

An ISO 9001 : 2008 Certified Company



040-2374 6666 | 23734842

info@nareshit.com

www.nareshit.com

nareshit

nareshitech

nareshit

Find Latest IT Jobs  
SeshaJobs.com

Opp. Opp. Satyam Theatre, Durga Bhavani Plaza, Ameerpet, Hyd-16 d - 16

# CORE JAVA

## Chapters 1,2,3

by

# Mr. Hari Krishna

for Online Training Call / WhatsApp : 8179191999

# Learning Project Development Using Java

## JAVA 10

Including

Java 4, 5, 6, 7, 8, 9 Features

Core Java.

with Free SCJP (OCJP) and

JVM Architecture

& Kotlin

By: Mr. Haji Krishna  
Nasiresh Technologies

- As part of Core Java, we will learn developing programs Using Java language, also we will learn developing projects using Java language and its technologies.
- As part of SCJP (OCJP) we will learn the same Core Java Topics, but more in detail several test cases to understand complete Java Syntaxes and Execution flow points.
  - SCJP is a Core Java based Certification exam name.
  - It is conducted by SUN Microsystems (online)
  - Now a days, we are calling this exam as OCJP, Because Java language is bought by Oracle Corporation.
- As part of JVM architecture, we will learn how a Java program is executed and How much memory is allocated and when it is destroyed. With the JVM architecture knowledge we can develop fast executing applications, and also we can solve big performance issue easily.

## Najresh IT Websites

- For new batch Updates

Website : najreshit.com

FB : fb.com/najreshit

- For latest job update

Website : seshajobs.com

FB : fb.com/seshajobs

- For Placement information

Website : nacreservices.com

- YouTube Channel

www.youtube.com/najreshit

- Our Websites

- \* For articles and tutorials

Website : hkpiprogrammingworld.com

FB : fb.com/hkpiprogrammingworld

- \* For doubts discussion

fb.com/groups/javahasikrishna

Name : javabyhasikrishna

Closed Group

fb.com/groups/hkpiprogrammingworld

## Software Industry Interview Process

In Every Java Based interview, we all have minimum below 4 rounds :-

1. Written Test Round.
2. Technical Round.
3. Project Manager Round.
4. HR Round.

1: In Written Test Round we will have object to based questions, we have to find answer for the question from the given option.

In this round we will have 2 papers :-

- (i) Aptitude (CRT)
- (ii) Technical (C, DS, CJ, AJ, Ostacle)

If you want to pass written test come we must think like a compiler and JVM at syntaxes and execution flow level.

2. In Technical round, Face 2 Face Interaction will be there, you should answer (speaking) to the question and also should write programs. If you want to pass technical round you must think as a faculty (speaker) & programmer (Developer).
3. In project manager round also you will have Face 2 Face interaction, you should answer to the questions, In this round, question related to project, design, developing and managing. If you want to pass this round, you must have very strong knowledge on concept level subject and think like a manager.

#### 4. Last, HR Round, Most Dangerous Round.

In this round, your attitude will be checked also your strength, Hobbies, behaviour will be check.

- To pass every Java Interview, By End of the Course, you must become

- (i.) Compiler, JVM.
- (ii.) Faculty and Programmer.
- (iii.) Project Designer.

#### Complete Java Course Division

According to SUN Microsystems, Java language divided into 3 parts, They are:-

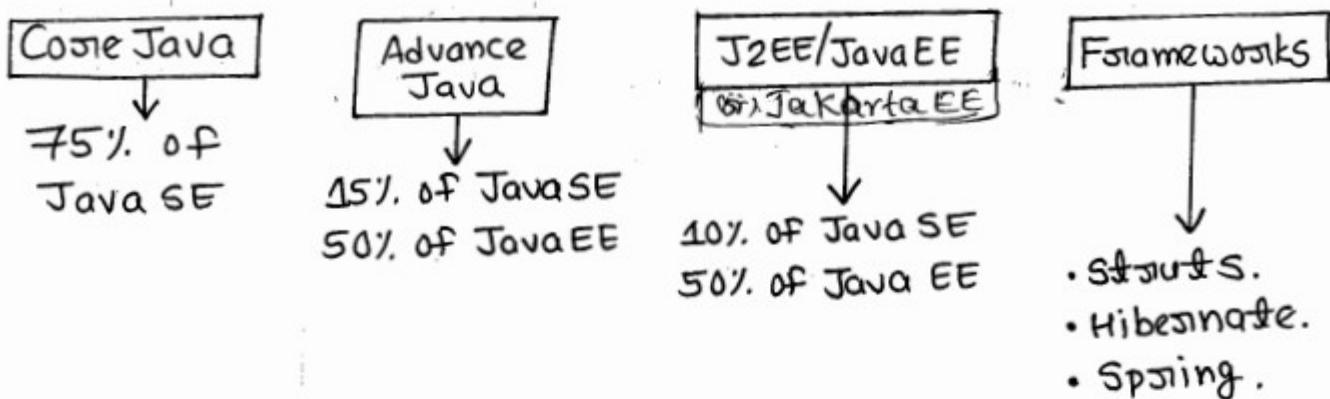
- 1. Java SE
- 2. Java EE
- 3. Java ME

We are here to learn only Java SE & Java EE.

But, According to Amesipet, Java is divided into 7 parts:-

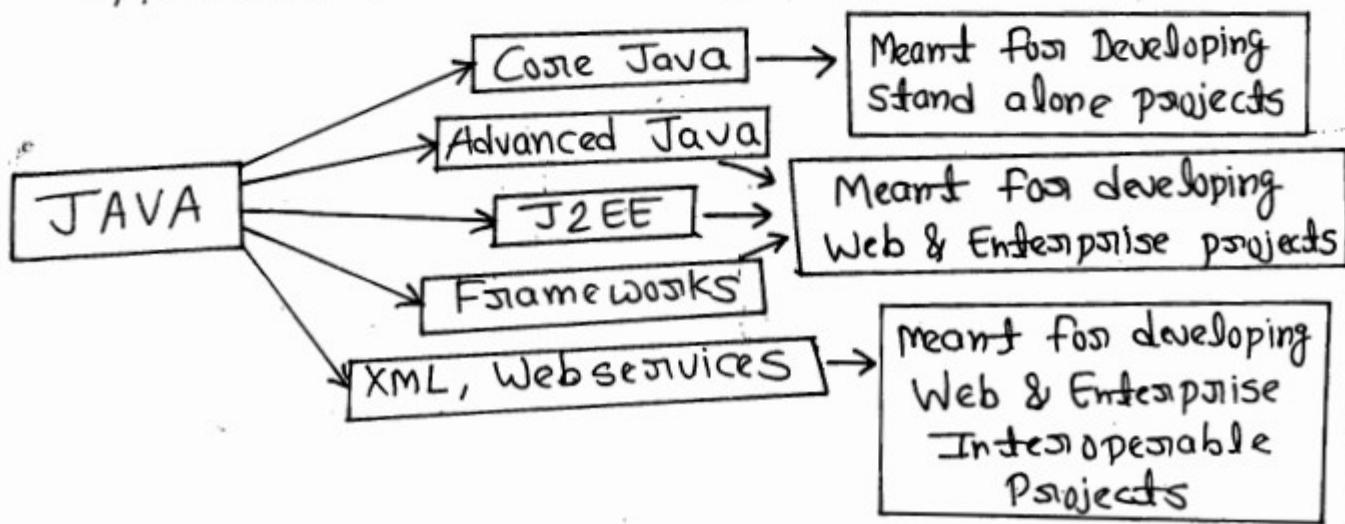
- 1. Core Java.
- 2. Advanced Java.
- 3. J2EE / Java EE.
- 4. Frame Works.
- 5. XML, Web services.
- 6. Java Tools.
- 7. Design Patterns.

- Below diagram show you, the mapping of SUN and Ameesipet.



- \* Frameworks - Struts, Hibernate, Spring are not part of Java language, it means they are not developed by SUN Microsystems and they are not installed with Java Software.
- \* Frameworks are projects developed by using Java language for developing other projects in software company fast and easy. These frameworks are developed by different companies. Struts developed by "Apache Company", Hibernate is developed by "Gavin King" and spring is developed by "Rod Johnson".

- Below diagram show you Java concepts and the application we can develop using those Concept:-



## Must Learn Topics

To get the job and sustain in job and to live happy life in company, We must acquire :-

1. Problem Solving skills.
2. Logical Programming skills.
3. UI Programming skills.
4. Database programming skills.
5. Object Oriented Programming skills.
6. Web Programming skills.
7. Different Programming languages  
Communication Programming skills.

We can get the above programming skills with the help of below courses :-

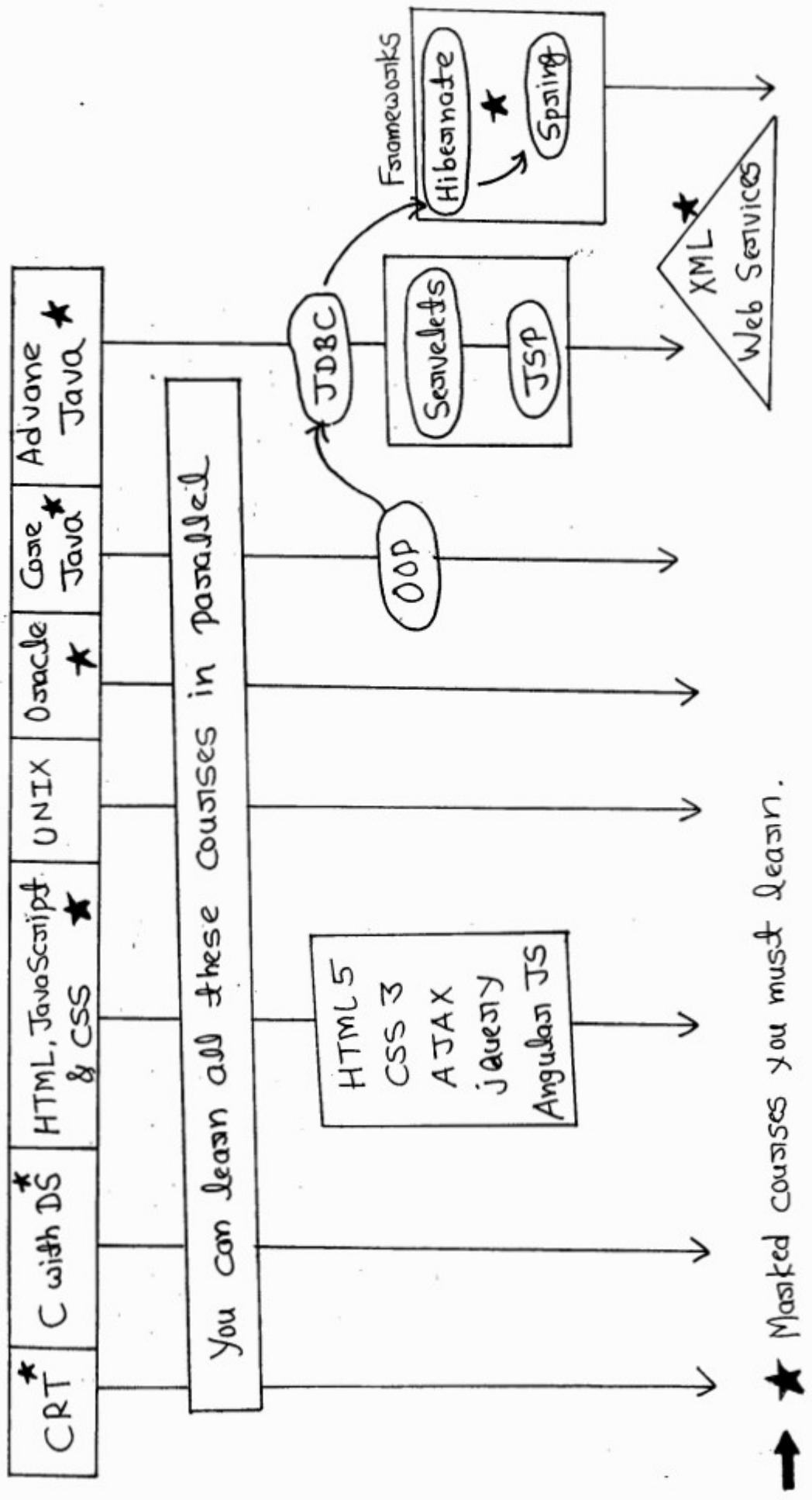
1. CRT (A,R,P,V,TC).
2. C with DS.
3. HTML, CSS, Javascript.
4. Oracle (SQL, PL/SQL).
5. Core Java.
6. Advance Java.

← Mandatorily

7. Frameworks,
8. XML, WebServices.
9. Java Tools.
10. Design Patterns.

← Essential

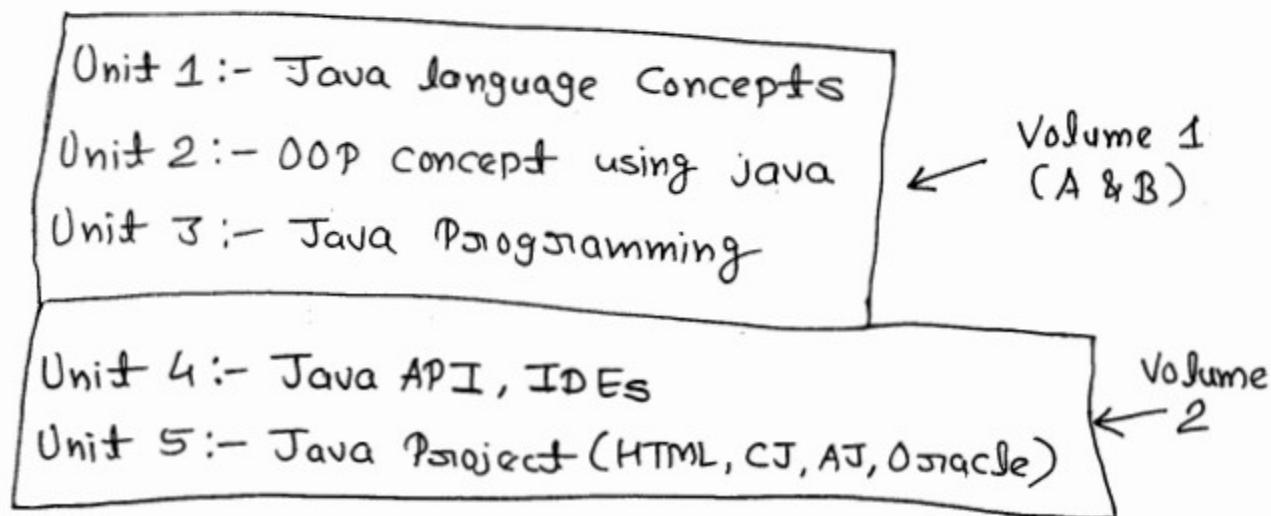
# Java Learning Road-Map





## Course Java Syllabus

We are all here ~~for~~ to learn developing project using Java Programming language. So we should not only learn Java concepts, we should also learn project architectures and developing this project using java language. So we should learn java in a systematic way, in the below 5 units order:-



By end of course completed, we must become project full stack Developer. It means Full Stack Developer must able to develop all parts of project Development. Starting from HTML screens to Java logic to Database scripts.

14/07/17

## Chapter #1 Introduction To Java

### Part #1 : Java History

1. What is Java, why Java ?
2. Who invented Java ?
3. Prerequisites to learn Java ?
4. Java history and features ?
5. Java Abbreviation .
6. Java Versions and Released date .
7. Definition of Java .
8. Where Java is used ?
9. When should be use Java and C ?

Q. 1. What is Java ?

Ans:- Java is a platform independent Object oriented Programming language .

Q. 2. Who invented Java ?

A) Java is invented by a scientist James Gosling in SUN Microsystems for developing internet application.

Q. 3. Why Java ?

A) It is invented to achieve platform independency for developing internet based business applications

Using Java, we can develop both stand alone and Internet application .

#### 4. Prerequisites to learn java.

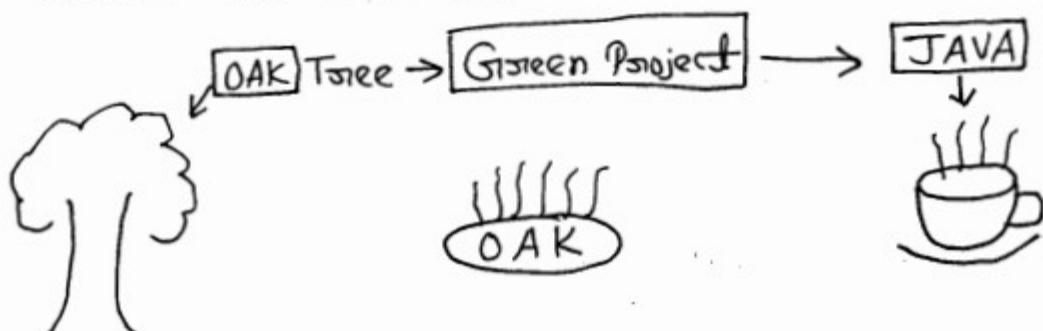
- A) There are no prerequisites to learn java, we can start java as first programming language.
- No need to have C, C++ programming knowledge.

Note: C language knowledge is not required for learning java, but C language knowledge is compulsory for indepth logical programming and to pass interview rounds to get job.

- Java has prerequisites, we must have awareness on internet usage and mathematical calculation using +, -, \*, /, %. operators, because java is meant for developing internet based business app if in business operation maximus will have adding and subtracting money, for e.g. Bank Business.

#### 5. Java History.

- Ans:-)
- James Gosling, Mike Sheridan and Patrick Naughton initiated the java language project in June 1991.
  - Java was originally designed for developing software for network consumer electronics devices.
  - The language was initially called OAK. OAK is a tree that stood outside Gosling's office.
  - It went by the name Green later and was later renamed Java.



#### 6. Java Abbreviation.

There is no abbreviation for java. The development team of java has just chosen this name. The name Java specifically doesn't have any meaning rather it refers,

- to Hot aromatic drink coffee.
- Because the name derived from Coffee cup, Java icon is "Coffee Cup".

### 2. Java Versions and Released Date.

Ans:- SUN Microsystems released the first public implementation as java 1.0 in Jan 23, 1996.

- It promised "Write Once, Run Anywhere" (WORA) on popular platform such as Windows OS, Linux OS, Solaris OS, Mac OS.

- ~~By~~ adding more features to java lang support industry Java language is updated and released in the market with the different versions numbers. The first version of Java is 1.0 and latest version of Java is 1.8, the new version of Java 1.9 is about to release, expecting in this year.

For complete details of version no. and their introduced feature refer youtube.com/nareesit - Java 8 features by Hasi Krishna (video 1 and Video 2)

### 3. Java Features.

Ans:- It has 10 main features, they are :-

- |   |                             |
|---|-----------------------------|
| <u>1. Simple.</u>   | <u>6. Object oriented.</u>  |
| <u>2. Secure.</u>   | <u>7. Multithread.</u>      |
| <u>3. Robust.</u>   | <u>8. High Performance.</u> |
| <u>4. Portable.</u>                                       | <u>9. Distributed.</u>      |
| <u>5. Architecture neutral.</u><br>(Platform independent) | <u>10. Dynamic.</u>         |
| <u>11. Byte Coded</u>                                     | <u>12. Interpreted</u>      |
| <u>13. Garbage collected</u>                              | <u>14. Open Source.</u>     |

## Java Features

It has 10 main features, they are,

- 1. Simple
- 2. Secure
- 3. Robust
- 4. Portable
- 5. Architecture Neutral
- 6. Object Oriented
- 7. Multithreaded
- 8. High Performance
- 9. Distributed
- 10. Dynamic
- 11. Byte Coded
- 12. Interpreted
- 13. Garbage Collected
- 14. Open Source

1. Simple : Java is simple from Learning point of view and also from Development point of view

Learning Point of View Simple because its syntax are similar to C/C++ and omits many of the features those make java language complex, confuse and unsafe. Hence in java we do not have multiple inheritance, structure, union, template etc. Java doesn't support pointer creation in a program, but java software has pointer internally.

From Development Point of View Java is simple because it provides rich set of predefined library (API). Because of this huge predefined API, the development will be fast and easy, because we no need to write code.

2. Secure : Java is secure from the Code point of view and Data point of view.

→ From Code point of view because java compiled program is not directly executed, before its execution

it is verified by bytecode verifier. If there is any malicious code or virus program execution is terminated.

→ From Data Point of view because of accessibility modifier and encapsulation. It means in java the data available in one program can be hide from another program from direct access using accessibility modifier, private, default, public, protected and we can provide data access to another program via method i.e encapsulation.

3. Robust: java is robust it means it is strong in Type checking and terminating execution safely.

→ Java is strictly typed checking language it means it does not allow storing high range value in lesser range variable, if we try java compiler will throw error.

Eg: int x = 10.5; x C.E.

→ Java guarantees fail safe by using exception handling mechanism it means when program execution is failed, the program is not terminated immediately we do not lose the value generated in the previous lines. With exception handling mechanism we can stop terminating program execution, we can save the results eg: Mobile Recharge option through Paytm.

4. Portable: Java is portable, it means we can take java program from one computer to another computer and we can execute this java program in this computer. Java program is portable because:

i) Its memory sizes are fixed across all OS and processor hence we will get same result in all computer.

ii) It can run in any OS because of platform independency nature.

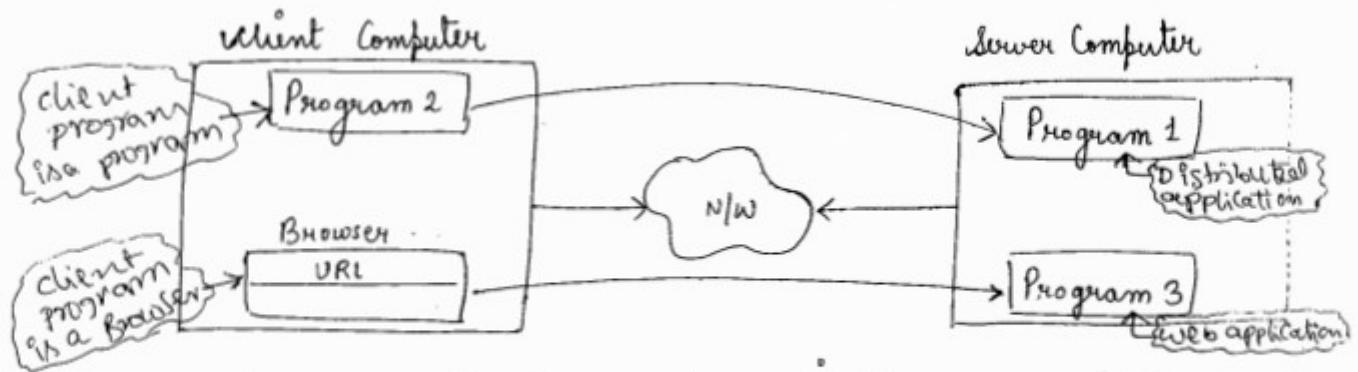
5. Architectural Neutral: It means it is platform independent it means irrespective of other computer hardware devices and software devices and OS Java Program can run in any computer and any OS because of JVM.

6. Object Oriented: Java is an object oriented programming language it means in Java every program must develop around an object by using a concept called class, every class based programming language is an OO programming language.

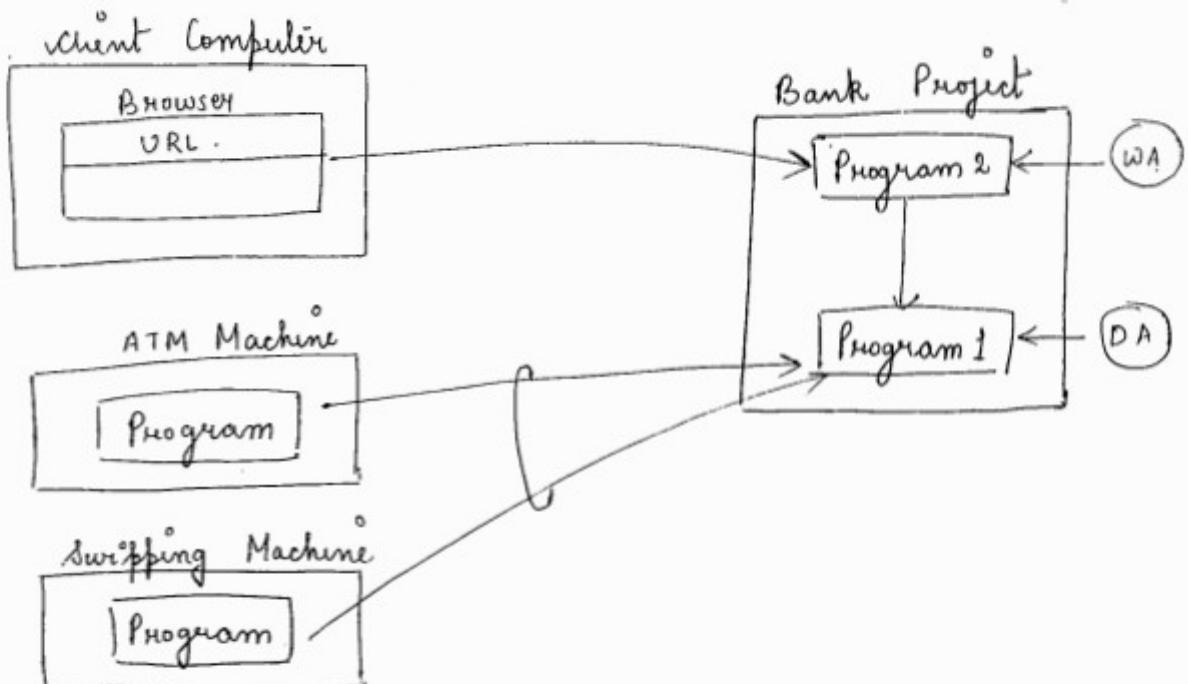
7. Multi-threaded: Java is multithreaded programming language it means in Java we can execute multiple tasks concurrently.

8. High Performance: Java is a high performance programming language it means Java programs are executed fast because of multi-threaded nature and Jit compiler.

9. Distributed: Java is a distributed programming language it means the program we placed in one computer that can be accessible via network from another computer directly, from the program.



- If a server program that can be directly accessed from a program from client computer then it is called distributed application (DA).
- A server program that can be indirectly accessible only from a browser through URL it is called web application or web component. All bank application are distributed and web application.



10. Dynamic Programming: Java is a dynamic programming language from two point of view (i) memory point of view (new keyword) (ii) functionality point of view (inheritance and polymorphism).

