

d. `setName("Shaw")`

`System.out.print`

`String name = d.getName();`

`System.out.println(name);`

Constructor: It is defined as a Method
name of same class name & no return type.

5 Don't we need, ~~std~~ std.Student() ✓
in object creation pass through ↓

Ex:- `Student std = new Student(" ")`

→ No void ✓
→ public as per our requirement ✓

* public, protect → Access Specifiers ✓

* public, protected → Access Specifiers ✓
* when ever there is no constructor in class
* when ever there is no constructor in class
then built in built `JVM` can create it \downarrow \downarrow
`Student std1 = new Student()`

for `Student2 std2 = new Student2()`

* I could use only ~~0~~ parameterize
when no constructor in class and also

* No parameterized constructor in class
is known as default constructor. \downarrow no parameterized constructor in class
`Student std2 = new Student2()`

Default looks like
constructor

- * When constructor creation
 - ↳ No JVM for default zero parameter
 - ↳ when NO constructor creation
- * When constructor Overload
 - ↳ JVM will give zero parameters

↳ Same Method name ✓
↳ Same parameters ✓

↳ Different parameters ✓
 $(\text{Super} \rightarrow \text{default})$

* In constructor

- ↳ whether we write in first line or not
- ↳ whether we write in first line or not

↳ there could super() or this()

↳ only one either \uparrow or \downarrow

this \rightarrow call same class constructor ✓
super \rightarrow call parent constructor

* Constructor changing \downarrow call one more stack or
constructor changing \rightarrow call one more stack or
stack over flow \rightarrow may stack

return type explicit ?
this \rightarrow call same class
constructor

* Inside one constructor required this \rightarrow
this \rightarrow call same class
constructor

* refer to current object ✓
↳ Method

Method

Constructor

- * call explicitly by calling name
- * when object is created constructor is called.
- * will not participate in inheritance

No return type

explicitly

not participating

will not participate in [during class loader system]

static keyword

static variables can execute

In first static block

After

Static Method

non-static block

constructor

method

instance variable

method

Java Stack (Normal)

will execution one after another

Constructor

when object is created

constructor is called

when object creation [Memory for instance variable]

first program like this

int a;

int b;

static

S.O. print ("static block");

Print Main Method

static int a;

int b;

first program like this

int a;

int b;

will execution one after another

constructor is called

when object creation [Memory for instance variable]

first program like this

int a;

int b;

first program like this

* Static variable also called as class variable.

- * Eclipse → static examples ✓
- * String Assignment ✓

Class Demo

4

INT B

* print com

Demo C

Count++

۵

public law main

```
public static void main ( )  
{  
    // new Demo ();
```

Mark created

* static block → same Memory allocated →

xvii

(count + 1)

Democracy

10

Jas
Moor

Avoid this weak static int

* wing object reference ✓

using older references

Static Methods
can be called
using class name

not provide
Object creation
Mandatory

* 4
↳ static variable y must be
↳ static variable y must be
↳ instance variable in Private

* Good practice is all in one Method which

* exceeding
effect modularity:

Radiation or Soot

* In static ~~part~~ block, if any mistake taken
then it results exception.

place, when it results

Inheritance

Code re validity

Runtime Polymorphism

Abstract

MORNING

... in flavor, we can never

Between ~~countries~~
~~countries~~

by
extending

Ex- Class Demo 2 extension Relationship

19
Karak...
— (noz-A)

* nutrient - child relationship

* patient = ...
' - Sub class

* Bass class - 3 sec
derived

* existing class - definite

* Inheritance

— [christian] [resistivity]

Rules: [Imperialism] is allowed (one class) ~~excluded~~ ^{is many}

* single inheritance

first class is parent of all classes

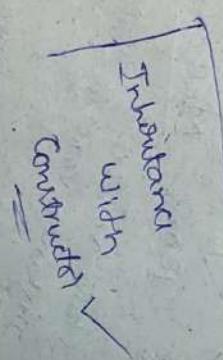
* Multilevel individual vs. group

* Mr. DeWitt Bass can have only

class / sub class.

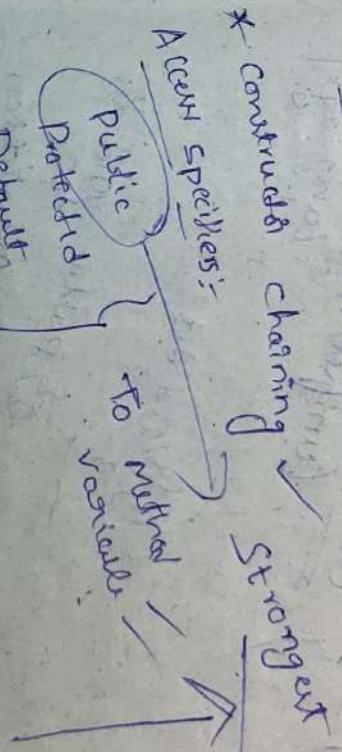
~~Multiple inheritance~~

X and Clark 6/16. J
11:00 a.m. + May 2d X and 123 extnd 321

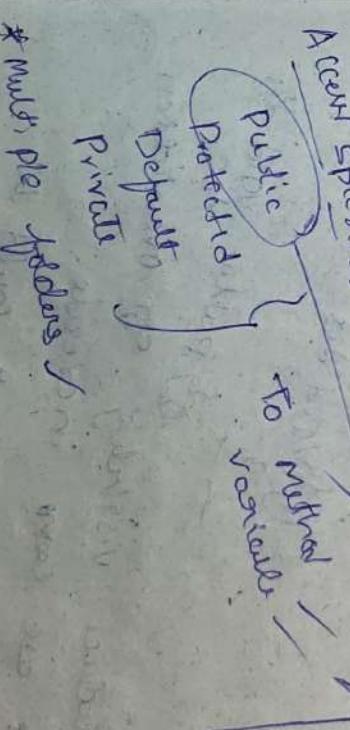


- Interface ✓
Between classes we can build relationship
by extending ✓

- * Constructed will not participate in inheritance
however in acting there could be super-
in the construction that will call.



- * constructs chaining → strongest Access Specifier



- Access Specifiers:

```

graph TD
    AS[Access Specifiers] --> P[Public]
    AS --> PR[Protected]
    AS --> I[Private]
    P --> D[Default]
    P --> MI[Multiple Inheritance]
    PR --> MI
  
```

* Multi ple InheritancE

Within in class (is-A relationship) Different package

Inherited methods → As it is we say child from Parent

✓ Overriding method → from child to parent for overridden child modified ✓

✓ Specified method → In child there methods which are not in parent class

+ → public

→ protected

~ → default

- → private

UML (Unified Modeling Language)

Point
i=10

child
i=20

→ S.O.P[1] → 20
Super. → 10 ✓

Overide method

Rules:-

1) we cannot reduce visibility of overridden

method but we can increase.

method but we can't change inherited methods could be

2) when over overriding mainclass in public

mainclass in public overridden method must

3) Return type of overridden method must be same, not change in return type

4) Return type of different as that of parent class can be different return type ✓

5) Parameters of overridden method must be same as that of parent else it will be considered as not overridden

super → it will invoke constructor of parent class

super → it will call parent class

super keyword → To call parent class instance variable

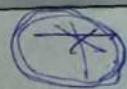
Point
i=10

child
i=20

→ S.O.P[1] → 20
Super. → 10 ✓

final keyword → Class Method ✓

Method Variable ✓



final keyword → Class Method ✓

* final class will not participate in inheritance

* final class can't override method ✓

* yes final method can't override method ✓

but we can't change its

* final variable acts as constant but it

not modified ✓

Runtime-Polymerism

* reference must be same in child ✓ [tight coupling]
* reference must be differ in parent ✓ [loose coul.
* reference must be differ in parent & child
* runtime poly morphism

[child class c1 = new child1]

* type of object & reference to be

same → Tight Coupling [child c1 = new child1]

* type of object & reference to be

different → Loose Coupling [parent aa = new child2]

(invoke)✓ one time will give

polymorphism. [call methods one time which
methods ok ✓ are present in
child] ✓

* overridden

Method ok ✓

* Inherited

Method ok ✓

* But specialized Method X

[directly not occur] X ✓

(indirect) [By Add cast ✓] [Down casting] ✓

Abstraction

↳ abstract keyword ✓
↳ interface ✓ (100%)

for abstract class we
don't object ✓

actual implementation

↳ which means hiding
showing existing only features.

↳ NO Body & implementation,
but abstract public void methods;

{ X } Method is in abstract

* In the class, one method is in abstract
then class name should be in abstract

class ✓

* Variables cannot be abstract X

* Methods ✓ Create

* we can't object for abstract class

* abstract class can have both abstract
& concrete methods ✓

* abstract class can have only concrete ones

* no problem ✓

* constructor cannot be made abstract X

Abstract class
↳ incomplete class [because no creation of obj]

Incomplete class [because no creation of obj]

Abstract Method Final X (illegal combination)

Learn + practice + explore

IS-A relationship: relationship means extends or
implements
↳ inheritance b/w 2 classes which
are polymorphic

Can we have constructor in abstract class?

204

Has-A relationship [most imp for Spring boot framework].

e.g. class Engine //Dependent object

Bentley

1
one Assisted me
to me

Lm IDE (Proctu)

practic

bliss classes)

1

law can // Target object

Has-A relationship

3 Engine engine; // instance variable

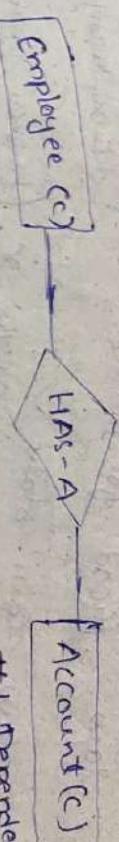
Dependency injection: The process of injecting dependency into object to target object.

- * Constructor Dependency injection [through Constructor]
- * Setter Dependency injection [through setter] ↗

1) one to one association (Established by a new clause
to many association through a clause

) Many to one Association

) Many to Many Association - Assessor



Inner class: Good for Maintenance (which reduce documentation).

* Inner class can static ✓
* Outer class cannot be static X (because how we call).

⇒ when we use inner class name as static
then we use directly our outer class a name.

⇒ when we don't use inner class as static,
then we go for object to class inner *

↳ [refon 1 Dec class → last code snippet]
* Inside Main method we can Create a

class without name known as
Anonymous.

\$ → Interface Specification (SRS) ✓

Interface :- Service requirement specification

✓ one with a code and run it anywhere
over different servers, databases ✓

(API)

GUI → Graphical user Interface Application

Customer and Banks Application

Contract between client and provider.

Contract
between
this
interface

so, introduce after a contract
b/w service provider

first of all you will interface
with abstract (no body) then
after write class which
implements interface, then → overriding
we write body for those methods which
are present in interface write in class

abstraction if we want to promise
100% of abstraction it we want to provide

in Java

interface Account

// It is 100% Abstract
methods

public void deposit();

public void withdraw();

public void checkBalance();

3 Abstract class ✓

100% pure Abstract interface ✓

because of → Interface

interface (Example) → It is interface
by default public abstract

{ Method default to public &
abstract
void m1(); } Method default to abstract
void m2(); } Method default to abstract

j SampleImpl implements ISample

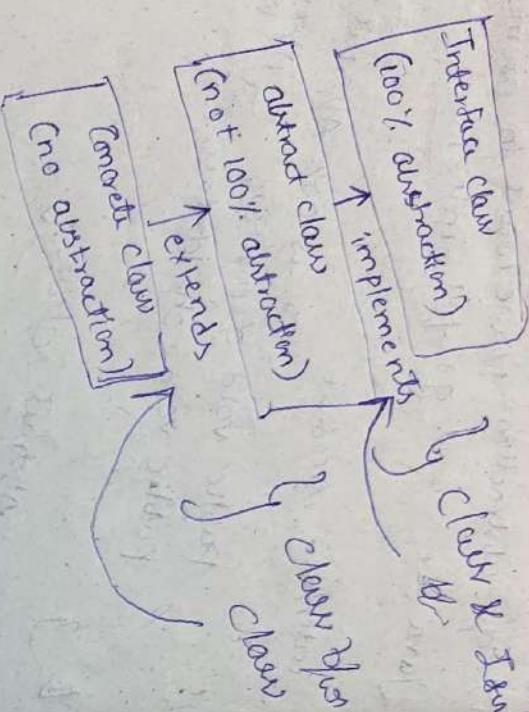
class SampleImpl {
 public void m1() {
 System.out.println("Print m1");
 }
 public void m2() {
 System.out.println("Print m2");
 }
}

3 No default access modifier inside of class
No default access modifier inside of class
No default access modifier inside of class

Object obj = new SampleImpl();

obj.m1(); // (Sample) m1();
obj.m2(); // (Sample) m2();

SampleImpl obj = new SampleImpl();
obj.m1(); // (Sample) m1();
obj.m2(); // (Sample) m2();



- // class extends class
- // class implements interface
- // class extends class & implements interface
- * interface extends interface, interface ✓
- // implement interface, interface ✓
no. of interfaces ✓
- // class
- * one interface can extend ~~two~~ interfaces ✓

* one class extend at a time ✓

* one class implements two interfaces

* one class implements two interfaces

* class commonImp implements One, Two by implementation given by MySQL, Oracle -

Ex:-

① class can extend as well as can implement

② class can extend as well as can implement

on interface. → first pref.

↳ access modifiers in java

↳ in this mainly public, private, protected.

↳ Inside interface, we can define variables. ✓

* Every variable present inside the interface

* by default public static final.

* Inside interface, we can implement [SR]

why final → variable name can't be modified [SR]

why static → without implementation of class obj name

can implement any no. of interfaces simultaneously

Inside interface, we can implement [SR]

contd normal methods, →

interface Remote

* when we final compulsory you have to initialize

Otherwise compile time error.

interface ISample

```
{ int a = 10; // default public static final }
```

} public class TestApp implements ISample

```
{ Main method
```

```
{ int a = 0; // local variables
    System.out.println(a); // 0
    System.out.println(ISample.a) // 10 }
```

Marker interface (or)
Tag interface (or)

Aug interface

interface known as
Marker interface.

Adopter class [It is a design pattern allowed
to solve the problem of direct implementation
of interface methods]

Concrete class
Abstract class } when you have to

interface ↓

* when class implements with two interfaces,
simultaneously? except in both the interfaces
→ you possible, ~~exist~~ in both the interfaces
method signature should be same, but
different return types.

Difference b/w

Interface ↓
Abstract class abstraction

No private, when some signature with different
return type.

public abstract

⇒ 100% abstraction

⇒ need public abstract

2 public abstract

separately ✓ knows JVM ✓
cloneable ✓

marker interface - which doesn't contain any
abstract methods inside a

interface known as

Note:- When in interface all about abstract Method
Coming to abstract it could be there
Then what purpose of interface?

Ex:- abstract class Sample

```
{  
    public abstract void m1();  
    public abstract void m2();  
}
```

interface ISample

```
{  
    void m1();  
    void m2();  
}
```

Because of interface performance high ✓
abstract performance low ✓
* if everything abstract we recommended interface

- * no object creation for abstract class.
- * `toString()` → returns the hashCode value of the object.
- * `Integer i = new Integer("Ten");` ~~// Number Format exception.~~

Wrapper classes

`int a = 10`

↑
Primitive type

`integer a = 10`

↑
reference type

class contain two

* Every wrapper
Constructors.

* String cannot be participate in characte,

Boolean

* Wrappers class → Immutable✓

Note:-

⇒ In wrapper class, `toString()` is overridden to print the data.

⇒ In wrapper class, `equals()` is overridden to check the content.

⇒ Just like String class, Wrapper classes are also treated as "Immutable classes".

Can we make our user-defined class as immutable?
⇒ Yes. possible ✓.

Need of wrapper classes: To store primitive data
in the form of objects.

Number \Rightarrow [parent] { Byte, Short, Integer, Long,
float, Double }.

Object \Rightarrow [parent]
{ Boolean, Character }.

* Object:
* `toString()` return hashcode of object

`equals()` \Rightarrow Compare the reference

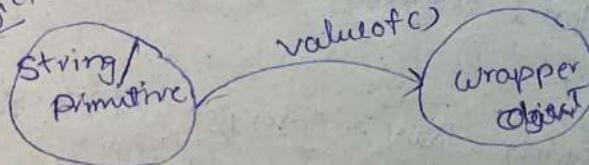
String \Rightarrow `equals()` \Rightarrow Compare content
StringBuffer/Builder \Rightarrow `equals()` \Rightarrow Compare
reference.

Wrapper class:

`toString()` \Rightarrow Content of the object point
to

`equals()` \Rightarrow Compare the data content

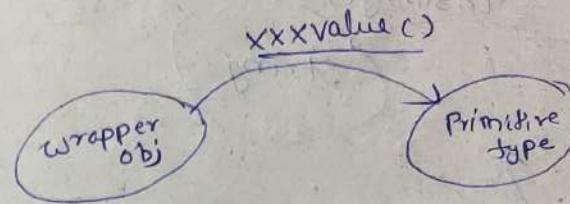
Valueof():



To convert String primitive to wrapper class we
need Valueof method ()

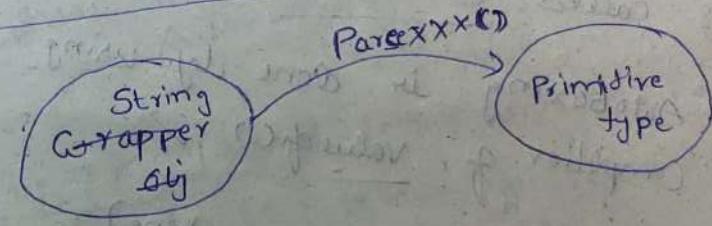
Ex- `Integer i = Integer.valueOf("10")`
`s.o.println(i);`
 $\Rightarrow 10$

\hookrightarrow when String signature, it could be
number format exception \hookrightarrow `Ex- Integer.valueOf("ten")`



xxxvalue() \rightarrow which wrapper obj to primitive type
 \hookrightarrow which is not applicable for character &
Boolean.

