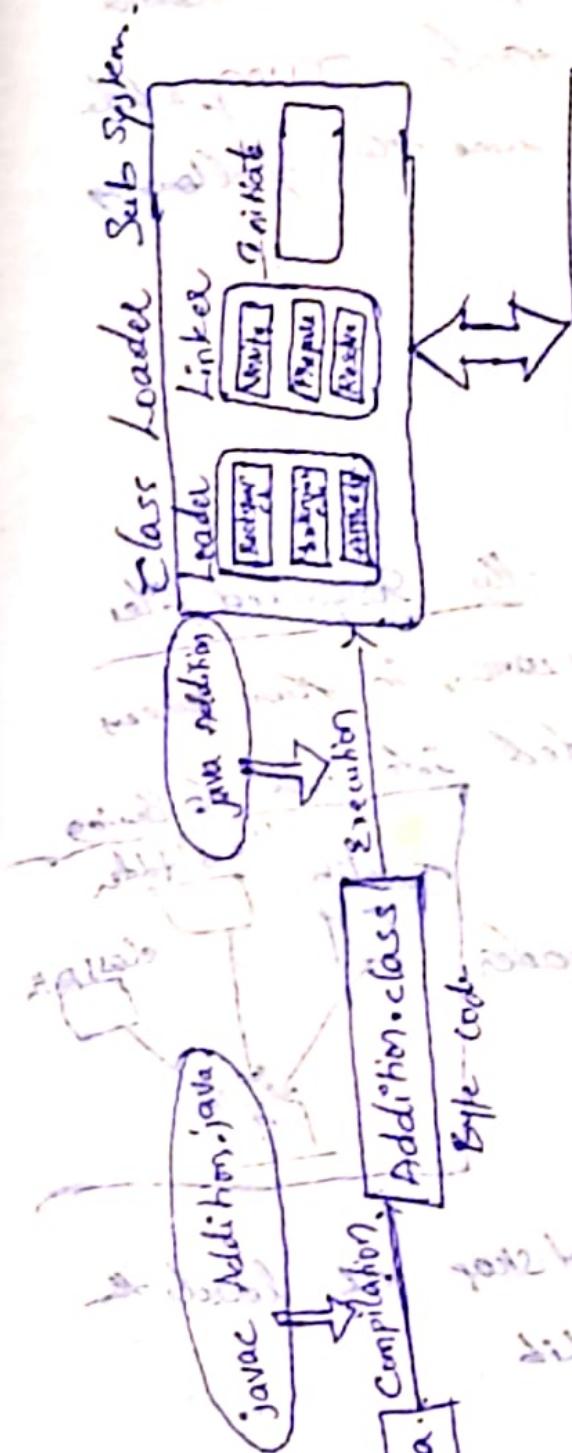
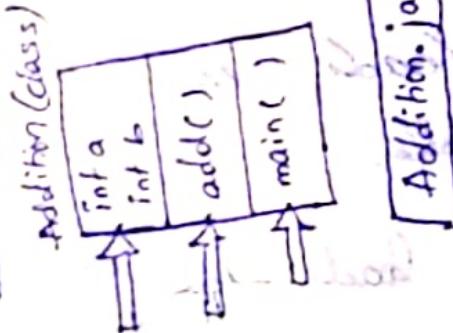
Virtual Machine

JVM Architecture

→ JVM internally categorized into three parts:

1. Class Loader Subsystem
2. Runtime Data Area

JVM Architecture



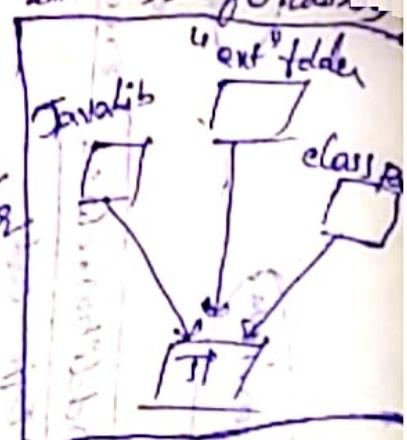
(a) Loader:

Loader will load into current running program, in the Loader is sub-divided into the following Sub-Loaders.

(i) Bootstrap class Loader

(ii) Extension CL

(iii) Application CL.



(i) Bootstrap class Loader:

Bootstrap CL loads the required files from JavaLib.

(ii) Extension CL:

Extension CL loads the required files from "ext" folder.

(iii) Application CL:

Application CL loads the required files from "classpath".

file
as need,
the following

-component will check the loaded
modules and the required components are same
or not, which is known as verification process.

(ii) Prepare :

Prepare component performs decoding process, in
this process identifies the type of programming
components (classes, variables, methods, ...).

(iii) Resolve :

Resolve component identifies the programming
components belong to class & belong to object using
"static" keyword in code like below

Note:

Based on "static" keyword the programming components
are categorized into two types.