→ Java is dynamic from memory point of view. It means we can allocate multiple copies of memory for the same variable at the execution time by using new

### Keyword

- Java is dynamic from functionality point of view it means once we develop project we can add more features and we can modify existing features dynamically without modifying source code. It is possible because of OOPs concept inheritance and polymorphism.
- If we want to call a language is dynamic we must allocate multiple copies of memory and add more features without modifying source code.
- C is a static programming language because in C we can allocate multiple copies of memory for the one variable declaration and we cannot add more functionality without modifying source code.

Eg : GSM mobile is an example for dynamic nature  
It can accept any SIM without modifying its software

CDMA mobile is an example for static nature. It can accept only one type of sim. If we want to change sim to another company, we must modify software in mobile.

- In any programming language compiler will not allocate memory for variable memory is allocated only at execution time by the runtime system of this language. Compiler is only responsible for verifying the syntax and converting the source code either into machine language code or bytecode.

Eg : A person is dynamic will have full common sense. A student is dynamic outside the classroom, have full common sense to take decision but inside classroom student is static, never use brain and common sense  Just kidding.

11. Byte Coded : Java program is compiled into a special instruction set called byte code, Java program is not compiled into machine language code like in C/C++. Hence Java is byte coded.
12. Interpreted : Java is called interpreted programming language it means like in C/C++ program Java program is not directly executed, because its compiled code is not machine language code rather it is byte code. In Java to convert bytecode to machine language code and further to execute this machine language code, inside JVM Java uses interpreter. Because Java program is executed by interpreter, Java is called interpreted programming lang.
13. Garbage Collector : Java is a automatic memory management programming language it means in Java we no need to destroy object memory it is automatically destroyed by a special program called garbage collector. But in C++ programmer only responsible for creating and destroying objects.
14. Open Source : Java is a open source software. It means
- it is freely available to download
  - its source code is available to programmer either for reading or modifying
  - any third party company or programmer can suggest and add more functionalities to the software
  - the other popular open source softwares are Tomcat server, eclipse, hadoop, etc.

All products developed from Sun Micro System and Apache Organization are open source products.

## Definition of Java

- Java is a very simple, high-level secured, multithreaded, platform independent, object oriented programming language.
- It was invented by "James Gosling" in 'SUN Microsystem' for developing networking application.

Where will we use Java?

We use Java in developing :-

1. Desktop application - calc, Trades console.
2. Web application - seshajobs.com, tiedbus.com.
3. Enterprise application - hdfcbank.com.
4. Interoperable application - paytm → bank site.
5. Mobile application - ~~Android~~, WhatsApp etc.
6. Gaming application - ClashofClans, Candy Crush etc.
7. Robotic application - Chitti.

Using java language we can develop any type of application for any platform, e.g. computer desktop, internet, mobile, electronic device level any application we can develop. that why java is called full fledged developed programming lang.

When should be use C and when should be use Java?

- We use C for developing system level and application level software which are stand alone and platform dependent. For example: OS, Drivers, ms-office.  
Where as
- We use Java only for developing Application software (business operations) which can be executed in different OS either as stand alone or via internet.

17/07/17

That's all about Part 1 - Java History related points. Before moving to the next topic do revision by reading the main bullet points, given in Agenda Section.

Now lets learn next topics related to project

### Part #2

#### Project Design Architectures

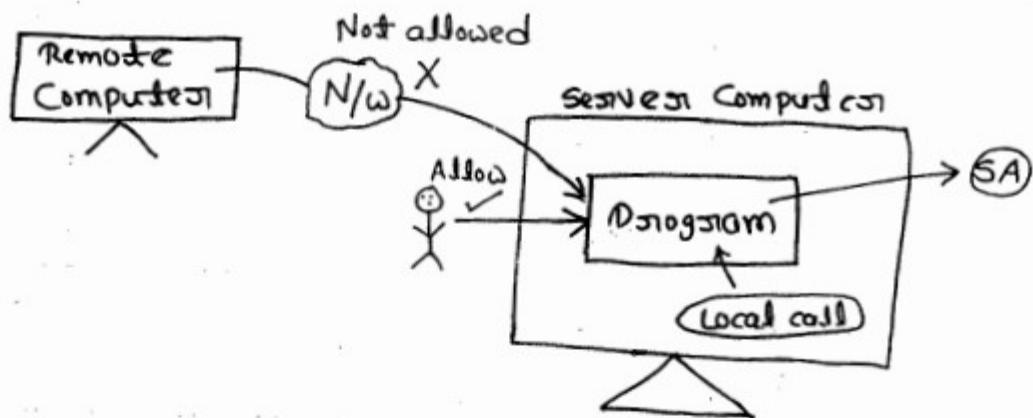
1. Types of applications.
2. Types of stand alone applications.
3. Types of internet applications.
4. Gmail Architecture.
5. M-V-C Architecture.
6. LC-RP Architecture.

#### 1. Types of Applications.

- Java supports two types of application Development
  - (i) Stand alone applications.
  - (ii) Internet applications.

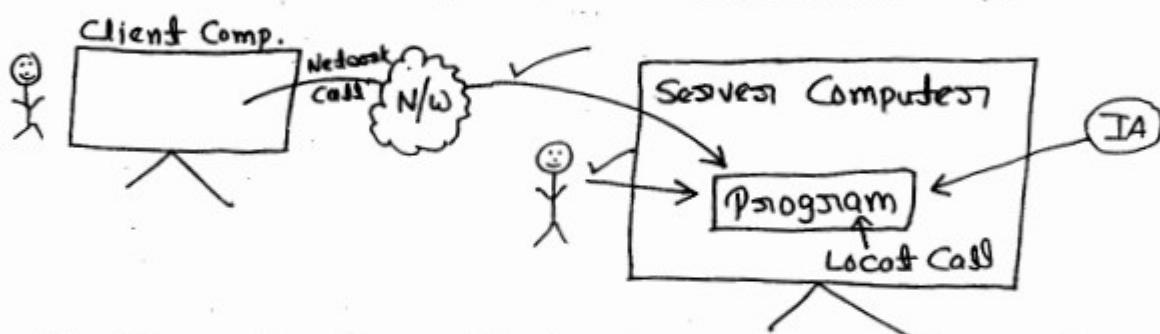
##### (i) stand alone application.

A single<sup>user</sup> and single computer application is called stand alone application. It means a program that can be accessible only in one computer in which it is installed by single user is called stand alone application. A stand alone application can not be accessible from remote user via network.



## (ii.) Internet Application

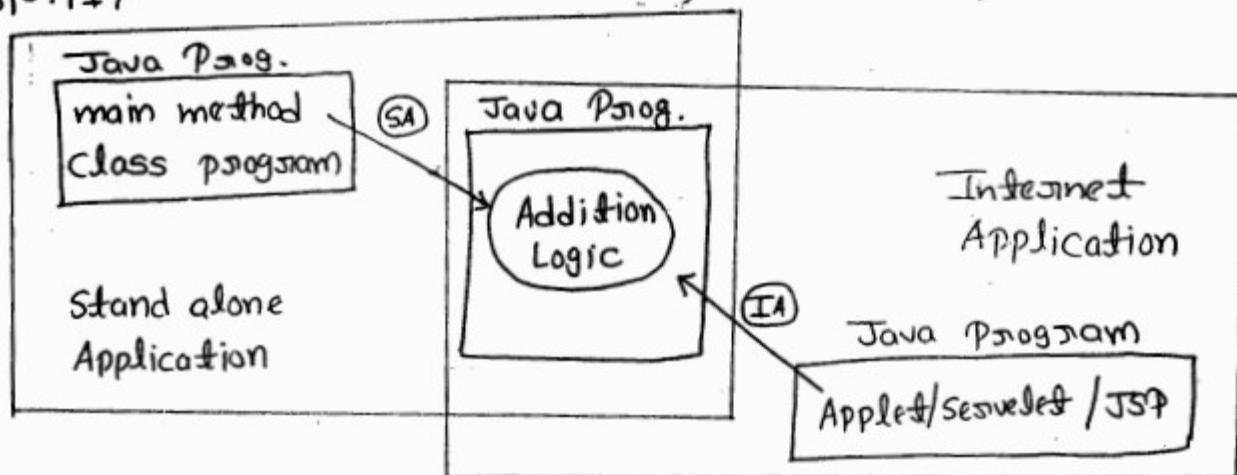
Internet application is a multiple computers and multiple users application. It means a program i.e. accessible locally and also remotely from different computers through network call by multiple users at a time, is called Internet Application.



In Java, a main method class is a stand alone application, it can not be accessible from other computers through network (LAN/WAN/MAN)

- In Java Applet, Servlet and JSP and Frameworks is developed on top of these technology based program os Internet or web applications.
- By default every java program is a stand alone application. If we attach or plug this program from Applet, Servlet or JSP then this application can be accessible and executed via internet from other computers.

18/07/17



- All core java programs are stand alone program means main method classes.

## 2. Types of Stand alone Application.

We have 2 types of stand alone application

(i) CUI based applications.

(ii) GUI based applications.

GUI based applications are also called window based application or desktop application.

CUI stands for Character User Internet application.

GUI stands for Graphical User Internet application

### CUI

(i) An application that takes input from end user by providing text based character messages is called CUI Application

### GUI

An application that reads input values from end user by providing separate window with required no. of labels, text boxes, buttons is called GUI Application.

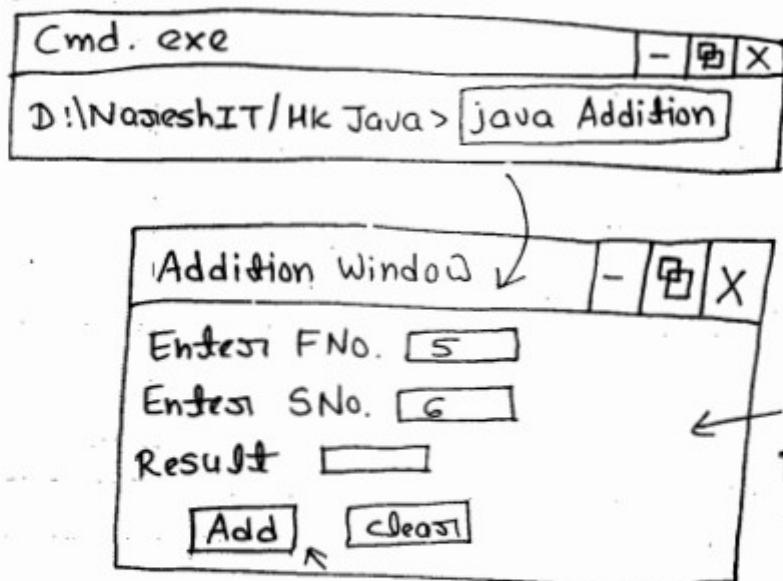
(ii) Cmd.exe

D:\NagesIT\HKJava>Java Addition

Enter FNo: 5  
Enter SNo: 6  
Result: 11

Character based interaction so this Addition program is called CUI based application.  
They do not have special windows.

## (iii) GUI

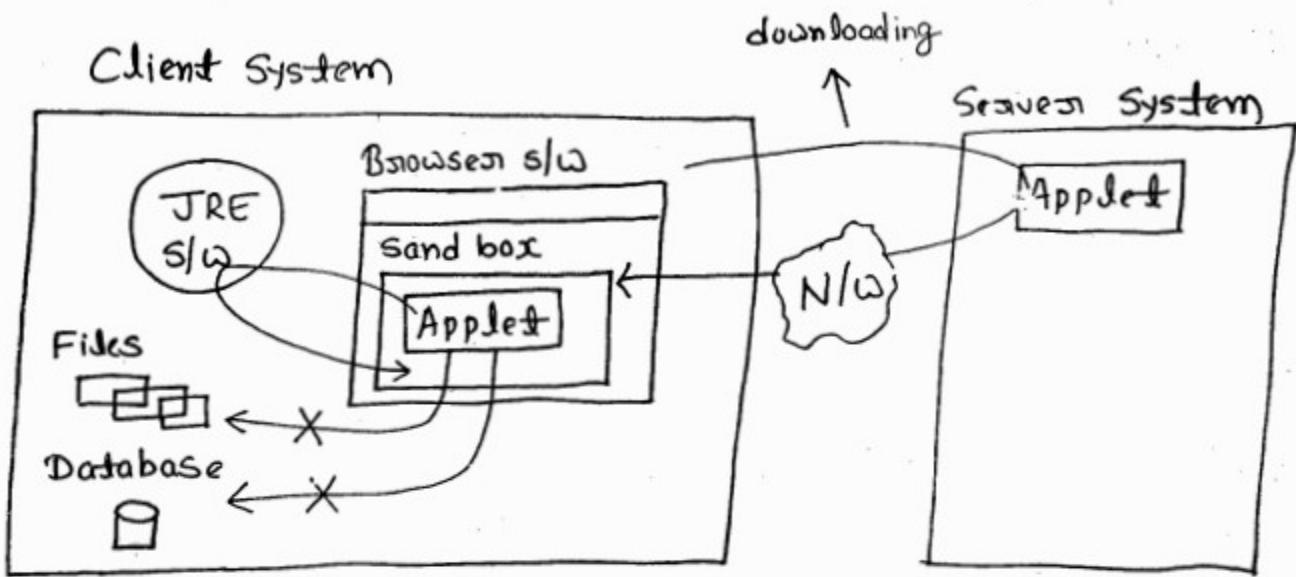


Because inputs are reading by providing window this Addition program is called as GUI application.

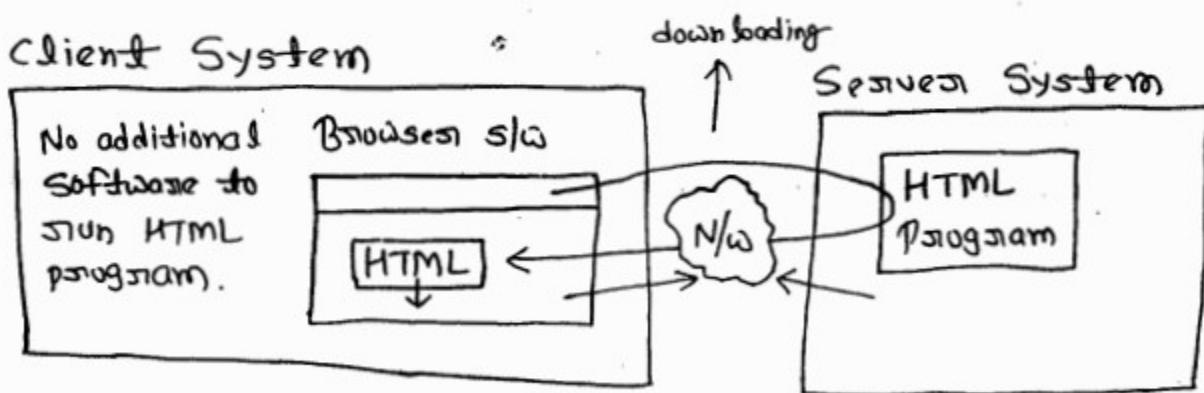
To develop CUI application we create main method with Scanner API (Predefined program)

To develop GUI application we must create main method class by using AWT or Swing API

- By default every internet application is a GUI application while because in internet application from client system to server system we must pass all ~~as~~ input values at a time, it is ~~not~~ possible in CUI application.
- Internet based GUI application is developed by using Applet (~~Java~~ Java program) but now a days we are not using Applet, it gives less performance, it takes more downloading time and also in client computer we must install java software, also should do lot of ~~co~~figuration to run applet program in our computer. we should do ~~co~~figuration to run applet, applet by default will not contain permissions to access file system and database in our computer because it is downloading from internet. Applet will run in a special ~~not~~ place called sandbox.



In modern web applications, in place of applet we are using HTML, CSS and Javascript to develop GUI application ~~for~~ for collection input from user. Because HTML is light weight it means it doesn't carry additional program along with it, so downloading and execution is fast and also no need to install any additional software, browser is sufficient also no need to do any additional configurations because HTML program by default can not access file system or data base.



## Difference between CUI & GUI

### CUI

1. Doesn't have Special window directly executes in Cmd window.
2. It reads input from end user by providing character based text messages.
3. It doesn't have different font, colours, graphics images.
4. It can send one value at a time to application.
5. It can not provide complete information to end user.
6. After output display entering input, we can not change it. Because value is already submitted to application.
7. After output display, program execution is terminated, we must run this program for other inputs.
8. It can developed by using CMD line args application and Using BufferedReader, or Using Scanner or by using console.

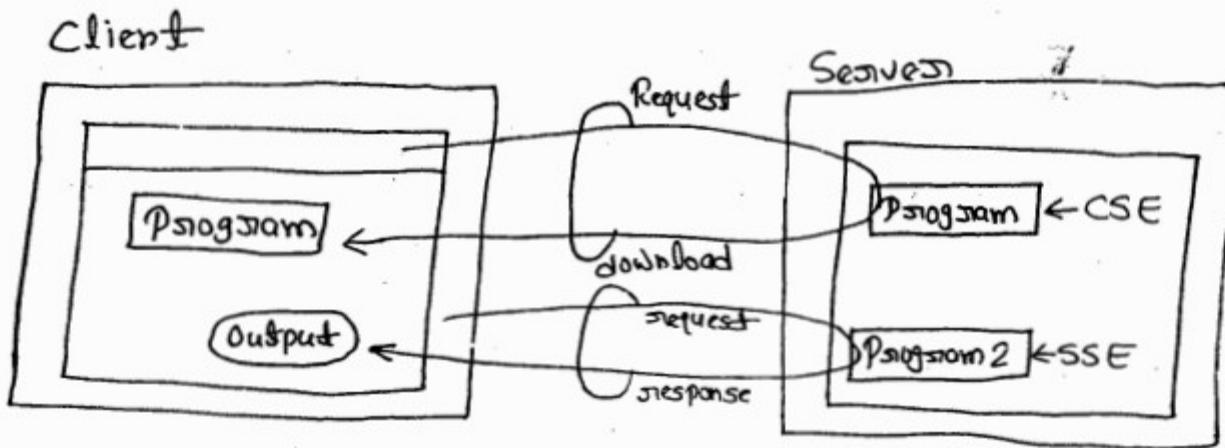
### GUI

1. Provides separate window.
2. It reads input from end user by providing labels, text boxes and buttons.
3. It can provide different font, colours, graphics and images.
4. It can read and pass all input values at a time to application.
5. It can provide complete information and more clarity to end user.
6. We can change value, bcoz value is not submitted to application.
7. After output display, program is not terminated, we can use same currently running program for new inputs.
8. GUI application is developed by using
  - (i) ~~AWT~~ AWT } Stand alone
  - (ii) Swing } GUI application
  - (iii) Applet } Internet
  - (iv) HTML } GUI application.

19/07/2017

### 3. Types of Internet applications.

- We have 2 types of internet applications
  - 1. Client side executing applications.(CSE IA)
  - 2. Server side executing applications.(SSE IA)



- An application that resides in server system, when client send request if it is downloaded in client system browser, it is called client side executing internet application.  
e.g. login page, registration page and every HTML page is a client side executing internet application, also mobile apps installing inside our mobile are also client side executing internet application.
- An application that resides in server system, when client send request if it is executed directly in server system and sends only output as response to client, it is called server side executing internet application.  
Java, .NET, PHP and other internet supporting ~~not~~ languages programming code is server side executing application.  
For example, user registration, login data validation and some calculations, data storing retrieving updating and deleting from database programs are server side executing internet applications.

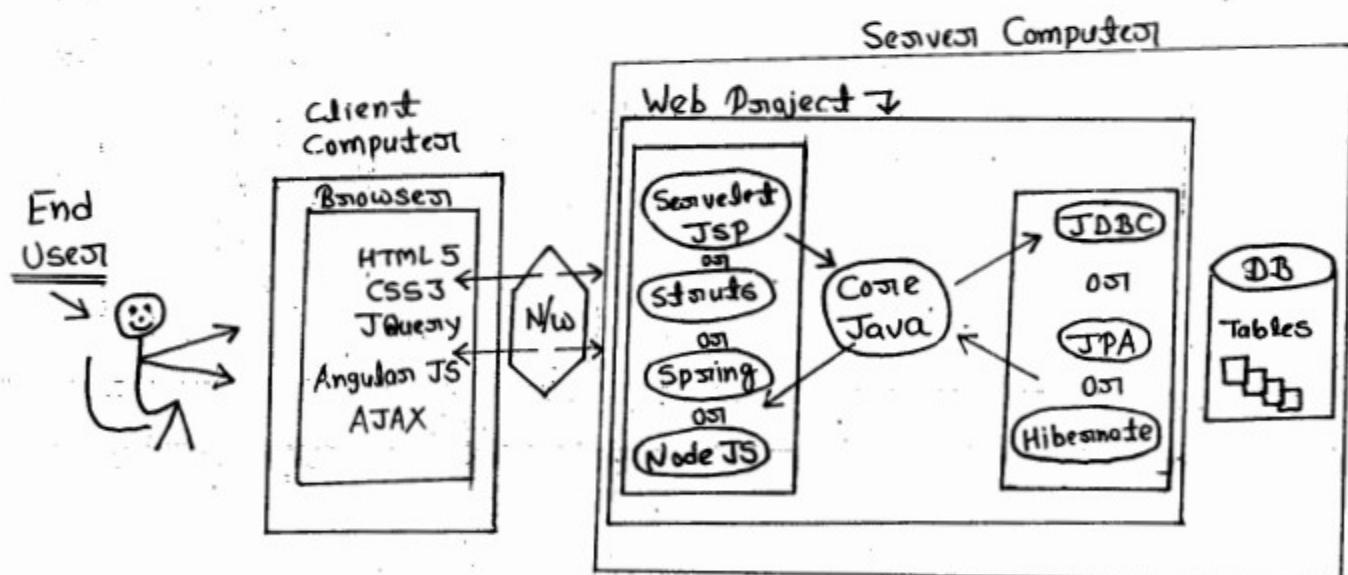
- Purpose of client side and server side application:-

- Client side execution application is used for running a program in client system to read input and display output to a remote user throughout the world.
- Server side executing application is used for processing request and preparing response.

\* Processing request means reading inputs from remote user, verify or validating these inputs right or wrong, if right (correct) performing some calculation by interacting with database, generating output, is called processing request.

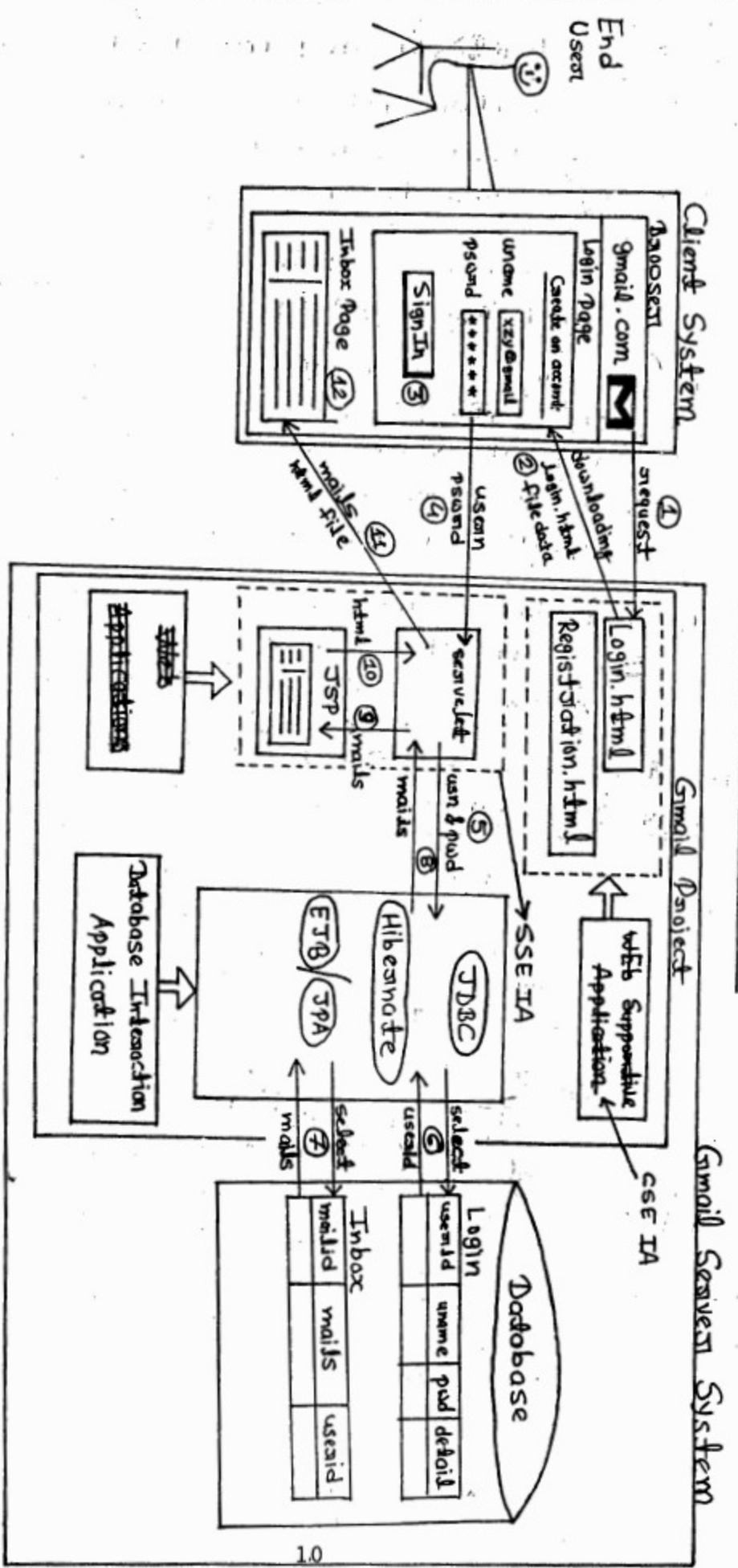
\* Preparing response means generating Dynamic HTML page with the given output is called preparing response.

- Below diagram will show you internet application development by using Java language concepts and technologies:-



20/07/17

## Gmail Architecture



With respect to ~~the~~ diagram, Conclusion Points:-

1. Every stand web project minimum 3 programs (Components)

- (i) Client side execution component.
- (ii) Server side execution component.
- (iii) Database interaction component.

2. To develop above components in Java we use

- (i) HTML/Applet :- to develop client side execution component.
- (ii) Servlet and JSP, Spring-MVC :- To develop Server side execution component.
- (iii) JDBC, JPA, Hibernate, Spring-DAO :- to develop database interaction component.