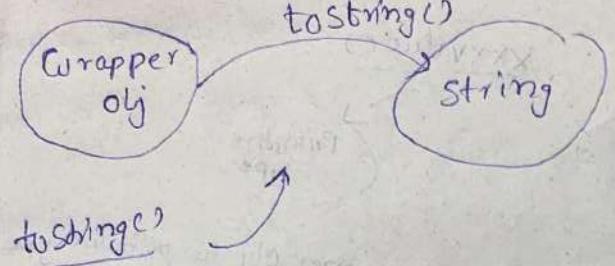
\rightarrow `Integer i = new Integer(130);`
`System.out.println(i.byteValue());`
 $\Rightarrow -126 \checkmark$



ParseXXX (): String \rightarrow Primitive type

\rightarrow `int i = Integer.parseInt("10");`
`s.o.println(i);`
 $\Rightarrow 10$

\hookrightarrow Here also when String signature it could
be number format exception



- * static factory method ✓
- * instance factory method ✓

Autoboxing → Automatic Conversion of primitive type to wrapper object by the compiler is called "AutoBoxing".

Note:- Autoboxing is done by using Compiler of valueOf()

Ex:- Integer i1 = 10; (primitive type)
↓ (After compilation)

Integer i1 = Integer.valueOf(10); (wrapper obj)

Autounboxing:- Automatic conversion of wrapper obj to primitive type by the compiler is called Autounboxing.

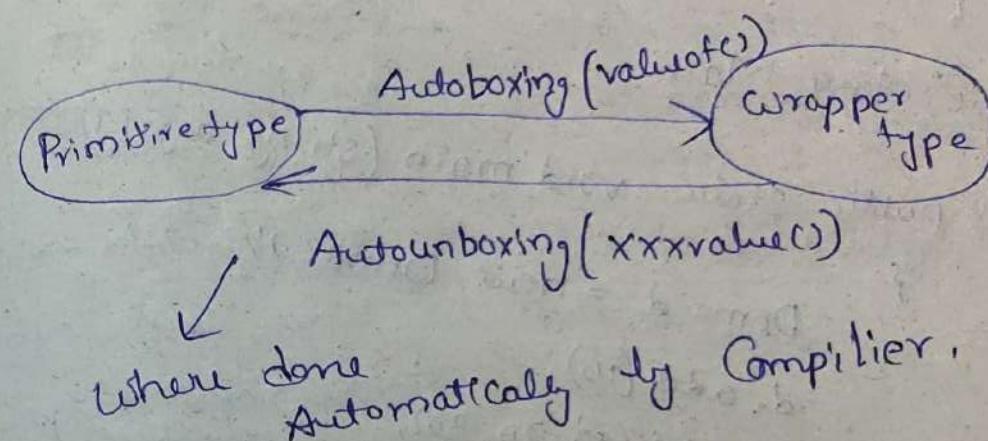
Ex:-

```

Integer i1 = new Integer(10);
int i2 = i1 (wrapper obj)
      ↓
int i2 = i1.intValue(); (Primitive type)

```

* Wrapper → primitive ⇒ xxxvalue()



* int i1 → 0 {default because primitive type}
* integer i1 → null {because of reference type}

* for null there is no Method take place then it could be nullpointer exception

* ALL wrapper classes are Immutable

* Buffer Concept range ⇒ -128 to +127 ✓

Class Demo

```
{  
    public void add (int... x)  
    {  
        System.out.print ("var-args");  
    }  
}
```

public class Main

```
{  
    public static void main (String args[])  
    {  
        Demo d = new Demo();  
        d.add (10);  
        d.add (10, 20, 30);  
    }  
}
```

Var-arg Method:

The Method which can take no. of arguments
in case you write ~~one~~ in one Method

like add (int... x)

↳ Eclipse

→ Can we mix normal argument & var argument → Yes
→ add (int x, int... y)

Note:-

methodOne (int... x)

→ we can call this method by passing a group of int values, so it becomes 1D Array

refers var-arg [lanch04]

methodOne (int[]... x)

→ we can call this method by passing a group of 1D int[], so it becomes 2D Array.

NOTE: Child reference can be collected by Parent type which we call as

Implicit Type Casting

NOTE: Widening followed by Autoboxing X
Autoboxing followed by widening ✓

new Vs newInstance() :-

↳ refer Eclipse IDE ✓

Without creating
new object
while we write super

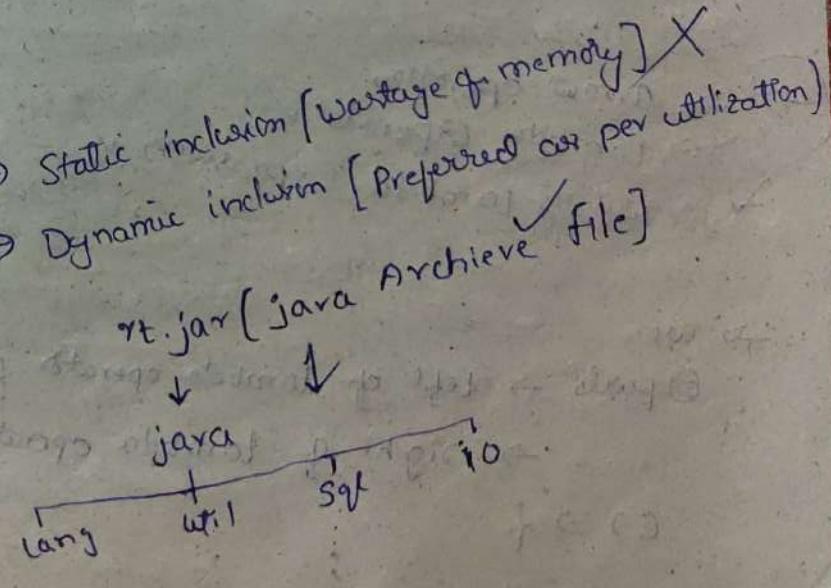
↓

While writing different
exceptions take place

ClassNotFoundException
 ClassNotFoundException
 → new once checked
 → new
 → load .class file automatically and search in Current work directory
 → Don't load .class file automatically and manually search load by Class.forName()
 Then searching happen within the directory
 → When search required .class file not found then ClassNotFoundException
 → The purpose of NewInstance for loaded class create an obj.
 → When it is creating obj it expect 0 args otherwise it InstantiationException
 → When constructor is private

import java.util.ArrayList // informing ArrayList class in [java.util]
 Main
 {
 Student obj = new Student();
 for this
 ✓ ArrayList arr = new ArrayList();
 Without having className we have to be
 Create a object by ArrayList

② import statements
 → Explicit class import (Ex: import java.util.ArrayList)
 → implicit class import (Ex: import java.util.*;)
 ↓
 C/C++ → Static inclusion [wastage of memory] X
 Java → Dynamic inclusion [Preferred or per utilization]



* If a class contain every Method static
that could be called as Utility Method)

Ambiguity problem by static ✓

only

* One abstract Method in interface known as
"functional interface".

⇒ By Combination of
we write Lambda
expression.

@functional interface → Eratation (which is
interface Demo
{ void disp();
}

→ refer Eclipse ✓

→ Arrow operation
→ Lambda Operator ✓
→ for writing Lambda Expression ✓

→ we

② parts → left of lambda operator parameters
right a lambda operator body

* In Lambda expression Ignore { } X
when having one statement

Ex: C) → ; ✓

C) → { ; } ✓

* to write Lambda exp we [C] → ✓
→ Ignore int type in parameters like (a,b) ✓
→ Ignore parenthesis () when having only one
Parameter. ✓
→ Ignore return type are also ✓
→ for single statement remove { } X

* Static methods participate in inheritance ✓
Method hiding

↓
which we write static method in Inheritance
it looks like don't override it and

Exception Handling

- * abstract and final Illegal expressions
- abstract means incomplete and final means we can't override and we can't inherit.
- Then abstract and final no one can implementation then what purpose?
- * final class in inheritance X
- * final method in inheritance yes but we can't override.
- * we can't create obj for abstract class.

Exception handling:

→ Exception refers to a mistake that will occur during runtime of application which will result in abnormal termination of application which a developer needs to handle.

Exception handling:-

→ It means a process of handling exception in such a manner because of which abnormal termination of application could not happen.

try, catch, throws, throw, finally --
↳ key words used in exception handling.

* where particular exception takes place in different methods, Main method that method static area will create exception object.

↓
when it takes place JVM comes to picture by default exception handle.

Unchecked exception: If compiler is not checking whether exception takes place or not. ✓
during compile time

How to handle exception?

⇒ try {

} → Enclose risky statements

Catch (Inbuilt classes)

{ ; }

} handle try ✓

↓

→ write the statement which is occurred due to exception.

When exception only catch block executes ✓

* If exception one of line or try block --- then below lines in between try block & catch block will not execute.

Catch (ArithmeticException ex)

```
{
    // When takes place for denominator with zero
}
```

Catch (NegativeArraySizeException ex)

```
{
    // When takes place negative size
}
```

Catch (ArrayIndexOutOfBoundsException ex)

```
{
    // When out of Boundary takes place
}
```

* which are specific exceptions for different exceptions to catch ✓

* In Java single try block multiple catch blocks are allowed ✓

Catch (Exception e)

```
{
    System.out.println("Wrong Input.");
}
```

Generic Exception (Parent)

not must ✓

↓ Below we have to write

* Separate try Catch is allowed ✓

Whenever there is an exception ?

→ Handle Exception (try { } - catch { })

→ Duck the exception (throws)

→ Re-throwing an exception (throws, throw, finally). ✓

* whenever not handling and after handling is method is throwing exception to caller in such case we write

Ex:- void alpha() throws ArithmeticException

unchecked :- Compiler will not force it checked during compile time.

checked :- It will be checked by compiler.

* user duck the exception of checked exception.

* Ducking & handling

* whenever this exception will never occur go for Ducking ✓

* when handler present any method, Again it not go for calling Main Method. ✓

Similarly, when there is no handler in method, then it go for Main Method ✓

Re-throwing Exception:- Throwing already handled exception obj to the caller (Main) \Rightarrow throw keyword \rightarrow use of re-throwing critics in catch block.

* Lines below throw will not be executed.

for avoiding to execute below throw keyword we write in finally {

}

* throws:
↳ whenever excepted obj promoted to the other stack, then in Method signature atleast one throw whatever exception
* \rightarrow when unchecked (throws).

* throw:
↳ Excepted obj finally

handled object throw ✓

throw
 \Rightarrow write in catch block

\Rightarrow Statement below throw not executed

\rightarrow rethrow

throws

\Rightarrow write in Method signature

\rightarrow here executed

\Rightarrow Duck & Method signature

Methods to print exception information:- (Exception e)
 \rightarrow e.getMessage() :- prints description of msg. Ex:- by zero
 \rightarrow e.toString() :- prints name and description of msg.
Ex:- ArithmeticException / by zero.

\rightarrow e.printStackTrace() :- prints name and description of exception along with stack trace.

Ex:- ArithmeticException : / by zero at Demo.alpha().

finally :- * If no exception ✓
* If exception & catch match ✓

* write return before statements not executed like throw only

↳ Avoiding by finally
finally also write along with try & catch only ✓

* At any critical statements want to execute go for finally ✓

* finally dominate return, throw ✓

↳ will throw compulsorily there could

Nesting is allowed in try - catch blocks ✓

try

{

try

{

catch

{

}

Catch

{

}

} allowed ✓

Nesting any
no. of times
allowed ↗

* Below the catch block until the handler
it will not execute ✓

* Without try - catch block there is no finally ✓.

* Two try blocks and one catch Block X

* One try block and Multiple Catch blocks ✓

* try and finally are allowed ✓

* catch and finally are not allowed ✓

* object to exception is "Throwable" ✓

Parent

by try catch

throws :- handle the exception and throw it back
the exception object to the caller.

throws :- Method signature and commonly used
for "checked exception"

Synchronous Exceptions

↳ occurs at specific program statement ✓

Asynchronous Exceptions

↳ occurs anywhere in the program ✓

Runtime Exceptions

↳ Out of Memory ✓

↳ StackOverflow ✓

Exceptions

↳ Arithmetic ✓

↳ ArrayIndexOutOfBoundsException ✓

↳ Negative ✓

Custom Exception :-

↳ our own exception ✓

Resource :- An object which we create used in
code and we perform the operation ✓

try-with-resource :-

→ In this automatically ~~closed~~ at the end of try
it closed normally & abnormaly

→ which it is not required to write finally block

→ so length of code would be reduced &
readability improved ✓

→ All resources are said to be AutoCloseable
resources if they ~~close~~ implements an

interface called "java.lang.AutoCloseable".

→ directly or indirectly.

→ JVM will call close() ✓

* When we write try without finally - by default compiler will check for the checked exception - Compiler will check for the handling code only then compilation is successful.

* Nesting with resources is allowed ✓

② Try with multi-catch block :- // for better understanding of multiple try & catch block given code snippet

Ex:-

```
try{  
    ...  
}  
catch(ArithmaticException e){  
    ...  
}  
catch(NullPointerException e){  
    ...  
}
```

Catch (Arithmatic Exception) | NullPointerException

In this place not could be
Parent - Child X

Catch (Arithmatic Exception) | NullPointerException

Catch (Arithmatic Exception) | NullPointerException

Catch (Arithmatic Exception) | NullPointerException

No limit what we have to write in
the parent - child

* eg:- 10Exception , SQLException.

uncheckedException:- Compiler will not check for handling code , and JVM will come into picture.

eg:- RunTimeException and its child classes Error and its child classes

Rule of overriding catch exception :-
Inherited exception, child overriding the class method throws, then compiler should throw the parent class exception

Some checked exception

Ex:-

public void main() throws IOException
{
 ...
}

↳ If we want to ignore exception hierarchy
for which

Ignore unhandled exception

instanceof [refer Eclipse]

- * Runtime object what type of it we can check it by instanceof ✓ (at beginning it is available)

(Ex:- public class Main

{ main method

```

    {
        Thread t = new Thread();
        System.out.println(t instanceof Object); // True
        System.out.println(t instanceof Thread); // True
        System.out.println(t instanceof Runnable); // True
    }
}

```

↳ refer to same object by instance

instanceof

- * isinstanceof() is a method, to checked whether the given object is particular not, we don't know at the type of beginning it is available.

↳ load class file

not known

↓ known from beginning

```

    {
        System.out.println(Class.forName(args[0]).isinstanceof());
    }
}

```

instanceof

what type of it we can check it by instanceof ✓ (at beginning it is available)

Multi-threading :-

Multitasking

↳ process Based Multitasking ✓
↳ Thread Based Multitasking

① process Based Multitasking:-

* which can execute several tasks simultaneously known as process Based Multitasking

* At "OS level" ✓

② Thread Based Multitasking:-

* Independent line of execution is controlled by "JVM" ✓

↳ several tasks simultaneously chew

* Executing several tasks independently each task is a separate independent part of the same program is called Thread based Multitasking.

* Every thread is under controlled by JVM :-

- * Each independent part in "Thread"
- * Multi-threading will reduce response time of application to improve performance

The main purpose of multitasking is to utilize the CPU time effectively, which can be used to improve or code

* Java provides built support to work with threads through API called Thread.

Runnables, Thread Group, ThreadLocal —

Threadscheduler

* JVM controls those threads by support of

Thread scheduler ✓

* If multiple threads are waiting, Threadscheduler decides which will first execute.

Multithreading:-

* While writing code, in class must be. Given inbuilt API (i.e. Thread), the task which could be write under sum method, (sum()) ✓, but Main Method after creating obj, we call Start method,

Start method,

(Start())
↳ Refer Eclipse ✓

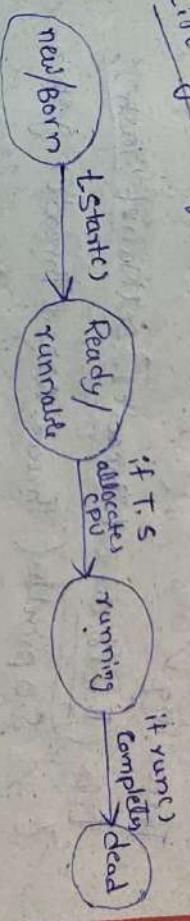
* When we call start(), Two threads are executed one in main thread & one in thread ✓

* First prefer Main thread to execute

first ✓

when we are using two args in main method of t.start()
t.start() → It results in runtime error → SegFault

Life cycle of Thread:-



* Comeback is not possible like

In Main method,
In {
obj.start();
}
Obj.start();

↳ Which gives compiler error
IllegalStateException ✓

}

↳ Refer Eclipse ✓

* Why implementing Runnable is good approach?

Because, the inheritance concept is directly taken place in implementing Runnable interface.

When class extending is not possible X

run method ✓

↳ Refer Eclipse

start method calls call sum method, where could be having (0 args) or run method it could be having (0 args) or run method call by explicitly it done ✓

In case we call by explicitly it done ✓

Volatile

* Is there any guarantee $t.setPriority()$ will change $t.getPriority()$

- * Thread t = new Thread c)
- ✓ Thread t = new Thread(Runnable r)
- ✓ Thread t = new Thread (String name)
- ✓ Thread t = new Thread (Runnable r, String name)

- * Setters and getters are allowed in.