(i) static programming components

(ii) Non-static programming components

(a) Static Programming Components:-

The programming components which are declared with "static" keyword are known as "Static Programming Components".

Note:-

These static programming components will get the memory within the class and accessed with the className.

(b) Non-static Programming Components:-

These are known as the programming components which are declared without static keyword are known as "non-static prog. comp."

Note:-

These non-static prog. comp. will get the memory within the object and access with this object name.

(c) Initiate:-

Initiate component performs Initialization process. In this process one run time data area is generated and which is loaded with the "class code" (class).

④ Runtime Data Area: - internally divided into following partitions. (a) Method Area (b) Heap Area (c) Java Stack Area

- (d) PC Registers Area
- (e) Native Method Area.

11/12/19

(a) Method Area:

- The memory location where the class is loaded is known as method area.
 - While class loading the static programming components of the class will get the memory within the class, in this process main() will get the memory within the class. and the main() is copied onto java stack area to start execution process.
 - the execution engine will come to java stack area to find the method having the following format or not. :- `public static void main(String args[])`

- If the format is available then the execution is started. If not the execution is not started.

(b) Heap Area

- The memory location where the objects are created is known as 'Heap Area'
- While Object creation, the non static

members of the class will get the memory within the object.

Simple Structure of Java Program

- ⇒ In JavaLang we cannot write a Java Prog. without class
- We use "class" keyword to declare classes in JavaLang.

```
class Class-name {  
    // variables  
    // methods  
    public static void main(String args[]){  
        {  
            // set_of_instructions;  
            {  
                // function body starts  
            }  
        }  
    }  
}
```

Method or function call to the library in

final part

The main method program or
code part is stored in Execution Area
while all others reside in

Example Prog.:

- WAP to display the msg as "Welcome to Java"

```

import java.lang.*;
class Display
{
    public static void main(String args[])
    {
        System.out.print("Welcome to Java");
    }
}

```

Writing, Saving, Compiling and Executing Java Prog.

Step 1: Create one folder (destination folder) part of any drive: E:\Demo100

Step 2: Open editplus (any text editor) and type the prog.

Open editplus → Browse & select destination folder → click on file → New → Java and type the prog.

Step 3: Save the prog. in destination folder

click on file → save → name the file as

Display.java → click 'Save'

Note:

⇒ Use "Command Prompt" to perform compilation and execution process.

⇒ To open Command Prompt, go to destination folder.