3. To develop above component in .NET we use

- (i) HTML : To develop CSE component.
- (ii) ASP.NET : to develop SSE component.
- (iii) ADO.NET : to develop DIA component.

| <u>Java</u>       | <u>.NET</u> |
|-------------------|-------------|
| 1. Core Java      | C#.NET      |
| 2. Servlet<br>JSP | ASP.NET     |
| 3. JDBC           | ADO.NET     |

4. ~~Both~~ CSE is responsible for <sup>creating &</sup> providing UI and report to End user

~~Both~~ for collecting input from end user, then pass it to SSE component and then for gathering output from SSE component, finally displaying it to end user.

5. SSE component (Servlet and JSP) is used for

- (i) Processing request.
- (ii) Preparing response.

Processing request means reading input from network i.e send by end user via HTML forms then Executing business logic validation and calculations by using this input then finally generating output and sending it to JSP for preparing response.

- 7.
- Preparing response means generating Dynamic HTML page by using the output given by the servlet.
  - Returning the result HTML page to servlet.
  - Finally Servlet returns this response ~~to~~ HTML page to browser to display output to end user.

Note:

In projects we will use servlet  
(i) Servlet for processing request.  
(ii) JSP for preparing response.

8. Database Interaction Components (JDBC, JPA, Hibernate)  
~~SE~~ is used for performing CRUD operation on persistence operation on database to store, read and <sup>delete</sup> update data permanently on database.

CRUD operations Means,

Database Terminology

C - Create → Insert

R - Read → Select

U - Update → Update

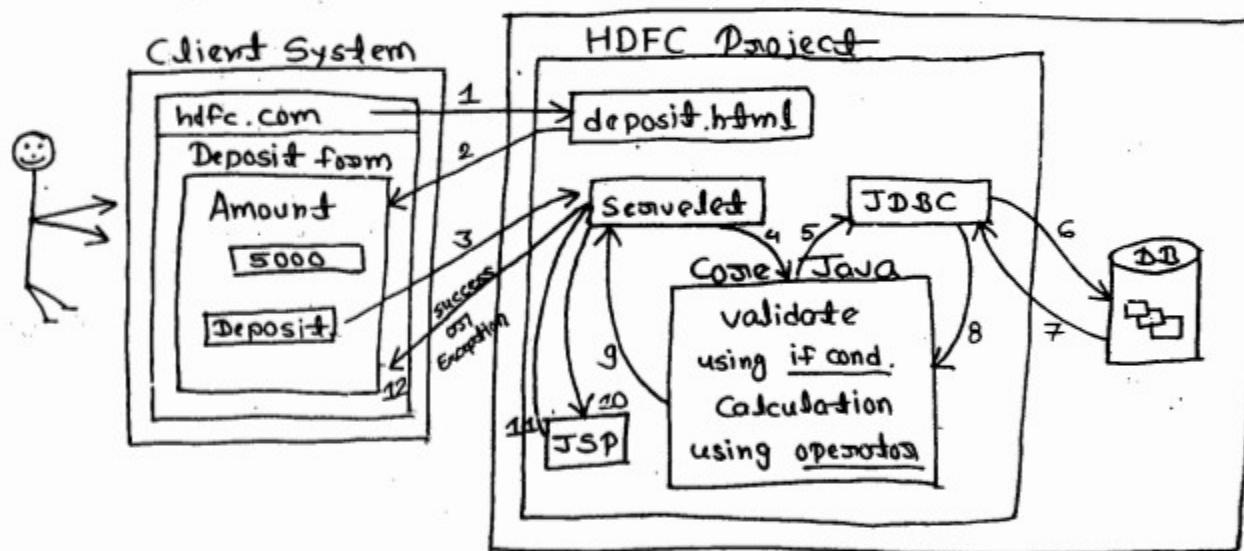
D - Delete → Delete

This is the basic Idea on real time projects architecture, different components exist in projects and different technologies and frameworks use to develop these components, and project execution flow

Q. In above project (Gmail Arch.) we have use HTML, Sessionlet, JDBC, JSP, Hibernate, Spring; Where, is Core Java in this project?

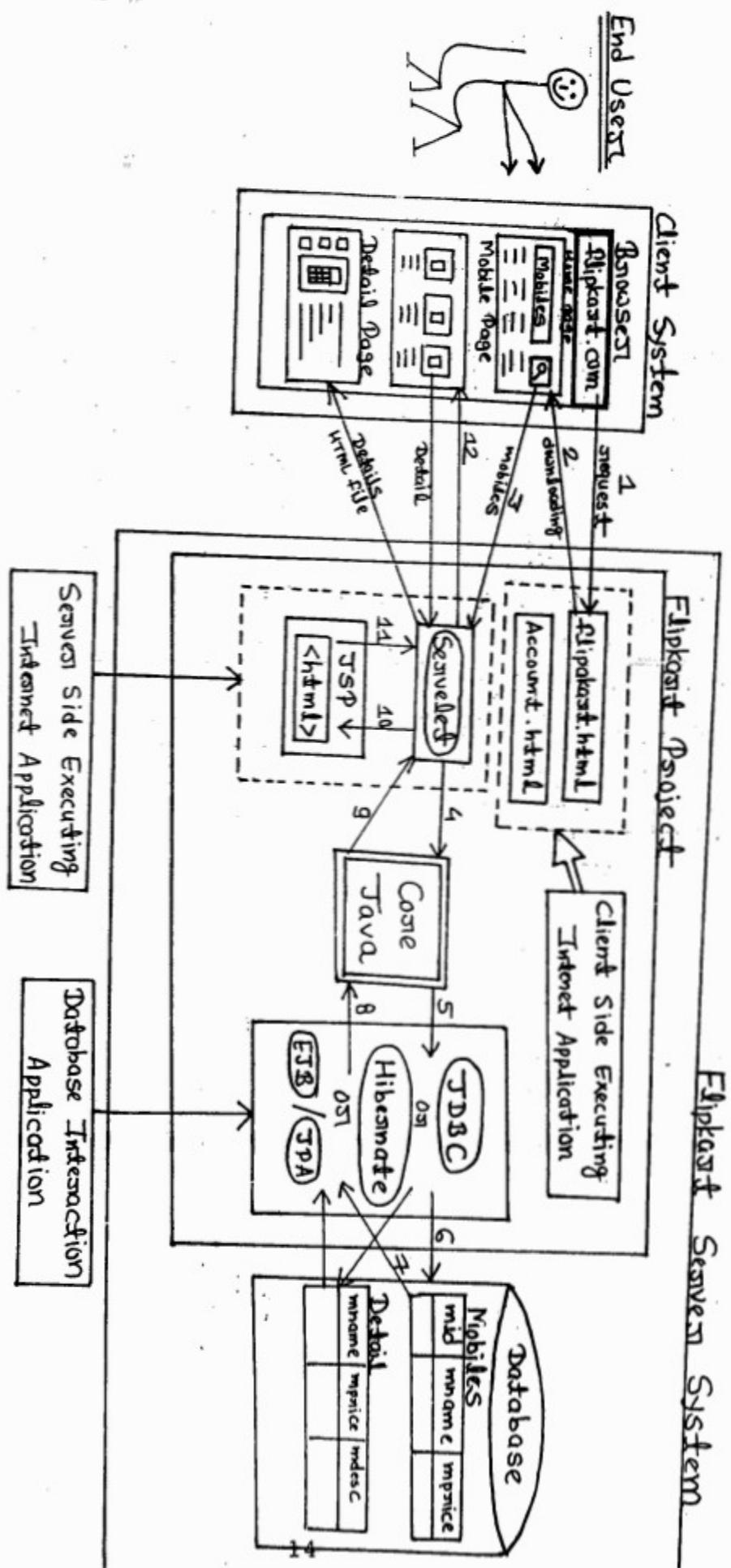
Ans:- Core Java is a GID, you can not see directly.

Core Java is the basic Java language the complete syntaxes rules and guidelines to develop a program is provided by core Java. Sessionlet program, JSP program, JDBC Program, Hibernate, Spring programs are itself core java programs, and the background logic developing in this program using if conditions, loops, variables, object, operators, exception handling, multithreading, collections, everything is Core Java.



Assignment: Visit any website on your choice, flipkart.com. Observe different operations you can do on this site starting from home page URL and observe all our applications involvement in this website. Draw Gmail Project kind of Architecture diagram for this flipkart website as it is appearing on browser.

# Flipkart Architecture



## MVC Architecture

1. The standard architecture used to design a project is MVC.
2. The standard architecture used for developing project is LCRP.

### Main Point of MVC :-

1. Abbreviation and definition of ~~the~~ MVC.
2. Central idea of MVC.
3. Advantages of MVC.
4. How MVC can support parallel development and separation of roles of code development.
5. ~~What~~ Architecture is a layer and what ~~be~~ will be there in model, view and controller layer?
6. MVC Architecture Diagram.
7. MVC Execution flow with java technologies.
8. Why this architecture is named as MVC?
9. Technologies combination used in project development?
10. The purpose of developing project, types of operations, logics developed in project and the technologies used in developing these different types of logics?
11. Software engineering skills.
12. About software industry and important terminology, Types of projects, Types of teams and types of companies.

1. Definition: MVC stands for "Model View Control".

MVC is a standard architectural design pattern for developing World Wide Web applications. Initially this architecture is used for developing only web applications, Now a days we are also using this architecture for developing stand alone applications also to use its advantages.

→ Design pattern is a solution to reoccurring problem. MVC is the solution to project architecture level problem, so MVC is called architectural design pattern.

2. The central Idea behind MVC is separation of roles of code development to achieve code reusability, parallel development and fast development.

3. Advantages of MVC :-

- (i.) Separation of roles of code development (modularity).
- (ii.) Code reusability.
- (iii.) Parallel development.
- (iv.) Development is fast.
- (v.) Time and Money is saved.

Because of above advantages MVC become standard architecture for developing all project using any language (Java or .NET or PHP).

4. How MVC can separate roles of code development ?

- Based on the types of operation we performing project.
- MVC can separate roles of code development

In Every project we will perform 3 operations.

They are:-

- (i) Reading input and displaying output.
- (ii) Performing validation & calculations to generate output by controlling execution flow.
- (iii) Storing output permanently in a database or in files.

Above 3 operation can be developed separately by creating separate programs. Hence based on above 3 operations developing style MVC can separate roles of code development and allows multiple programmers to develop above 3 operation logic in parallel, so that project development is fast.

Hence, if you design project by following MVC architecture, the project will contain 3 layers

- (i) Model layer.
- (ii) View layer.
- (iii) Controller layer.

## 5. What is a layer?

The logical separation of code in project is called layer, each layer contains one or more programs as one group to perform particular operation.

MVC architecture divides project into 3 layers:-

- (i) Model    (ii) View    (iii) Controller layers.

(i) Model layer : A program or a group of program performing CRUD operation on DB are collectively called Model.

- JDBC & Hibernate comes under model layer.

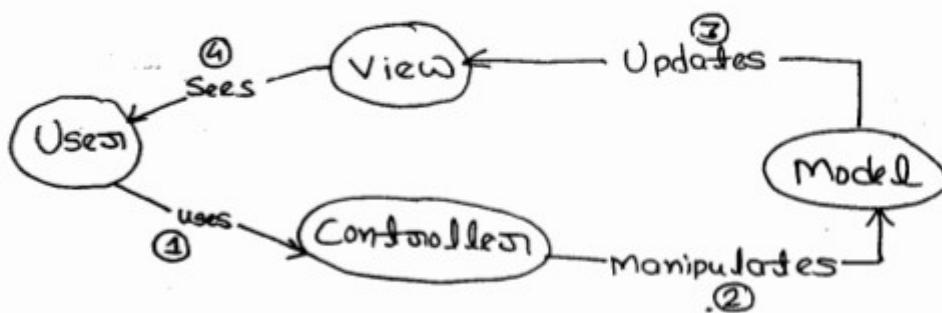
(ii) View layer: performing data presentation operation (reading and displaying) by creating GUI and report are collectively called View.

- HTML and JSP programs are comes under View layer.

(iii) Controller layer: A program or a group of programs performing controlling, validations and calculations operations are collectively called Controller.

- Servlet and Core Java logic programs are comes under Controller layer.

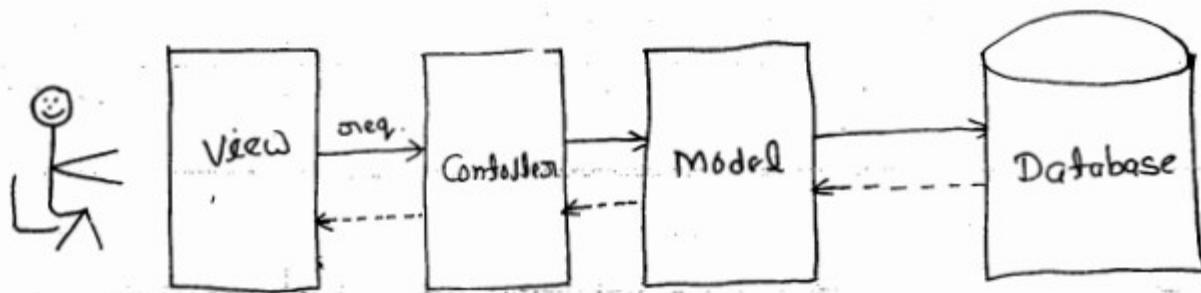
## 6. MVC Architecture Diagram



The meaning of above diagram is :-

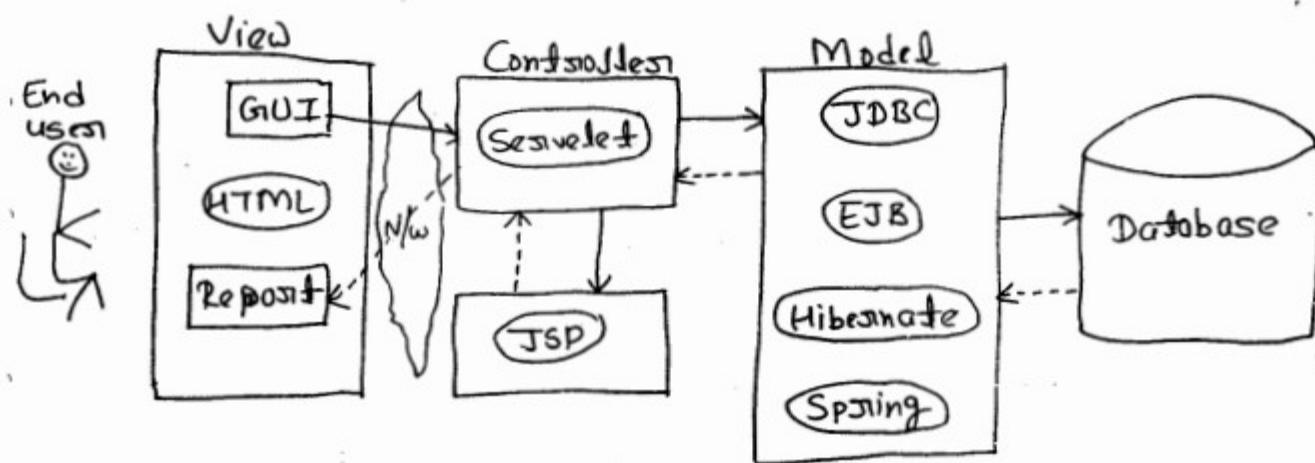
- ① User uses controller for sending input to project
- ② Then controller will perform manipulations (V & C) by using model components by interacting with database.
- ③ The model generated output will be updated in ~~JSP~~ view pages in HTML tags.
- ④ Finally the dynamically generated HTML page will passed to browser displays output.

Above diagram is not showing you actual execution flow of project, below diagram will show you MVC architecture Execution flow.



Above Two architecture diagram are common to all projects and all languages.

I: Below diagram will show you MVC architecture based project development in java language.



Q: Why this architecture name is M-V-C?

This architecture name is as MVC, based on the sequential order this architecture suggest us to design project components. This architecture suggest to design components in the below order

1. First we should design DB objects, following by Model component to perform CRUD operations.
2. Then next we should design view components to read input from user and to display output to user.

3. Finally we should design controller components for integrating view and model components to exchange input and output values.

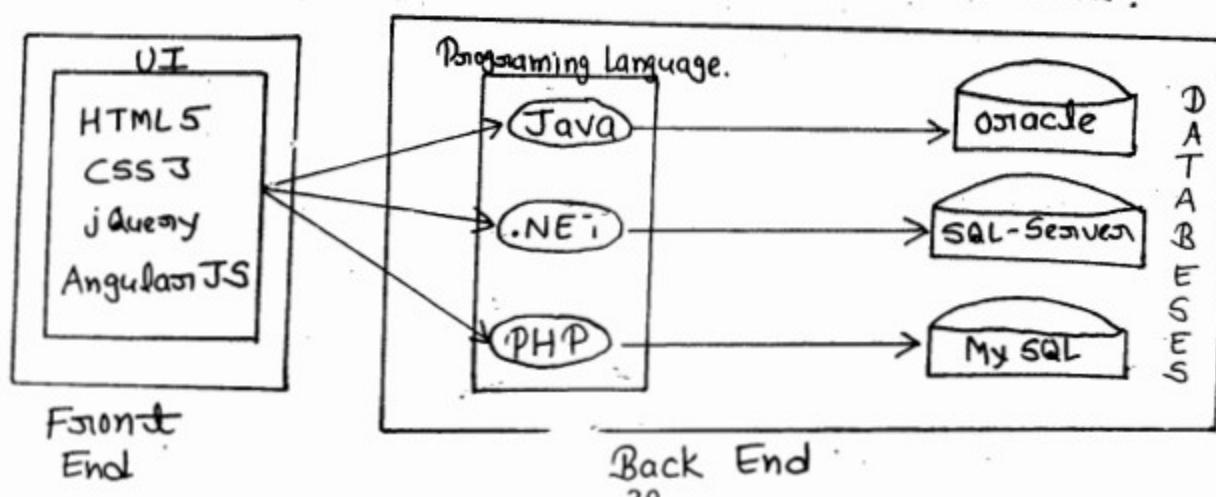
~~Because this architecture suggest us to design model components first, then next view components and finally controller components, it is named as MVC.~~

Q: Oye, MVC architecture provides parallel development, but in above statements you are telling MVC is Sequential, what is this non-sense?

Ans:- MVC based project design is sequential,  
Development is parallel.

- Hence, Component designing means just listing out no. of programs and their names. We can not design components in parallel.
- Developing components means developing code for this project operations with the given names.
- Once names are decided, multiple programmers can develop them in parallel.

9. Technologies combination used in project Development.



Once a project come into software company, project designer and manager will decide front end and back end development technologies based on the scale(size) of the project.

- High Scale project (large) is developed by using Java and Oracle.
- Medium Scale projects are developed by using .NET and SQL Server.
- Low Scale projects (small) are developed by using PHP and MySQL.

Large scale projects means its data require high security. And if contain many functionality, generally large scale application are developed by using Java.

But according to real world, we can use any of the above 3 languages to develop any scale based application.

According to Ameejipatany <sup>(Learning)</sup> programming language can communicate with any database, but in project we will use above combination for best results.

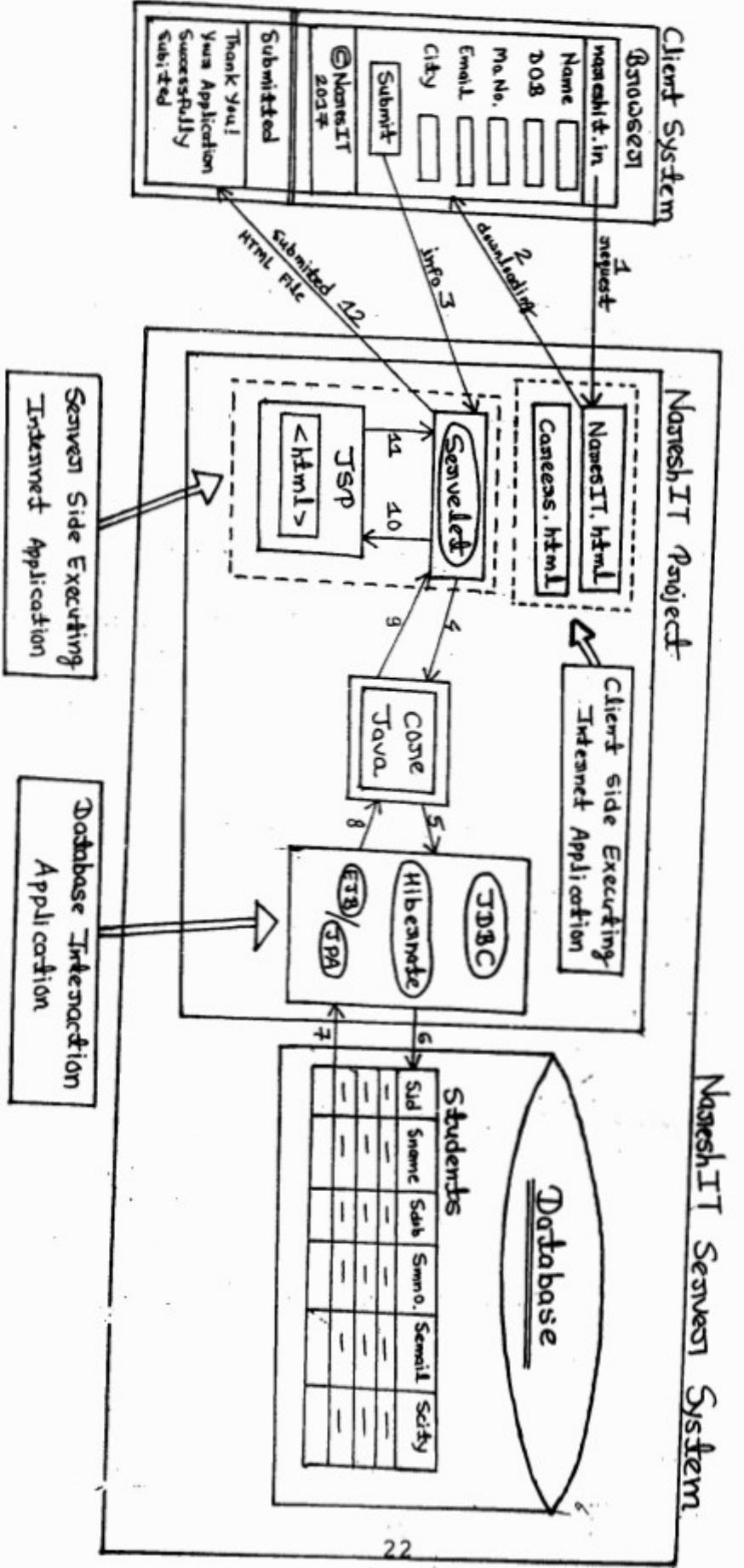
## 11. Software Engineer Skills

To become software engineer, to develop MVC based any layer component, we must acquire below skills.

1. One programming language - Java/.NET/PHP
2. UI preparation language - HTML, CSS
3. Validation Scripting language - JavaScript
4. One database - Oracle/SQLServer/MySQL
5. Server OS - unix commands and shell scripting
6. Problem solving skills - CRT and C with DS
7. Presentation skills - MS Office (Doc, PPT, Excel)
8. Communication skills - English.

## Assignment

By following MVC architecture Draw the Designing of NareshIT students information maintenance system.



Q. How should we understand real time projects which are already developed and running successfully? and to solve errors and add more enhancements.

- In software company projects are already developed and running successfully. As a new joinee our role is fixing errors or in project execution and adding more functionality to this existing project.
- For fixing the errors, for adding more functionality, we must thoroughly understand project execution flow.
- The clue to understand project execution flow is
  - (i) Identify no. of forms and reports exist in this project and running in browser.
  - (ii) Then should identify which form data is storing in which table column and which table data is retrieving and displaying through report.

To find this execution flow we must follow below procedure

- (i) Identify which servlet class is calling from this HTML call by using <form> tag action attribute value

```
<form action = " /msgseveld" >
-----  

-----  

</form>
```

Servlet program  
calling from this  
html form

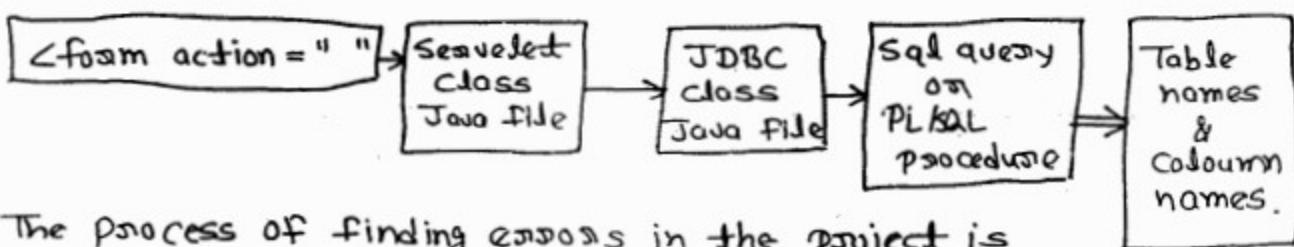
- (ii) Open this servlet program java file, find the JDBC program name used in this servlet class.
- (iii) Open this JDBC class java file, find the query running from this JDBC program on database.
- (iv) Find the table names and column name used in this query, that's it we ~~found~~ in which table, columns form data going and storing.

- Apply above procedure every form and report to find their background connected database tables and respective classes.

- Once we found database tables, their respective classes, we can easily modify existing code to solve errors and also we can add more code for adding enhancement to the project.

### Summary:

To understand project execution flow to fix errors in project and to add more enhancement code, we must find Table name, columns and their respective java classes those are connected to html forms and repositories running in browser in front of us in the below procedure:-



The process of finding errors in the project is called "Debugging", above process is also called debugging approach.

### About Software industry & its Technologies

Q. What will you do in software industry?

A. We will develop a software for automating business operations, For example Bank.

#### • Types of softwares

Every software comes under one of below types:-

1. Project.

2. Product.

#### • Difference between Project and Product.

A project is a software that is developed for a particular client requirements.

For example:- A software development for HDFC bank.

A product is also a software that is developed common for many client for the same requirement

- Types of Companies:-

We have two types of companies based on project and product development, they are:-

1. Project based / Service based companies.
2. Product based companies.

Project based companies develop only projects, these are also called as service based companies. such as TCS, Infosys, CTS, Wipro etc.

Product based companies develop products, they are Oracle, IBM, Apache, Microsoft etc.

- Maintenance Project

Project development comes under two levels:-

1. New Projects.
2. Maintenance projects.

A project under developments is called New project.

A project which developed and delivered to the client is called maintenance project. It is also called project under production. In this stage this project is used by end users for doing their transactions.

- Types of Teams

Based on the above two types of projects development we find two types of teams, they are:-

- (i) Development Team
- (ii) Supporting Team or Maintenance Team.

- (i) Development team sole is developing new projects.
  - (ii) Supporting team sole is fixing the bug and adding more enhancement those are deposited by the customer from the maintenance project.
- Common Terminology used in Maintenance Project

#### 1. Bug/Issue.

An error occurred in the project execution is called Bug or Issue.