Threading ✓ \checkmark

like Thread.currentThread().setName("Yash")

S.o.println(Thread.currentThread().getName())

↳ when there is no reference of a

Thread ✓

↳ Thread hierarchy

Runnable (T)

Obj(c)

1

↳ Thread (c)

Thread priority :-

- * Default Thread priority is always 5 ✓
 - * Based Thread priority is based on Priority, which will used for giving Precendence.
 - * Particular Thread priority
 - Min- Priority (1) ✓
 - Max- Priority (10) ✓
 - Normal Priority (5) ✓
- ↳ give 1, 5, 10 etc or to JVM, but when we give 100, 50 → JVM sudden like → IllegalArgumentExpection
- ↳ Anywhere you write anywhere you have to write ↳ In child thread ↳ to write ↳

- * Thread.yield() ↳ In child thread ↳ to write ↳
- ↳ pause your execution sometime and check any other thread waiting and those thread priority and your priority will same, then you will give chance. ✓ (refer eclipse)

* Thread.join() (d) join(c) ✓

↳ If the Thread has to wait until the other thread finished its execution then

we need to go for join() (refer eclipse)

→ join() → pause time like 1000 millisecond

→ but join() → pause time like 2000 millisecond

↳ when program goes for InterruptedException ↳ opposite deadlock

* Infini wait leads to Deadlock never want to take place in Multithreading ↳

* Synchronized is cause for Deadlock, so whenever we using synchronized block

Special case.

Thread sleep :-

↳ The method which is used to wait for some time and not depend on other thread.

Some time and not want to perform exception. (Any operation amount of time)

thread

↳ Thread.sleep() giving particular time.

Interrupted Exception :-

↳ When one thread is sleep, we can interrupt other thread.

Both these cases will occur InterruptedException.

↳ Signature

public void interrupt()

* we can write by lambda expression (λ).

Anonymous class

Synchronized :-

↳ only one thread can synchronize in a

↳ when you mention synchronized until that method, no thread will enter until that thread.

↳ execution takes place of particular thread,

↳ which affects increasing waiting time,

and also performance in low memory

↳ Remember string buffer, (synchronized)

↳ String Builder(§ 5)

In this

* Internally synchronization concept is implemented by using lock concept and also never recommended to use

↳ synchronized keyword.

↳ class level

↳ object level

* In Java there are two locks, we have

↳ Class Lock :- when in method go for static synchronized so for

to use class lock.

↳ Object Lock :- when it could be in object level

↳ synchronized → object class.

↳ These both are different and there is no relationship.

Synchronized block :-

↳ class, where we mention

↳ If this is inside Method, then all threads,

↳ synchronize then it will not all threads, that

↳ by avoiding these where problem occur, that

↳ region you will put synchronized.

Mention where

Ex:- void m1()

{ . . . ; }

↳ Problem occurred

↳ Synchronized block → which will not use synchronized block which will not entire lock thread

Synchronized method

- * When you mention synchronized in a method, synchronized will execute no thread will execute until current thread exits to other place call obj level lock

- * Realtime Application rarely used X
- * Realtime Application used ✓

So only thread can access this region on its critical.

Wait

synchronized(this)

- * whichever thread wants to get the resource/information from another thread, that thread should call "wait()"
- * whenever thread updates the resource/information to other thread, that thread should call "notify() / notifyAll()"

wait(), notify() / notifyAll() all present in object

private void synchronized(lock) {
 ...
}

Note: lock concept applies only for objects and class types, but not for primitive type, which will end up unexpected type

e.g:-

```
int x = 10;
```

Synchronized(x) → unexpected type

Integer x = 10;
Synchronized(x) → warning it is a wrapper class.

InterThread Communication (refer eclipse) ✓

- ↳ Two threads can communicate each other with obj.wait(), obj.wait(100), obj.wait(100, 10)

- ↳ wait() ✓
- ↳ notify() ✓
- ↳ notifyAll() ✓

- * If we mention synchronized inside a particular action takes place first

- * Realtime Application rarely used X
- * Realtime Application used ✓

So only thread can access this region on its critical.

Wait

synchronized(this)

- * whichever thread wants to get the resource/information from another thread, that thread should call "wait()"
- * whenever thread updates the resource/information to other thread, that thread should call "notify() / notifyAll()"

wait(), notify() / notifyAll() all present in object

private void synchronized(lock) {
 ...
}

Note: lock concept applies only for objects and class types, but not for primitive type, which will end up unexpected type

→ Thread will call any type of objects [because if obj type is parent of any class]

→ If a thread wants to call wait(), notify(), notifyAll() then compulsory the thread should be owner of object, [that thread should lock object or synchronized method, otherwise result in illegalMonitorStateException ✓]

notify() → If there are multiple threads, then notification sent to only one thread, for which thread, Thread Scheduler will decide.

notifyAll() → When we using notifyAll(), the notification is given to all the waiting threads, and it will executed one by one. And also we don't know which thread will notified first because of Thread Scheduler.

Garbage Collector Thread ✓ Based on Priority changes

↳ Supporting thread
↳ Technically called an "Daemon Thread".

↓
Similar Synchronized(this)

↓
class Demo
{
 public void main()
 {
 synchronized(this){
 System.out.println("Accessing, if there
 is arguments or, though
 it no go access through
 this key word");
 }
 }
}

↓
Demo obj = new Demo();
obj.main();
↳ when not passing any
arguments
↳ this

Exception Core Java Revision

* In main method, we print something, that print method of a printStream class

* In public class 'Name', that file class name or same as file name because when you mentioning public, anyone can access it. Because of public, anyone can access it. If you make inner class as private, just not outer class.

* When you mention private to Instance variable.

* The variables, inside method → Local variable. The variables, outside method & Inside class → Instance variables.

* When you mention private to Instance variable you can access by creating methods that one method is set the value and one method get the value. (getters & setters)

* No return type, while creating a construction.
↳ don't be return type
↳ public calc(int m, int n)
↳ where passing
↳ num1 = m;
↳ num2 = n;
↳ arguments
↳ result
↳ Main method obj = new calc(5, 6);

In instance

```
int num1;  
int num2;
```

public calc(int num1, int num2)

{

```
this.num1 = num1;  
this.num2 = num2;
```

}

while writing this keyword it will
works on current object.

* Java does not support multiple Inheritance ✓

A < B < C ✓
A B C
C J X

* It one method to be abstract and class
should be also abstract ✓

* Default value for int → 0 ✓
String → null ✓
Boolean → false ✓

* Inside Interface, we can write different
methods static, private methods, different
methods in later version of jdk.

TypeSafety ✓

* In Java, Arrays →
Can't be store Heterogeneous Data ✓
→ store only Homogeneous Data

Difference b/w Interface & class?
→ Interface means a SRS [Software Requirement Specification]
Whereas class is Prototype to know to
implement a object.

Collection :- At [1-2 address] / Collection API / framework
Present in [its package] [No TypeSafety]
it known as classes :-

- 1) Requirement of storing large Data ✓
↳ ArrayList [Dynamic Array D.S.]
- 2) LinkedList [Doubly Linked D.S.]
- 3) Array Queue [Double End Que D.S.]
- 4) Array Deque [Double End D.S.]
- 5) Priority Que [min heap D.S.]
- 6) TreeSet [Binary Search Tree D.S.]
- 7) Hashset [Hashing D.S.]
- 8) Linked Hashset → only list

↳ existing memory is growing ✓

1) ArrayList :- [Refer Eclipse] → internally Dynamic Array D.S. [Not fixed size].

↳ Internally Dynamic Array D.S.

* In ArrayList, we can store both Homogenous
data & Heterogeneous data ✓ [All data types allowed]

* In Collection, if you add any data that

could be stored on object

* we can add different numbers by giving
their index position [like front, middle, rear]
But front, middle not recommended because not
efficient.

Collection

* index Band occurring allowed in ArrayList

Contains Method

* Add method

* indexof method

* linkedlist { Both, list Interface & Deque Interface }

ref Eclipse

L) It follows doubly linked list Data Structure

* Homogenous Data ✓ & Heterogeneous Data ✓

* stored as object ✓

* No shifting of data ✓

* which one methods in ArrayList, In linkedlist
are also there & and also add first() &
add last() ✓

* index Band occurring Allowed ✓

In linkedlist

* As compare to ArrayList, In linkedlist

that addfirst()

they present two methods that
addlast() ✓

* As compare to ArrayList, linkedlist is more

efficient ✓ In ArrayList & linked list duplication allowed ✓

ArrayList

* Polist method → same like above but

the element no more in collection.

* It affected element ✓

1) when size off Array
known

Not required methods or
it is not required

2) only Homogenous
It allows both homogenous
& heterogeneous data ✓

3) It is faster than.
It is slower

* Refer Different Inbuilt methods in
Eclipse IDE.

ArrayList
ArrayList
for word, for more
than 1000 words
then it is better to
use arraylist

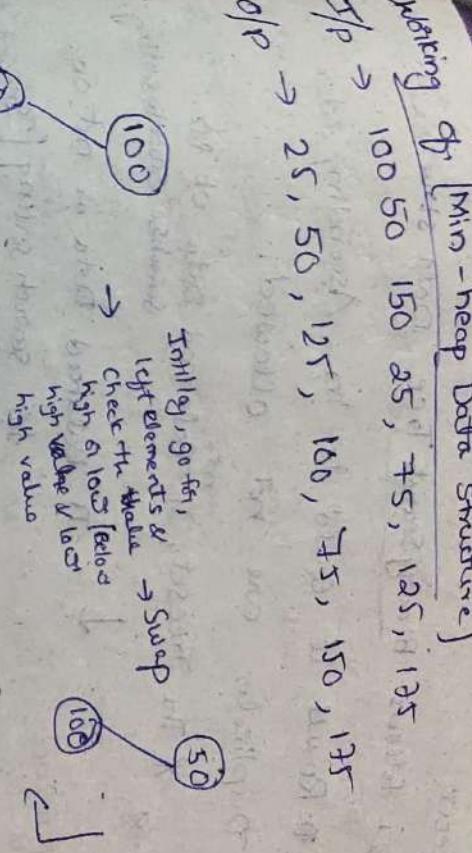
Array Deque: (double ended que D.S.)

- * Duplicate Allowed ✓
- * Insertion & deletion at front end & rear end is allowed ✓

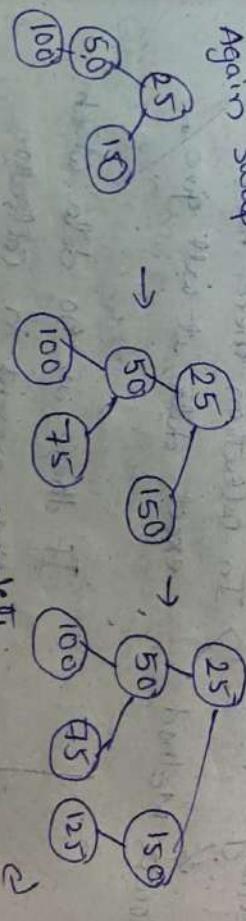
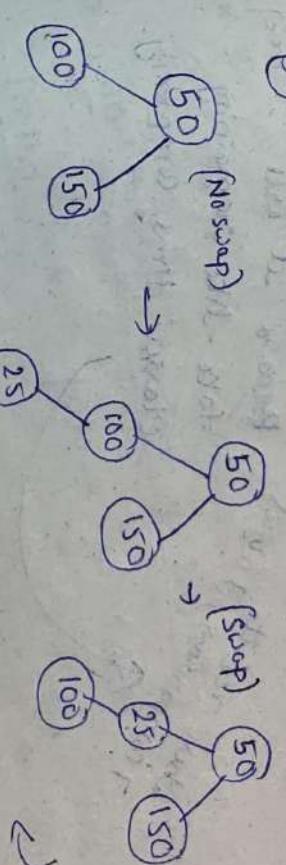
- * Index Base occurring in not allowed
- * implementing Deque Interface ✓

Priority Queue

- * It follows [min-heap D.S.], It implements Queue Interface.
- * Index Base occurring is not allowed ✓
- * Duplicates are allowed ✓
- * Heterogeneous not allowed ✓



Again swap



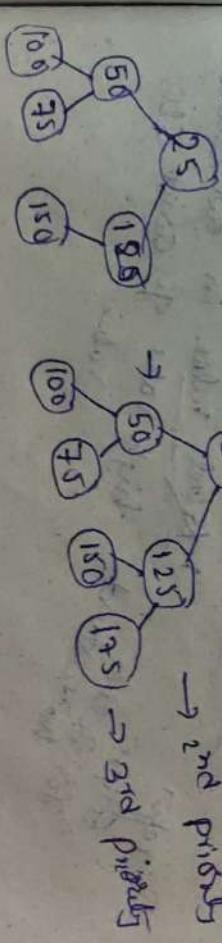
Swap →

complete

1st priority

→ 2nd priority

→ 3rd priority



→ 25, 50, 125, 100, 75, 150, 175

Treeset:-

* It follows Binary Search Tree Data structure.

* Result it could be in Ascending order.

* Duplicates are not allowed.

* The Treeset, If ^{not} balanced Data ok for ^{Search & Search sorting}.

↓
If balanced Data is not ok

for search setting [skewed tree]

results will be same it will take like remaining
comparisons
↓
lower time complexity.

Ceiling method

&
lower method

In collection there have

exact data it will give.

If there is no data which

is present in collection,
it will result to give

refer value in collection
for better and. high value in collection

Hashset:-

* It follows fit Hashing algorithm.

* It is a combination. Hash function & Hash table

* Index Board according not Allowed.

* No Order of insertion.

linked hashset:-

* It is also follows same as like hashing algorithm.

linked hashset follows the order &

insertion

Q) If you want order Insertion & hashing.
Algorithm which class you go?
A → linkedhashset

* collections which store in the form of objects and follow Autoboxing.

* Common method in all class.

Iterator method

→ with cls list Iterator obj = obj.iterator()

→ Descending Iterator

→ Present in linkedlist / ArrayList / TreeSet

while travelling, we want to fetch the data go for

Iterator

In this,

there are next method → accu^{ing} the data

next Method

→ next data

* ListIterator is only present in ArrayList &

linkedlist only.

* ListIterator → which mean you want to fetch data in reverse direction.

from previous Method → when reverse data taken place.

Review Method

Iterator: {3 types} → return data

→ Iterator → It present in All class
→ List Iterator → It present only ArrayList & linkedlist

→ Descending Iterator

→ Present in linkedlist / ArrayList / TreeSet

In this method
present in
linkedlist
not required
size

[By default it start from
beginning]. Backwards

Before, 7 classes of collection, there could be one class known as "Vector". In this Enumeration which acts like Iterator. And this Enumeration only in Vector class

Iteration(I)

Collection(I)

Map(I)

Set(I)

Queue(I)

Stack(I)

Vector(I)

Enumeration(I)

Vector implement Collection interface to in loose coupling

~~* fail fast~~ - when Concurrent, structural model

↳ suddenly fails for own program.

↳ exception thrown
Concurrent & Structural modification is while Accessing data

↳ your modifications in collection.

✓ refer Eclipse

↳ fail fast

↳ fail safe

↳ fail safe not fail ✓

↳ fail safe is concurrent & structural

↳ when doing modification (if not occur Exception)

* fail safe by CopyOnWriteArrayList

↳ Under Concurrent package with out

Under Concurrent package with out

Collection: * It is a interface and it is a framework. ↗ hierarchy and

↳ Collection it is a hierarchy and interface ↗

↳ it is a set. ↗

↳ Collection ↗

↳ Collection ↗

(VS) Collections: - It is a individual class

↳ Collections ↗

* In ArrayList, we want to sort by using

collections class

↳ Collections.sort(objs);

↳ The use for simple static like only same type.

↳ when we use for complex objects for direct

when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

↳ when we use for complex objects for direct

* In collections they having add method ✓

But in Hashmap it could be put method ✓

* Similarly, In HashSet it not maintain order
of insertion, by avoiding then we will maintain
hashset which is subset of HashSet.

(1.2) * When we want to maintain order
of insertion, by avoiding

Similarly, for HashMap it is not maintain
order insertion, we will use LinkedHashMap

* When we want to maintain order of value form,
simply go Map of HashMap ✓

S.o.println(obj1); // Hashcode / address of obj

s.o.println(obj1.ref) exact value

) sometime this output & sometime below output
for object reference.

Why?

{ for avoid result in hashcode, by
overriding toString method, ex: public String
toString()

* when we printing object reference because
of "toString Method", when Inherit Method

override toString Method it will give exact value
override toString Method

Demo:

obj1 ✓

↓
values

Collection c = obj1.values();

Iterator Itv1 = c.iterator(),
while (Itv1.hasNext())

{ Itv1.next(); } ←

when using
Iterator in
Maps before
Iteratable value
Iterable could be
step could be
because value return type is
collection, iterator
that be there
go to iterator
for collection, iterator
be same

⇒ output only values ✓

→ output

for key set s = obj1.keySet();
Iterator Itv2 = s.iterator();

while { ←

Then, 3

same

Set es = obj1.entrySet();

Iterator Iter3 = es.iterator();

where
& is
whole { ←

Same

When you suddenly want to terminate your

Program, then System.exit();

while in every program, the Garbage collector is present, while Garbage collection takes internally help of finalize Method ✓.

But, HashMap dominating Garbage Collector

↓ invoke ↓

In this finalize not By Avoid this we

WeakHashMap ✓ instead of HashMap

↓ finalize Method in
In this invoke ✓

Difference b/w HashMap / WeakHashMap

→ In HashMap WeakHashMap

→ Not allowed → It allowed
finalize Method

→ No Garbage Collection taken place

↓

↓ That the reason why key values are

unique ✓

Hashtable obj = new Hashtable()

b) It is legacy class [it means it from beginning i.e.]

b) HashMap & Hashtable are same .

b) But HashMap → 1.2v

Hashtable → 1.0v [legacy class]

b) In Hashtable all methods are "synchronized"

But HashMap all methods "not synchronized"

* Synchronized [Multithreading not possible]

* non-synchronized [Multithreading is possible]

* Hashtable → Thread safe

HashMap → Not Thread Safe

* Hashtable → low performance ✓

HashMap → high performance ✓

* TreeMap which looks TreeSet [Sorting]

In TreeMap, best sorting is Based

on key values ✓

Finalizer → HashMap ✓ (overridden equals() , hashCode()) ✓

GENERICS

Type safety] and type casting in

* In Collection not required, then use Generics feature

available from Jdk 1.5 ✓

* Generics deal the data through API at

* Generics available at Compiler level.