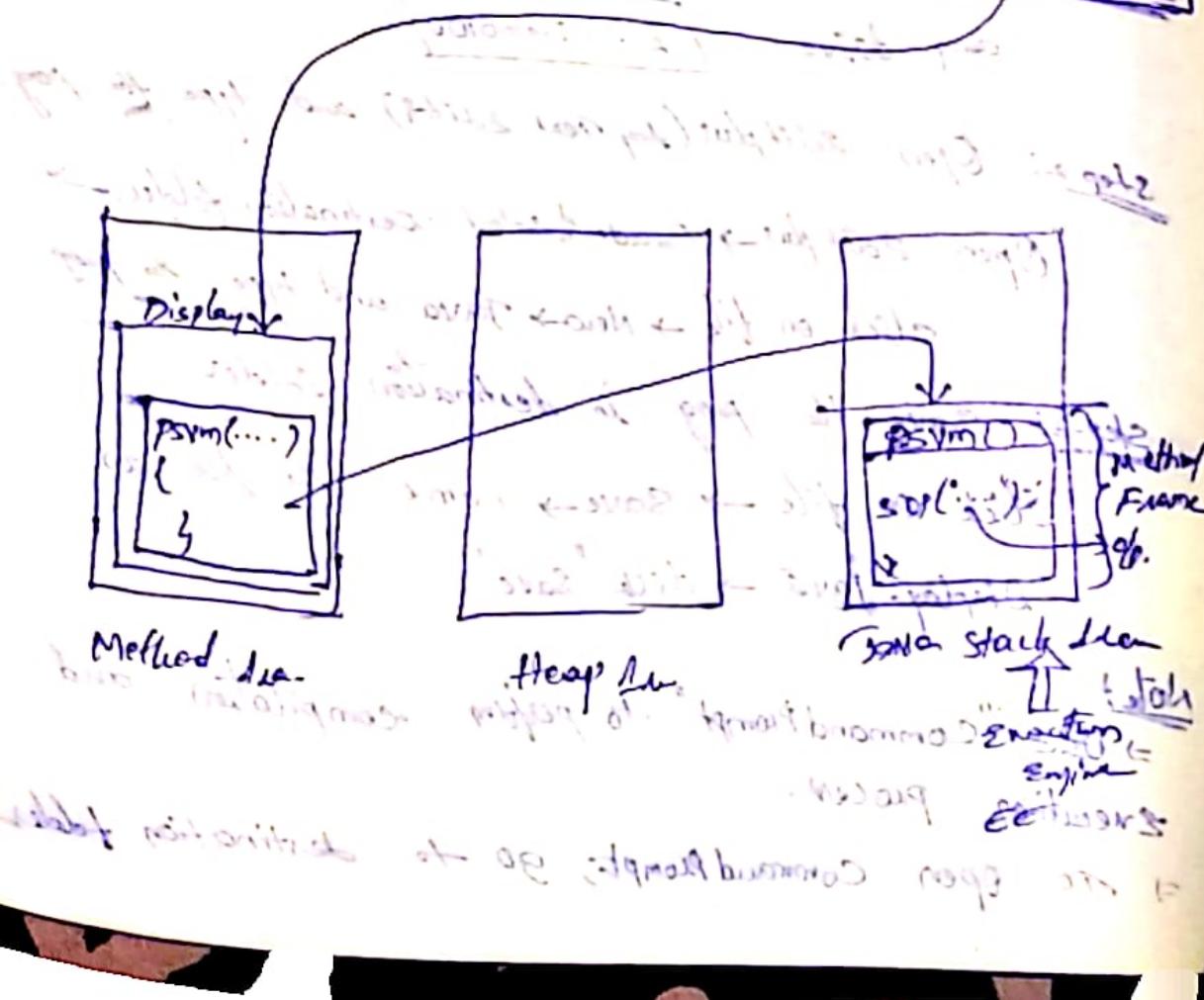
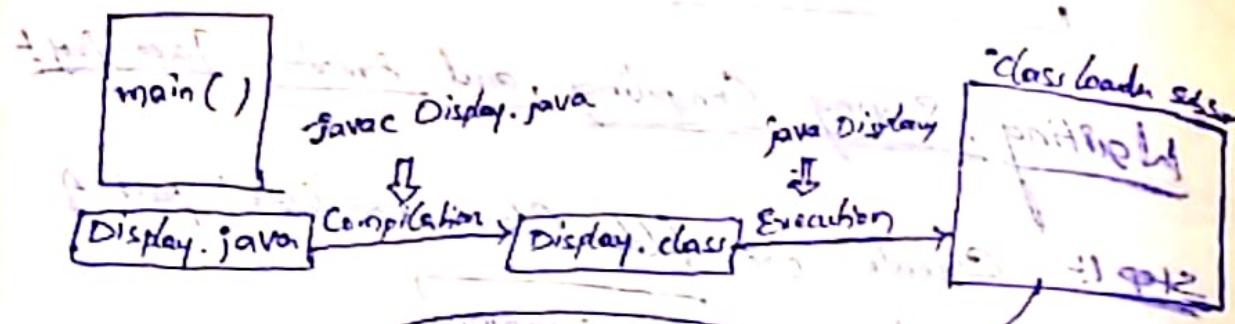
and type "cmd" in address bar and press Enter.

Step 4: Compile the program as.
javac Display.java

Step 5: Execute the program as.
java Display

5/10/19

Execution flow of above program:-



Eg.

Define Method Frame.

→ The partition of java stack area where the method is copied for execution is known as Method Frame.