#### 2. Replicate.

~~Checking the bug in local system.~~

Running the project in local system, and getting bug in our local system is called Replicate.

#### 3. Resolve or Fix Bug.

Solving the bug in the program either by adding new code or by changing or by removing existing code is called Resolving or Fixing Bug.

#### 4. Debugging.

Finding in which program this bug is occurring is called Debugging.

#### 5. Patch / Service pack.

This modified program is called patch / service pack. After fixing the problem we will send this modified code (patch) to the customer then customer will apply this patch in the production system in existing project. So this bug is also resolved in clients production system.

#### 6. Ticket

A unique number i.e. used to identify An entry created by client to report an issue to the software company is

issue an update the status of this issue and customer will also track the issue status by using this entry.  
When issue is resolved, Ticket is closed.

## I. Working Domain & Platform.

The language using which you are developing project is called working platform, e.g. Java, .NET, PHP, Python, Hadoop..... are platforms.

The Business on which we are developing project is called Domain. For e.g. Banking, Health, Hotel, data analytics, PO, are domains.

Note:- Platform is also called Technical Domain.

Domain is also called Functional Domain.

## 10. Purpose of developing projects

27/07/2017

The basic need of developing a project is to establish communication between an end-user and Database for

- (i) storing
- (ii) retrieving
- (iii) updating &
- (iv) deleting data remotely.

### • Three types of logics

For implementing above three/four operations, we will develop below three types of logics, they are:-

- (i) Presentation logic.
- (ii) Business logic.
- (iii) Persistence logic.

(i) Presentation logic :- The logic that is used for developing GUI and Report for reading input from end-user and for displaying output to end-user is called presentation logic.

(ii) Business Logic:- The main logic of the project that contains validation & calculations for generating output from the given input value is called Business Logic. It is also called B logic.

(iii) Persistence Logic:- The logic that is used for storing output permanently in a persistent media such as Database and files is called persistence logic.

#### • Technologies for developing above 3 logics

For developing presentation logic and persistence logic we have ready made technologies:-

\*. The technology given for developing presentation logic are called Rendering Technologies

Ex. HTML5, CSS3, jQuery, AngularJS, JSP, ASP.NET.

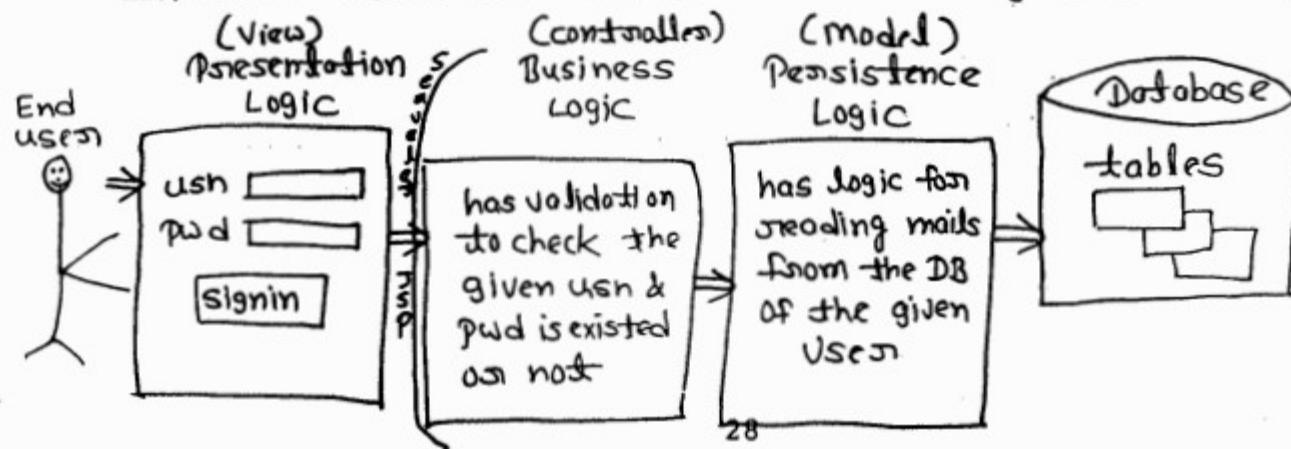
\*. The technology given for developing persistence logic are called persistence technology.

Ex JDBC, JPA, Hibernate, ibatisis, Active JDBC, ADO.NET.

We do not have predefined technologies for developing B logic, Because B logic will be changing from project to project. We must develop B logic from zero level.

For developing B logic, we must use language concepts such as Datatypes, Operators, Control flow statements, Exception handling statements.

In Core Java we will learn Developing Business Logic.



In above project if we don't use Servlet & JSP programs, if we directly access business logic from presentation logic, this project is become stand-alone application.

So, That's all about MVC architecture.

### Part #3: Java Concepts

28/07/2017

Q. What is Java Exactly?

multithreaded

- Java is a platform independent object oriented programming language and also it is a platform and Technology.

1) Java is a programming language, because it provides its own syntaxes and semantics for developing new programs.

2) Java is a Platform, because it provides its own execution environment called JVM. Java Program is not executed directly by OS, It is executed by JVM (Java Virtual Machine).

3) Java is a Technology, because it provides steady made OS pre-defined rich set of API (library) for fast and easy development.

Note:- C and C++ are only languages, they are not platform and also not technology because C++ and C has provided only Syntaxes and Semantics to develop new program, but it doesn't provide execution environment and huge set of steady made library, So it is not platform and technology.

- Java has concepts for developing desktop, web and mobile & electronic devices based applications. So Java is called Technology
- For easily maintaining and downloading, java concept are divided into three editions (division).

## Java Editions

Java has 4 Platforms (Editions) they are :-

1. Java SE - For standalone application development.
2. Java EE - For Internet and Enterprise.
3. Java ME - For Mobile Application Development.
4. Java fx - For RIA (Rich Internet Application).

1. Java SE stands for Java Platform Standard Edition.  
It provides concepts for developing desktop (standalone) CUI and GUI application, Applets, database Interaction application, Distributed Application and XML passing apps.
2. Java EE stands for Java Platform Enterprise Edition.  
It provides concepts for developing Web and Enterprise applications and Interoperable application.
3. Java ME stands for Java Platform Micro Edition.  
It provides concepts for developing mobile and electronic device level application. This edition is called Micro, because this edition programs are embedded in small chips and inserted in chip only program embedded in chip is called micro (small).
4. Java fx stands for Java Platform ~~EFFECTS~~ (~~(P)~~ FX). It provide concepts for developing rich internet applications with more graphics and animations. Java fx is an ~~extension~~ extension to swing applications of Java SE.  
Note:- Because Java fx is an extension concept to Java SE swing applications, Java fx API is included as part of Java SE software. Just by installing Java SE, we will also get Java fx API.

We are all here for learning Java SE & Java EE.  
Because we are here for learning Computer Based Applications.

## Small History about Java Editions

1. In the begining version of Java 1.0, we do not have editions because in initial version java doesn't support server side web programming and enterprise applications development. It will support only stand alone applications and applets (client side web programming) development. In begining versions even it is not supported database interaction applications.
2. In the next version, in 1.1, DIA support development is supported.
3. From Java 1.2 version onwards server side programming support is added By using the technology Servlet and JSP.
4. Since Java is supporting all three flavours of application development (SA, Web, mobile) ~~SUN~~ people decided to divide java concepts into three parts for easy maintaining and downloading.
5. From Java 1.2 onwards, Java has Three Editions with the names
  - (i) J2SE (Java Platform ~~2~~ Standard Edition)
  - (ii) J2EE (Java Platform 2 Enterprise Edition)
  - (iii) J2ME (Java Platform 2 Micro Edition)
6. In the above edition names the number 2 doesn't have any serious significance. It is included in the edition name only to inform programmers this java editions core concepts are given in Java 1.2 version.
7. In the next versions 1.3, 1.4, 1.5 this number 2 is confusing to the java beginners because it is not matching with version numbers.
8. Then to avoid this confusion from 1.5 version onwards this number 2 is removed from edition name and renamed as:-
  - (i) Java SE (ii) Java EE (iii) Java ME

g. In Java 1.4 version, we have new edition is given called Java fx, for developing RIA application.

## Complete Java

According to SUN Microsystems, Complete Java means :-

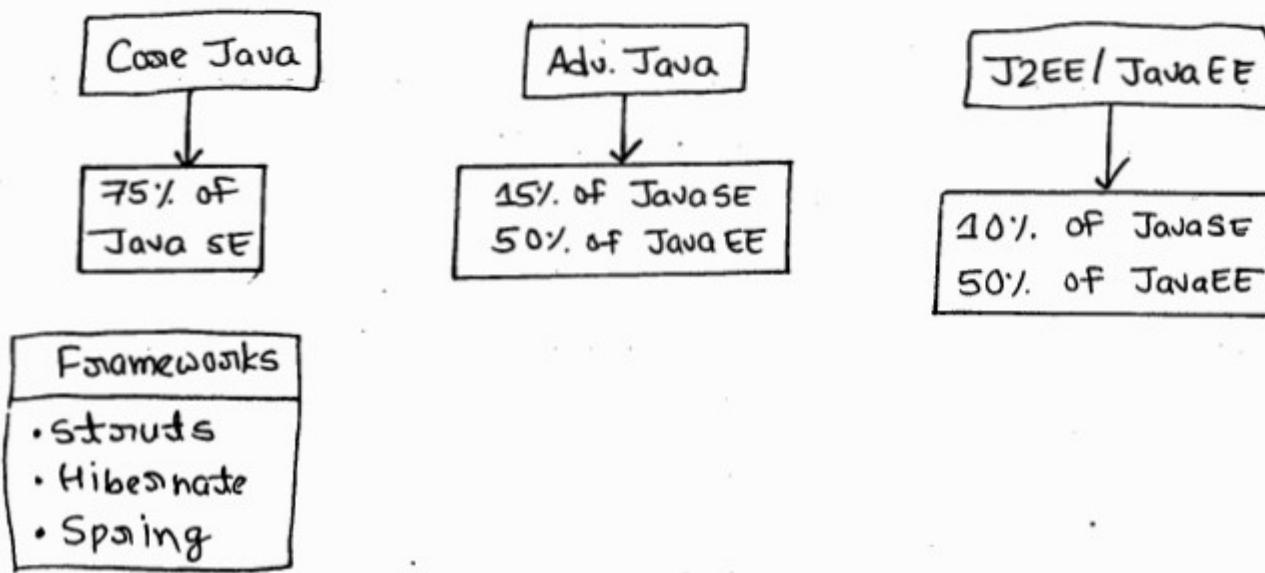
1. J2SE / Java SE / JSE
2. J2EE / Java EE / JEE
3. J2ME / Java ME / JME

We are here to learn Java SE and Java EE.

According to Ameespet, Complete Java (For easy learning)  
We divided Java SE and Java EE into below courses :-

1. Core Java
2. Advance Java
3. J2EE / Java EE
4. Frameworks
5. XML, Webservices
6. Java Tools
7. Design Patterns

### Ameespet Complete Java Concepts Division



## Java Technologies

- \* Java is a programming language, platform and Technology.
- \* Programming language concepts syntaxes and semantics is common to all types of Java applications development.
- \* So Java SE has programming language concept, Platform and its own technologies.
- \* Whereas Java EE has only its own platform and its own technologies, Java EE doesn't provide programming concept (syntaxes and semantics).
- \* So to develop and compile Java EE programs, we must install and use Java EE SE Software (JDK).
- \* Java Programming language concepts are datatypes, operators, control flow statements, Exception Handling statements, class, variables, methods, constructions, blocks, objects, arrays, strings, multithreading collections, AWT etc. concepts ~~nothing~~ comes under Java Programming concepts.
- \* Java SE and EE has their own technologies for developing different types of Applications.

| Java SE               |   | Java EE         |
|-----------------------|---|-----------------|
| 1. JDBC               | ✓ | 1. Servlet      |
| 2. JNDI               |   | 2. JSP          |
| 3. Connection Pooling | ✓ | 3. Web Services |
| 4. XML                |   |                 |
| 5. RMI                | X | 4. EJB          |
|                       |   | 5. JSF          |

Note: RMI, EJB and JSF no need to learn. These technologies we have alternate concepts, Frameworks, Spring and Hibernate.

## Java Platforms

- For Compiling and executing Java applications every edition has their own execution Platform.
  - (i) Java SE platform is JRE.
  - (ii) Java EE Platform is Server Software (Tomcat, Weblogic, etc....).
  - (iii) Java ME platform is MDK.
- For Developing Stand alone application and compiling, executing, we must install JDK with JRE software.
- For Developing, Compiling and executing Java EE web and Enterprise applications, we must install server software with JDK. JDK for compiling and server software for executing.
- For Developing, compiling and executing Java ME applications we must install MDK software.

Note: Now a days for developing Mobile applications we are not using Java ME rather we are using Android with Java SE (Java language). From May 2017 onwards for developing Android based mobile application the official language from google is Kotlin.

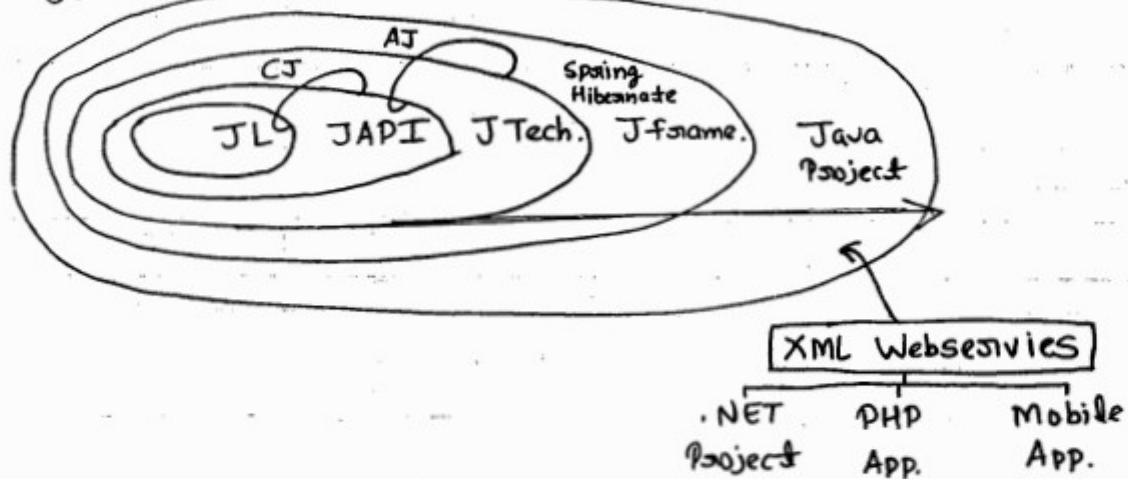
| Java Editions | Platform (Syntax & Semantics) | Platform (Execution Environment) | Technologies (Predefined API) |
|---------------|-------------------------------|----------------------------------|-------------------------------|
| 1. Java SE    | ✓                             | ✓ (JDK)                          | ✓ (JDBC)                      |
| 2. Java EE    | ✗                             | ✓ (Server)                       | ✓ (Servlet, JSP, ...)         |
| 3. Java ME    | ✗                             | ✓ (MDK)                          | ✓                             |

From the above points we can conclude Java EE and Java ME are developed on top of Java SE, so for using Java EE and Java ME we must also install Java SE software (JDK).

Java EE and Java ME

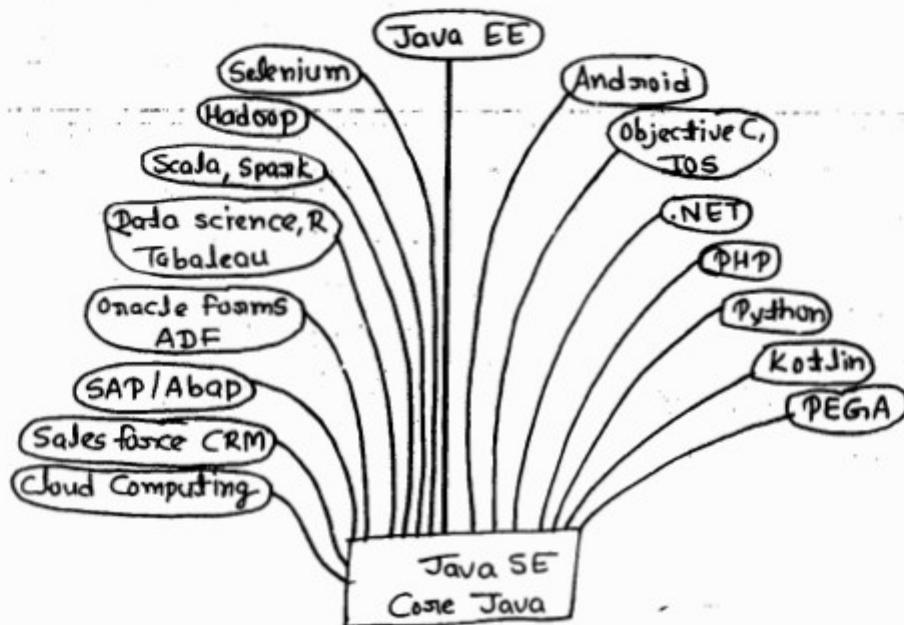
## Java Concepts Utilization order in Project Development

- Every project is developed using Java language (JL) concepts, Java API, its technologies and framework for fast developments.
- All the Java concepts are used in the below dependency order



## Java SE Tree Diagram

- Now a days Java SE become most technology or concept for developing any new technology based software.
- If you are strong in Java SE concepts, we can easily shift from one technology based projects to another technology based projects because all technologies are similar to Java.
- Below is Java SE tree diagram, that shows different technologies developed on top of Java SE.



## Summary

As part of Java core Concepts, we learn

### Complete Java Overview

- (1.) What exactly Java is?
- (2.) Java Version and editions history?
- (3.) What is JavaSE, JavaEE, JavaME, and JavaFX?
- (4.) SUN complete Java
- (5.) Ameespet Complete Java
- (6.) Difference between:-
  - CoreJava & JavaSE, JavaSE & JDK.
  - Adv. Java & JavaEE, JavaEE & Seaver Software.
- (7.) Java Platforms and Java Technologies.
- (8.) Java Concepts utilization order in projects development.
- (9.) Java SE Tree diagram and Java Learning Road Map.
- (10.) Java Project Architecture with all java concepts.

## Assignment

List out all differences between Java SE and Java EE as a table

### Java SE

- (i) Java SE stands for Java Platform standard Edition.
- (ii) It is a specification provides rules and guidelines for developing stand alone application and applets and also provides rules and guidelines for developing JDK software for compiling and executing stand alone applications and applets.

### Java EE

Java EE stands for Java Platform Enterprise Edition.

It is also a specification, it provides rules and guidelines for developing web and enterprise applications and also provides rules and guidelines for developing seaver software for executing web and enterprise applications.

- |  |  |
|--|--|
| (iii) It is a base edition for all java editions.                                      | (iii) It is developed on top of Java SE.   |
| (iv) JDK is sufficient for Compiling and executing Java SE based applications.         | (iv) We must have Seavse and Jvn softwares for compiling and executing Java EE based applications.   |
| (v) Java SE provides programming language concepts, its own platform and technologies. | (v) Java EE provides only its own platforms and technologies it doesn't provide programming language concepts.   |
| (vi) Java SE is implemented as JDK software by SUN.                                    | (vi) Java EE is implemented as seavse software by SUN and also by many other companies like apache, IBM, BEA, JBoss etc.   |
| (vii) Using Java SE, we will develop logic for a business operation.                   | (vii) Using Java EE we will not develop any logic, rather we will provide an environment to execute Java SE development logic through internet as web application. |

That's all about

Java  
Concepts



31/07/17

## Part #4

### Introduction to Core Java Concepts

As part of core Java, We will learn developing logical and object oriented programs as part of business operations using Java.

As Beginning , We will learn below Topics

1. Common Terminology used in all languages.
2. Platform, Platform dependency, independency
3. Why C, C++ language are platform dependent ?
4. How Java achieved platform independency ?
5. Important facts on Java Program and Java software.
6. Types of Java software
7. Difference between JDK, JRE, JVM, JIT.
8. Installing JDK, its folder structure
9. Setting JAVA\_HOME, path and class path
10. Difference between path, classpath

Part  
#05

### 1: Common Terminology used in all languages.

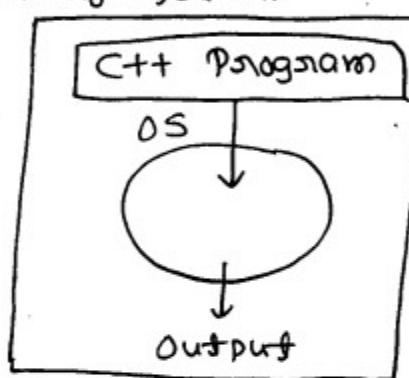
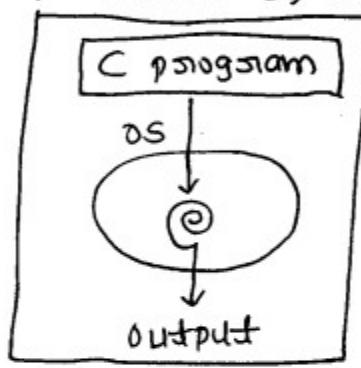
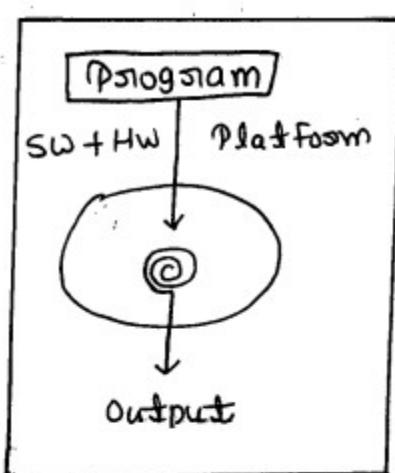
- |                     |     |         |         |           |
|---------------------|-----|---------|---------|-----------|
| 1.) Source code     | DWC | Abc.C   | Abc.CPP | Abc.java  |
| 2.) Compiled code   | CGC | Abc.obj | Abc.obj | Abc.class |
| 3.) Executable code | CRE | Abc.exe | Abc.exe |           |
|                     |     | ↓       | ↓       | ↓         |
|                     |     | O/P     | O/P     | O/P       |
- 4.) Compilation      SC  $\Rightarrow$  CC
- 5.) Execution      Run Compiled Code (CC)  $\Rightarrow$  O/P
- 6.) Compiler      TS converts Source Code  $\Rightarrow$  CC
- 7.) Interpreter      TS converts CC  $\Rightarrow$  Machine Language (ML) in execution.
- 8.) Compiled PL      Does not use interpreter
- 9.) Interpreted PL      Use both Compiler and interpreter.

Refer page No. 2 in Volume - 1A, for more detail

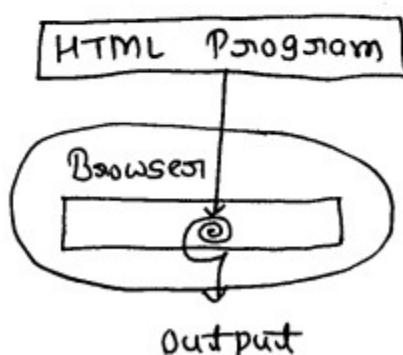
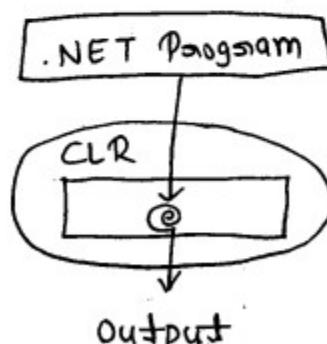
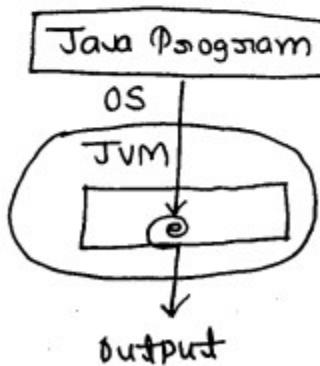
## 2. Platform, Platform Dependency, Platform Independence.

### • Platform

- A platform is an environment, In which a program is loaded & executed.
- A computer platform is OS + HW devices. A platform can be software only or Hardware only or both.
- For C and C++ programs, platform is OS. It means C and C++ programs are directly executed by Operating System.

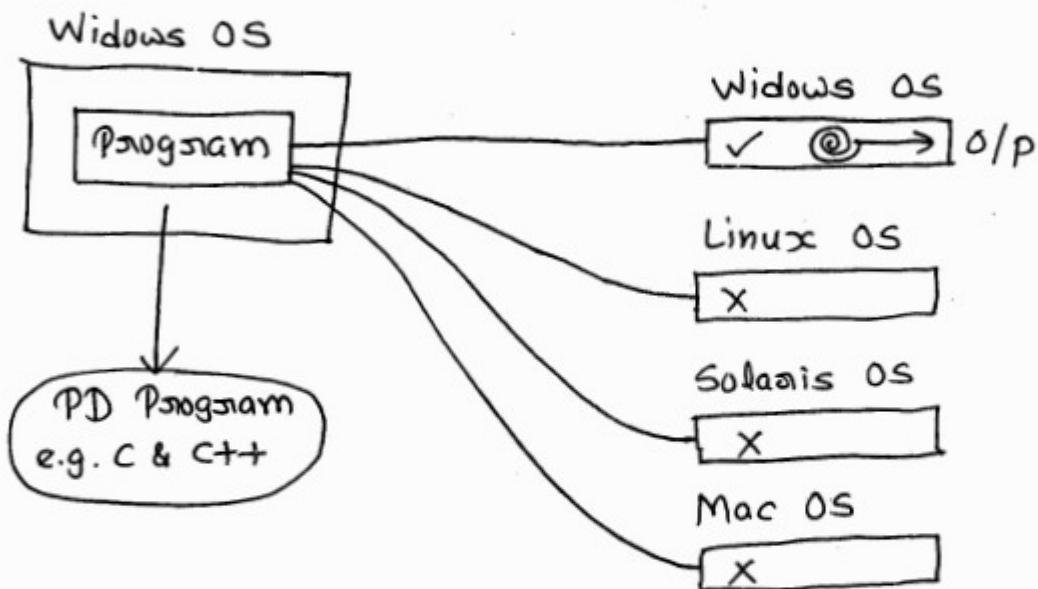


- Java program platform is JVM and .NET program platform is CLR and HTML program platform is Browser. It means these three programs are not directly executed by OS.



