

Keyboard

Mouse

- Ram

HDD

TPU

etc

Software

System S/w
(machine)

OS

- mac

- windows

- solaris

- unix

- android

- linux

- etc

Application
S/w

client
server

Application Software →

window app

local app.

global. app

web. app.

global app

→ run on particular
system

run on server

offline

Ex

- Tally
- ms office

online

Ex - gmail

- yahoo
- amazon

Language →
Medium of Communication

Translator →

Assembler

Single character
by
single character

Interpreter

single row
by
single row

Compiler

whole

* lowest speed

medium speed

highest speed

Computer oriented languages →

Programming

structured

Scripting

markup

- Business logic

- Backend

- client side

eg

- code construction

- data storage

- validation on

based

- validation

- logical format

- the web

- DBMS (SQL)

- page

Programming

local

C

C++

Java

global

Java

3

.Net

-
-
- python
- etc

.Net

PHP

- python
- etc

Scripting →

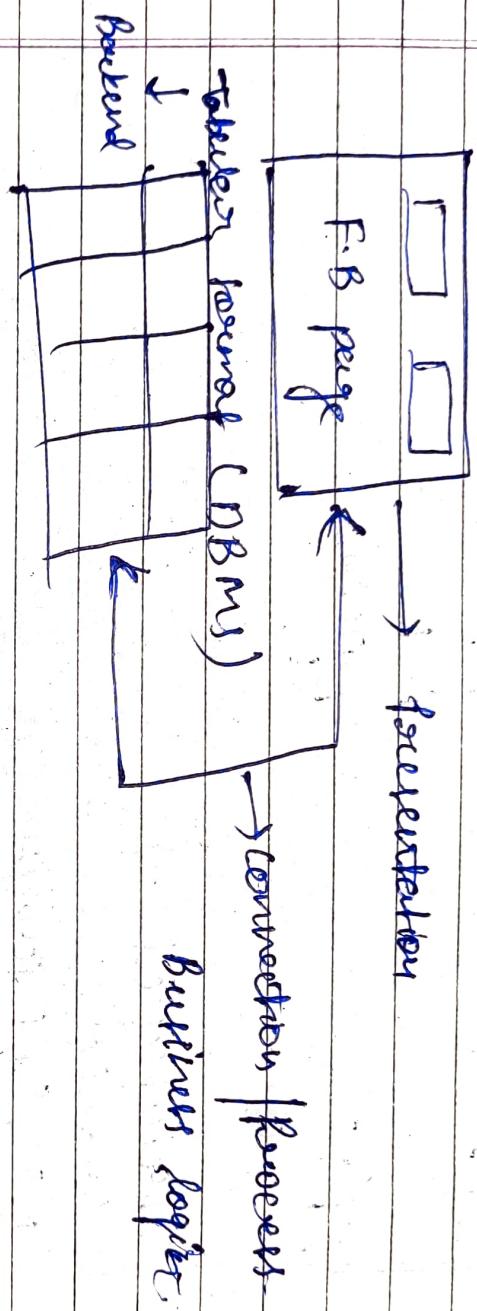
- Oracle
- MS Sql Server
- MS access
- DB2
- Sybase
- SQLITE
- etc

Scripting -

- JAVA Script
- VB Script
- Jquery
- Angular JS
- Node JS
- etc

Markup -

- HTML
- HTML5
- CSS
- Bootstrap
- Templates
- XML
- etc



- Front end →
- 1 - Presentation / view / interface / design / look & feel / animation etc (Scripting / markup)
 - 2 - process / Business logic (Front end to backend communication) (Programming.)
 - 3 - Backend I
data stored in logical personal forms / SQL
(structured / DBMS / SQL)

Evolution of programming languages

- | | | | |
|-----|------|---------|--|
| (1) | 1957 | Fortran | numerical computation |
| (2) | 1960 | Algol | Algorithm based. |
| (3) | 1963 | BCPL | Basic combine programming language. |
| (4) | 1967 | CPL | Combine Programming language. |
| (5) | 1970 | B lang. | developed by:
Sir Martin Ritchie. |
| (6) | 1972 | C lang. | developed by: Sir
Denis Ritchie. |
| (7) | 1980 | C++ | Object oriented lang
developed by:
Sir Bjarne Stroustrup |

Disadvantages of C++ →

- local
- platform dependent
- inefficient

Objective -

- global
- platform independent
- efficient

6

1991 Java developed by Sun Microsystems
Sir James Gosling.

* OAK → Real name.

* 1995 , Java → new name.

by Sun Microsystems

Since 2006
ORACLE (Take over)

We are defining you

JAVA

→ - Java 2 standard edition (Core Java)

- local / desktop, run on particular system.

Ex. GUI, CUI

→ J2EE → Java 2 enterprise edition (Advanced Java)

- global / web app, run on server, online, Ex → Servlet, JSP, Java Bean, Expression lang etc.

→ J2ME → Java 2 micro edition, wireless technology

- mobile app, - handheld, - data card etc

→ EJB → Enterprise Java Beans
- session beans, - entity beans, - message driven beans).

→ Framework →

- JSF → presentation (model)

- Hibernate → Backend (storage)

- Spring → business logic (controller)

- STRUTS - Frame work

MVC based

Java platform

* MVC → Model view controller

Vendor → Apache tomcat

7

JAVA Architecture →

Java 2 way process →

Step - 1

1st process

Step - 2

2nd process

Step - 3

Source code

• java file.

Byte code

• class file

Machine code

developer

user / client

compile

execute / run

compiler

java

javac

JVM

JDK

(Java development kit)

Install,

location :

C:\program files\java\jdk\bin\javac

How Carry Java application development -

Type 1

Manual

- Notepad.

- command prompt.

Type - 2

JDE

(Integrated Dev.
Environment)

- Netbeans.

- Eclipse.

- IntelliJ Idea.
etc.

How to install JDK.

Class demo

{ void func()

System.out.print("First program");

public static void main(String args[])

{ demo obj = new demo(); // object creation
obj.func(); // function calling.

}

// public → for access outside a class
// static → bind with the class name

(bcoz static function calls from the class name)

Interview Question

// void → no return type of main() method
// main() → predefined method (it is a starting point when the translator for the execution)

// string args[] → command line arguments

// String → predefined class

// args → name of array (user friendly)

// [] → array,

// funcl → user friendly function

// System → predefined class.

// out → predefined variable.

// print() → predefined function (with something)

// class() → keyword.

// demo() → default constructor.

// {} → open braces

// {} → closing braces

// ; → double quotes

// ; → service colon

// ; → semicolon

// ; → command line arguments

// static functions call from the class name.

Or our object both.

// Non-static → function call is from our object.

* printer → to change the line and print