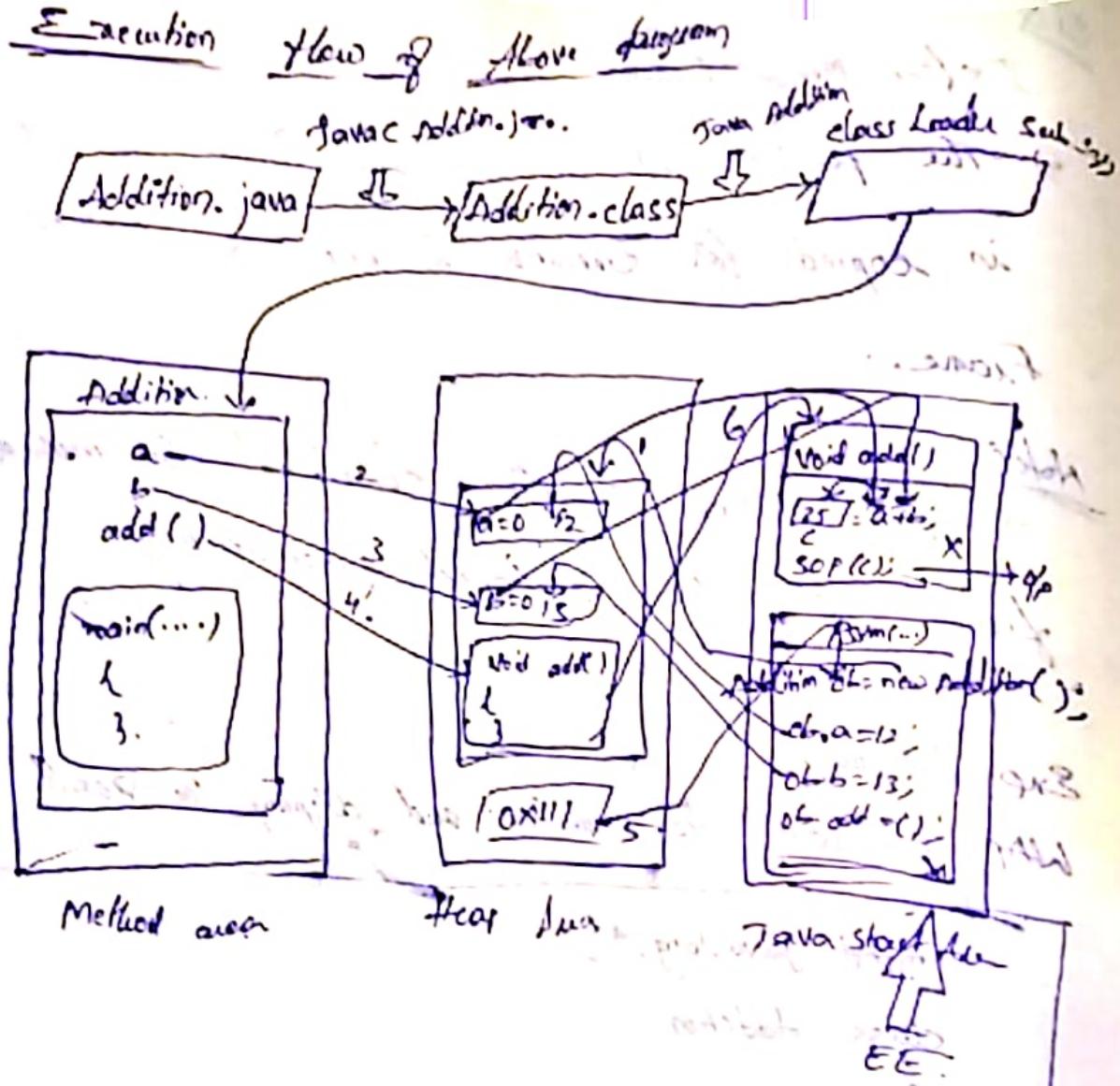
Note:-

→ After method execution completed the method frame will be destroyed automatically.

Ex. Program:

WAP to add two numbers and display the result.

```
import java.lang.*;  
class Addition  
{  
    int a,b; //Non static variable (instance Variable)  
    void add() //Non static method  
    {  
        int c=a+b; //C is nonstatic Variable  
        System.out.print(c); //Local Variable  
    }  
    public static void main(String [] args) //static method  
    {  
        Addition ob=new Addition(); //Object creation.  
        ob.a=12;  
        ob.b=13; //Non static variable add static  
        ob.add(); //method call  
    }  
}
```



6/12/19

Define Object:-

Object is a storage related to

a class part of heap area holding the non-state members of the class.

Note:-

After object creation the object will have the following (i) state

(ii) Behavior

(iii) Identity

(i) State:-

The variable part of Object is known as state of Object.

iii) Behavior:

The Method part of Object is known as Behavior of Object.

iv) Identity:

The Variable which is holding the reference of Object is known as Reference Variable or Object name, which is known as Object Identity.

Note:

We use "new" operator to create objects in JavaLang.

Syntax of Object Creation:

[class name obj-name = new Class-name();]

Example:

Addition ob = new Addition();

Execution Behavior of New Operator:

1. "New" Operator specifies the execution control to create reference part of Heap Area.
2. "new" Operator specifies the execution control to check the required class is available on Method area or not.
3. If the class is available on Method area then the non-static members of the class are binded to the reference variable via this association.
4. The reference part of object will be added onto Method area.

Define Variables:

= Variables are the data holder; the data can be used.

Value of Reference

→ Based on static keyword the variables are categorized into two types.

1. Non-Static Variables.

2. Static Variables.

1. Non-Static Variables:

→ The variables which are declared without static keyword are known as Non-Static Variables.

→ Based on the declaration & memory location where non-static variables are declared categorized into two types.

(a) Instance Variables

(b) Local Variables,

(c) Instance Variables:

Def: the variables which are declared outside the methods without static keyword are known as Instance Variables.

ed by the

Example:-

Then Instance Variables are available until the object is available on heap area.

Note:-

When the Instance Variables are not initialized with any value then they are assigned with Default Values.

(ii) Local Variable :-

The Variables which are declared inside the methods without static keyword are known as Local Variables.

Memory Location:-

These local Variables will get the memory within method frame while method Execution.

Scope & Visibility:-

These local Variables are accessed only inside the methods. (Within the methods)

10/11

execution is

Note :-

- the Local Variables will

- - - - -

Note :-

In the process of Constructing a

we have One Mainclass and any number of
Subclasses.