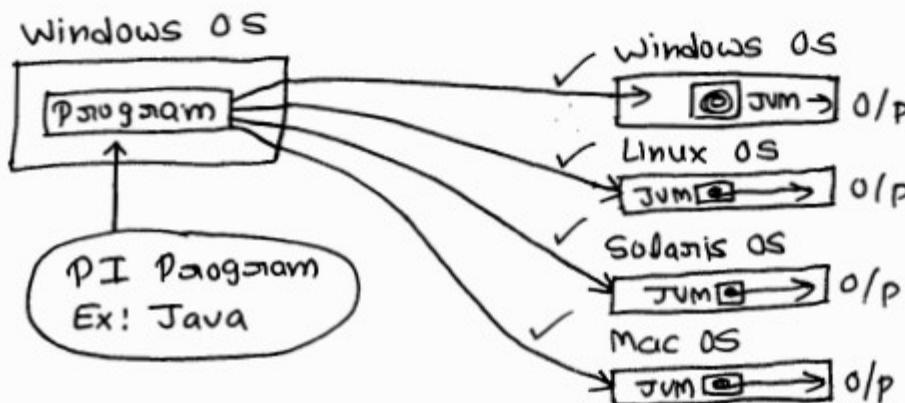
- Platform Dependency.

A program that we developed and compiled in one OS, If it is not executed in different OS, we call this program as platform dependent program. and the language using which we will develop this program is called Platform Dependent Programming language. For ex. C and C++ programs are platform dependent program.



- Platform Independence

If a language's program is executed in all different varieties of OS, irrespective of OS in which it was compiled, we call this program as Platform Independent program and language using which this program is developed is called PI language. e.g. Java is PI.



Note: If any language program is directly executed by OS then it is called PD.

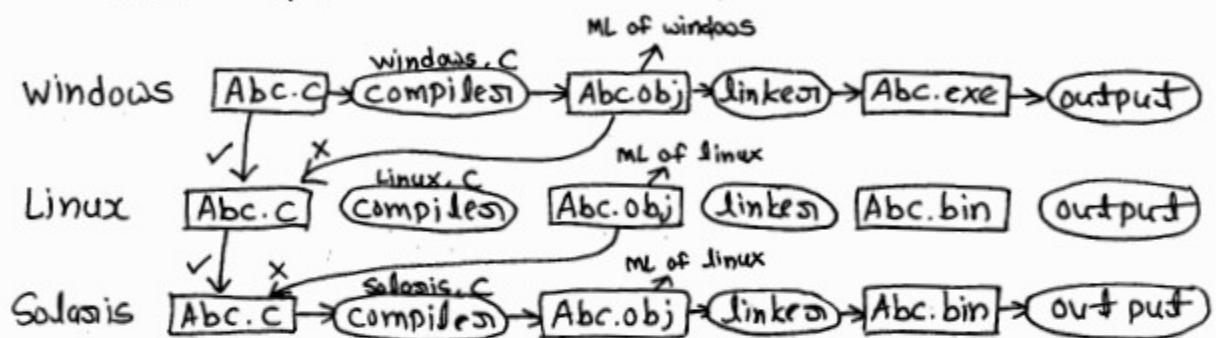
If any language program is not directly executed by OS, rather if it is executed by a mediation layer software like JVM, this program is Platform Independent.

01/08/17

### Q. Why C, C++ languages are platform dependent?

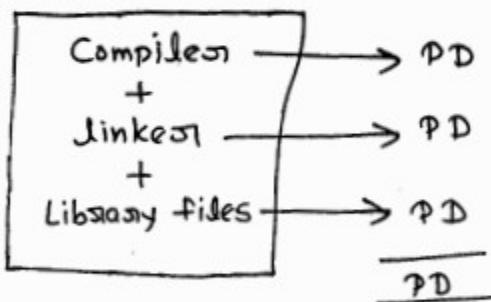
Ans:- In C & C++ languages program is compiled targeting to one OS. This compiled code contains machine language whose format is understandable to only current OS. ~~since~~ This machine language format is not understandable to different OS. C & C++ programs are not executed in different OS. They become platform dependent.

Below diagram will show you C, C++ program platform dependency :-



Not only C program, C software is also platform Dependent. Because all its components (compiler, linker, library files) are platform dependent.

### C Software

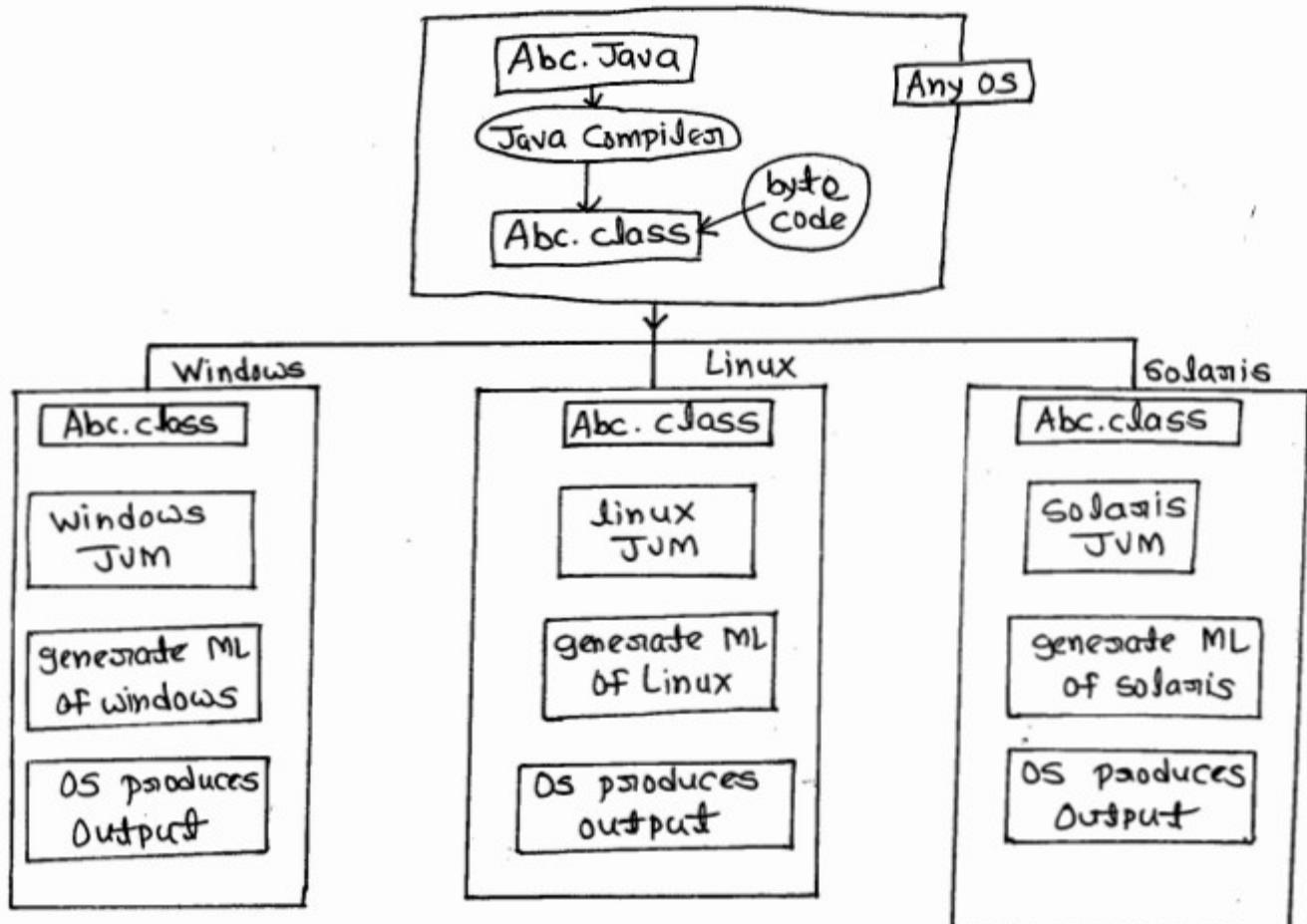


So we will have separate C software for every OS. Then we must download and install in our computer based on our system OS.

#### 4. How Java achieve Platform Independence?

For achieving platform independence java program is not compiled targeting to current OS, rather java program is compiled into a code which is common to all OS, later it is translated into Machine language of client OS in which Java program is running. So Java achieved platform independence by generating machine language for client OS by moving machine language generation from compilation phase to execution phase by introducing new compiler software, bytecode and JVM.

Below diagram show you Java's Platform Independence



Summ

- Java achieved platform independence by moving machine language generation from compilation phase to execution phase by introducing new compiler software bytecode and JVM.
- Bytecode is an instruction set to the JVM about java program code. Bytecode is an intermediate language,

It is a native language of JVM.

- JVM is a java platform, it is responsible to run java bytecode by converting it into current OS machine language.
- SUN Developed JVM for every OS to run the same compiled code in all popular OS.
- Note:- .NET program also platform independent, .NET programs also compiled into intermediate code to all OS, for others it is translated into current OS machine language by CLR.

Here .NET problem is, Sun Microsoft gave CLR only for Windows OS. Because CLR is not available for Linux, Solaris and Mac OS. .NET program can not be executed in these OS.

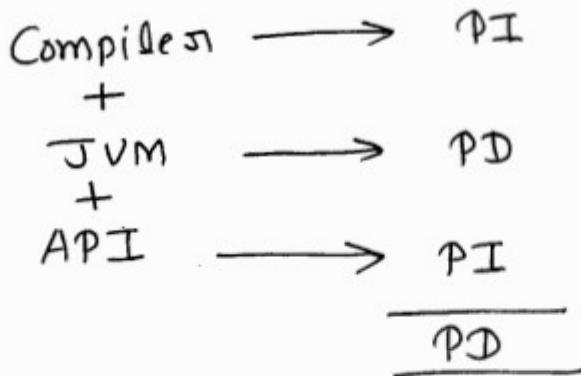
### Java Program Development, Compilation and execution procedure

- In the java programming language, all source code is first written in plain text files ending with the ".java" extension. Those source files are then compiled into ".class" files by the javac compiler. A ".class" file does not contain code that is native to your processor; it instead contains bytecodes - the native language of the Java Virtual Machine. The Java launcher tool then runs your application with an instance of Java Virtual Machine.
- Below diagram shows Java program compilation and execution :-



## 5. Important facts on Java Program & Java Software.

- (i) Java Slogan: WORA - Write Once Run Anywhere.
- (ii) Compiler is responsible to take java source code, convert it into bytecode and saving this bytecode in a separate new file with ".class" extension with class name as file name.
- (iii) JVM is responsible to take the bytecode, execute this bytecode by converting it into current OS ML. After execution completed generated ML is destroyed.
- (iv) Java supports only 2 extension files .java and .class. In java we do not have a file with machine language code. Java does not support .exe file, because of security reason point of view and over always compiled code is downloaded from servers, ML code saved in client system is useless.
- (v.) Java compiler is os independent because both input and output files source code and byte code are java related files.
- (vi.) JVM is os dependent, because it should generate os dependent ML from the given bytecode.
- (vii.) Java Software contains



- (viii) Java software is platform dependent because JVM is platform dependent. Because java software is platform dependent, we will separate java software for every os. Hence based on our system OS and processor we must download and install java software.

(ix) Java program is platform independent and java software is platform dependent, both because of JVM.

⇒ Java program platform independent, because JVM is available for every OS, so java program can be executed in all OS.

⇒ Java Software is platform dependent, because we have separate ~~JVM~~ JVM for every OS, so the same java software can not be installed on all OS.

(x) For all OS, JVM is developed by SUN(Oracle), but not by OS vendors.. So, JVM will not be installed inbuilt along with OS installation. We must install JVM explicitly in our computer after OS installation.

(xi) Programmer is responsible for developing, ~~compiling~~ compiling and executing a java program, whereas

(xii) Customer/ client is responsible for only executing java program.

(xiii) Based on programmer and customer responsibilities Java software is divided into two types.

### 6. Types of Java Software

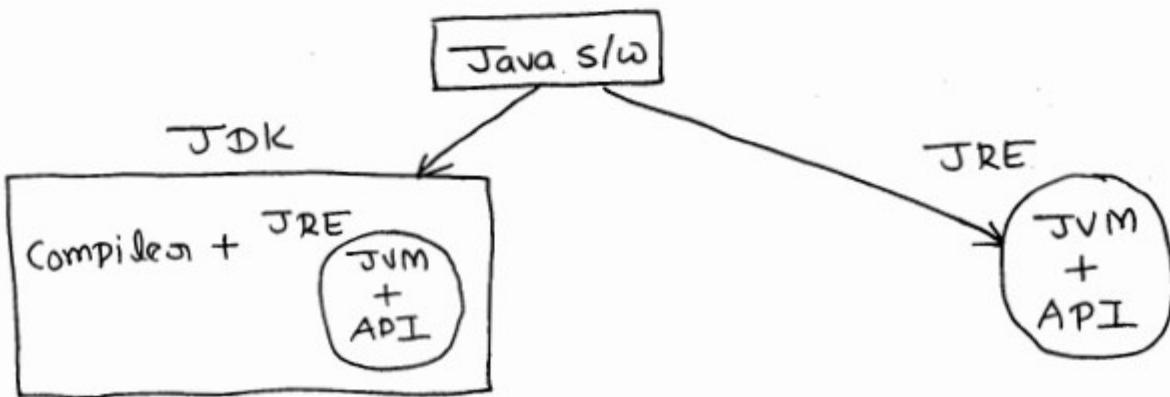
#### Java Programming Concepts

Part #05

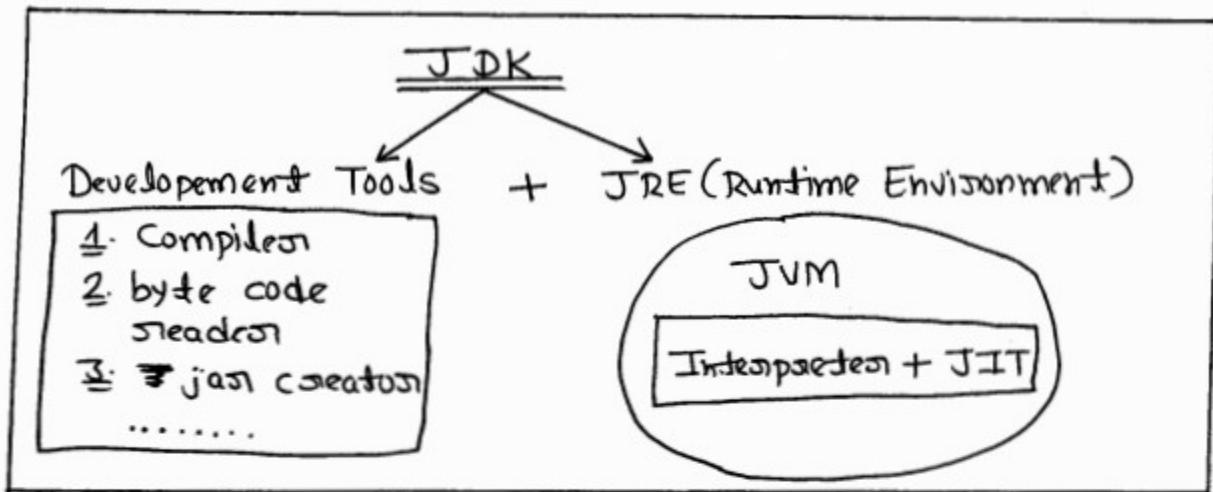
Java software is divided into two principle products

(i) JDK (Java Development Kit)

(ii) JRE (Java Runtime Environment)



### 3. Difference between JDK, JRE, JVM and JIT.



Collect Notes for this section from [Youtube.com/Nageshit](https://youtube.com/Nageshit)  
Java by harshikrishna /  
playlist

1. Place :- JIT is part of JVM, JVM is a part of JRE and JRE is a part of JDK.

2. JDK :-

- JDK stands for "Java Development Kit".
- It provides development tools and Execution Environment. Development tools means small-small software programs.
- JDK contains Development tool compiler and run time environment ~~JRE~~ JRE.
- So using JDK we can develop, compile and execute new applications and also we can modify existing applications.

### 3. JRE :-

- JRE stands for "Java Runtime Environment."
- JRE provides only Runtime Environment, it does not provide any development tools.
- So JRE contains only JVM.
- Hence Using JRE, we can only execute already developed applications, we can not develop new applications. and also we can not modify the existing applications.

### 4. JVM :-

- JVM stands for "Java Virtual Machine".
- It is a Java platform, because it is responsible to execute our Java byte code.
- JVM provides Interpreter + JIT using which we can run Java byte code by converting them into current OS machine language.

### 5. JIT :-

- JIT stands for "Just In Time" compiler.
- It is responsible to help Interpreter for executing Java byte code fast.
- Basically JIT will improve the performance of Java Program execution.

03/08/17

8. Java Software installation (JDK) and its folder structure.

⇒ Refer page no. 15 to 20.

After installing the software

(i.) binary - command - Running the current software program.

(ii.) library - Program files - develop program.

**Abc.java**

↳ Java C → compiler.

↳ Java → JVM.

05/08/17

9. Setting path and class path

⇒ Refer page no. 21 to 27 in volume - 1A

Installing JDK, setting above 3 environment variables in Linux, Solaris,

Refer page no. 26, 27.

| Windows OS  | Linux, Solaris, Mac OS |
|-------------|------------------------|
| set path    | export path            |
| ;           | :                      |
| %path%      | & path                 |
| echo %path% | echo & path            |

Windows OS

set path = C:\JDK 1.8.0-144\bin;%path%.

Linux/Solaris/Mac OS

export path = /home/Java/JDK 1.8.0-144/bin:&path

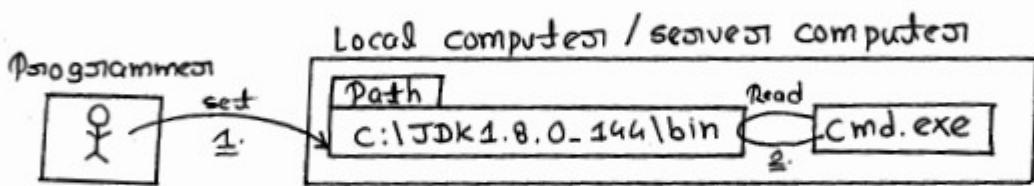
Two Rules:-

1. In class-path variable one must place .; at beginning at its value section. Otherwise class will not be executed.

2. In path variable value we must always place before oracle bin folder path. because oracle software also contains Java software, which is version then our JDK install software. If we placed oracle software path before JDK software path. programs are compiled but they are not executed.

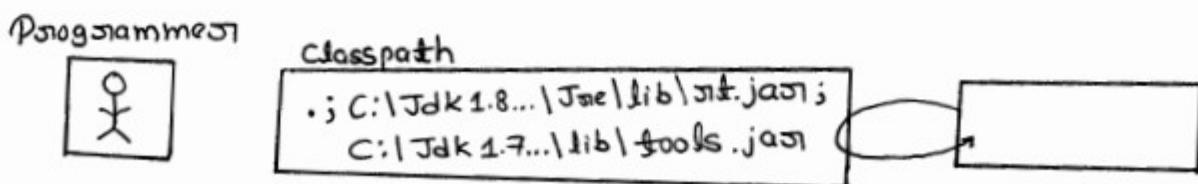
Q. Who will set path and uses path? What is path?

- Path is an environment variable, it is used for binding binary files of a software.
- Path is a mediator between programmers and command prompt window. Programmers will set path variable by storing binary file path (bin folder path), command prompt window will need path variable to find and execute binary files of a software.



Q. What is classpath, who will set & uses of classpath?

- \* Classpath is also an environment variable, it is used for finding library files (.class file) of a software.
- \* Classpath is a mediator between programmers and compiler. JVM programmers will set classpath variable by storing the software library file path. Compiler and JVM will need classpath variable while compile and executing our class for finding library files those are used from our class.



Q. What should we do if get Error: '-----' is not recognized?

Ans: We must set path variable by storing their files folder path.

Q. What should we do if we got an Error "Could not find symbol class" OR

Java.lang.ClassNotFoundException OR Could not find or load main class?

Ans:- Set classpath by storing this class/folder on Jar file path and also should store .;

Q. What should we do if we want access one folder classes from another folder classes.

Ans:- We must set classpath, by storing this folder path.

Q. Why we don't set path variable for running a game software?

Ans:- Because, we don't run game software from command prompt window, we will run game software from its own window by double clicking shortcut icon on desktop.

We must set path variable for a software only if we want to run this software from command prompt window.

10. Difference between path and classpath.

#### Path

1. used for locating binary files of a software.
2. Stores bin folder path.
3. It is used for CMD shell for finding binary files from different software configured in path variable.

#### classpath

1. used for locating library files of a software.
2. Stores lib/jar file path.
3. It is used for compiler and JVM for finding library files of different software configured in classpath variable.

4. It is meant for all s/w installed in our computer including OS for finding binary files.

5. '---' is not recognized error, we should set path.

6. .; (dot semicolon) is not required.

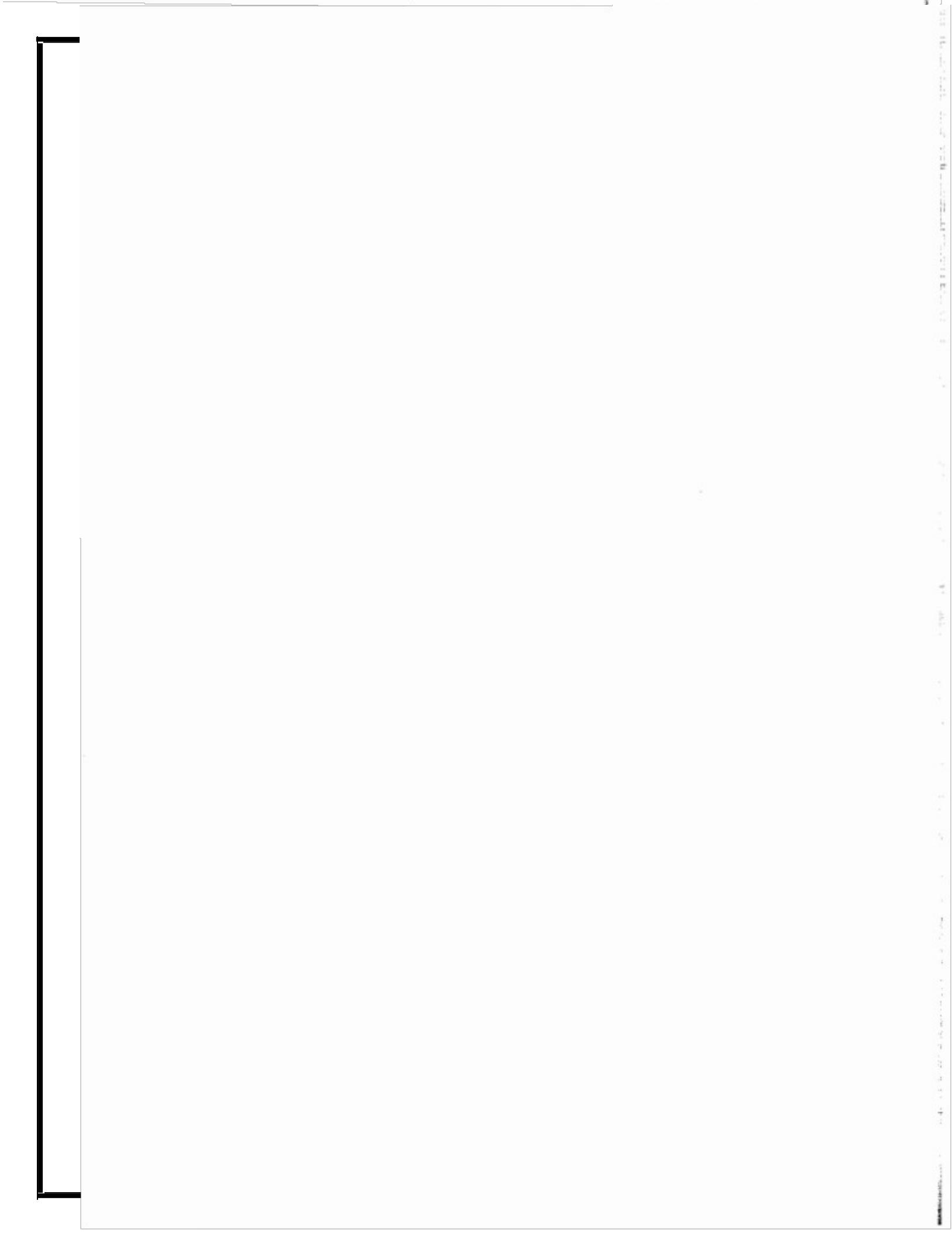
4. It is meant for only Java and its related software, but not for all software.

5. Can not find symbol or Could Not Found Exception error or could not load main class error, we must set classpath.

6. .; (dot semicolon) is ~~not~~ mandatory.

\* Software required for development, compilation and execution of a java program.

| Software                   | Purpose                     |
|----------------------------|-----------------------------|
| <u>1.</u> Edition Software | development (type and save) |
| <u>2.</u> JDK Software     | Compiler and JVM Software   |
| <u>3.</u> CMD Software     | Compilation and execution   |



\* Java programming Concepts.

- SWs required to develop Java program.
- Basic java programming elements.
- Essential statements of java program.
- Steps to develop java program.
- Hello, WORLD! App development
- Compilation and execution procedure.
- Compile-time and Run-time Errors.
- Some more examples.
- Interview and OCJP Questions.
  - > Main method interview FAQs
  - > New change added on Java 7 w.r.t main method
  - > Can we execute a class without main method.
  - > 60+ interview questions on source file, and class.
  - > Diff b/w S.O. print(), println() and printf() methods.
  - > Java coding standards and naming conventions.
  - > Java platform architecture
  - > Why java doesn't support pointers.

> To develop, compile and execute a java program we need three s/w to be install in our system.