Note only for ~~String~~ type of data →

* Type Safety means [Data be homogenous,

so type casting problem won't come]

enum :-

→ jdk 1.5 ✓

→ It is a group of named Constants (♂)

→ Predefined Constant [constant means no change

[value]

→ use enum keyword ✓

enum class with constants in

→ Inside enum class write constants in

UPPER CASE ✓

→ where that constants are static final ✓

→ own data type or particular Constant

→ own data type or particular Constant

Ex:- enum NORTH, SOUTH, EAST, WEST

MUST be in Capital S

→ // static final ✓

List < String > obj1 = new ArrayList<String>();

→ Allowed & Polymorphism to be child to parent ✓

At parametr is allowed ✓

ArrayList < Obj > obj1 = new ArrayList<String>();

↓ child.

↓ parent

→ Not allowed known in Type Parameter not

Allowed for polymorphism

→ extra information that we are embedding

on code it will reach till compiler

JVM ✓

But // comments are not reached that particular stage

Annotation → Annotation → parent of all Annotations

→ Built in

→ Custom (our own)

→ @ functional Interface ✓

→ @ Override ✓

Apply the Annotation for

→ class ✓

→ interface ✓

→ method ✓

→ fields (instance var) ✓

→ local fields ✓

→ Parameters ✓ → As a developer you know which annotations

enum ✓

class to use ✓

→ when we own custom

new annotations we have

3 'you can make', → String get (int index)
ArrayList <String> al = new ArrayList <String>();
al.add ("Sachin") → String.add ("Sachin");
al.add (new Integer(10)); →

String.add (new Integer(10));

→ Compile Error ✓

→ No Type Casting in Generics ✓

→ Type Casting allowed in ArrayList ✓

→ In ArrayList → It is also Generic class

In Generics → Type Safety ✓

→ In Generics → have Type parameter ✓

→ have Base parameter ✓

→ we can also write user defined Generics

class ✓

→ To get the solution

```
class Demo < T extends Number >
```

When we control Type parameter always
we extends by word.

In case instance we also we
we extends only ✓

Here we also
only we extend
only.

class Demo < T extends Runnable > implements

new HashMap< Integer, String> h =

```
h.put(27, "Taj");
```

class Demo < T extends Number, Runnable > / Invalid
→ class Demo < T extends Number, Runnable > / Valid

→ [Class & Interface ok ✓]

→ class Demo < T extends Number, String > / Invalid

→ [Class & Class]

→ class Demo < T extends Runnable, Comparable > / Invalid

I

T

→ [Interface & Interface ✓]

→ In Type parameter can we achieve polymorphism ✓
→ In Box parameter we also achieve
Polymorphism.

→ At Type parameter → No primitive ✓

→ valid

→ class Demo < T extends X > X Not valid

→ class Demo < T implements X > X Not valid .

→ class Demo < T super X > X Not valid .

→ In Method < & super > ✓ valid ✓

→ where only we extends only

Above X → class, X type on its child type

X → interface, X type on its implementation

✓ class type ✓

✓ known as Bounded type (means Control)

* Key points about Bounded types

→ As we can pass any no. of type
parameter need not to be one.

e.g. HashMap< Integer, String > h =

```
new HashMap< Integer, String >;
```

→ class Demo < T extends Number, Runnable > / Invalid

→ class Demo < T extends Number, Runnable > / Valid

→ [Class & Interface ok ✓]

→ class Demo < T extends Number, String > / Invalid

→ [Class & Class]

→ class Demo < T extends Runnable, Comparable > / Invalid

I

T

→ [Interface & Interface ✓]

→ class Demo < T extends Runnable, Number >

→ class Demo < T extends Runnable > I C X

I

C

X

→ [Interface & Class] ✓

→ class Demo < T extends Runnable, Comparable > / Invalid

I

C

X

→ [Interface & Interface ✓]

* Generic class → Normal class with Type parameter
Can you Type parameter at Method level ?

→ Yes, it is possible .

* At method level, we use super to control type parameter.

e.g.: ma (ArrayList< super > al)

Allowed during method level.

①

methodOne (ArrayList< String > al) → Method

al.add ("Teja"); // valid

al.add ("Kiran"); // valid

② methodTwo (ArrayList< ? > al)

al.add("Teja"); // Not valid

al.add (10); // Not valid

al.add (null); // Not valid

→ Because ? is in left side only not right side ✓

→ Generic will see only LHS No R.H.S

e.g.: ArrayList_{obj} = new ArrayList< Integer>;

obj.add (22);

obj.add (2.32);

Valid because compiler see only this (L.H.S)

ref: obj.add (2.32);

clips ↴

③ Generics Concept is available to applicable at compiler level and not in runtime (JVM)

ArrayList<? extends Number>();

↳ Invalid

↳ Because ? is in left side only not right side ✓

→ And also this method is useful whenever we are performing read operation

→ Because default could be null ✓

→ And also this method is useful whenever we are performing read operation

→ suited for only read operation. ✓

al.add(null); → valid ✓

(Bounded)

Generics [1.5 v]

→ for collections to Type safety and avoid Type casting b/w

→ Generic applicable at Compiler level not JVM level.

→ When we writing Generic, loose Coupling
is promote at only parameter not type param

→ Not applicable for superior or primitive type ✓
e.g. int Integer

→ We can also bound control of parameter ✓
T extends X.

Generic class Syntax <T extends X>

X → class → X type & its child type ✓

X → interface → X interface type & implementation its class

↳ interface → X interface having Type parameter.

→ Generic class mean Normal class having Type parameter.

→ ⑥ Method → refer Back Page.

↳ one method bounded More Parameter. Yes ✓

Comparable interface

// Natural sorting order

→ It is a functional interface Present in

java.lang package.

→ The interface usually used by TreeSet obj

during sorting process. And Method used

→ package

④ functional Interface Comparable < T > abstract

public interface java.lang. Comparable< T >

{

In side TreeSet obj, the data should be homogeneous → it uses Comparable to set the object

✓) The Object should compulsorily implements
on interface called "Comparable".

Jt said, result in classCastException → it works

NOTE: All wrapper classes and string classes
have implemented "Comparable" interface.

→ StringBuffer class not implemented Comparable interface ✓

Comparable interface ✓ In default alt Comparable()

* Binary Tree set im default super 21/12/22 [date] class

↳ By default it sorting to left Ascending

↳ By different setting to left Ascending.

↳ By different for strings & numbers.

↳ If customized setting

Comparable : if developer's

interface is used by developer with

their interface for used to be sorted inside

to make obj can to keep objects inside

Descending order. public element & abstract interface to implement

TreeSet.

→ Not functional interface.

→ Not balanced Binary tree set on that object

e.g.

class

introduce Comparable

```
public int compareTo(Object obj);
```

↳ Obj1.compareTo(Obj2) → returns

in details

2/12/2012 video

Introduce Comparable

1. public int compareTo(Object obj1, Object obj2);

Note: Integer i1 = (Integer) obj1; // After this
Integer i2 = (Integer) obj2; // stops char below

i1. compareTo(i2) ⇒ Ascending order

- i1. compareTo(i2) ⇒ Descending order

i2. compareTo(i1) ⇒ Descending order

As ascending order

- i2. compareTo(i1) ⇒ Ascending order

↳ insertion order

return 5+4 ↳ reverse of insertion order.

return -1 ↳ no insertion order.

return 0 ↳ only first element

↳ insertion boundary

Q for what purpose Comparable use?
→ Natural sorting order, use Comparable method.

Q for what purpose Comparable use?

→ Customized sorting order. Change sorting
order to be in Ascending order or Descending
order. If it is in Ascending order, the

obj can already implemented in Comparable
then directly go for compareTo Method.

→ If it is in user-defined obj, then
whether you want Ascending order or
Descending order go for compareTo

Stream API

→ different Interface methods are there

Note: when to go for predicate and when to go for function?

Predicate → To implement some conditional
Test method checks we should go for Predicate

function → To perform some operation and
to return some result go for function

for function → with out iteration.

?? → Starts like Method Reference

tomorrow
constructor reference on
eclipse

② Methods

→ static method ✓
→ instance method ✓

we can
use in
my current
code

everywhere
we use

constructor
reference

Stream API → different methods ✓

Also check
Consumer
Referring
to objects and
methods

To operations → using object when create only one
object once only created.
can't reuse or obj because JVM shutdown

Stream API → from JDBC → 1.8✓
java.util.stream
package

RAM memory (volatile)
Because memory not available
by using obj multiple times

→ Stream is a Interface
Configuring =
Processing =

* To keeping data in HDD & Database, whenever
we want to we known as "Persistence"
It is a mechanism

There are many Stream API's in
filtering
InBuiltpAPI

Map

* In HDD small storage in file
Takes

when
new
object
is creating
then
create
long
for long
instead of
int

* In Database persistence, go for
By performing above package
Input & Output operation. API's
available

* JDBC → Java.sql package ✓

* Some Database → MongoDB, MySQL,
implement at later

Agenda Practice on Comparable
↓
Method Reference /
Constructor Reference /

Input and output operations [io package]

1. file

2. FileReader

3. FileWriter

4. BufferedWriter

5. BufferedReader

- ~~X~~
- a) isFile()
 - b) isDirectory()
 - c) length()
 - d) mkdir()
 - e) exists()
 - f) Create Newfile()
 - g) list()

↓

FileReader()

→ If you want to perform Reader Operation on a file, then we need to use "FileReader"

FileWriter(): → Want to perform writer operation on

→ same a file, then we need to use "FileWriter"

* directory & filename both same ✓

* java.io API classes are built using the

Standard of Unix OS.

* In Linux/unix os both directoryName, filename.

Directory

↳ which takes for position Position

↳ location

↳ Directory name.

file

File() an existing file/directory we just
performed the operation like -

In Detail, [I/O operations]

⇒ when we use the object permanently and also we at some ever use it

I/O operations.

Come to picture the object which

Basically the mechanism the object which

HDD or database, then whenever

could be in known as "Persistence".

We want to use I/O operation.

Then persistency is performed by I/O operation.

package in io package,

→ I/O operations

→ I/O package & have Inbuilt API

In io package

function.

In API Mainly you discuss about

→ file ✓

→ FileReader ✓

→ FileWriter ✓

→ BufferedWriter ✓

→ BufferedReader ✓

* whenever you specify any name without extensions, it treat as directory.

directory → implements ✓

filename → implements ✓

for directory result false output, then

f.mkdirs()

go for method f.mkdirs() (make directory)

* By unix OS, then go package 'java

com represent both files & directory.

→ When we write a code where having both files & directory, then java will take separately by using isFile Method

isDirectory Method ✓

so here, it is no more H.E. (without)

In file, on a existing file/directory we just perform operations like

isfile() ✓

isDirectory() ✓

length() ✓

(exists() ✓

mkdir() ✓

CreateNewFile(), or read, or view
list(), or intro, traverse, etc.

fileReader: If we want to perform reader operation on a file, then we need

to use "fileReader".

fileWriter: If we want to perform writer operation on a file, then we

need to use "fileWriter".

SQL:

- Structured Query language.
- Ways keeping large volume of info Data in one place

Known as "Database".

→ DBMS → Store data in form of rows & columns (Table)

→ Every row (record).

(RDBMS)
Relational DBMS

→ Every person information in
Multiple tables.

MySQL
→ Oracle
→ SQLite

Popular
SQL

It could be in
structured way

MySQL is a database (place).

SQL is a language.

→ Using SQL language you can store in a

database (mysql).

→ Structured way, in form of Multiple
tables

NoSQL → Not in a structured way and also

it could be in different ways like

(key-value, Documents, graph --).

SQL →

JAVA → Case Sensitive.

→ Case Insensitive

some commands are same

→ ① Creation Database [some commands are same]

use Database names

② Create table

operations perform [CRUD operations]

③

→ semicolon ; Good practice ✓ partition name it

→ when statements/rewriting partition name it capital, then no problem.

to be any type initially Capital letters no problem.

after writing in small letters no problem.

→ It recommended to use in Capital letters.

→ It recommended ✓ (recommended)

CREATE TABLE

student (----); ✓

Create table

Student(-----); ✓

✓ true case
Predefined in MySQL

④ step → if you want show that above

then

desc student; ✓

student

describe student; → mean the table

drop student; → mean will gone ✓

numerical → int

string → Varchar

size → Varchar

name Varchar(10)

age int(4)

city Varchar(7)

char allowed 255 ✓

mysql -u root -p

* MySQL has a workstation viewer.
↓
when you (root) use
the command

show databases;

/ show schema;

inuron;

①

CREATE DATABASE

(81)

inuron;

CREATE SCHEMA

(81)

inuron;

CREATE TABLE

(81)

student (id int, name, Varchar

Create DATABASE inuron;

(a)

✓ CREATE TABLE inuron.student(id int(6), name

varchar(20),

for any database working, in that database
you have to use that particular table so
for drop Table student;

creation of

table.

CRUD operation → creation of database &

* It will come after creation of table creation.

C → insert statements (eg.: Create)

R → select statements (eg.: Read the data)

U → update statements

D → delete statements

Better write along as above

use inuron;

↪ INSERT INTO(id, name

* value(?, ?, ?) → place holder.

→ Unique → PRIMARY KEY NOT NULL

↓

Column Name

↓

Unique Value

↓

Null not allowed

!!

⇒ Main key points

→ Auto-Increment, PRIMARY KEY

→ ID [given]

NOT NULL

→ which gives like NO NO NO

which will not to

mention id in

Insert values

column

In Null

Unique Key

Given unique for particular column.

Multiple tables are allowed because

↪ Rows

* while mentioning UNIQUE KEY, duplicate values are not allowed.

e.g. number int(5) UNIQUE KEY,

file

→ select * from table-name;

→ entire queried fetched.

✓ → Select columnname, from table-name,

eg:- Select fname from table-name;

Select fname from table-name;

✓ → Select * from ~~table~~ table-name where age = 21

fname,

give what you want

- drop & delete?

drop entire table ✓

delete will delete content inside the table ✓

CREATE TABLE talk-name

CRUD

INSERT INTO ~~table~~ talk-name ()

values ();

✓

It shows once create table then give multiple values for that table then ultimately

it gives multiple tables known as RDBMS

use database;

use database;

update // Alias

select ~~table~~ firstnamenew [give name]

from table-name;

where age = 18;

update student -> set age = 18 where

age = 28;

use database;

alter table student -> drop column city;

which deletes that particular column

use database;

alter table student -> add column city varchar;

which adds that particular column.

Difference b/w update / Alter R structure

which you want to modify you only want to perform any

DDL

→ Data definition language → Data Manipulation by



table structure

⇒ Create, Alter,

Drop, truncate

→ Create table

→ once table created
you want to manipulate it

Insert

Truncate

⇒ delete entire table

Content in table →
table →

⇒ Roll back is
not possible

unique key

→ unique value

→ null value allowed

→ multiple column

⇒ Roll back is
possible

Foreign key

→ multiple column

→ same value

→ no null

→ foreign key in one table
for truncate →
table birds will be
deleted by rows

then not efficient of
removing first rows
then continue

DML

→ Data manipulation language



table structure

⇒ Insert, update,

delete

Drop, truncate

→ Insert, update,

→ once table created
you want to
manipulate it

Delete

Primary key

→ unique column

→ no null value

→ one in a table

→ unique value

delete how it is rollback?

→ set autocommit = 0;

use database;

delete from table-name;

rollback;

transaction

* operations

Suppose we have file Beamer.html

- * Select * from table-name order by id asc;
- * Select * from table-name where age not between 19 and 23 order by salary desc;

- * ~~Select~~ distinct * from table-name;
- ↳ Mention what column you want.

- * Select distinct * from table-name;
- ↳ Mention what column you want.

- * repetition is not possible ✓
- ↳ mentioning distinct & no repetition in not possible ✓
- ↳ filtering the data go for distinct.

- Joins :
- Inner join ↗ refer Mac for
- Left join
- Right Join
- Cross Join
- are allowed
- are not allowed

- ↳ No. of tables at a time in RDBMS

JDBC

Java Database Connectivity.

→ Java database knowledge are built with core Java knowledge are built

* Standalone Applications (Java writing)

* Standalone Applications means which are running on single machine.

* for Web Applications — we need "Advanced Java".

e.g: Calculator, MS-Word, notepad

* Web Internet.

Web Applications following technologies:

Develop Web Applications (Java database connectivity)

- a) JDBC (Processing) (Access the data and get result)
 - b) Servlet (Processing) Frontend
 - c) JSP / Trymeler JSP (Presentation) Java server pages.
- www.Astrology.com

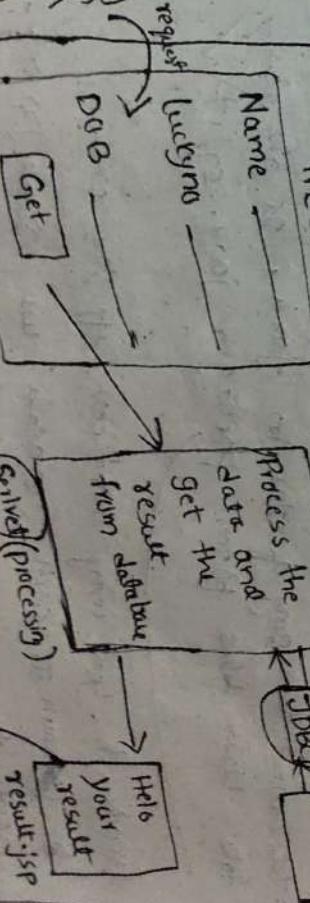
Process the data and get the result from database

index.jsp

ProcessServlet.jsp

JDBC

database



Ultimate Enterprise Application

8) Servlet: If we want to communicate with database go for JDBC.

e.g.: whenever some processing logic is required then we need to go for Servlet.

e.g.: verify user OTP enter ✓

JSP (Java Server Pages):

→ presentation logic is required to display something.