Main Class :

Main class means which holding "psn() main() method"

Sub Class :-

→ SubClass means which holds Variables & methods.

Example :-

What to display emp detail using "class-object" concept.

Emp Data : Employee with variable details like

⇒ eId, eName, eDesg, address

⇒ Void getEmpData()

Emp Address : Employee Address like

⇒ hNo, sName, city, pincode

⇒ Void getEmpAddress()

Main Class

:-

```
    System.out.println ("eid:" + eid);  
    System.out.println ("ename:" + ename);  
    System.out.println ("eDesg:" + eDesg);  
    System.out.println ("edge:" + edge);
```

{

```
public static void main (String [] args)
```

1. $\text{M}_1 \text{O}_2 \text{S}_2$ + H_2O + Na_2CO_3

Liquor made by Na_2CO_3

sol. in H_2O

(Dissolving salt)

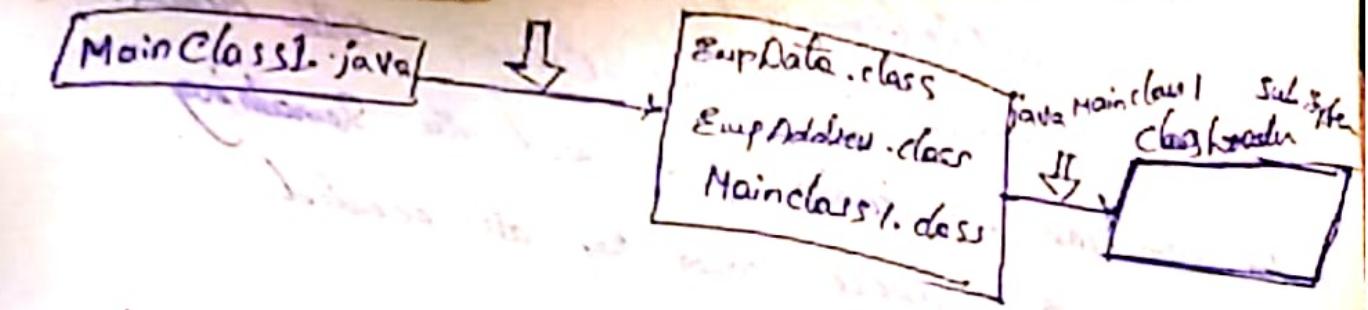
Water + Na_2CO_3 , which has Na_2CO_3

[(andrea + andrea) + water] has Na_2CO_3

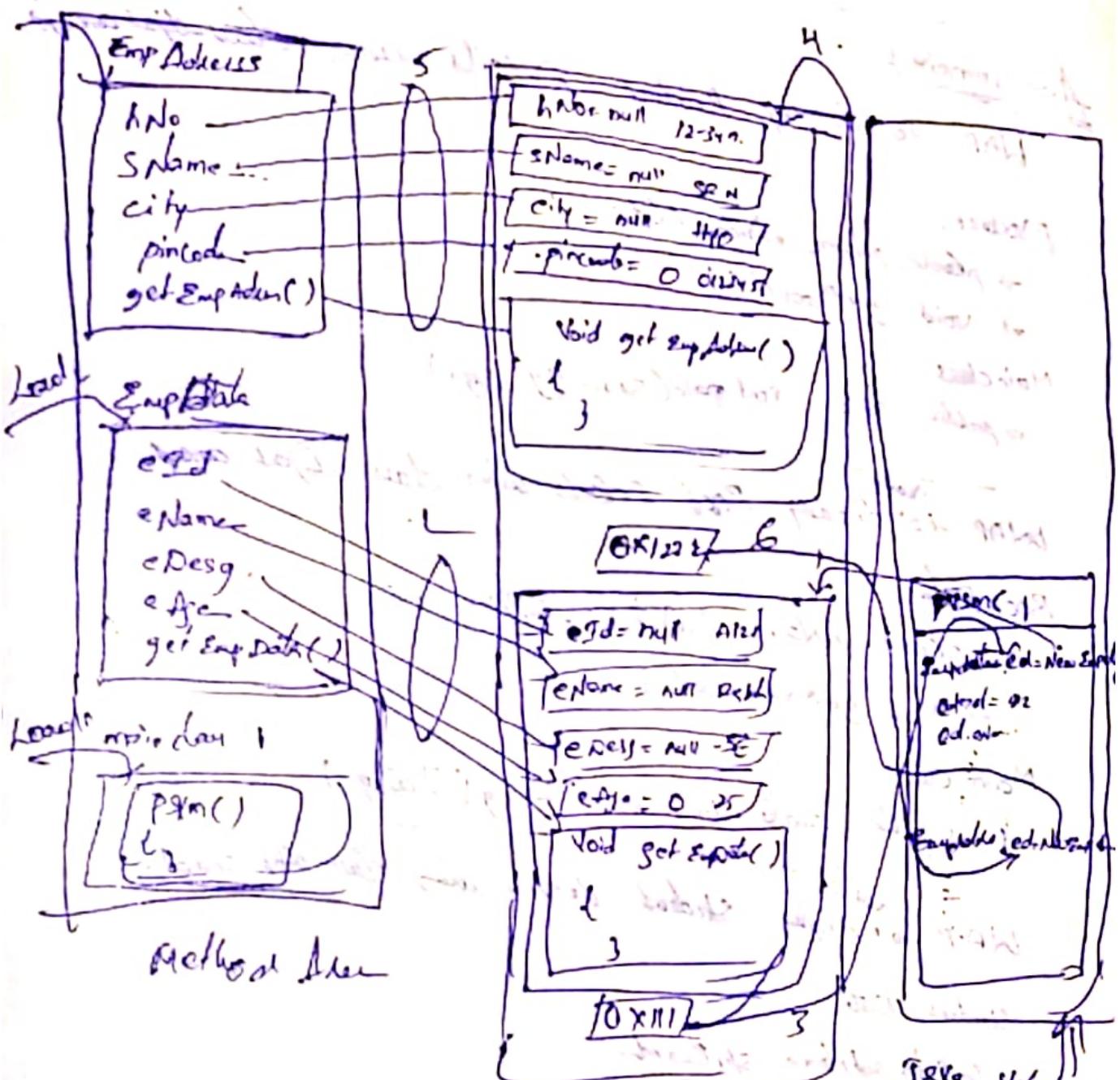
[(andrea + andrea) + water] has Na_2CO_3