(1) Editor S/W

(2) JDK S/W

(3) Command Prompt S/W

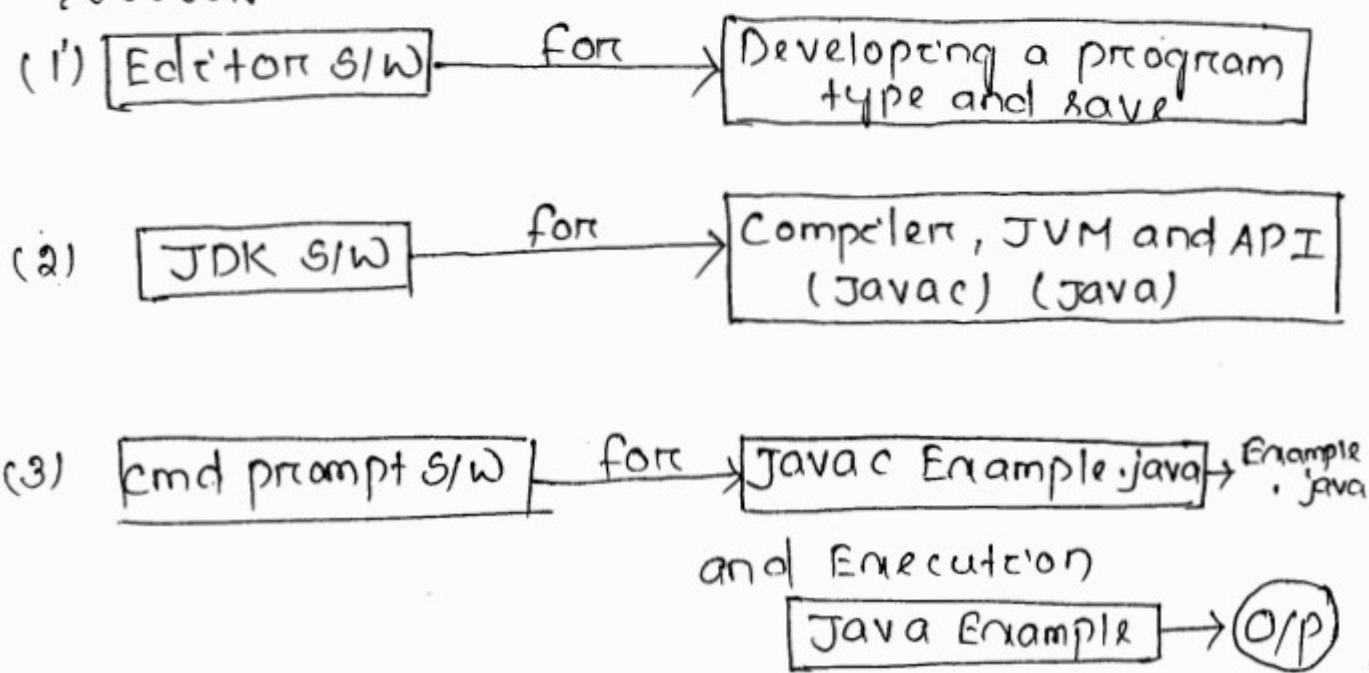
\* We need editor S/W for developing java programs. We can use any editor S/W like notepad, notepad++, Edtplus, IDE S/W, eclipse, my eclipse, Netbeans, IntelliJ, jbuilder etc.

\* JDK is used for getting compilers and JVM S/W into our system to compile and execute java programs.

NOTE:- Just by installing JDK we can't compile and execute java programs, we must run javac and java commands from cmd prompt window.

\* cmd prompt window is required for compiling and executing java programs by running 'javac' and 'java' commands.

### Summary



**NOTE:-** In built in windows O/P. We will get editor s/w Notepad, cmd window. We need to install JDK expressively and should setpath.

## Basic java programming elements

Java supports 10 basic programming elements.

(1) Module (From java-9 onwards)

(2) Package

(3) Class

(4) Interface

(5) enum ← (from java 1.5 onwards)

(6) Annotation ←

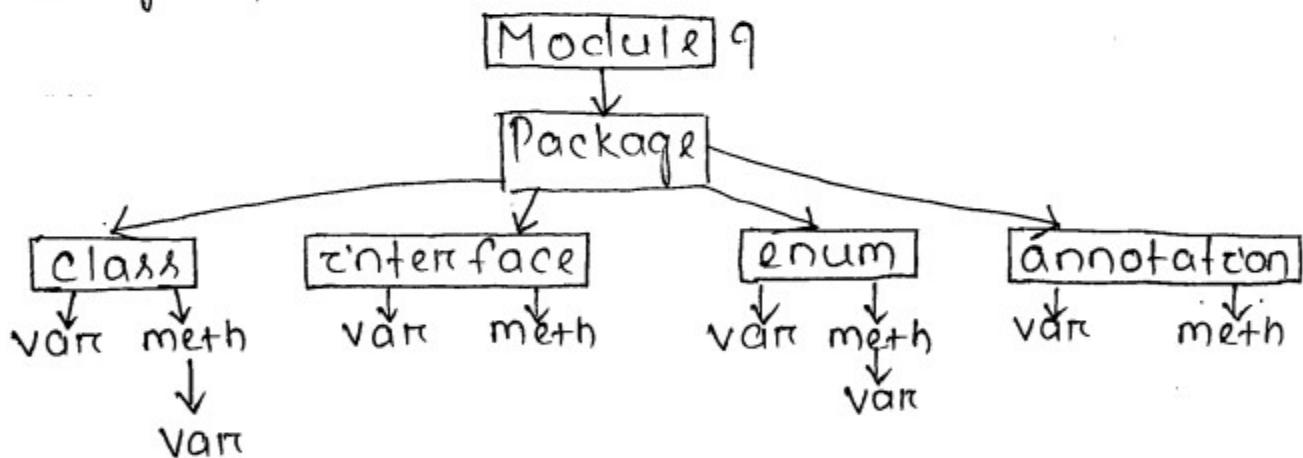
(7) Variable

(8) Method

(9) Constructor

(10) block

All above programming element are organized as group as shown below.



**NOTE:-** Constructor and block are allowed only inside a class and enum.

(1) Java supports all above 10 programming elements for implementing OO programming concepts by using Java language

(2) According to OO programming we must develop all programs around objects.

(3) An object is a real world thing that you can see and use.

Ex:- Person, Animal, Vehicle, mobile, Bank A/c etc are objects.

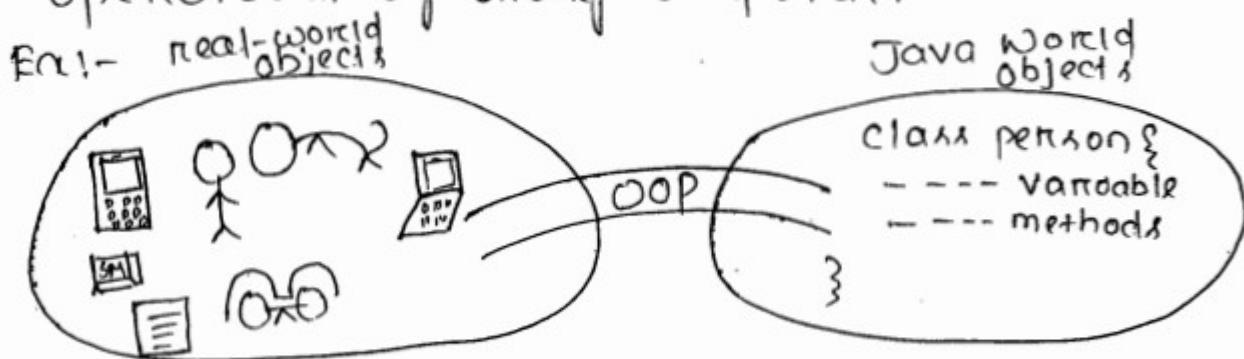
(4) Developing a program around an object means we must create a program.

- > to represent a particular object (Ex. person)
- > by storing that object values (Ex. name, height etc)
- > by implementing its operation (Ex. sleep, eat)  
by using a class, variable and methods, respectively.

(5) → Class is meant for representing object

→ Variable is meant for representing value  
→ Method is meant for representing operation.

(6) Hence by using Java language above programming elements we must bring real-world objects into programming world to perform their operations by using computer.



(7) As per above diagram we can conclude that;

- > We must create multiple classes as many different type of objects as we want to create.
- > In each class we must create multiple variable based on the multiple values of the object we want to store.
- > Also in each class we must create multiple methods based on the multiple operations that object performs.

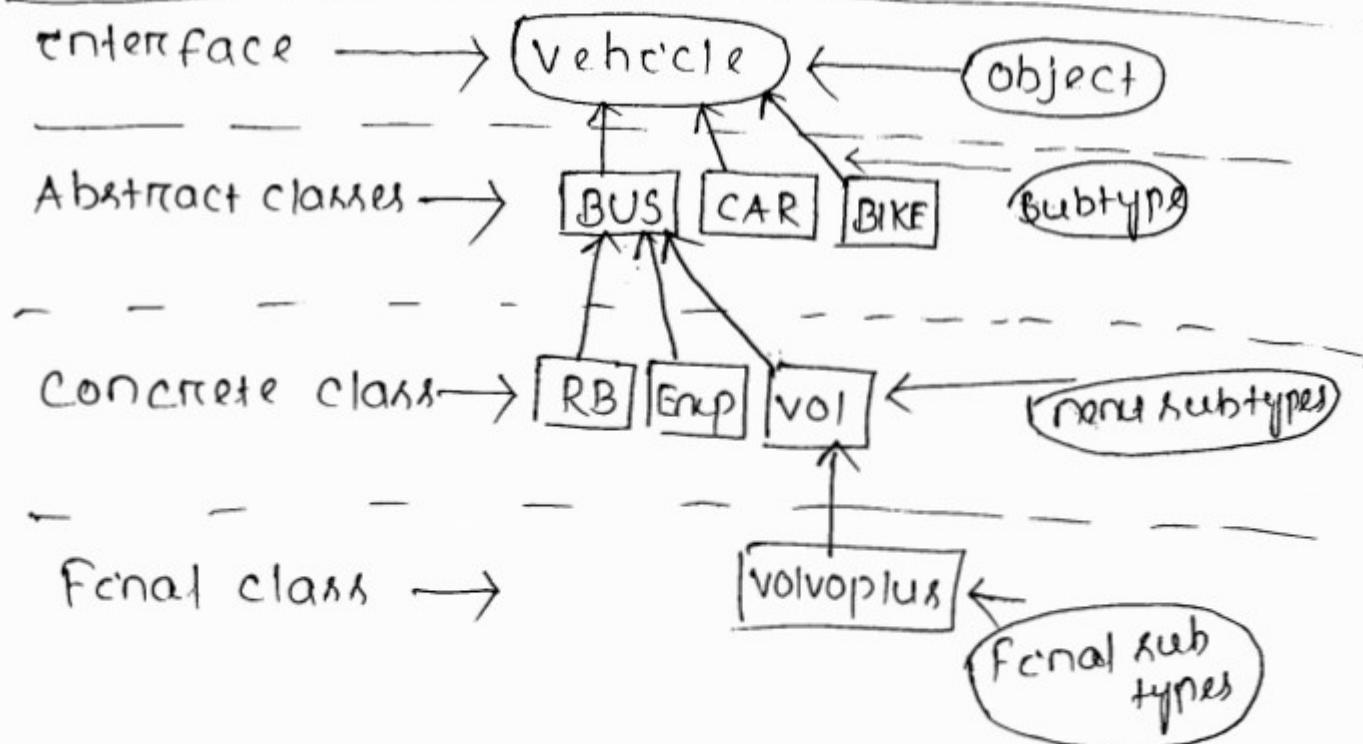
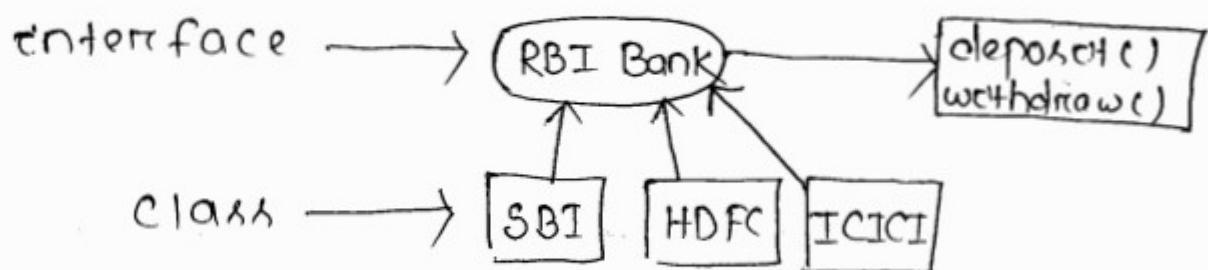
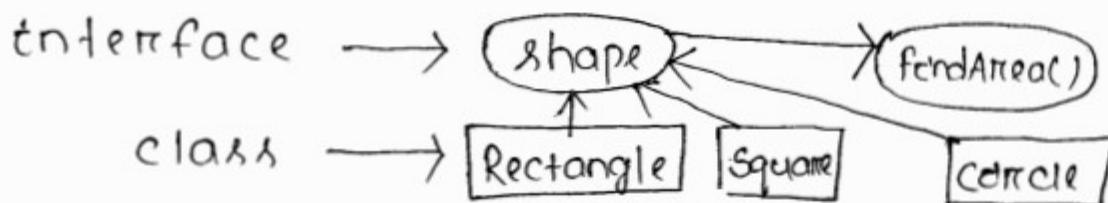
For example as per above diagram we must create 7 classes with the required number of variables and classes.

Below table will explain all programming elements and their purpose.

| <u>Java Programming Element</u> | <u>Purpose</u>  |
|---------------------------------|---|
| (1) Module                      | → Used for / meant for grouping related packages and their classes.   |
| (2) Package                     | → Meant for grouping classes and interfaces.  |
| (3) class                       | → Meant for representing real world object<br>> Interface → Meant for only declaring object operation<br>> abstract class → Meant for declaring some operations and implementing some others.<br>> concrete class → Meant for implementing all operations.<br>> final class → Meant for implementing all operations and to stop creating sub classes<br>> enum → Meant for declaring named constants.<br>> annotation → Meant for providing description and configurational values. |
| (4) variables                   | → Meant for storing values of an object and its operation values  |
| (5) method                      | → Meant for developing logic for implementing operation of an object.   |

## Java programming element

- (6) Constructor → Meant for initializing object
- (7) Block → Meant for representing inner object of an outer object
- (8) Inner class → Meant for representing inner object of an outer object



DT- 16.11.2017

## Essential statement of java program

→ To develop, compile, execute and display some message on console we must use below three statements they are;

- (1) Class
- (2) main
- (3) System.out.println();

\* Class is essential because java is an object oriented programming language, every program must start with a class and moreover only class allows us to define method with logic.

\* Main method is essential because java program execution starts with main method. JVM can execute only main method from a class.

**NOTE!-** We can compile a class without main method, but we can't execute.

\* System is essential because in java we can print any message or data on console by using System statement.

## Steps to develop java program

**Program1!-** Develop a program to display Hello,World! on console.

- (1) Open Notepad
- (2) Type below program

```
Class program1 {  
    Public static void main(String []args) {  
        System.out.println("Hello, World!");  
    }  
}
```

(3) Save above program with the name program1.java.

in D:\01 corejava\01javaBasic

(4) Compilation and Execution:-

> Open cmd prompt

> change drive (G:)

> change directory (cd 01 corejava\01javaBasic)

> compile (D:\01CC\01JB> javac program1.java)  
→ program1.class

> Execute :- java program1  
→ O/P:- Hello, World

\* Develop a program to display your name.

program2.java

```
class program2{  
    public static void main (String [ ] args){  
        System.out.println ("Sudheswar Parida");  
    }  
}
```

compile :- > javac program2.java  
1- program2.class

execution :- > java program2.class  
1- Sudheswar Parida

program3.java

```
class program3{  
    public static void main (String [ ] args){  
        System.out.println (" * * * * * * * * * ");  
        System.out.println (" *Name Kridhna * ");  
        System.out.println (" * * * * * * * * * ");  
    }  
}
```

\* Develop a program to display name Haré krishna using single System statement, but o/p should come as: Haré  
krishna

program4.java

```
class program4{  
    public static void main(String[] args){  
        System.out.println("Hare\nkrishna");  
    }  
}
```

\n:- Newline char, it will bring next characters to next line

Main Method

DT-17.11.17

(1) What is main method?

Ans:- main method is a user defined method with predefined prototype syntax.

(2) What is the use of main method?

Ans:- main method is used for starting the execution of an application as stand alone application by using the command java.

> main method is the critical point of a class execution. It means JVM will start class logic execution with main method. If class doesn't have main method it won't be executed. JVM will throw error.

NOTE!- main method is mandatory for execution only. for just compiling a class main method isn't required.

(3) What is the syntax of main method?

Ans:- public static void main (String[] args)

(4) Can we change the parts of main method?

Ans:- except args we aren't allowed to change or remove the main method other parts. We will get either CE or RE.

- > In place of args we can place any name.  
Ex:- harry
- > We can use [] either after String, before args and after args, but not before String.
- > We can use... in place of [] from java 1.5v onwards.
- > We can use public and static in any order but must be used before void.

(5) Why main method has public?

Ans:- to be accessed by JVM software from outside of our project directory main method must has public keyword.

(6) Why main method has static?

Ans:- To access and execute main method logic without creating object , main method must have static keyword.

(7) Why main method has void?

Why main method return type is void?

Ans:- Returning value from the main method is useless because the value returning from main method will be return to JVM, JVM won't do any work with this returned value hence to stop returning value from main method main method return type must be void

(8) Why main method name is main?

Ans:- According to coding standards main method name is chosen as main because of the central point of class execution. Any method run generally called as main

> And moreover the name `main` is conventional name derived from C and C++, because Java is descendant of C and C++ language.

(9) Why `main` method has parameters?

Ans:- `Main` method has parameters to receive input values from end-user at program execution time from command line.

\* WAP to read and display values from command line from end user?

`Test.java`

```
class Test {
```

```
    public static void main (String [ ] args) {  
        System.out.println (args [0]); "ab"  
        System.out.println (args [1]); "cb"  
    }  
}
```

cmd

```
> javac Test.java
```

```
> java Test ab cb
```

args[0], args[1]

(10) Why `main` method parameter is Array?

Ans:- To allow end-user to pass multiple value.

(11) Why `main` method parameter type is `String [ ]`?

Ans:- Any value sending from keyboard will passed into a program as `String` type by OS, and moreover any values can be represented in terms of `String` and further it can be converted to its original type. Hence SUN chosen `main` method parameter type as `String [ ]`.

(12) Why main method parameter name is args?

Ans:- The input values sending to a method parameter is technically called as argument, so it is the short-cut name of argument.

\* Develop a program to read a name from keyboard and display on console.

Fifthprogram.java

```
class Fifthprogram {
    public static void main (String [] args) {
        String name = args [0]; // ⑤
        System.out.println (name); // ⑥ "Harry"
    } // ③
```

> javac Fifthprogram.java  
> java Fifthprogram // ① Harry  
→ Harry

> java Fifthprogram Srbuna  
→ Srbuna

Rule:- We must pass input value to execute this class else JVM will throw error,

> java Fifthprogram //  
Exception: ADOBE! 0

\* Develop a program to read a number, name, course, FEE of a student from end user from keyboard and display them on console

### Serathprogram.java

```
class Serathprogram{  
    public static void main(String[] args){  
        String number = args[0];  
        String name = args[1];  
        String course = args[2];  
        String Fee = args[3];  
  
        System.out.println(number);  
        System.out.println(name);  
        System.out.println(course);  
        System.out.println(Fee);  
    }  
}
```

```
> javac Serathprogram.java  
> java Serathprogram 101 Harry Java 1000  
101  
Harry  
Java  
1000
```

Rule:- To run above program we must pass minimum four values, else JVM will throw error.

## Java Printing Statements

→ Java has 3 printing statement to display some message on CONSOLE they are;

- (1) System.out.print( ); ← available from java.io
- (2) System.out.println( ); ←
- (3) System.out.printf( , ); ← java 1.5v

→ All above three methods are defined inside a predefined class PrintStream in java.io package as shown below

Package java.io;

public class PrintStream {

    public void print (String msg) {

        --- --- --- --- logic for

        --- --- --- displaying given  
    msg

}

    public void println (String msg) {

        --- --- --- --- logic for

        --- --- --- displaying given  
    msg

}

    public PrintStream printf (String format,  
                              Object ... args) {

        --- --- --- --- logic for displaying

        --- --- --- --- given values on  
                             the given format

}

}

- To display messages on console we must call and execute any one of the above three methods.
- All above three methods are non-static methods. Hence in order to call these three methods we must create PrintStream class object connecting to system console.
- We cannot create PrintStream class object, connecting to console by ourself because we need to access OS native library.
- We half of us SUN Microsystem created PrintStream class object connected to console in a predefined class 'System'. This object is created with the name 'out' as shown below;

```
Package java.lang;
import java.io.PrintStream;
public class System{
```

public static final PrintStream Out=.....;

PS class  
object creation  
statement  
connecting to console

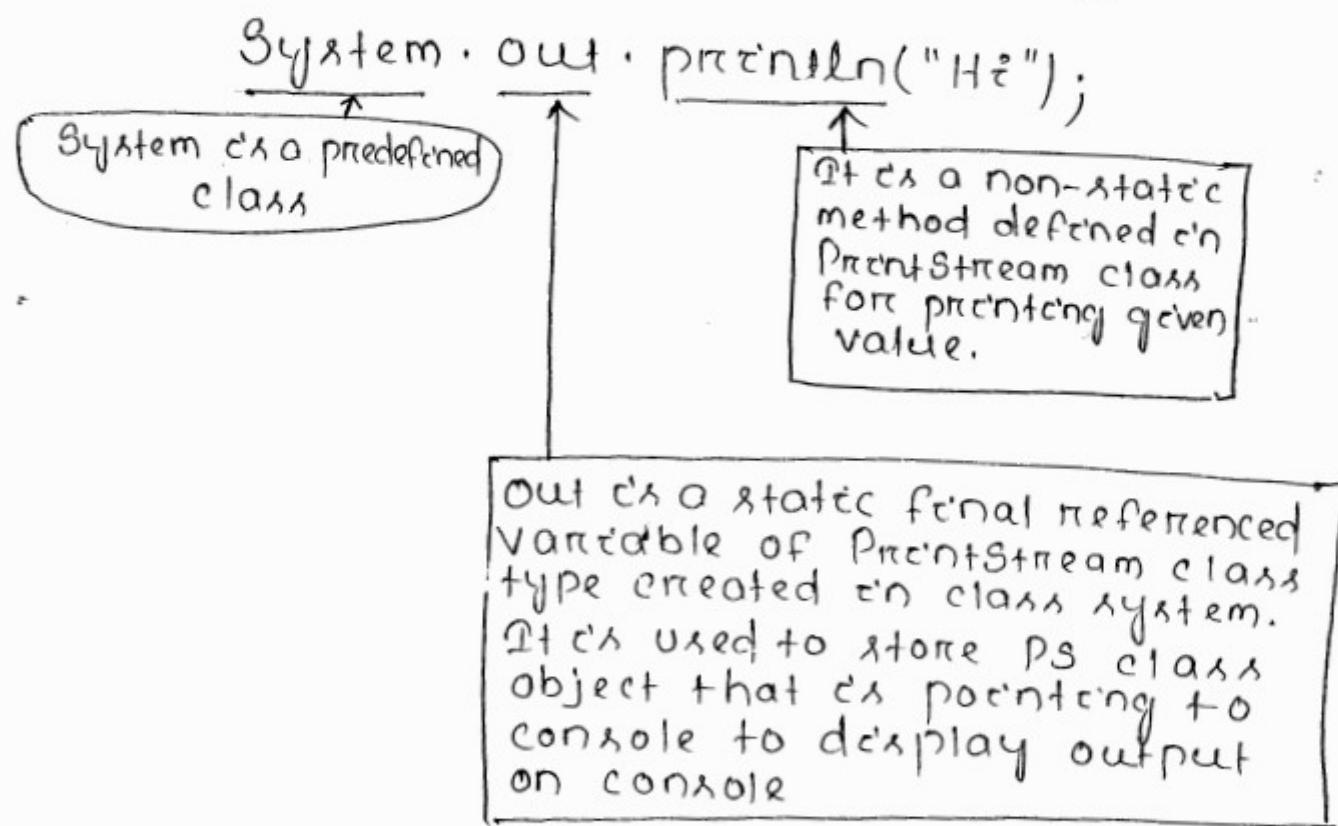
Rest of the statements  
available on class  
System.

}

→ Hence to invoke and execute print, println and printf methods, we must use the object System.out to display values on console. Hence, these three method calls statements are formed as shown below;

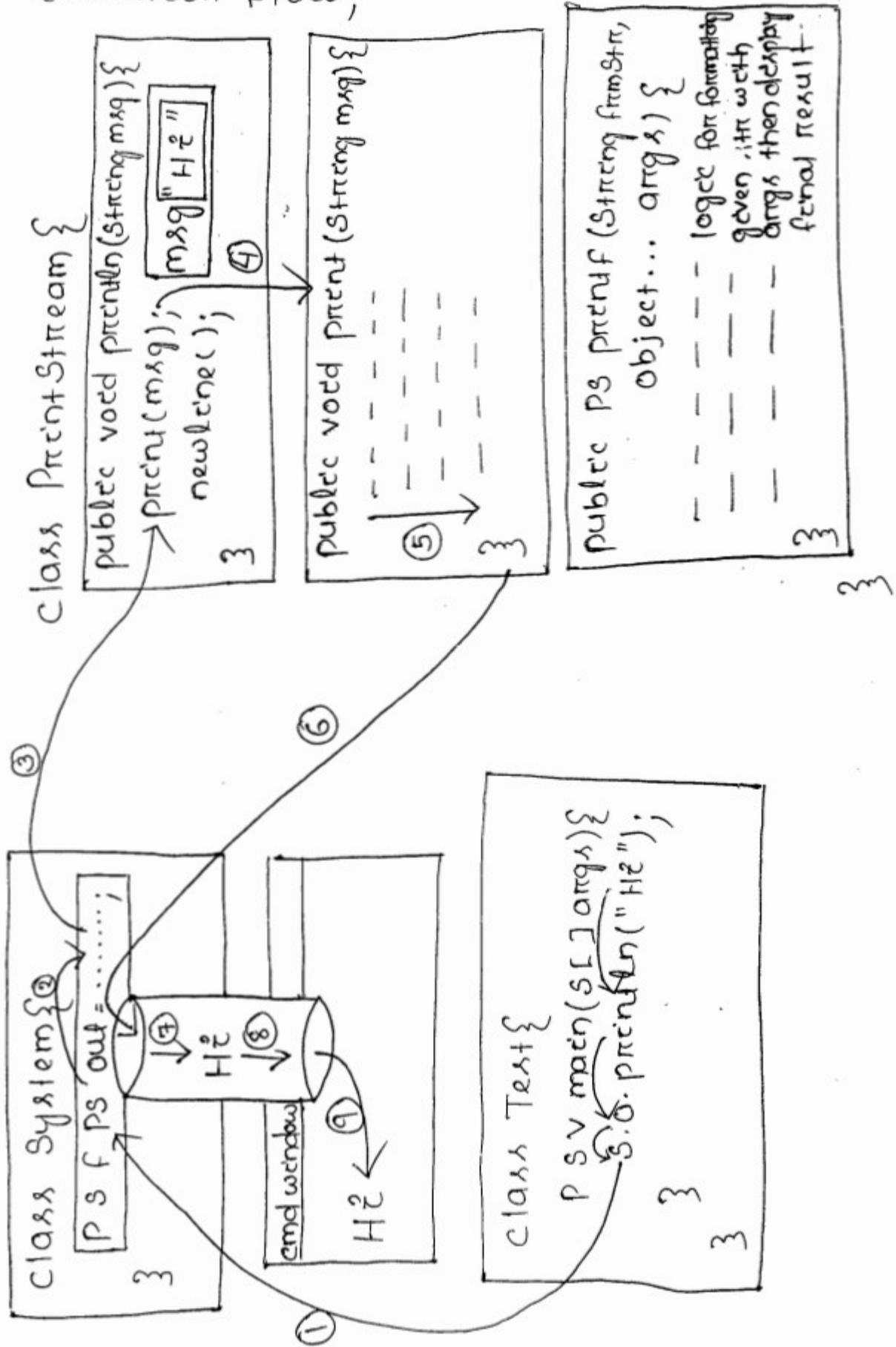
```
System.out.print("He");
System.out.println("Hello");
System.out.printf("HRU?");
```

\* Explain System.out.println("He"); statement



→ In order to display values on console we must call `println` method with the `out` object as available in `System` class. Hence the statement is formed as `System.out.println("—")`;

Below diagram will explain you System.out.println(" "); execution flow;





Difference between `print()` and `println()` methods?