→ JSP stands for View Component

e.g.: display login page

display error page

JDBC

.java

(Driver) \rightarrow Jar \rightarrow Oracle

steps involved to communicate with Database:-

1. Load and register the connection with Database.
2. Establish the Statement object and execute the query.
3. Create ResultSet.
4. Process the ResultSet if it gets generated.
5. Handle SQLException if it gets generated.

6. Close the Connection.

Write the JDBC code the API given (we intend) insert a loop & coupling

jar in java, as we always

* we cannot use any of classes from packages java.sql & java.awt take

→ for requirement of jars go through their coditions like my sql & check it like my sql & check it

Journey of Database (JDBC)

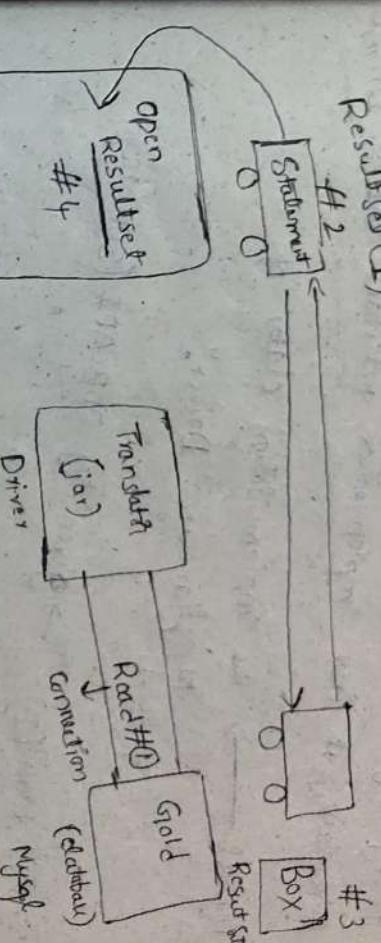
Interfaces

Connection(I)

Statement(I)

ResultSet(I)

steps ✓



① Load and register the Driver!

→ we need to load & register the driver or

Put the DB requirement, or need to set the

→ As per DB specification, or need to set the

JRE environment with DB environment.

To say any class & DB vendor, we say it is

Driver class, it has implemented interface called "Driver"

MySQL → The implementation class of

Mysqljar in "Driver"

Driver (I) → given by JDBC API (SUNMS)

Driver (I)
↑
implements

Driver (C)

Driver (C) → given by
mysql - connector - j - 8 .XX .jar

(MySQL vendor)

✓ first step

Q How to setup .class file explicitly into JRE?

⇒ Class.forName ("com.mysql.cj.jdbc.Driver");

Driver class in
loaded, so in

Driver class in
loaded, so in

JRE JDBC

environment for

MySQL DB setup.

class Driver ✓ 2nd step

{ static

Driver driver = new Driver();

DriverManager.registerDriver(driver);

3

registered successfully

→ that means, we set up the

JRE environment with DB

environment ✓

Note,

from diagram above

javac, java are commands executed in an

OS will search for these commands in an environmental variable called "path".

Set path = "location of jdk/bin".

→ the packages don't create explicitly on

location (it.jar) ✓

NOTE:-
Since, To inform javac, java command about the

location of third party jar, SUNMS had given a new environmental called "classpath".

→ Export PATH = \$PATH;. Paste SQL class
class

Step②

2. Establish the connection with Database -
→ In step 2, must have username & password because accessing of database.

Two types of Establish the Connection:-

Protocol

- ① DB specific protocol ✓
- ② Well known protocol. ✓

① Database specific protocol:-

ProtocolName : dbengiename://ipaddress of db:portNo of db

Instead,

"jdbc:mysql://localhost:3306/testiran"
↓
DatabaseName

Protocol

Connection connection = null;

if step②

String url = "jdbc:mysql://localhost:3306/testiran"

String user, Name = "root";

String password = "root123";

Connection = DriverManager.getConnection(url,

new java.util.Enumeration(),
new java.util.Properties());

System.out.println("Connected");

Step:-3

Create statement object and send the query.
String sqlselsetQuery = "Select * from student";
Statement = connection.createStatement();
statement.executeQuery(sqlselsetQuery);

Cut off creating

Above line you are creating connection object, but be created? // so connection, statement, resultset

be created? // so connection, statement, resultset
Answer - No, for an interface instantiation is not possible.

In above line, an object to a class which

implements an interface called Connection is created and we hold the reference of the object with instance name.

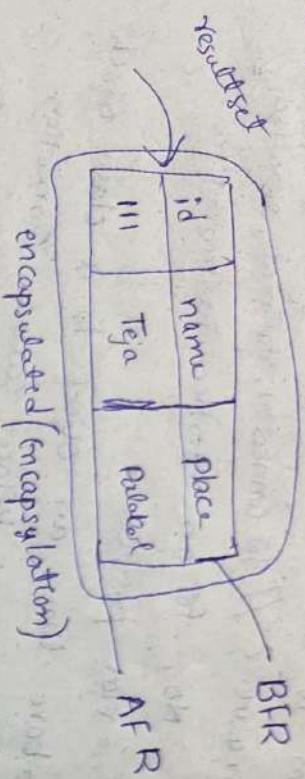
This is done to promote loose coupling.

The above code would also represent "abstraction".

If you pass the selectQuery

Result set object appear

④ process the resultset



which means wrapping the associated

data and its relation into single

unit is called "Encapsulation"

→ To access the data we need to use "getters"

To set the data we need to use "setters"

Already

data are been there

data are used

we want to use data)

we want the data)

getters (Get the data)

Setters (Set the data)

cursor next

next (if true)

(otherwise false)

```
while (resultSet.next()) {  
    Integer id = resultSet.getInt(1);  
    String name = resultSet.getString(2);  
}
```

```
Solving City = resultSet.getString(3);  
}  
connection.close();  
System.out.println("Data inserted");
```

⑤ Handle the exception
↳ Already done in step ① & step ②

⑥ Close the connection

finally {

```
// closing the resources  
if (connection != null)
```

```
{  
    try {  
        connection.close();  
    } catch (SQLException se)
```

```
{  
    se.printStackTrace();  
}
```

```
{  
    se.printStackTrace();  
}
```

Step 01 // Load the register driver

```
class.forName("com.mysql.jdbc.Driver");
System.out.print("loading the driver - - -");
```

Step 02 // Establish connection

```
String url = "jdbc:mysql://localhost:3306/tjaktion";
String username = "root";
String password = "liran@5327";
Connection connection = DriverManager.getConnection(url, username, password);
```

Resultset = statement.executeQuery("Select id, name, city
from studentinfo");

```
System.out.println("ID\tNAME\tCITY");
while (resultset.next())
```

```
{
```

```
    int id = resultset.getInt(1);
    String name = resultset.getString(2);
    String city = resultset.getString(3);
```

```
    System.out.println(id + "\t" + name + "\t" +  
                      city);
```

Step 03 // Create Statement obj and send query

Step :- 2

Optional because JDBC 4.0 autoloading feature is

→ JDBC 4.0 autoloading feature is enabled. ~~disabled~~.

→ In eclipse META-INF where services

you observe ~~it.~~ you observe in JDBC

Q) What is autoloading in JDBC?

→ Loading and register the driver is

done automatically.

Behind the scenes

→ In Eclipse

→ go to META-INF

→ service

→ Shows com.mysql.cj.jdbc.Driver ✓.

Note:-

→ Using ResultSet object, we can retrieve column names

the records based on the numbers

also instead of numbers

→ If java program and database engine is

running in the same program

make string url="jdbc:mysql://localhost:3306/test"

According to DBA Specification, all SQL commands

or →

a) DDL (Data Definition language)

e.g:- CreateTable, alterTable, dropTable -

b) DML (Data Manipulation language)

e.g:- update, insert, delete -

c) DQL (Data Query language)

e.g:- select

d) DCL (Data Control language)

e.g:- Alter password, grant access -

e) DA Command (Database Administrator Commands)

e.g:- Start audit

Stop audit

f) TCM (Transaction Control language)

e.g:- commit, rollback,

According to Java Developer point of view, SQL

Operations are into 2 types

→ Select operation (DQL) ✓

→ Non-select Operation (DML, DCL -)

In Step 4 we observe Statement to object

to send to query, Step 4, we need to call

where executeQuery, we need to call

following methods.

- 1) executeQuery() ✓
- 2) executeUpdate() ✓
- 3) execute() ✓

① executeQuery() :-

* This method pertain Select operation ✓

* Because of above Method execution, it will
get will get a group of secondary
which represented as ResultSet object.

Eg:-

Resultset resultset = statement.executeQuery("

Select id, name, city from student")

```
if (value == true)
{
    // selectQuery
}
else
{
    // non-selectQuery
}
```

② executeUpdate() :-

* This method is used for "Non-Select operation"

(like insert|update|delete) ✓

* we won't get group of records ✓
* we will get a numeric value which
represents no. of rows affected - so
return type is "int" for this method.

Eg:- int rowsAffected = statement.executeUpdate("insert into student values('Rahul', 'Mumbai', 22)

(3) execute():

* we can use this method for both select
& non-select operations.

* if you don't know the type of Query
at beginning and at dynamically at
runtime then we use this method

to execution.

→ return type is "boolean". we use execute()
To execute stored procedure we use execute()
Eg:- boolean value = statement.executeQuery(dynamicQuery)

Formatting SQL Queries using dynamic input

1st Approach

from keyboard
it takes input
String id = sc.nextInt();

```
// name = hardhit;
// city = HYD;
```

String name = sc.nextLine();
String city = sc.nextLine();

```
if(name = "u"+name+"");
    sqlInsertQuery = "insert into student(id, name, city)";
    values ("u+id+", "name+", "city+");
    String format("insert into student(id, name, city)");
    values (%d, '%s', '%s'),
```

values (%d, "%s", "%s"),
String format("insert into student(id, name, city)");
values (%d, "%s", "%s"),

first single
double code → after double In this place

① id+,
② name+,
③ city+

2nd approach:

sc.nextInt();

int id = sc.nextInt();
String name = sc.nextLine();

String city = sc.nextLine();

name = "u"+name+
" " "

city = " " + city+
" " "

String sqlInsertQuery = "insert into student(id, name, city)
values (%d, "%s", "%s")", id, name, city);

3rd approach: Best approach
2 approaches are not recommended, to
* Above do formatting we prefer using String class

format()

public static String format(String format, Object... args)

int id = sc.nextInt();

e.g.,
String name = sc.nextLine();

String city = sc.nextLine();

String sqlInsertQuery =
"insert into student(id, name, city)", id, name, city);

String format("insert into student(id, name, city),
values (%d, '%s', '%s'),

values

%d,

%s,

%s");

↓

for int

(Not required)

single
code

↓

for string (single code required)

main steps in JDBC -
in the connection

→ establishing statement

→ Create resource

→ closing

→ closing way in every code is object value.

What would different select program
program

Common steps

→ establish connection ↗ common.

→ close ~~seen~~ resources

→ handling exception

In output space

for we give + live
System.out.print("1 2 3 4 5")

for
Space:

Properties:- (refer Eclipse)

* To store data in key value pairs ✓
* use get() method to get the value associated
with key.

Advantage of Properties:-

→ Making changes in SRC code.
→ no need of re-compilation
→ no need of re-run → import.java.io.*

Step ① Create FileInputStream → throw exception.

FileInputStream fis = new FileInputStream("↓
Path location")

of properties

Step ② Create obj Properties → import.java.util.Properties

Properties prop = new Properties();

prop.load(fis); → property
getProperties

String url = prop.get("url"); →
String user = prop.get("user"); →
String password = prop.get("password");

No. 1, Create file with properties
url = "jdbc:mysql://localhost:3306/
root"
user = "Kiran@5324"
password = "Kiran@5324"

life cycle of Query Execution:- [SQL Database working
with Java]

1. Compilation

- a) Query Translation ✓
- b) Query Parsing ✓
- c) Query Optimization ✓

2. Execution

- Once compilation successful the optimized query is
taken as input and the query will be
executed by query interpreter.

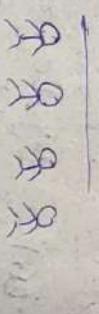
3. Fetch the result

- result will be sent in the form of
ResultSet (SelectQuery) or in form of relevant
(non-selectquery) to the Java application by

database engine,
Statement object

→ If you push the select query
appear

→ Resultset object we can't send it to
network because resultset objects are not
"Serializable"



In above context, same query will be executed multiple times with change in values

If we use statement object approach

- ↳ same query have to compiled every time for every client request, this could create performance issue.

To resolve this problem we need to use "Prepared Statement(I)" ✓

Advantage of Prepared Statement:

- Query will be compiled only once even though we are executing multiple times with different values.
- Performance increased.

PreparedStatement pstmt = connection.prepareStatement("insert into student
values(?, ?, ?, ?)");
pstmt.setString(1, id);
pstmt.setInt(2, name);
pstmt.setInt(3, age);
pstmt.setString(4, address);
pstmt.executeUpdate();

At this line query will sent to database engine

DB engine compile the query and store in database.

* Step ④ writing & PreparedStatement
Query to executing.

String SQLInsertQuery = "insert into student
values(id, name, age, address)
values(?, ?, ?, ?)";
PreparedStatement pstmt = connection.prepareStatement
("insert into student
values(?, ?, ?, ?)");
pstmt.setString(1, id);
pstmt.setInt(2, name);
pstmt.setInt(3, age);
pstmt.setString(4, address);
pstmt.executeUpdate();

Prepared Statement pstmt = connection.prepareStatement
("insert into student
values(?, ?, ?, ?)");
pstmt.setString(1, id);
pstmt.setInt(2, name);
pstmt.setInt(3, age);
pstmt.setString(4, address);
pstmt.executeUpdate();

PreparedStatement pstmt = connection.prepareStatement
("insert into student
values(?, ?, ?, ?)");
pstmt.setString(1, id);
pstmt.setInt(2, name);
pstmt.setInt(3, age);
pstmt.setString(4, address);
pstmt.executeUpdate();

* Statement(I) Prepared Statement(I)

Both are child relationship

Parent - Child relationship

change in inputs.

⇒ Using prepared statement can execute

the query multiple times with change

in the inputs.

Main.

Differences

Advantages of Prepared Statement -

→ R.T + E.T + R.T

1) Performance is very high because compact

to statement approach because query will

be compiled once

But In prepared statement object is

restricted to only one query, that query can be executed multiple times

with change in input

(e.g.: j@phone → jo).

2)

In this we don't send query multiple times b/w java application / database traffic will be reduced.

3) Dynamically we can send inputs to the query ↗

Static Query vs Dynamic Query :-

Static Query :- The SQL query without positional parameter (place holders) (?) is called

Static Query.

4) Inputs to query like java style, no need to perform formating or per the DB specification.

5) Best suitable for inserting Data Values

c) " for insertion of BLOB's and

CLOB's (image & PDF files)

Limitation of Prepared Statement :-

↳ one statement object can be used to execute multiple query but with no change in values.

[e.g.: SAMSUNG Phone → jo, ast, vi].

→ In statement, R.T + C.T + E.T + R.T

→ In prepared statement, No compile Time(C.T)

It prevent SQL injection Attack.

* Simple statement object is used for static queries.

Where as prepared statement object can be used for static queries and dynamic queries also.

stmt = con.prepareStatement("SQL Insert Query")

stmt = con.prepareStatement("SQL Delete Query")

X Stmt = Con.prepareStatement("SQL Insert Query")
Not Allowed as already PreparedStatement
exists with same SQL Statement

↳ Pre-compiled again

Instead of using

stmt1 is

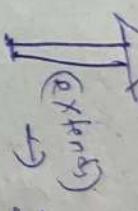
```
import java.util.Date;
```

java.util.Date → which is used to

Date (name)

Difference b/w java.util.Date & java.sql.Date ?

→ java.util.Date Class
→ General utility class to handle Date
in its own Java program.



→ It represents both Date & Time.

(Parent)

→ Especially Designed to Date
we write D.B. operations

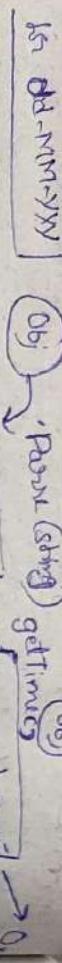
→ To give only Date

↳ In SQL package also there is Time class

available

For print of yyyy-mm-dd & dd-mm-yyyy
where you observe that, "dd-mm-yyyy"

This could be simpleDateFormat
java.sql.Date

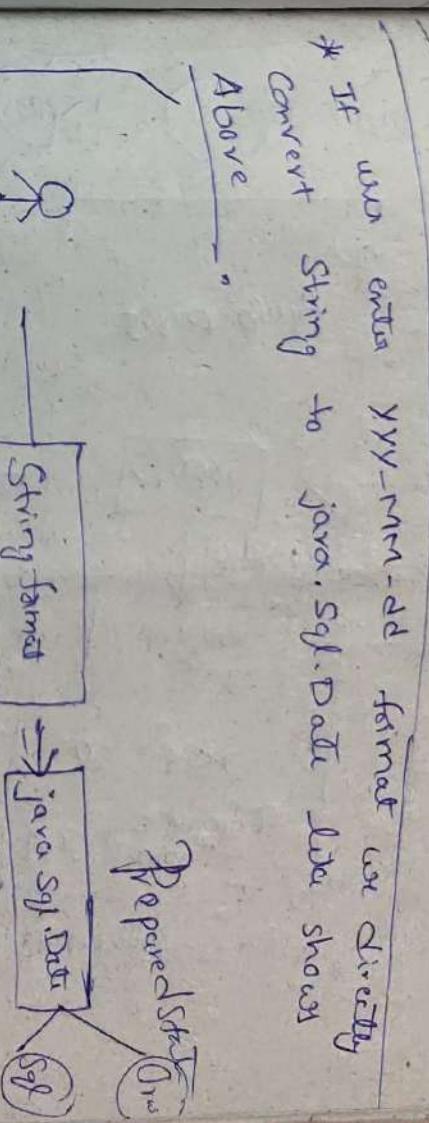


dd-MM-yy
yyyy-mm-dd

yyyy-mm-dd
it will take like this
MM-dd-yy

dd-mm-yy
it will take like this

Above complete action is because of
different Time zones may affect to SQL



ps.setString(2, java.sql.Date) *

L This method internally converts date value

into database supported format

format
yyyy-mm-dd [Input]

Scanner sc = new Scanner(System.in);

System.out.println("String standardInput:");
(yyyy-mm-dd);

String standardInput = sc.nextLine();

java.sql.Date sqlStandardInput = java.sql.Date.