[(andrea + andrea) + water] has Na_2CO_3

{ Na_2CO_3 + H_2O } from which acidic alum }
+ Na_2CO_3



Load 3



Note:

- Every class component of application will have its own .class file (class file).
- In the process of loading classes, the main class is loaded first and the remaining subclasses are loaded when they are needed.

Assignment :-

WAP → display Product Details using class object concept.

Product:

- pCode, pName, pPrice, PQty
- void getProd()

Main class

→ public static void main (String [] args)

WAP → display Book Details using class object concept.

Book Data:

- bCode, bName, bAuthor, bPrice, bQty

→ void getBookData()

Main class

→ public static void main (String [] args)

WAP → display Student details using class object concept.

Student Data:

- StID, StName, StBoard

→ void getStudentData()

Student Address

- hNo, sName, city, pinCode

→ void getAddress()

Main class

→ public static void main (String [] args)

(c) Java Stack Area:-

- The Memory block where all the methods are executed is known as "Java Stack Area". In this process main() method is copied onto Java Stack Area first and remaining methods are called after execution using main() method.

(d) PC Register Area:

- Program Counter or Register will record the state of method execution in Java Stack Area.
- Every method which is executing in Java Stack Area will have its own PC-Register.
- All these Registers are Opened in separate memory location known as PC-Registers Area.

(e) Native Method Area:

- The methods which are declared with Native keyword are known as Native Methods.
- These Native Methods are constructed from C/C++ code.
- When these methods are used part of java program, these methods are separated and loaded onto separate memory location known as Native Method Area.
- These Native methods are executed using JNI (Java Native Method Interface), while execution of JNI uses Native Method Libraries.

Ques Why Native methods are available part of Java?

- => When we want to interact with the resources outside Java, then the Java Application will take the support of Native Methods because of this reason Native methods are available as part of Java.

③ Execution Engine:

- Execution Engine will start the execution process from main() method available along Java application.
- => This execution engine will have the following tools Translator.

- (i) Interpreter (Interpretive mode)
- (ii) JIT Compiler. (Just-In-Time)

(i) INTERPRETER

- Interpreter starts the execution process and executes the normal instructions.

- (ii) JIT Compiler:
JIT compiler executes stream instructions.

(Stream = Multimedia files Audio, video, image & animation)

FAQ: What is the difference between interpreter and JIT compiler?

- => When we have interpreter in execution process then we can accept frequently in the middle of execution process, which is preferable.

Internal Application development.

Note:

Java is Interpreted Language because it uses
Interpreter in execution process

Naming Conventions in Java:-

→ The rules to be followed by the java programmer in writing a java program are known as Naming Conventions in Java.

Packages :-

→ packages in java must be in lowercase of abc

Classes & Interfaces:-

→ In classes and Interfaces the starting letter of every word must be uppercase.

Ex:- EmpData

EmpAddress

InputStreamReader

Variables & Methods:-

→ In Variables & Methods first word in lowercase and from second word onwards the starting letter must be uppercase.

hNo

rollNo

panCard

getProduct()

getEmpData()

readLine()

(A to Z) digit - 0 to 9

alpha - A to Z

alpha - a to z

alpha - e to E

- types.