Ans:- In, it means newline

- `print()` method will display given values on the same line and places cursor on the same. So that the next coming value will be also displayed on the same line.
- whereas as `println()` method will display the value on same line, then moves cursor to next line. So that the next displaying value will be printed on the next line.
- The "ln" property is applicable on next displaying values but not on current displaying value.

Hence we must choose either `print()` or `println()` method depending on the next value printing line.

- If we want print both current value and next value on the same line, we must display current value by using `print()` method.
- If we want to print current value and next value on different lines, we must display current value by using `println()` method.

Identify O/P pattern displayed in below cases.

|                                |          |
|--------------------------------|----------|
| <code>S.o.println("A");</code> | <u>A</u> |
| <code>S.o.println("B");</code> | <u>B</u> |

|                          |          |
|--------------------------|----------|
| <code>S.o.p("A");</code> | <u>A</u> |
| <code>S.o.p("B");</code> | <u>B</u> |

|                                |          |
|--------------------------------|----------|
| <code>S.o.p("A");</code>       | <u>A</u> |
| <code>S.o.println("B");</code> | <u>B</u> |
| <code>S.o.p("C");</code>       | <u>C</u> |

|                                |          |
|--------------------------------|----------|
| <code>S.o.println("A");</code> | <u>A</u> |
| <code>S.o.p("B");</code>       | <u>B</u> |
| <code>S.o.println("C");</code> | <u>C</u> |

- \* Develop a program to display the below 5 statements in 3 lines as shown in the below pattern.

Answer Sixteenth program

I/P:-

A  
B  
C  
D  
E

O/P:-

A, B  
C, D  
E

Seventh ~~Seventh~~ program.java

```
class Seventhprogram {
    public static void main(String[] args) {
        System.out.print("A");
        System.out.println("B");
        System.out.print("C");
        System.out.println("D");
        System.out.print("E");
    }
}
```

\* Read, write above program to display same O/P by using only print method, don't use println method.

### Program8.java

```
class Program8 {  
    public static void main(String[] args) {  
        System.out.print("A, ");  
        System.out.print("B\n");  
        System.out.print("C, ");  
        System.out.print("D\n");  
        System.out.print("E");  
    }  
}
```

NOTE! - `println("B")`; = `print("B\n")`;

The other difference between `print()` and `println()` methods is,

> We can't call `print()` method without passing arguments, but we can call `println()` method without passing arguments.

• `System.out.println();` statement displays empty line.

### \* Class Program9 {

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

`System.out.print();` & CE

`System.out.println();` ✓

`System.out.print(2,3);` & CE

} `System.out.println(2,3);` & CE

3.

both print()  
and println()  
methods are  
one parameter  
methods, so  
we can't pass  
multiple parameters

\* Develop a program to display A B C on three different lines with one line gap.

### Program10.java

```
class Program10 {  
    public static void main(String[] args) {  
        System.out.println("A");  
        System.out.println();  
        System.out.println("B");  
        System.out.println();  
        System.out.println("C");  
    }  
}
```

### Summary on diff. betw print() and println()

| print()   | println()   |
|---|---|
| (1) Prints value and places cursor on the same line                 | (1) Prints value and places cursor on the next line                 |
| (2) To display multiple values on the same line we must use print() | (2) To display multiple values on diff. lines we must use println() |
| (3) It has 9 overloaded methods. doesn't have no-param method       | (3) It has 10 overloaded methods. It has no-param method.           |
| (4) Sop() statement leads to C.E                                    | (4) Sopn() as a valid stmt displays empty line.                     |

## Difference between print() and printf()

- f, f stands for formattting
- print() method will display the given text as it is.
- printf() method will display the given text in the given format by substituting the given values in the text.

for example:-

System.out.print("He"); → O/P He  
System.out.printf("He %d", 5); → He 5  
System.out.printf("He" + 5) → He 5

\* Why we need printf() when we have already print()?

- printf() method is actually available from java 1.5 version onwards.
- It is added to java language to make it programmer comfortable to java language.
- Actually printf() is not required in java language. print() and println() can display data exactly as printf() can display.
- To make printing operation easy and to avoid the usages of format specifier like %d, %f etc.. print() and println() are defined with overloaded forms,

DT-21.11.2017

## printf method points

- (1) It is available from 1.5 version onwards
- (2) It is given in Java to display data with some special formats as we are displaying data in C language.
- (3) This method defined with two parameters as below:

```
public PrintStream printf(String format,  
                           Object... a)
```

Var-arg  
parameters

- (4) The second parameter of printf method is var-arg parameter, it means it accepts 0-n number of values.
- (5) We can call printf method by passing only one argument of String type, second argument is optional

```
System.out.printf("He");
```

```
System.out.printf("He");
```

- (6) If the string argument contains formatting specifiers (%d) we must pass second argument, as many %d we have.

```
System.out.printf("He %d", 5);
```

```
System.out.printf("He %d %d", 5, 6);
```

(7) When we run printf() method with formatting specifiers, Inside printf() method logic all formatting specifiers are replaced with given values in the order from Left to Right the final result will be displayed, as shown above.

(8) Rules in calling a printf() method is

- \* First argument String is mandatory
- \* Second argument is optional
- \* If String argument contains %d, we must pass second argument, as many %d we used those many args.

Identify CEs and O/P on the below lines:

|    |                                       |                    |
|----|---------------------------------------|--------------------|
| CE | → S.O. printf(); →                    | CE                 |
| ✓  | → S.O. printf("He"); →                | He                 |
| X  | → S.O. printf("He %d"); →             | RE<br>He Exception |
| ✓  | → S.O. printf("He %d", 5); →          | He 5               |
| X  | → S.O. printf("He %d %d", 5); →       | RE He 5 Encap      |
| ✓  | → S.O. printf("He %d %d", 5, 6); →    | He 5 6             |
| ✓  | → S.O. printf("He %d %d", 5, 6, 7); → | He 5 6             |

## Difference b/w printf() and printf() method

| printf()   | printf()  |
|--|---|
| (1) displays as it is  | (1) displays by formating given text with given value                             |
| (2) It is available from 1.0   | (2) It is available from 1.5 to support C style of printing result.               |
| (3) It is one parameter method and overloaded method. It has 9 forms | (3) It is two parameters method and it is also overloaded method. It has 2 forms. |
| (4) We can display a num or a string directly                        | (4) We can't display a num directly. We must displayed with string combination.   |
| (5) to display str and num we will " " and + combination             | (5) here we'll use %d and , combination   |

Below program will show you all differences between printf() and printf() method.

// program11.java

```
class program11 {
    public static void main (String [] args) {
        System.out.print(5); → 5
        // System.out.printf(5); CE! no suitable method found
        for printf(int)
        System.out.print("He"); → He
        System.out.printf("He"); → He

        System.out.print("He" + 5); → He5
        System.out.printf("He" + 5); → ("He5")
        → He5
    }
}
```

The common difference between printf(), println,  
and printf() methods is;

→ printf(), printf() methods don't have  
no parameter methods. We must pass one  
argument to call them, whereas println  
method has no parameter and parameter  
overloaded methods, hence we can call  
println() method without passing argument.

Ex:- Sopln(); ✓

Sopl(); X

Sopf(); X

→ If we want to move control to the next  
line by using prnu / printf method  
we must append \n to the next.

program12:- Develop a program to add two  
integers 5, 6 and display result on the  
pattern,

The addition of 5 and 6 is 11.

Display above result may be using both  
printf() and printf() methods in two  
separate lines.

```

// System.out.print("He",5); → CE
System.out.printf("He", 5); → He
System.out.print("He %d"); → He%d
System.out.printf("He %d"); → He:exception
System.out.print("He %d", 5); → CE
System.out.printf("He %d" + 5); → He5
System.out.print("He %d", +5); → He+5
System.out.printf("He %d", +5); → He:out
System.out.print("He" + 5 + 6); → He56
System.out.printf("He" + 5 + 6); → He56
System.out.printf("He %d", 5); → He5
System.out.printf("He %d.%d", 5); → He5.0
System.out.printf("He %d %d", 5, 6); → He56
System.out.printf("He %d %d", 5, 6, 7);
                                → He56
}
}

```

### ~~class~~ program12.java

```

class Addition {
    public static void main(Strong[] args) {
        int a=10;
        int b=20;
        int c=a+b;
        System.out.print("\n The addition of "+a+" and "
                        +" "+b+" is "+c);
        System.out.printf("\n The addition of %d and %d "
                        +" is %d", a, b, c);
    }
}

```

- # \* Interview Questions
- (1) Role of compiler?  
 (2) Role of JVM?  
 (3) What is compile time error?  
 (4) What is runtime error?  
 (5) Why CE?  
 (6) Why RE?  
 (7) Who will throw CE?  
 (8) Who will throw RE?  
 (9) Java file can't found in the given name?  
 (10) Class file is available but still we get error class not found why?  
 (11) Class file can't found in the given class name?  
 (12) main method can't available? Which time error?  
 (13) Main method mandatory for CT/RT?  
 (14) Can we compile without main method?  
 (15) What is java source file structure? Which are allowed in source file?  
 (16) What are mandatory and optional in source file?  
 (17) What is the order to place them?  
 (18) Where package statement is allowed?  
 (19) How many package statement?  
 (20) Multiple package allowed or not?  
 (21) Where import statement is allowed?  
 (22) How many import statement is allowed?  
 (23) Where interface, abstract class, concrete class, final class, enum, annotations allowed?  
 (24) How many?  
 (25) Can we create empty java file?  
 (26) Can we create empty class?  
 (27) Main method mandatory for CE/RE?  
 (28) If Java SFI = class name,  
 (29) If JSFI = classname(public)  
 (30) Why JSFI must be equal to ICSI?  
 (31) Why this can't applied for non-public class?  
 (32) In a single java file how many class we create?  
 (33) What are the rule for multiple class?  
 (34) If a java file has multiple class what should we have it?  
 (35) Can we create multiple public class?  
 (36) How many public class we can create?  
 (37) Why not multiple public class?  
 (38) How can we create multiple public class?  
 (39) If we compile multiple public class how many class file are generated?  
 (40) Can we compile multiple classes at a time?  
 (41) Can we execute it at a time?  
 (42) Can we compile all/multiple java file at a time?  
 (43) In addition to main method can we create our own method.  
 (44) What is the diff b/w predefined and user defined?  
 (45) What is the meaning of user-defined class and predefined class?  
 (46) Syntax to userdefined method.  
 (47) If we created own method it gets automatically executed.  
 (48) What is method calling syntax?  
 (49) Rule in calling a method?  
 (50) How can we provide memory?  
 (51) Diff. b/w static k/w & New k/w?  
 (52) Just by declaring method by static doesn't executed automatically or not?  
 (53) Why main method has static k/w?  
 (54) How main method executed automatically?  
 (55) Why JVM call only main method?  
 (56) Main purpose of main method.  
 (57) How many user defined method inside a class?  
 (58) Rule for userdefined method.  
 (59) What is overloaded?  
 (60) What is the order of method execution of all the method?  
 (61) I have a class with only user defined method, no main method then how can we run?



CH #2

DT- 27.11.2017

Comments, Identifiers and keywords with Java 9 New features.

- (1) What is comment, types of comments, programs FAQs.
- (2) What is identifier, types of identifiers, rules, FAQs.
- (3) What is keyword, different operations, keywords list, FAQs.
- (4) Java 9 newly added keywords - From Java 9 we have 51 keywords & 64 reserved word.
- (5) What is java tokens, types of tokens.
- (6) What is separator, number of separators.
- (7) What is a reserved word, number of reserved words difference between reserved word and keyword?

## Comment and types of comment

- (1) A comment is a special syntax using which we can inform to compiler some part of the code no need to compile. And also using comment we can provide description to a programming element or a statement for other programmer or only for us to understand it in future.
- (2) We have three types of comments
  - > Single Line comment → //
  - > Multi-line Comment → /\* \*/
  - > Document Comment → /\*\* \*/
- (3) Single line and multi-line comments are meant for commenting programming elements and statements, whereas document comment is meant for providing description or documentation for programming element.
- (4) We must use single line comment for commenting variable. We must use multi-line comment for commenting method and we must use document comment for providing description to a programming element.
- (5) We must not use single line or multi-line comment for providing description to a PE. Hence there is no CE, but the description will not be present on API doc file. Hence to provide description we must use only doc comment.
- (6) Similarly we mustn't use doc comment for commenting variables and methods. Hence also there is no CE, but those commented variables and method will be present on doc file.

(7) Hence we must use

- \* SLC :- for commenting one line → `/*-var/Syntax*/`
- \* MLC :- for commenting multiple lines → `/*-meth statements*/`
- \* Doc C :- for providing description to be present in documentation file.

(8) Below example will show you the usages of all three comments,

Ex:-

```
/**
```

- \* This class is used for adding given numbers and returns the result.
- \* author: HK
- \* Version: 1.0
- \* clients: Alankar Technologies
- \*

```
public class Addition {
```

```
/*
```

- \* This method will add given two integers
- \* returns result

```
*/
```

```
public int add(int p, int q) {
```

```
// adding given numbers and returning result
```

```
return p+q;
```

```
}
```

```
}
```

## Java 9 New features in API Doc file :-

(9) We got enhancement in API doc file generation that Java 9 onwards API doc file generation purpose Oracle Corporation is using HTML5 and jQuery. and also in java doc file they provided search option to find any class easily.

(10) Where are we allowed to place comment in java file?

We can place comment anywhere in java file.

- \* We can place in source file directly
- \* We can place inside a class
- \* We can place inside a method
- \* and even we can place inside a class / method / variable name, but with care.

Ex:-

→ //java source file level comment ✓

class Example{

    → // class level comment ✓

        void m1() {

            → // method level comment ✓

        }

    }

Four compile time errors w.r.t comments

### Example.java

→ /\* \*/ \* / & CE: c, e or unexpected

```
class Example {
```

→ /\* \*/ \* / & CE: illegal start of type

```
void m1() {
```

→ /\* \*/ \* / & CE: illegal start of  
expression

```
/*
```

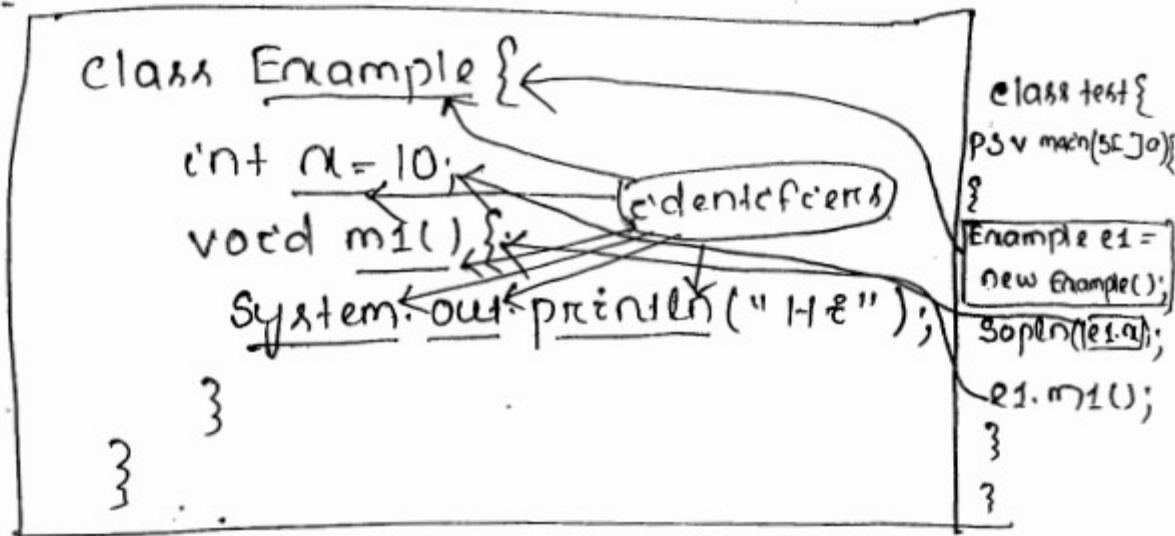
& CE: unclosed comment

```
}
```

## Identifiers, types, rules and OCJP bts.

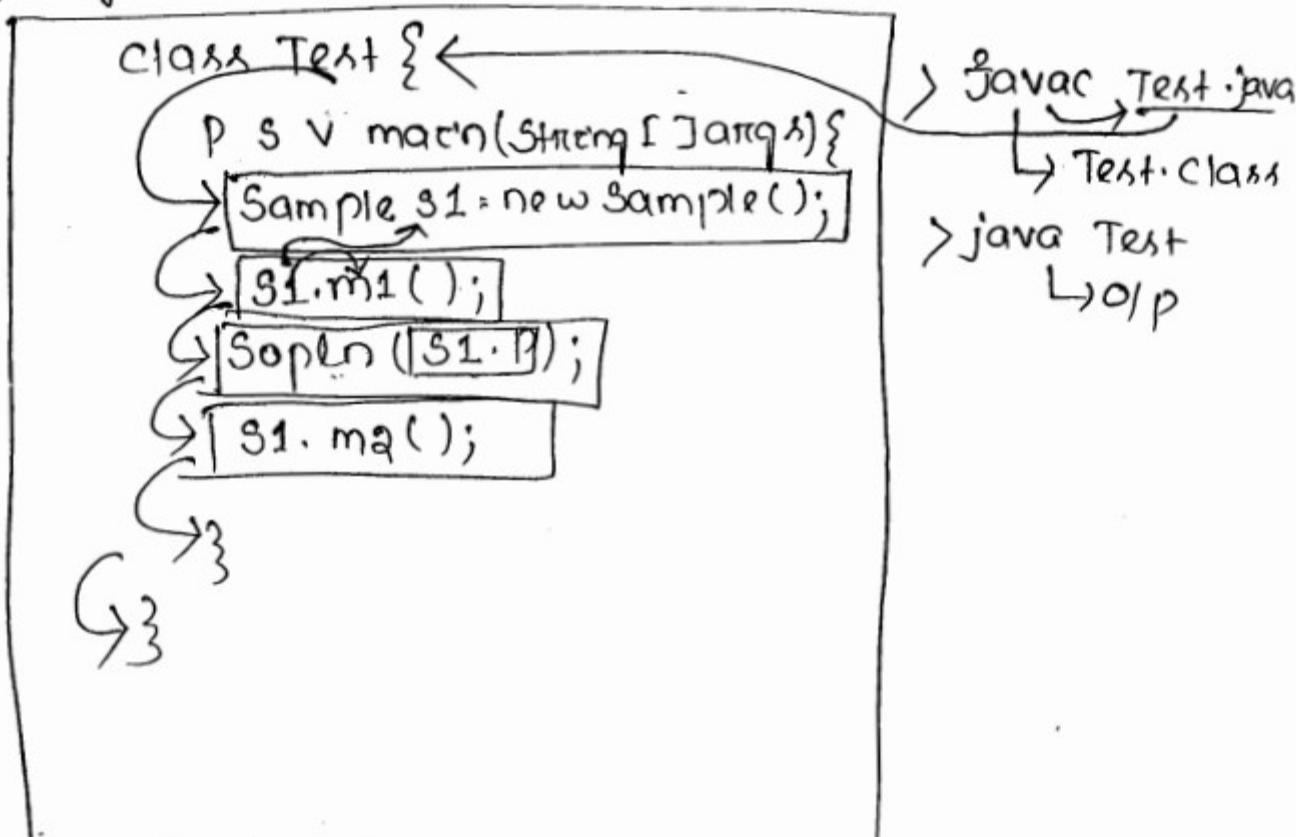
- (1) A name of a programming element is technically called identifier.

Ex:-

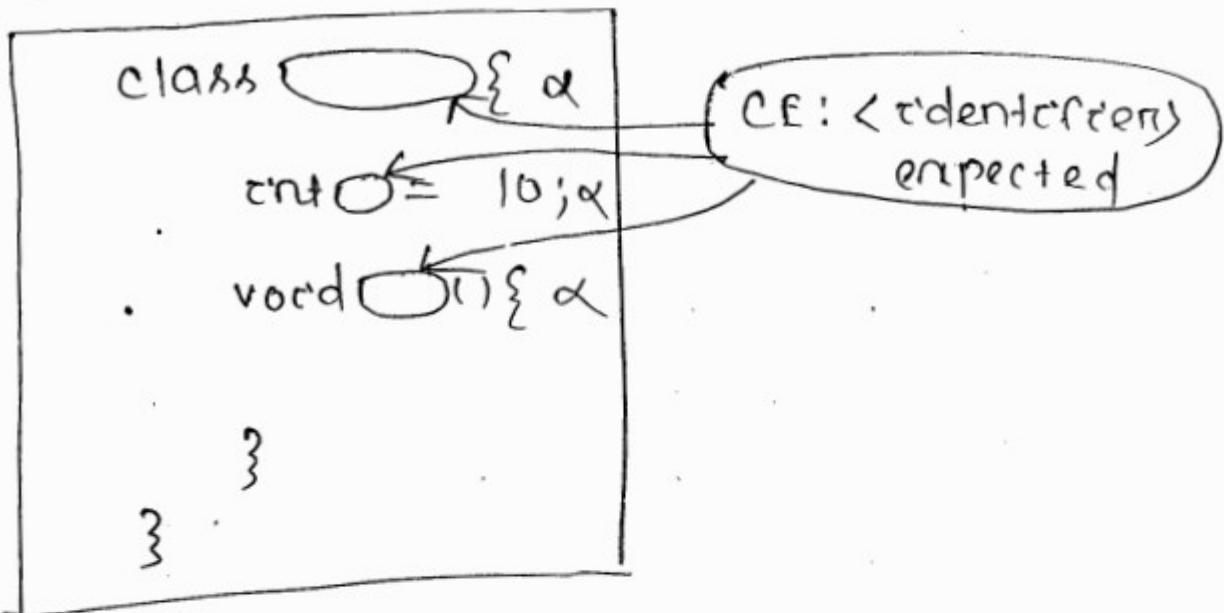


- (2) Identifier is used for identifying or finding a PE from other parts of the program or project

\*What compiler and JVM will do in the below program?



Is below compiled?



Different Identifiers or types. I | Rules on Identifier

- (1) module
- (2) package
- (3) interface
- (4) abstract class
- (5) concrete class
- (6) final class
- (7) enum
- (8) annotation
- (9) variable
- (10) method
- (11) inner class
- (12) label
- (13) generic parameter

name → the same below  
name → 7 rules are applied  
but dot(.) is allowed between words

name  
name

We must follow 6 rules + 1 rule while providing name to the programming elements.

## Rules:-

- (1) An identifier can contain,
- > alphabets (A-Z, a-z)
  - > digits (0-9)
  - > and special characters only \$ and \_.
- (2) An identifier can not start with digit, but it can be used from second position onwards.
- (3) An identifier can't contain special chars. except \$ and \_.
- (4) From Java 9 onwards we can't create identifier just with '\_' but it is allowed upto Java 8 version.
- (5) An identifier can't contain space in the middle of characters.
- (6) In Java identifier is case sensitive (a!=A)
- (7) Reserved words and keywords can't be used as identifiers.

Identify valid identifiers in the below list

- (1) abc → ✓
- (2) abc 123\$ → ✓
- (3) 234 Abc → X
- (4) \$\$ - \$\$\\$ - \$\$ → ✓
- (5) \_ → X
- (6) ---- → ✓
- (7) \_\_abc → ✓
- (8) \_\_\_123 → ✓
- (9) macn → ✓
- (10) class → ✓
- (11) static → X

From Java 9 version onwards we can't use single '\_' as identifier but we can use multiple '\_'s or '/' with combination of alphabets / digit / dollar is allowed and also we can start identifier as '\_'

NOTE!- But we can use predefined class name, variable name, method name as user defined identifier, and also there is no limit of the identifier length.

Q. Why we can't start an identifier with digit? and why we can't use single under score as an identifier.

(1)

`CLASS 1A { }`

(2)

`CLASS 1 { }`

(3)

`CLASS A1 { }`

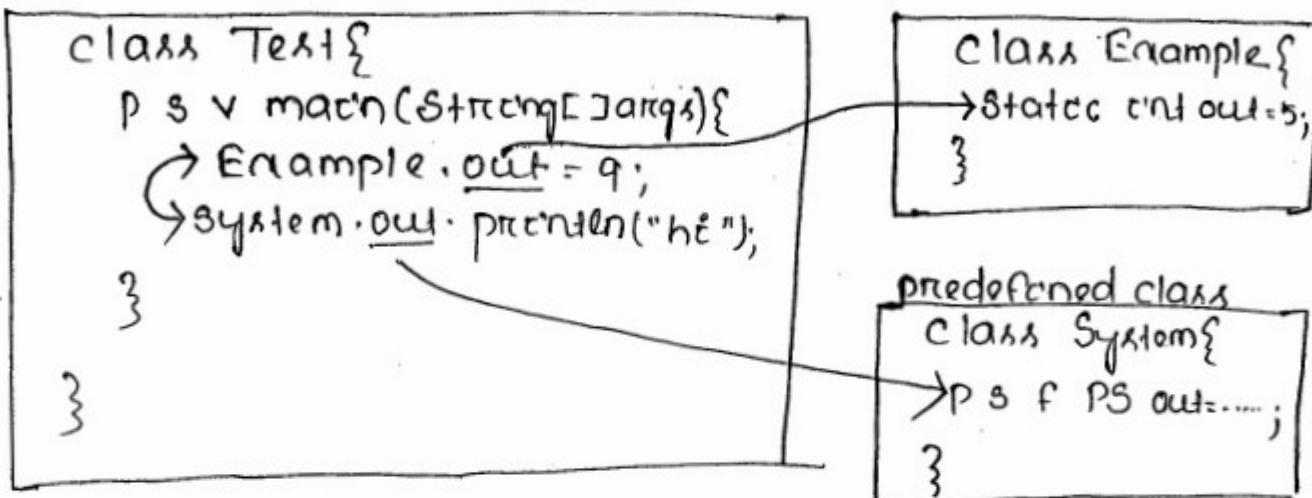
\* If we choose

Ans:-

\* If we choose predefined class name, variable name, method name as userdefined identifier, how can we differentiate predefined member from user defined member.