ValueOf(standardInput);

System.out.println(sqlStandardInput);

// 1993-03-27; → standardInput

ResultSet

getAt(1)

SimpleDateFormat

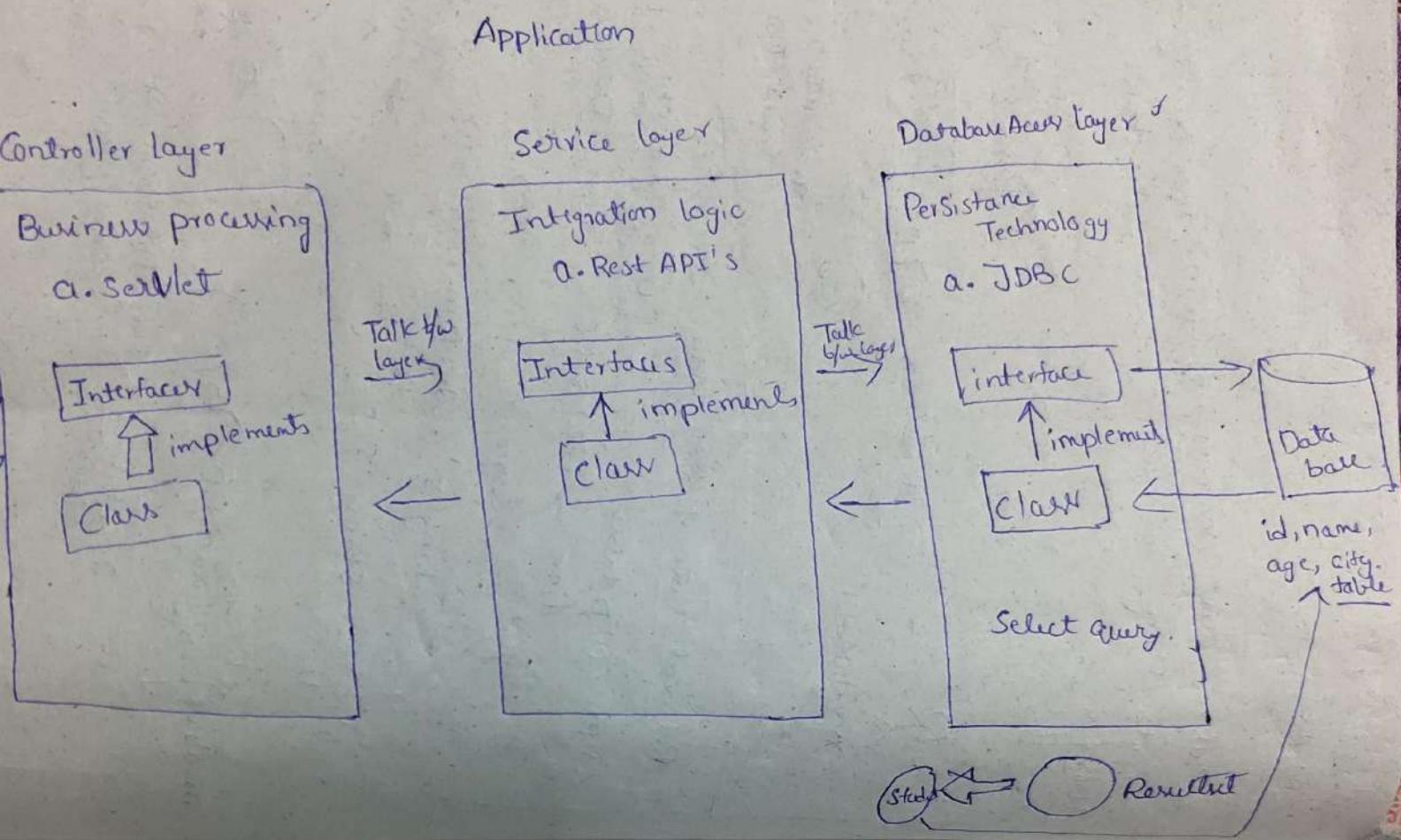
format(pattern)

java.lang.String

Database

java.sql.Date

Customer



Need of DTO in project :-

- Data Transfer Objects
- DTO is used for transferring the object from one layer to another layer in the application.
- * The data from database is transferred in the form of objects.
- * Every layer has its own implementation of interface with class.
- After implementation of student object if we have to use it in "Serializable", we have to understand the remaining layers.

Twin Back Better Understand

* Every layer we should do "interface injection"

implementing class.

e.g.: like `IStudent` → `IStudentImpl`

Every layer should separate package.

① Controller layer

→ in-injection. controller

② Service layer

→ in-injection. service

③ Database access layer

→ in-injection. persistence

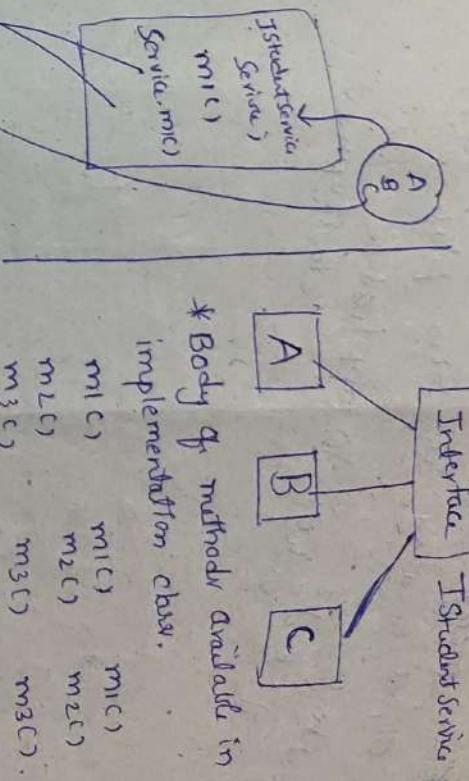
Surround with `in-injection`

having student object
and also student object
that have "Serializable".

It call any method of `m1, m2, m3`,
that could be have object of A,
object B & object of C.
Object creation not possible by its own.

object injection take [objects A, B, C]
which called "Dependency Injection".

use factory class



* In package `inmemory`.
Object creation is taken place ✓.

* In package `inmemory`.
Object creation is taken place ✓.

Object creation is taken place ✓.

Object creation is taken place ✓.

* By JDBC, we see how to insert data remaining images, files, audios etc.

Large Object:-

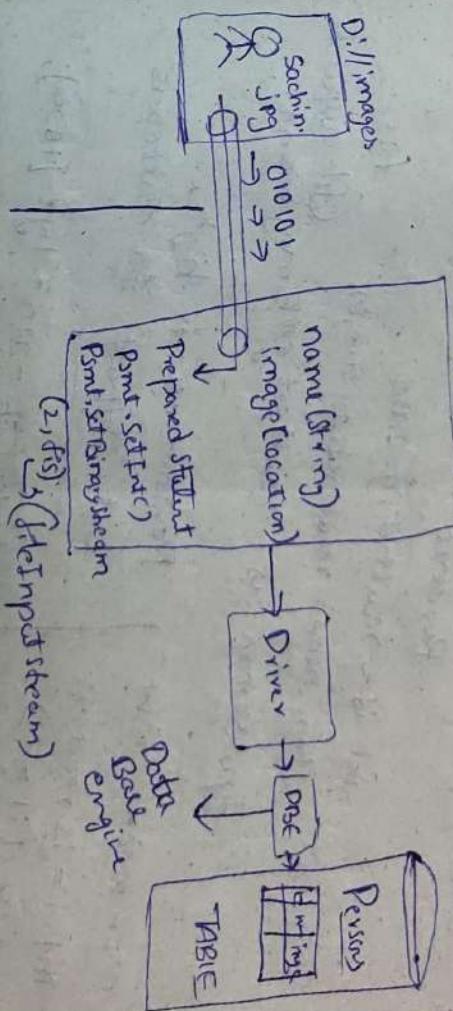
a) Binary Large Object (BLOB)

e.g.- images, audio files, video files ---

b) Character Large Object (CLOB)

e.g.- txt file, pdf file, .xml files ---

java



fileInputStream

✓ which will take place from local to java program in the form of 010101 → Avoid this type by writing setBinaryStream method.

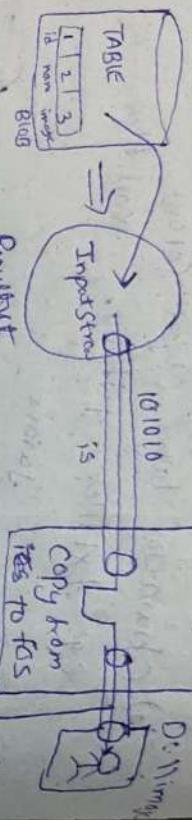
public abstract void setBinaryStream(int index,
Inputstream
(fr))

Query :- String SQLQuery =

// BLOB

Retrieval operation

→ The term retrieval means take place from database to local .
java



int id = resultSet.getInt(1);
String name = resultSet.getString(2);
InputStream is = resultSet.getBinaryStream(3);

In SQL
longblob
Timage datatype → BLOB → JDBC → Binary
RichText datatype → longtext

Performance low

Performance is high,
but difficult for developers

for (int i = 0; i < is.read(); i++) {
 byte[] b = new byte [1024];
 int n = is.read(b);
 if (n > 0) {
 fos.write(b);
 }
}

Retrieval CLOB operation

Reader reader = resultSet.getCharacterStream(3);

foutis.copy(I/pstream, opstr)

// CLOB → character large object : images

{ PDF file }

PreparedStatement.setBinaryStream()

→ PDF file

PrintWriter.setCharacterStream()

new FileReader(new File("pdfloc"));

longblob

Image datatype → BLOB → JDBC → Binary

RichText datatype → longtext

In Java (JDBC)

character

for (int i = 0; i < is.read(); i++) {
 char c = (char) is.read();
 pw.print(c);
}

File file = new File ("hi.htm");

* When we use statement object, we get the positional parameters.

Positional Parameters

Select count(*) from user where name = ? and password = ?

Prepared Statement Object

SQL Injection:-
Comments in SQL are of 2 types:-

Comments in SQL are of 2 types:-

- Single Line Comment
- /* Multi Line Comment */

*

SQL Injection Example?

Q) Can you explain SQL Injection?

Eg: Select count(*) from user where name = 'Sachin'
and password = 'Hindukar'.

No problem of adapt.

Select count(*) from user where name = 'Sachin' -- and
password = 'Hindukar';

Statement object

- Query compiled ('--won't come into picture')
- Query executed (Select count(*) from user where name = 'Sachin' -- and password = 'Hindukar')

SQL Injection

In detail you -- after that
it should be
here typing
Comments,

SQL Injection

PreparedStatement (Java)

SQLInjection → end user manipulated the
query with Special Syntax.

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "root");
Statement statement = connection.createStatement();
String query = "select * from user where name = ? and password = ?";
PreparedStatement preparedstatement = connection.prepareStatement(query);
preparedstatement.setString(1, "Sachin");
preparedstatement.setString(2, "Hindukar");
ResultSet rs = preparedstatement.executeQuery();
while (rs.next()) {
System.out.println(rs.getString("name") + " " + rs.getString("password"));
}

By using PreparedStatement
multiple it goes
to database

* In programming if any code is repeatedly required, we can define the code inside the method & we call that method multiple times as per our requirement.

Similarly, In database programming if any group of SQL statements are required than we define those SQL statements in a single group and call that group as per our requirement.

This group of SQL statements that perform

a particular task is nothing but "StoredProcedure".

* StoredProcedure is just reusable Component at database level.

Stored Procedure:

If we want to group of SQL statements that perform particular task, no return value needed.

→ which are permanently stored inside database for future usage.

→ Stored procedure created by DatabaseAdmin → Stored procedure has its own language to Every stored procedure.

Stored procedures
Oracle → PL/SQL / MySQL → Stored procedure language ✓

Stored Procedures

(3) Parameters

- IN parameter (to provide input values)
- OUT parameters (to provide output values)
- INOUT parameter (to provide & collect output)

for ex:-
select pid, name from product where pid = ?;

for usage of above statement go for

"Stored procedure"

"Stored procedure" → this one variable remains of p

procedure only this input variable remains of p

Eg:- PROCEDURE

PROCEDURE PRODUCT_DETAILS_BY_ID
(P.GET_PRODUCT_DETAILS_BY_ID, OUT NAME VARCHAR(40),
C.INT, INT, OUT NAME, OUT QUANTITY, INT);

OR RATE INT, OUT QUANTITY, INT);

BEGIN SELECT NAME, PRICE FROM
PRODUCTS WHERE ID = P.GET_PRODUCT_DETAILS_BY_ID;

PROCEDURE

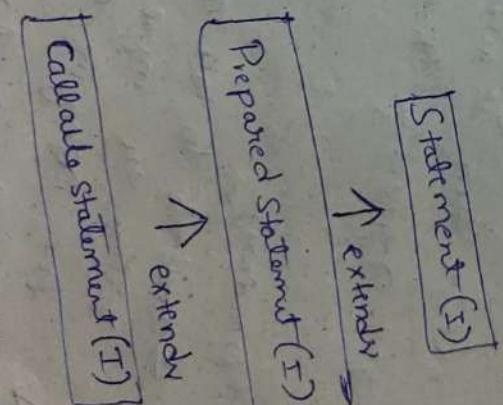
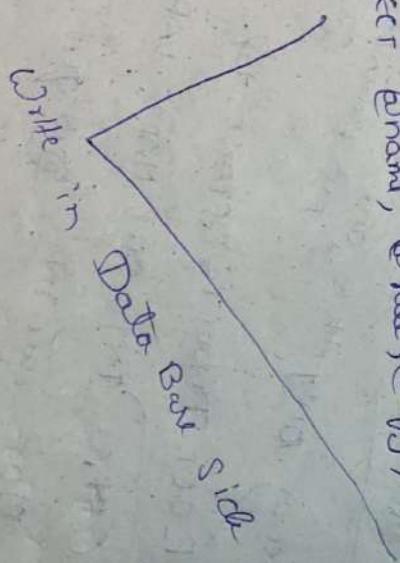
P_GET_PRODUCT_DETAILS_BY_ID
(IN id INT, OUT NAME VARCHAR,
OUT RATE INT, OUT QTY INT)

```
begin
    select name, price, qty into name, rate,
    qty from products where
    pid = id;
end$$
```

* To call the stored procedure we use the following

Syntax as shown, write in query

```
call P_GET_PRODUCT_DETAILS_BY_ID(2, @name, @rate, @qty);
select @name, @rate, @qty;
```



Q) If we want to execute only stored procedure can we use statement / prepared statement ?

Statement & prepared statement are written by java developer

Prepared statement → Executing the query (query will be given to developer by java developer)

Call statement ⇒ Executing the query (Query will be coded by DBA and present inside database)

Through this we not perform Insertion, ~~Update~~ Deletion

Mainly used for Retrieval part.

Statement in parent from preparedstatement
Statement in parent from CallStatement

* Our duty to specify where particular datatype used. Compare to database.

↳ P_GER_PRODUCT_DETAILS_BY_ID(1, ?@name,

@rate, @%

By JDBC (callabale Statement)

ing storedProcedureCall = (CALL P_GER_PRODUCT_DETAILS_BY_ID(?, ?, ?, ?));

Cstmt = connection.prepareStatement(storedProcedureCall);

When jvm encounters the above line, jvm will

the call to database.

Then DB engine will check whether the

procedure is available or not.

If it is available then it will return

Statement object representing that procedure.

JDBC (stored procedure)

= setxxx() → available to take care of

getting the input values of

setting the input values of

DB engine datatypes

e.g. setInt(1, id) → int ✓

setString(2, name) → Varchar ✓

Registering the output variables of storedProcedure

a) To map the Java datatype and Database

specific datatypes we need to some mechanism

b) The mechanism used is "JDBC Types" which

is also known as "Bridge Types".

e.g:- java datatype = string.