(1) Primitive Datatypes.

(2) Non-Primitive Data types.

10/12/18

(1) Primitive Data-type

→ the data types which are existing to ~~standard~~ standard
form of single valued data format are known as

Primitive Datatypes.

⇒ Primitive datatypes are categorized into following

(a) Integer Datatype

(b) Float Datatype

(c) Character Datatype

(d) Boolean Datatype

(a) Integer Datatype:

the numeric data which are ~~existing~~ standard without decimal point representation are known as

Integer Datatypes and given as follows. like bytes.

⇒ Then Integer datatypes are categorized into following.

(i) byte - 1Byte (8 Bits)

(ii) short - 2 bytes

(iii) int - 4 bytes

(iv) long - 8 bytes

(i) byte

(ii) short

(iii) int

Final application, "Gang"
will like phone no, Card no,...
'short' is used for stream data (multimedia data)

(4) Float Datatypes:

- ⇒ The numeric data with decimal point representation is known as float datatype.
- ⇒ Float datatypes are categorized into following:
 - (i) float - 4 bytes.
 - (ii) Double - 8 bytes.

Note:-

The number which is assigned with decimal point is automatically considered as double data; if we want float data then we must use 'F' & 'f' in the RHS of declaration.

(c) Character Datatype:

- ⇒ The single valued character which is represented with single quotes is known as Character Data.

Expt
'A', 'B', ...

Type: char - 2 bytes

FAQ:

Why 'char' is 2 bytes in 'javalong'.

- JavaLang uses Unicode (universal code) representation, which is ready to accept all the language codes in the world, because of this JavaLang is 2 bytes for a character.

Exp Codes:

Java char = 2 bytes

$2^{16} = 65536$ - for any language

ASCII & Chinese chars

ASCII (American Standard Code for Information Interchange)

→ for the United States

ISO 8859-1 for Western European language

KOI-8 for Russia.

GB18030 and BIG-5 for Chinese

(d) Boolean Datatype:

→ the datatype which is existed is the form of true or false is known as Boolean datatype.

Type:

Boolean - 1 bit

Exp Program:

```
import java.lang.*;
public class Datatype {
    public static void main (String [] args) {
    }
```

byte b = (Byte) 125;

short s = (Short) 31760;

int i = 123456;

long l = 709584455L;

float f = 2345.25E1F;

```
double d = 1234.678;  
char ch = 'A';  
boolean b1 = true;  
System.out.println("byte data : " + b);  
System.out.println("short data s : " + s);  
System.out.println("int data i : " + i);  
System.out.println("long data l : " + l);  
System.out.println("float data f : " + f);  
System.out.println("double data d : " + d);  
System.out.println("char data ch : " + ch);  
System.out.println("boolean data b1 : " + b1);
```

}

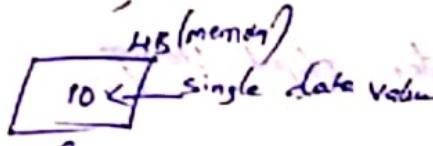
3

Expt :- Define Field storage

→ The Memory which is generated to hold single data value is known as "field storage".

Note :- → The primitive datatypes part of Java program generate field storage.

Expt :- `int a = 10;`



(2) Non Primitive Data types (or) Referential data types

→ The data types which are existing in the form of "Group Valued Data formats" are known as Non-primitive Data types.

⇒ These non-primitive datatypes are categorized into the following:

(a) Class

(b) Interface

(c) Array

(d) Enum.

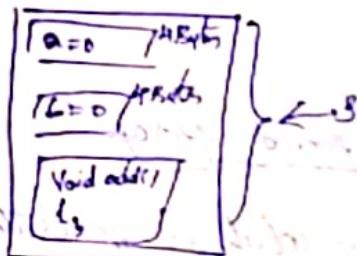
Ques:

Define "Object Storage".

⇒ The memory which is generated to hold group members of the class is known as object storage.

Addition of object storage

Addition of = new Addition()



Note:-

⇒ When we use non-primitive datatypes, java program generates object storage.

Note:-

Based on datatype in java lang, the variables are categorized into two types.

(i) Primitive Data Type Variable.

(ii) Non primitive Data Type Variable.

=> Primitive datatype variables will hold values and
non-primitive datatype variables will hold references

11/12/19

Expt: Define String:

=> "String" is a class from `java.lang` comes under non-primitive datatype, but we can use 'String' like primitive datatype because of this reason "String" is a special datatype in java.