Ans:- By using class name and package name.

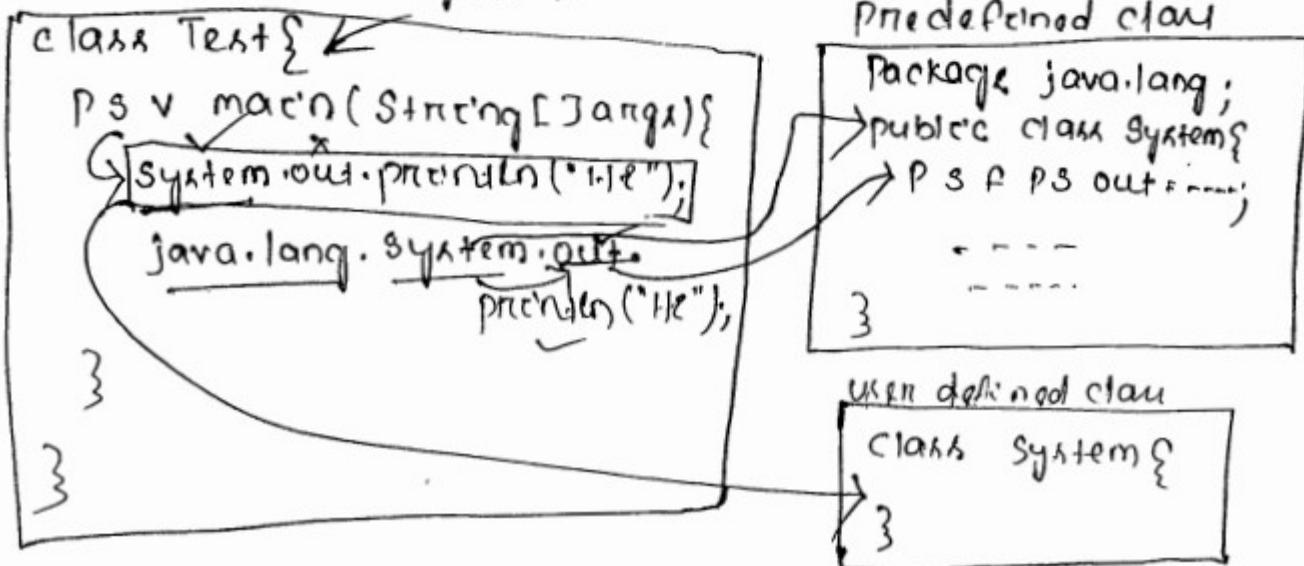
Ex:- (1) Created variable with name out



→ If we access "out" variable by using class name example, It will be accessed from class Example as end type variable.

→ If we access the variable "out" by using the class name system, It will be accessed from the class system as PS class type variable.

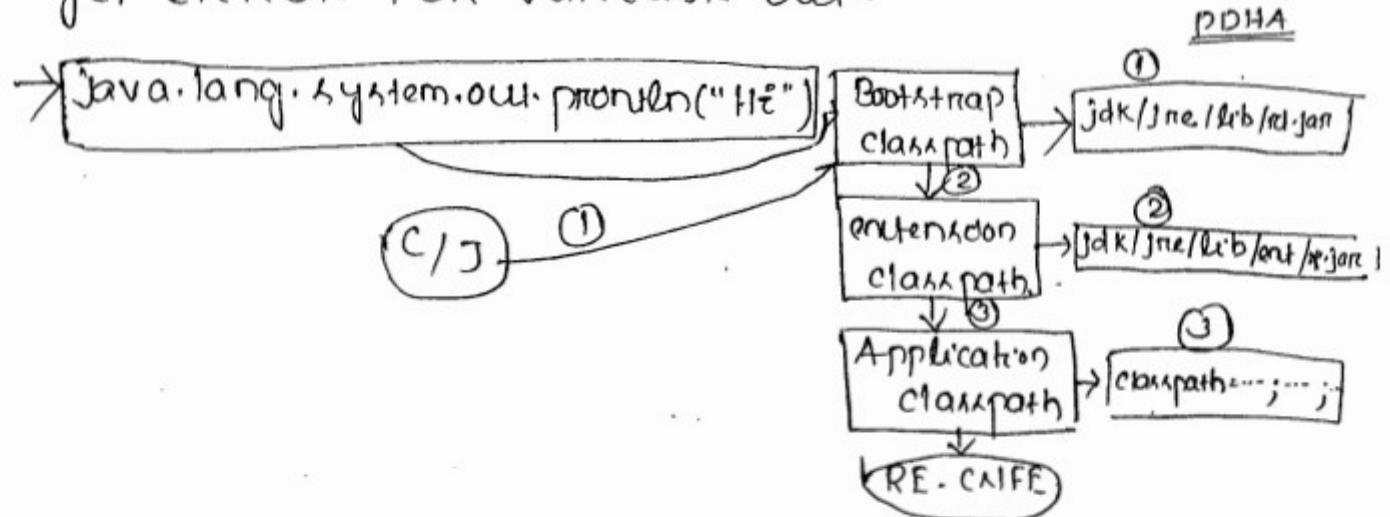
Case #2:- creating a class with predefined class name system.



- In above program we will get CE: C F 8 variable in the class system, because in the statement `System.out.println("He");`
- Computer software will access local class / our user defined class system. In that class we haven't defined the variable out, hence we get CE.
- To differentiate predefined class from user defined class, we must access predefined class by using its package name as shown above

Hence in the statement,

`[java.lang.System.out.println("He");]`, the class system in predefined class, we don't get error for variable out.



\* Identify below program is complete or not?

```
class A {  
    int System=30; ✓  
    int staticc=40; X  
    float Doubiz=50; ✓  
    String String="String"; ✓  
}
```

```
class B {  
    System String;  
    String System="String";
```

\* Findout valid identifiers from the below list.

- (1) maker
- (2) annotation
- (3) enum
- (4) \_
- (5) -----
- (6) \_\_123
- (7) \_\_abc
- (8) abc\_\_bbc
- (9) Strng
- (10) \_\_\_\$\$\$\$
- (11) 43abc

Keywords, rules, list of keywords, Java package.

- (1) A predefined identifier that has special meaning in a java program outside comments and string is called a keyword.
  - (2) A Keyword is a reserved word in java language made compiler and JVM that perform a unique and special operation.
- Rule: We have two rules on keyword.
- (3) → We can't use a keyword as user defined identifier.
- We must use keyword with all its characters lower case, because every keyword is defined with all characters lower case.

NOTE! - We have two levels of predefined identifier

(1) Reserved words and ~~Keywords~~

(2) Class, variable, method names,

> A word that is created as part of compiler and JVM softwares to perform one operation is called Keyword and a word that is created as part of compiler and JVM softwares to represent a value is not called as keyword, but is just called as reserved word.

> A word that is created as part of library (API) isn't called as keyword, it is just called as predefined identifier. And these identifiers are enclosed or under the scope of class and package.

> Whereas as keywords aren't under any scope or not enclosed by any thing. Hence keywords are directly accessible in our program.

Therefore we can not use keywords or reserved words as userdefined identifiers, because we and compiler can't differentiate.

But we can use predefined identifiers as user defined identifiers, because we can differentiate them using C/P.

(4) A keyword is used for communicating with compiler and JVM to perform one special operation on our program.

Ex:-

```
class Example {  
    static int a = 10;  
}
```

In this program we have used three keyword

\* By using class we are informing to compiler and JVM create Example as a class type program.

\* By using the keyword int we are informing to compiler and JVM create a int type variable.

\* by using the keyword static we are informing to compiler and JVM to allocate memory directly without object creation.

(5) Below are the 10 operations we will perform in a Java file by using keywords.

- \* package creation and usage
- \* class creation
- \* variable and method creation
- \* memory allocation
- \* controlling execution flow
- \* Setting access level permissions
- \* Setting execution level permissions
- \* establishing inheritance relationship
- \* representing object and its members
- \* handling user mistakes

(6) To perform above 10 operation Java supports 51 KWS.

(7) Among all those 51 keywords

- \* In java 1.0v we have 47 keywords.
- \* In Java 1.2v we got new keyword "String"
- \* In Java 1.4v we got new keyword "assert"
- \* In JAVA 5v we got new keyword "enum"
- \* In Java 9v we got new keyword "\_"

(8) Below is the list of 51 KWS with their operations:-

\* (10) package creation and usage (2)

- (1) package
- (2) import

\* class creation (3)

- (3) class
- (4) interface
- (5) enum (1.5)

\* Data types and return types (8+1)

- (6) byte
- (7) short
- (8) int
- (9) long

- (10) float
- (11) double
- (12)

- (12) char
- (13) boolean

(14) void → It is used as only return type

\* memory allocation (2)

static

- (15) new

\* control flow statement (11)

➢ Conditional

➢ Loop

➢ Branching

### > Conditional (5)

- (16) if
- (17) else

- (18) switch

- (19) case

- (20) default

### > Loop(3)

- (21) while

- (22) do

- (23) for

### > Branching / Transfer (3)

- (24) break

- (25) continue

- (26) return



### Accessibility Modifiers (4)

DT-01.12.2017

- (27) private

- <default> / <package>

- (28) protected

- (29) public

### \* Execution Level Modifiers (8)

- (30) static

- (31) final

- (32) abstract

- (33) native

- (34) transient

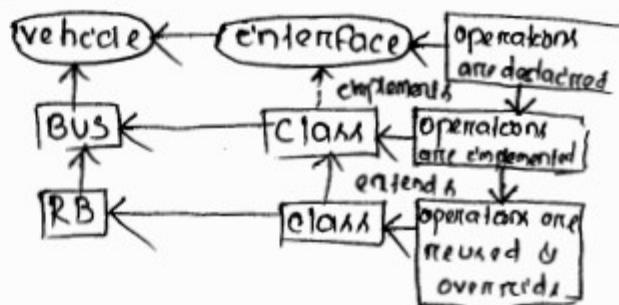
- (35) volatile

- (36) synchronized

- (37) strictfp (1.2)

## \* establishing inheritance (2)

- (38) extends
- (39) implements



## \* representing object and its members (3)

- (40) this
- (41) super
- (42) instanceof

## \* Exception handling (5+1)

- (43) throw
- (44) throws
- (45) try
- (46) catch
- (47) finally
- (48) assert (1.4)

## \* Un implemented and unused keywords (2)

- (49) goto
- (50) const

## \* Java's version new keyword

- (51) \_ ← underscore

## Default Literals

> boolean literals

(1) true

(2) false

> Null literals

(3) null

→ These three words aren't keywords. They are just reserved words representing values.

→ Even though these three words aren't keywords, we can not use them as user defined identifiers, because they are reserved words.

## \* Java 9v new features with keywords

We got 2 new features with respect to keywords.

(1) — is added as keyword

(2) 10 restricted keywords are added for module system.

From Java 9v onwards we can't use single '—' as identifier either for naming a class/ variable/ method or any other programming element, compiler will throw error.

The — is added as keyword in Java 9v, because for not to get conflicts with other languages like Python. We can use Java 9v JVM either for running Python or java based programs. Hence to run Python programs,

In JVM SW a new rule is created that ''' can't allowed as identifier as per python rule. Hence the same rule is implemented in Java compiler SW to stop Java programmers using python.

## Java module System

→ To support Java module system and to create and use module, in Java we got 10 restricted keywords.

→ Restricted keywords means these keywords are meant for using only in module programming. These keywords aren't regular keywords and they don't have any meaning in regular programming inside a class / interface program.

The 10 restricted keywords are;

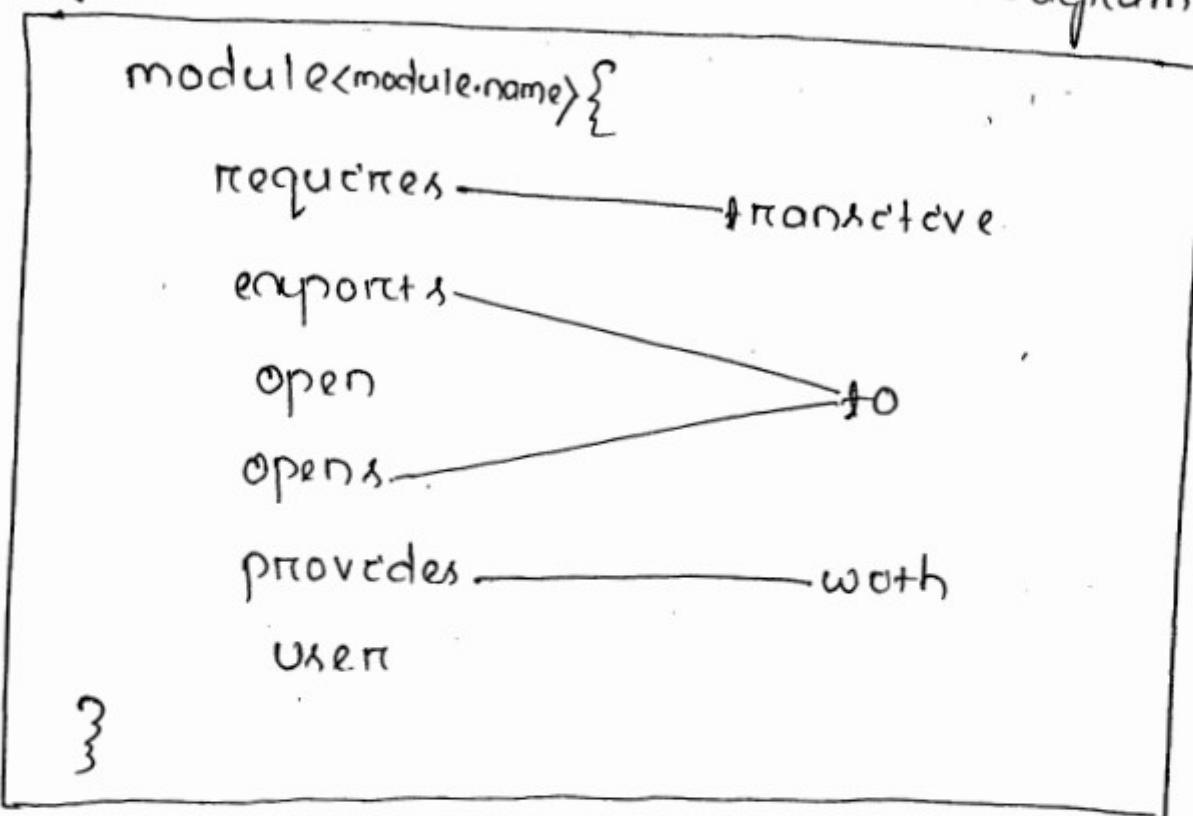
- (1) module
- (2) exports
- (3) requires
- (4) transitive
- (5) Open
- (6) opens
- (7) provides
- (8) to
- (9) with
- (10) uses

Above 10 restricted keywords are considered as keywords only in module programming. It means outside module programming inside class / interface we can use them as user defined identifiers. This facility added to support backward compatibility.

It means for not getting errors on the project those are developed using Java 8 or earlier version.

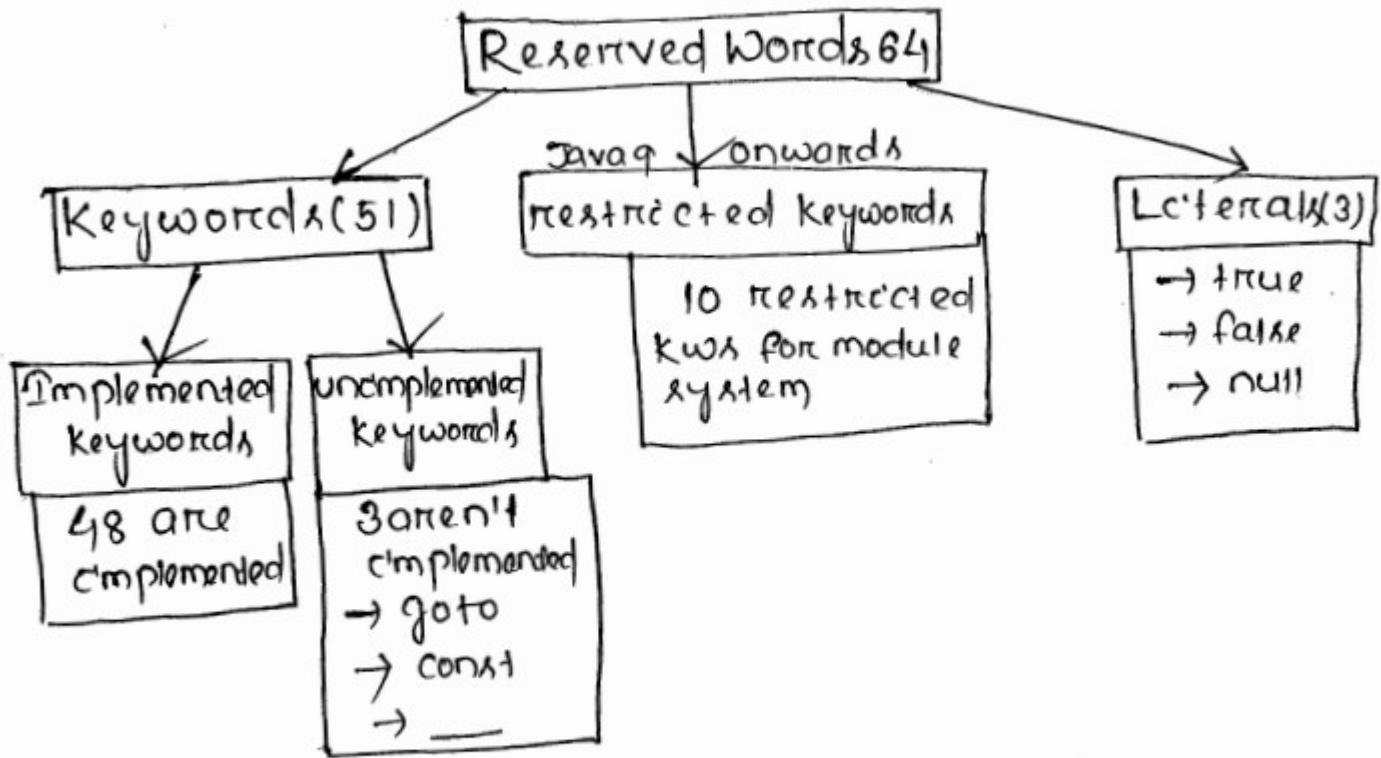
DT-02.12.2017

All above 10 keywords can't used individually some keywords must use in combination with other keywords. All keywords will go under module keyword as shown in the below diagram.



→ We must use transitive keyword in combination with requires keyword, we must use the keyword to in combination with either exports or opens, and we must use the keyword with, must use the combination with the keyword provides.

→ We will discuss the complete functionality of above keyword in the chapter module system after packages chapter.



Q. Why goto, const and — are not implemented?

Ans:-> Java language doesn't require these three kws hence they aren't implemented.

- > Even though they aren't required, java designers created them as keywords for stopping java programmers to use them as user defined identifiers in their programs.
- > If they are created as keywords, compiler will throw error when we use them as identifiers or here known as unknown. Only for getting error from compiler they are created as keywords and because they aren't required to use in java language they aren't implemented.
- > Java designers don't want allow java programmers to use goto, const and — as identifiers in java program from other language point of view C, C++ and python

> If we use these three words in Java program as identifiers, the programmers those are coming from C, C++ and Python will get confusion. Hence they shouldn't be allowed in Java program.

Q. What are the diff b/w reserved word and keyword? Can call a keyword as reserved word?

Ans:- Every keyword is a reserved word, but every reserved word isn't a keyword.

→ For example,

\* null is a reserved word but it is not a keyword.

\* class is a reserved word and also a keyword.

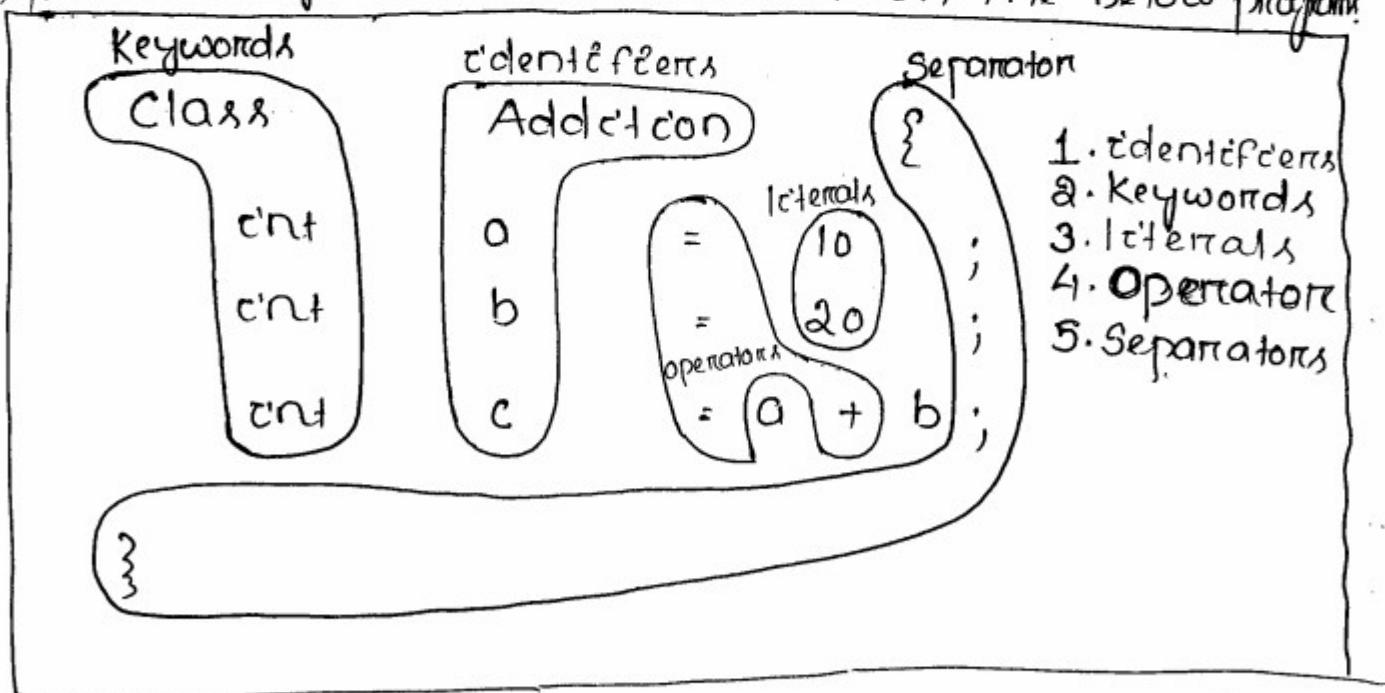
→ A reserved word that performs some special operation in a Java program is called a keyword. Some keywords won't be performed any operation, they are created as keywords - only for generated better compilation option when they used as identifiers.

→ We have 64 reserved words, among them 51 are keywords, 3 are literals, 10 are restricted keywords.

## Java tokens and types of Java tokens

- A token is a smallest unit in a program, that can not be divided further.
- Java supports five different tokens they are:
  - (1) Identifiers
  - (2) keywords
  - (3) literals
  - (4) Operators
  - (5) Separators.
- Except comments and white space anything we write in java program will be treated as a token. Hence any word or symbol we write in a program will comes under any one of the above five tokens.

\* Identify above five token in the below program



**1. Token categories (10)**

- module, package, class
- interface, enum, annotation
- variable, method, table
- generic parameter name

**2. Keywords (51)**

- class, interface,
- static, void, int,
- long, ...

**3. Literals (7)**

- \* Integer → 10, 20, ...
- \* Floating point → 10.0, 20.9, ...
- \* Character → 'H', 'K', '1', ...
- \* Boolean → true, false
- \* String → "HK", "IIT", "1", ...
- \* Class → Strong, class, A, class
- \* Null → null

Java  
Tokens

**Note :-**

- (1) :: (double colon) isn't a operator in java, it's a separator for representing method and constructor.
- (2) The arrow(→) is not a separator in or Lambda expression operator.
- (3) The separators ... and @ given in 1.5 v

∴ given in 1.8 v  
operator → given in 1.8 v

**4. Separators (12)**

- { } ( ) [ ]
- ; , . ... @ ::

**5. Operators (10)**

- + - \* / %
- < > <= >= instance of
- == != new → ?: etc....



# Working With Edtplus

DT-04.12.2017

- (1) What is edtplus?
- (2) Why edtplus?
- (3) When & should we choose edtplus?
- (4) How can we get edtplus?
- (5) Create a textfile, javofile and other program files.
- (6) Configuring computer and JVM

- (1) What is edtplus?

Ans:- An edtplus is a text editor software like notepad. It is used for developing text files and also any language based program files.

- (2) Why edtplus?

Ans:- Edtplus will provide different short-cut techniques to achieve fast development compared to notepad.

- (3) When & should we choose edtplus?

Ans:- We must use edtplus for creating and reading text based files. Edtplus isn't suitable for creating and reading doc files and image files.

\* What are the problems of notepad those forces us to use editplus or some other advanced editor software like Notepad++, Textpad, etc?

Ans:-

- Doesn't provide java program's default template.
- Doesn't provide code syntax highlighting (no color).
- Doesn't support code indentation (no alignment).
- Can't add .java extension automatically.
- Doesn't provide compilation and execution option.
- Doesn't support moving to source file automatically.
- Doesn't support auto saving.
- Doesn't support displaying line numbers.
- Doesn't support character case changes etc..

Editplus software is providing solutions to all above problems. So for developing sample programs as part of your learning process it is highly recommended to use Editplus or Notepad++ or Textpad.

(4) How can I get editplus?

Ans:-> Download and install the latest version Editplus software from [editplus.org](http://editplus.org).

> It is a licensed version software. you buy license. or you must use 30 days trial version.