JDBC datatype = Types.Varchar ✓

DB datatype = Varchar ✓

e.g:- java datatype = int

JDBC datatype = Types.Integer ✓

DB datatype = int ✓

Note:- To execute stored procedure we use execute method (execute());

Retrieves the records
the product name

```
CREATE
PROCEDURE
    P.GET_PRODUCT_BY_NAME
    (IN name1 VARCHAR(20),
     IN name2 VARCHAR(20))
BEGIN
```

```
    Select Pid, Pname, Price, Qty from Product
```

```
    Where Pname IN (name1, name2);
```

```
END
```

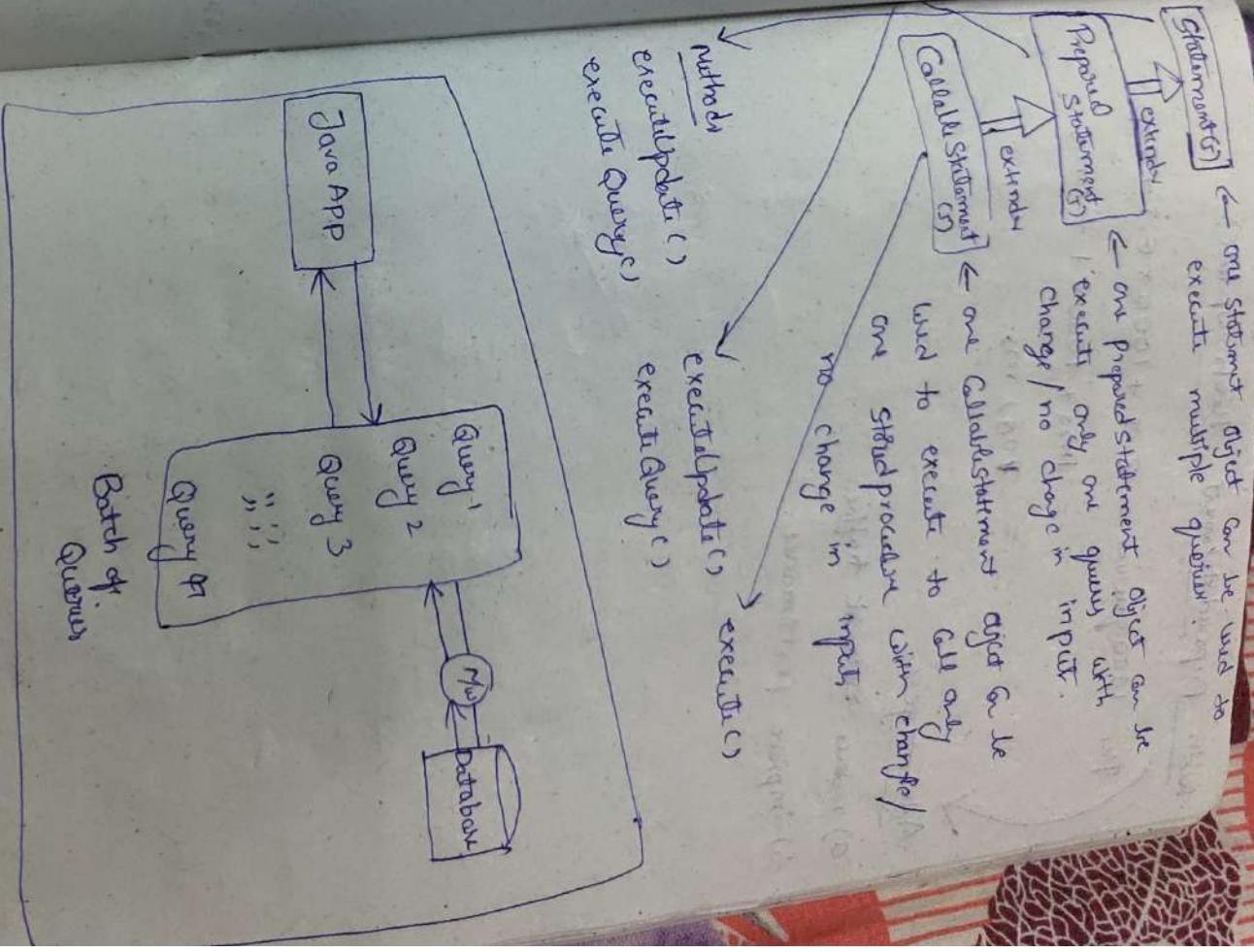
To call stored procedure follow,

```
CALL 'P.GET_PRODUCT_BY_NAME('medmix';Santos');
```

Output

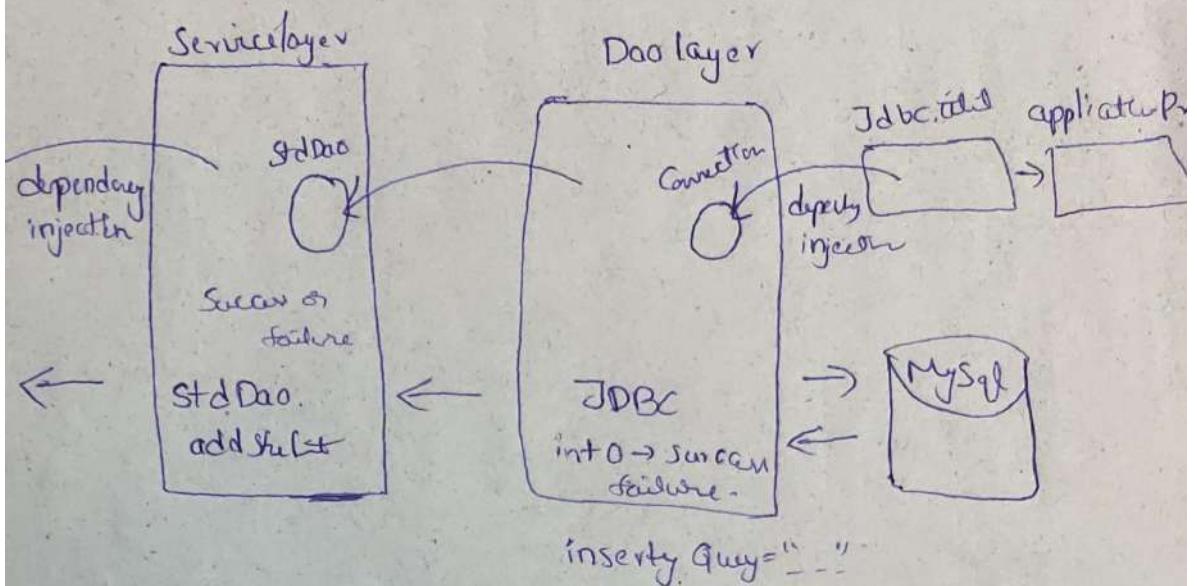
Pid	Pname	Price	Qty
2	Medmix	33	3
4	Santos	45	6

To DATABASE SIDE



→ which will avoid Network Traffic

→ Grouping all the related queries into single batch and we can send that batch at a time to the database.



Design pattern (singleton) :

one object is managed at any time.

and catcher by controller layer [i.e., name, age, address -- sent/entered].

layer object (only one object we want to play around) leave

~~object~~ → we pass three data, we need service object that

Dao layer, because that layer gives persistance logic.
injected, it also dependency injection.

database → for there JDBC will ~~other~~ needs.

dent whatever value needs that value to be send.
pass

→ success for failure -

record inserted successfully.

Transaction Management

→ Process of combining all the related operations into a single unit and executing on the rule "either all or none" as referred as "Transaction Management".

e.g.: Movie Ticket Reservation

- 1) Verify the station
 - 2) Return the Ticket
 - 3) Payment
 - 4) Issue Ticket
- All operations should be performed on single unit.

→ If some operations fail and some are success then may be "Data inconsistency problem".

Transaction properties:-

Every Transaction should follow the ACID Properties

A - Atomicity
C - Consistency

I → Isolation

D → Durability

Types of Transactions

- 1) Local Transaction
- 2) Global Transaction.

① Local Transaction

→ All operations can be done under same database.
e.g.: Transfer one acc to another acc which one belongs to same bank.

② Global Transaction

→ Could be done under different database.
e.g.: Transfer from one acc to another acc which are belongs to different bank.

ANSWER:

→ JDBC supports only Local Transactions.

→ If we want Global Transactions go for the framework

like Spring / Hibernate ✓

Reason: In

Problems of transaction

- 1) Disable - the auto commit nature of JDBC Connection. setAutoCommit(false);
 - 2) If all operations are completed means them we can commit the transaction using following method.
connection.commit(); ✓
 - 3) If any SQL query fails, then we need to roll back the operations which are already completed using the following method connection.rollback();
- After saving in database
→ reflects to what values in database
could be used
- Save point(I) → ~~do not~~
Connection.setAutocommit(false); ↗
Case Study I-
- Operation ①
Operation ②
- SavePoint sp = connection.setSavePoint();
- Operation ③
Operation ④
- if (balance < 1000)
connect.m.rollback(sp);

Division-1 Based on the type of operation performed

on Resultset

- a) Read only Resultset (Static Resultset)
- b) Updatable Resultset (Dynamic Resultset)

Division-2 Based on the movement of cursors

- a) Forward only Resultset (Non-Scrollable)
- b) Scrollable Resultset

- a) scroll Insensitive Resultset
- b) scroll Sensitive Resultset

absolute) → it works from Before first Record & from After last Record.

relative) → it works with respect to current position.

e.g.: `ResultSet.absolute(1)`
`ResultSet.relative(3)`

In Both the methods positive means move in forward direction, negative means move in backward direction

move in backward direction
move in backward direction (more toward update Table (more toward location step by step))

There are work under

Note: `rs.last()` and `rs.absolute(-1)` both are equal vs. `first()` and `rs.absolute(1)` both are equal.

* Note:
→ `Absolute()` is not depend on `relative()`

→ `relative()` is depend on `Absolute()` for taking records at particular point.

scroll Insensitive Resultset.

→ After getting resultset, if you change in database, if those changes are not reflect to resultset.

public static final int TYPE_SCROLL_INSENSITIVE

scroll sensitive Resultset.

→ After getting numbers, if you change in database, if those changes are reflect to numbers.

public static final int TYPE_SCROLL_SENSITIVE

In this you do refresh i.e. `resultset.refresh()`

Updatable ResultSet

→ It is possible to perform update operation using update operation using writing delete query.

result.updateRow()

→ This method deletes rows based on cursor position.

Insert operation done without writing query by using

ResultSet - [STEPS INVOLVED]

1) resultset.moveToInsertRow();

↳ Creates empty record

Id	name	age	city
"	Tom	35	New

2) resultset.updateXXX(int pos, object value);

↳ Insert the value based on the column data

Working with Excel:

3) resultset.insertRow();

↳ Record will be inserted to the table having values.

1	Excel	3.85
---	-------	------

Updatable operation done with using ResultSet without update operation using writing query :-
using resultset.getXXX(int pos)

1) resultset.updateXXX(int pos, obj new value)

↳ getting the old value from ResultSet

2) resultset.updateXXX(int pos, obj new value)

↳ update with new value.

3) resultset.updateRow();

↳ record will be updated to the

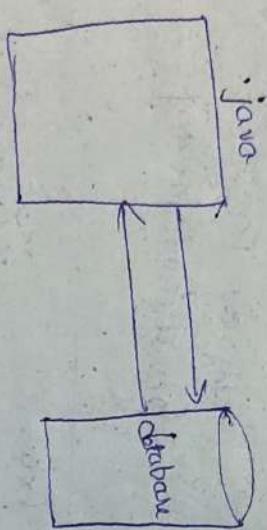
table as per the record status
(resultset information).

Similarly CSV files are also
comma separated files
↳ There we need join (Driver) ✓

Imp topic

Physical Connection:- Communicate with database by connection.

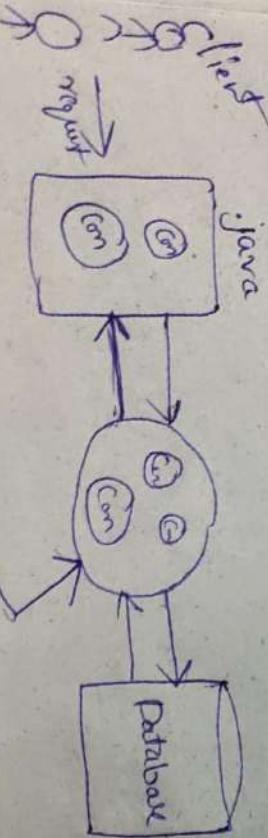
- * Directly communicate with database object.
- * Bad connection
- * performance low



Logical connection:-

- * Not directly communicate with database in between there is (connection pool).

* performance high.



Advantages:- want our client requests to get

→ we don't want object, the connection object is available pick from connection pool.

available pick from connection pool.

→ After using connection object, connection will be returned back to object pool so it supports

connection reusability.

"Connection Pooling"

* DB vendors also supplied mechanism in the jars.

* The best suited vendor who supplied mechanism is HikariCP.

connection pooling mechanism

It is a supplied default supported in spring boot.

which

helps us to improve connection approach

connection pooling

to use 2 jars

HikariCP is a "zero overhead" connection pool.

→ we need to use production ready connection pool.

a) hikaricp jar

b) slf4j jar

Note: Since Hibernate datasource is best
for connection pooling, In Spring,
Spring boot by default Hibernate connection

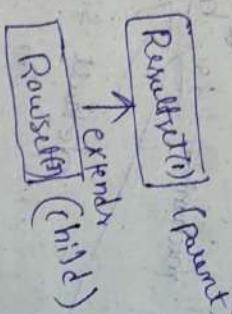
pooling is available.

Rowset:

- It is an alternative to ResultSet.
- We can use Rowset to handle group of records in more effective way than

records in more effective way than
we can use Rowset to handle group of

Rowset.



→ javax.sql.Rowset.

→ RowSet is a child interface of ResultSet.

→ By default Rowset is scrollable and updatable.

→ By default Rowset object implements Serializable, so Rowset objects can be sent over the network.

→ we can overcome limitations of Rowsets
by Rowset.

object my default connects object

→ ResultSet object of compulsory,

{ requires connection of compulsory }
Coming to Rowset either Connected/disconnected mode

2 Types of Rowset

a) Connected Rowset.
b) Disconnected Rowset.

All frameworks support Rowset → ResultSet(I) default: read-only & non-updatable

not give propery → ResultSet(I)

→ ResultSet(I) default: updatable & scrollable

(Don't use)

Connected

Disconnected

Sing DB vendor → ResultSet(I)
→ JDBC vendor → ResultSet(I)
→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

→ JDBC vendor → ResultSet(I)

Not preferred

→ JDBC vendor → ResultSet(I)

Note:- Creating of a new app.

Keyword is permitted in Real time Application

Instead of this, we use "factory Design pattern"

e.g:- Rowsetfactory rsf = RowsetfactoryProvider.factory

SERVLET APPLICATION

We know there

Types of Applications

→ Java Standard Edition

① standalone Application (JSE) → Java Standard Edition

a) CUI (console User Interface) Application

b) GUI (Graphical User Interface) Application

Note:- In Rowset, Implementation for Rowset

is provided by java vendor only,

not by db vendor.

Note:- which waiting with Cached Rowset, when

Copying All ~~the~~ Resultset to object data
to CachedRowset object we use "Populate"

CachedRowset.populate(Cresent);

Application logic over multiple no. of JMS

→ for web application we know Technologies

NOTE:- In disconnected Rowset [CachedRowset]
AUTO-INCREMENT is not supporting.

[CGI, Servlet, JSP] - ✓
→ The main purpose of web Application to generate dynamic response from Server Machine.

Note:- Disconnected Connection Not

recommended to Real Time Application

because driver support from db vendor

is not available -

Distributed Application:- [Business to business model]

→ It is a server side Application, it will be designed with distributing Application

logic over multiple no. of JMS.

→ We need Technologies [RMI, EJB's, WebService,

The main purpose - to establish the connection b/w local Machine and remote Machine to access the remote service.

E.g.: Banking sites take help of Third party Apps like Payments done by Banks, UPS, Cards -

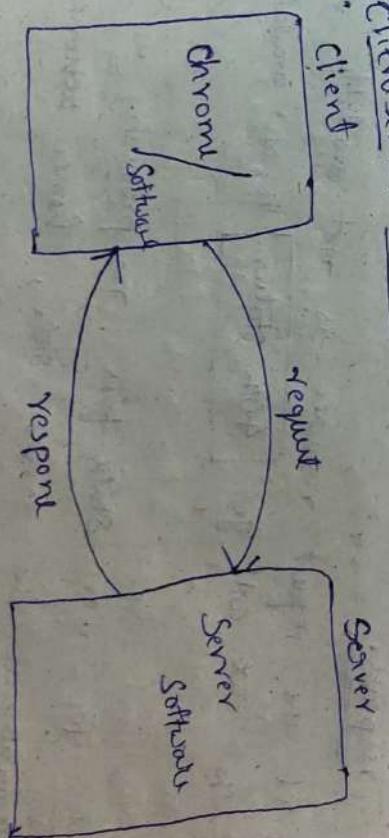
Client → The program which sends the request is called "client".

Server → The program which program the request & sends the response is called "server".

Web Application:

→ The Application which is developed only using web based Technologies like html, CSS, javascript, servlet, JSP etc is called Web Application.

Client Server Architecture:-



In detail, Client sends request they need client side

Client software (e.g.: chromium) Server sends response they need Server side

Deployment: of placing web application inside web server known as "Deployment"

Undeployment: of removing web application from web server known as "Undeployment"

Webclient: To send request from the user we need to have special software installed in the Client Machine

E.g.: chrome - etc.

Mostly used
webserver

The response which won't change from person to person, time to time called static response.