Expt:

`String name = "Raj";`

`String brand = "SCE";`

List of some operators used in JavaLang:

1. Arithmetic Operators

(Operator, Meaning)

+

Addition

-

Subtraction

*

Multiplication

/

Division

%

Modulo Division

(Return type)

`int a = 5;`

`a + b`

`a - b`

`a * b`

`a / b`

`a % b`

Expt:

$$a = 7 \quad b = 2$$

$$a + b = 7 + 2 = 9.$$

$$a - b = 7 - 2 = 5$$

$$a * b = 7 * 2 = 14.$$

$$a / b = 7 / 2 = 3 \text{ (quotient)}$$

$$a \% b = 7 \% 2 = 1 \text{ (remainder)}$$

$$\text{Ans} : 9) \quad 7 \quad 3 \quad \text{Quotient}$$

$$\begin{array}{r} 7 \\ \times 2 \\ \hline 14 \\ \hline 14 \\ \hline 0 \end{array}$$

Remainder

Q. Relational Operators

Operator	Meaning
>	Greater than
\geq	Greater than or equal
<	Less than
\leq	Less than or equal
$=$	is equal to.
\neq	Not equal to.

3. Logical Operations

Operator	Meaning
$\&\&$	Logical AND
$\ \ $	Logical OR
!	Logical NOT

Logical AND ($\&\&$)

A	B	$A \& \& B$
T	T	T
F	T	F
T	F	F
F	F	F

Logical OR ($\| \|$)

A	B	$A \ \ B$
T	T	T
F	T	T
T	F	T
F	F	F

Logical NOT (!)

A	$\neg A$
T	F
F	T

4. Increment-Decrement Operator

Operator	Meaning
$+$ +	Increment

Operator	Meaning
--	Decrement

Operator	Meaning
$- - =$	Indirect Assignment

Control Structures in Java:-

- The portion of the program which is controlled and managed by the program is known as Control Structures.
- ⇒ These control structures are categorized into three types →
 - (a) Selection Statements.
 - (b) Iterative statements
 - (c) Branching statements.

(a) Selection Statements:

The statements which are used to select the portion of the program for execution based on some condition are known as Selection statements.

List of S.S.:

(i) Simple if

(ii) If-else

(iii) Ladder If-else

(iv) Nested

(v) Switch case

(b) Iterative statements:

The statements which are used to execute the portion of the program repeatedly on some condition are known as Iterative statements (Looping Structure).

List of I.S:

- (i) While Loop
- (ii) do-while loop
- (iii) for loop

c) Branching Statements: These are the statements which are used to transfer the execution control from one location to another location.

Known as Branching Statements. Transfer Statement

List of B.S

- (i) Break
- (ii) Continue
- (iii) Return
- (iv) Exit

Object Oriented Programming Concept:

The process of constructing application using classes and objects is known as "Object Oriented programming".

Note: The process of controlling and managing the Non-primitive datatypes is known as "Object Oriented Programming concept".

The process of controlling and managing the Non-primitive datatypes is known as "Object Oriented Programming concept".

Levels in Object Oriented programming:

1. Object definition
2. Object creation.
3. Object location.
4. Object Components.
5. Object types.
6. Object cloning.
7. Object serializable
8. Object Locking
9. Object collection.
10. Objects Sorting.

11. Object holding Database Table Record

The following are the non primitive Data types or Referential datatypes:

a) Class:

- a) class
- b) interface
- c) array
- d) enum

- ⇒ class is a "structural Layout" generating objects.
- ⇒ class is loaded on to MethodArea of JVM architecture.
- ⇒ class is a collection of Variables & Methods.
- ⇒ Classes in java are categorized into two types.

1) User - Defined classes.

2) Built-In classes.

i) User-defined Classes:

The classes which are defined by the programmer are known as user defined classes or custom classes.

Ex1 Addition.java

Mainclass.java.

Ques:

2) Built-In classes:

- The classes which are "existing" from JavaLib are known as Built-In classes.

Sol:

Scanner class.

Define Scanner class?

- Using "Scanner" class we can read the data into JavaApp.
- "Scanner" class is available from "Util" package (Utility package).
- "Util" package is available from JavaLib.
- The following are some nonstatic methods from Scanner class:

```
public byte nextByte();
public short nextShort();
public int nextInt();
public long nextLong();
public float nextFloat();
public double nextDouble();
public boolean nextBoolean();
public String nextLine();
```

[char also an string]

which method can be used for removing the extra spaces in a string?

QUESTION

→ To use these Non-Static Methods, we have to create object of "Scanner" class

Syntax of Object Creation :-

class_name obj-name = new Classname();

Scanner s = new Scanner();

Scanner s = new Scanner(System.in);

Method Syntax

byte b = s.nextInt();

short s1 = s.nextShort();

int i = s.nextInt();

long l = s.nextLong();

float f = s.nextFloat();

double d = s.nextDouble();

boolean b1 = s.nextBoolean();

String str = s.nextLine();

Ex: Program :- Wait to read and display Product details using class object concept.

Product
→ pCode, pName, Price, Qty

→ void getProdut();

Main Class