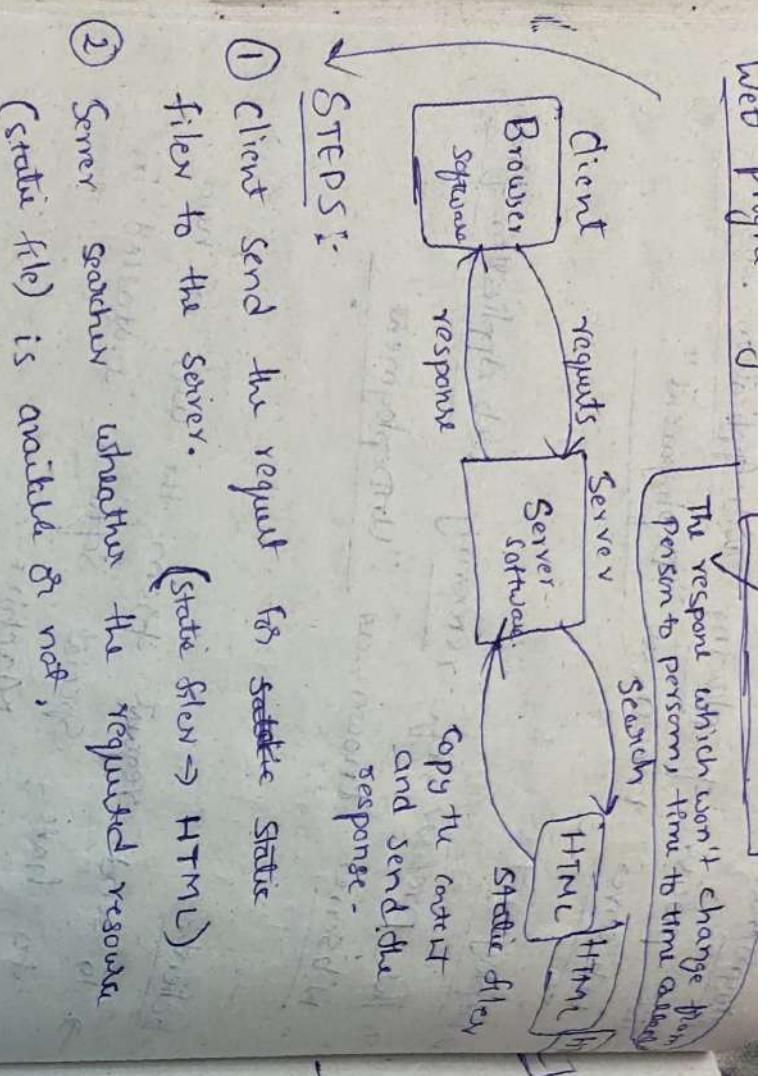
→ The response won't change from person to person, time to time called static response.

e.g.: Login page of gmail, etc.,

Dynamic response: The response which is dynamic from person to person, time to time.

e.g.: Dynamic response. e.g.: Bookmyshow, Index details.

Web programming for Dynamic response:-

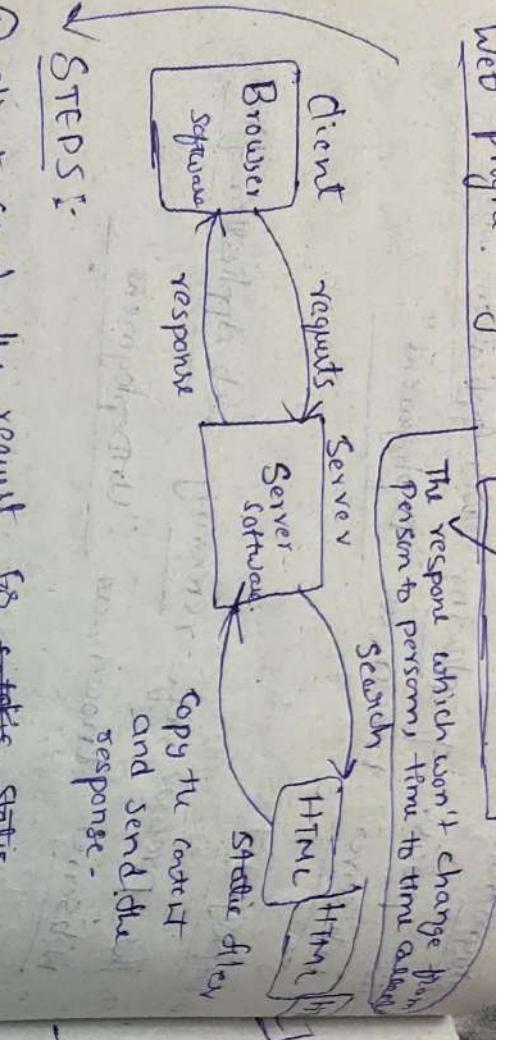


- ① Client send the request for static static file to the server. (Static file → HTML)

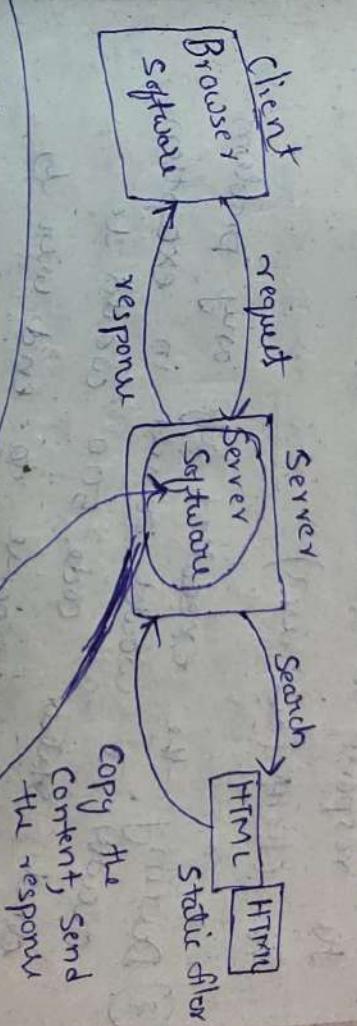
- ② Server searches whether the required resource (static file) is available or not.

- ③ If the required resource is available then server will provide that file as response.

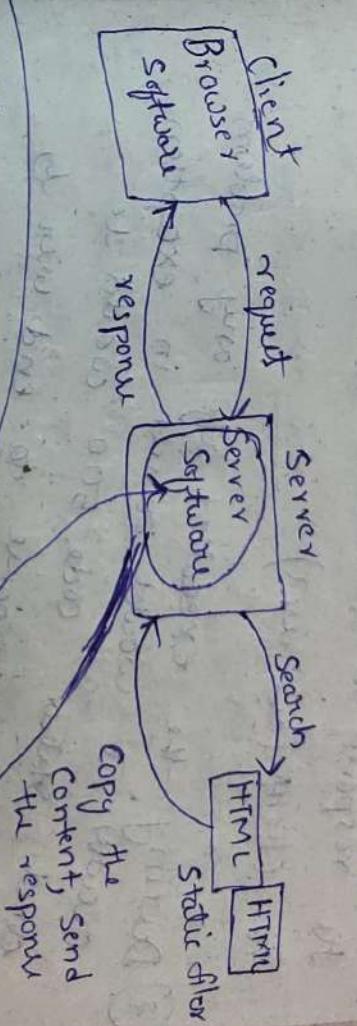
- ④ If the required resource is not available then we will get [404 status] i.e., Not available.



Web programming for Dynamic response:-



Web programming for Dynamic response:-



- NOTE:- To serve static files, no processing is required at the server side, hence unknown always know the static file.

④ If the requested resource is for dynamic information, then webserver will forward the request to web container.

⑤ web container will search for the helper application, which needs to be executed.

⑥ The requested helper application will be executed and it will be sent as the response to webclient and webserver to client (end user).

⑦ During the execution if any problem occurs it would return code 500 and status code 500 to end user by sending a response to end user.

Note At server side of dynamic response, we need helper application for these technologies like Java, JSP, etc.

we need some technologies like Java, JSP, etc.
→ Java ✓
→ JSP ✓
→ Server Page ✓

Q) To design web applications, we have CGI why?

Ans) CGI → Common Graphical Interface.

→ Not platform independent

/ process based technology
/ process is heavy weight component & results low of performance requests are coming from server handle more burden.

for these reason, we reduce burden of "Servlet"

handle → it comes into picture Servlet. Thread come to Picture because of light weight

Servlet :- Server Side technology

→ Servlet is a server side technology.
→ Servlet run top on "Java language".

→ built on top on "Java language".

→ built on top on "Java language".

→ built on top on "Java language".

→ Every request comes to separate thread on respective Servlet object.

Servlet why : JSP comes?

JSP

1) No java knowledge

only presentation skill.

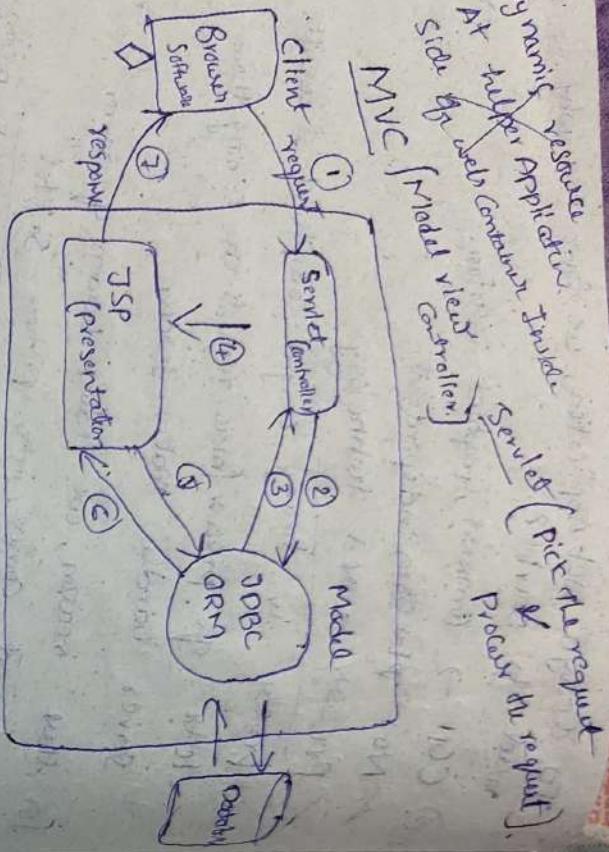
Frontend

① Good Java knowledge

② JSP for mainly in presentation.

Dynamically Application Inside
At Web Container Side
Side of web Container

Servlets (Pic the request & Process the request)
Tomcat Server



Web Application

- X Complete Project with only servlet → Tough X
- X Complete project with any JSP → Tough X
- ✓ Combination of both Servlet & JSP → Best ✓



- * servlet container Technical name → CATALINA
- * JSP container Technical name → JASPER.
- Are Tomcat server?
- It is a web server provided by Apache foundation.
- Every webserver have web Container.

It is responsible to manage & execution
Servlet & JSP.

- Servlet container (CATALINA)
 - To manage & executing servlet components
- JSP container (JASPER)
 - To manage & executing JSP Components.

→ It is an API which helps the programmers

to build web applications.

→ Servlet API provides 2 packages

- javax.servlet.*
- javax.servlet.http.*

a) javax.servlet.*

1. Servlet (I)

2. GenericServlet (AC)

3. ServletConfig (I)

4. ServletContext (I)

5. RequestDispatcher (I)

6. ServletRequest (I)

7. ServletResponse (I)

b) javax.servlet.http.*

- HttpServletRequest (I)
- HttpServletResponse (I)
- HttpSession (I)
- HttpServlet (AC)

By Servlet which want to process the request with the help of "server", then it could be webservice → one of that "Tomcat".

HTTP protocol [standard protocol].

Service protocol signature.

→ follows the protocol [service protocol].
→ Service-protocolname : //ipaddressofserver: PortNo-of-

the-program/resource-name.

HTTP Identification-no. of Server
Porting/
Port-no-id
(8)
Project-
name

Local host is given when Server & APP on same device

HTTP: // local host: 8080 / App1 .

Protocol rule [No change]

We can control (App name)

Protocol of send the request

Now, To send the request along with the data we need to use "Query String".

(App1)

It refers to name-value pair

data which will be sent by the user along with request.

Query string
 is send along
 with request

Query string [name value pair data separated with &]

From the client side to send request these are Big-7 methods:

1. GET
2. POST
3. HEAD
4. PUT
5. PATCH
6. DELETE
7. OPTIONS

1. GET
 2. POST
 3. HEAD

4. PUT
 5. PATCH
 6. DELETE
 7. OPTIONS

1. GET

→ If we want to get information from server then we need Get request.

e.g:- fetching train information

→ usually Get request is only for Read and at server side no Update takes place.

→ e.g:- <http://localhost:8080/App01/test?source=Bengaluru>

destination = Related

→ In Get request, In query string while writing any username / password it results in Not secured.

→ In Get request, Query string are sent in URL

2. POST
- Usually post request is to be for sending huge amount of data. [e.g:- Uploading Resume].
 - Usually POST request write only request and server side it could be update/insert operation taken place.

→ In this won't attached i.e., Query string not be there.

e.g:- <http://localhost:8080/App01/test>

→ In Post request, it could be secure because of Query string not visible. So, it is more

Secure.

- In Post request, Query string is not there, so large volume of data can be sent.
- Book marking & Caching is not supported in Post request.

Imp Q What is idempotent request?

- Q A It is By repeating request multiple times, if there is no change in response. Such type of request called "idempotent request".
- e.g:- Get request is not idempotent request

by repeating the request message many times, but there is no side effect at the Server side, then

it is called safe request

e.g:- & Get request is Safe request. But

Post request is not Safe request.

Note:- GET & Post request you want to

through HTML Page

Request line → Method type
Request line → Resource name
Request line → Protocol version

GET/POST /App-01/test HTTP/1.1

HTTP Request

Request line

Request headers

Request Body

Note:- When we send the request, automatically browser creates the HTTP request and sends it to HTTP protocol. The relevant

information will be assigned to HTTP Request Obj. Then it is sent to server, before those all happen

(from client to server)
HTTP Request (obj)

Request line

Request headers

Request Body

HTTP Response (obj)

Status line [e.g. 200]

Response header

Response Body

Status line

Protocol version.

HTTP/1.1 200 OK

Server side codes

Status code

Description of status code

100 - 199 → Information
200 - 299 → Success
300 - 399 → Redirectional
400 - 499 → Client error
500 - 599 → Server error

Request head:-
→ It will carry the information about the browser, it's supporting encoding & decoding type -

Request Body:-
→ To transmit binary, long complicated input

Response header:- It provides configuration information about the server, Content-type, MIME etc., Browser will use these details to display the response.

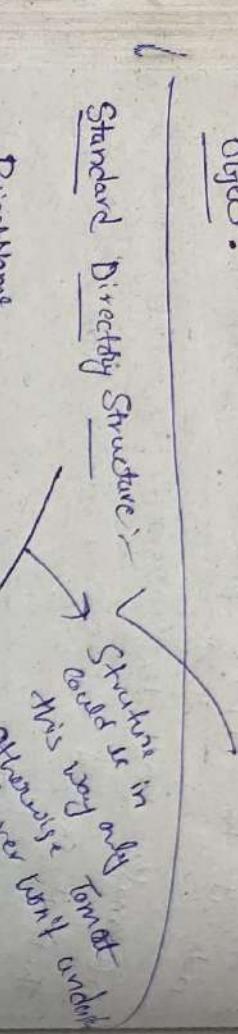
Object.

Automatically ~~HTTP~~ process

Whenever a ~~Java~~, automatically ~~Servlet~~ container will perform following actions:-

c) Depending on URL pattern for dynamic

Standard Directory Structure:



- b) Servlet Instantiation will happen
↳ Constructor will be called and object will be created

c) Servlet Initialization will happen

↳ Servlet Initialization will happen
↳ void init(ServletConfig config) throws ServletException.

d) Servlet request processing
↳ void service(ServletRequest, ServletResponse) throws ServletException, IOException

e) Servlet De-Instantiation will happen
↳ void destroy()

Q) How to create Servlet in Java?

A) There are three approaches.

a) Servlet(T) ✓

b) GenericServlet(AC) ✓

c) HttpServlet(AC) ✓

* for ~~Servlet~~, automatically to the written ~~Servlet~~

Matching should be provided and it should be informed to the container via XML.

Annotation.

http://localhost:8080/firstAPP/test → Avoid 400-499
firstServlet.java ends

XML

<web-app>
 Before Compilation
 Set Path ✓
 Set Classpath ✓
</web-app>

3) Before compilation set path and classpath environment variables

Set path = V-S code path
Set classpath = C:\Tomcat 9.0\lib\servlet-api.jar

* After compilation of first Servlet.java
Copy the .class file to classes folder present under WEB-INF/classes.

4) Now start the server by going to tomcat 9.0/bin
5) Now send the request URL by typing the URL as shown below

- Q) How to send the request to any application?
① Using URL pattern
e.g:- http://localhost:8080/[Name of APP/ContextRoot]/[URL pattern of the Request].
Once the Tomcat engine loads all the projects into execution area, it will create separate object for every project called "ServletContent" object.

- a) Loading
b) Instantiation
c) Initialization
d) Request Processing
e) De-Instantiation.

URL-pattern, then what actions will taken by tomcat engine?

- 1(A) e.g.: `http://localhost:8080/SecondApp/demo`

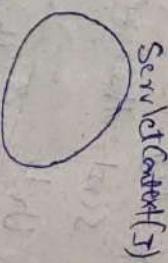
STARTING

- * when Tomcat server is start, then it scans all files present in folder "host" All files what they present in folder "host" know "deployment folder".
- * When Tomcat engine loads all projects into execution Area, it will create separate object every project known "ServletContext"(I).

Implementation class object
Created.

- * After it goes to web.xml, where checks URL pattern to respective project.
- * Then all do by Tomcat engine(webserver), which realize that either static or dynamic.
i.e., it read the URL-pattern for all the dynamic resource for future usage.

- * ABOVE ALL ARE DONE BEFORE request sent (they started we started our Tomcat engine)



Port No.

After now sent the request

- * By Browser we sent, client want to sent the request, it should know the GET Request

Url.

http://localhost:8080/SecondApp/demo

It is not
in our
Container

It is to be
in our
Container

* It sent through internet [N/W].

By protocol http protocol.

Because Client to send the request

* when HTTP picks the request,

HTTP request object

Request Line

Request Header

Request Body

Body

HTTP Request

Because we not
are there

* Once the http Request object hands over
to container after RequestDispatcher

- * After keep request obj, it will switch to the server. It takes SecondApp/demo formatted project want to state.

Because SecondApp/demo is a dynamic resource, for now that purpose Container will do remaining work. Then container same will do remaining work. Then container same web.xml file and identifier like below. And web.xml file and it takes demo and demo with respective from SecondApp/demo, web.xml where

check (web.xml) where

there SecondServlet



to execute ✓

Note Container (CATALINA) Life cycle actions, will be executed under static ✓

the requested URL pattern .Servlet

1. Servlet loading (.class file loading)

Class c = Class.forName ("SecondServlet")

2. Servlet Instantiation (for loaded class create a object).

SecondServlet obj = (SecondServlet)c.
newInstance();

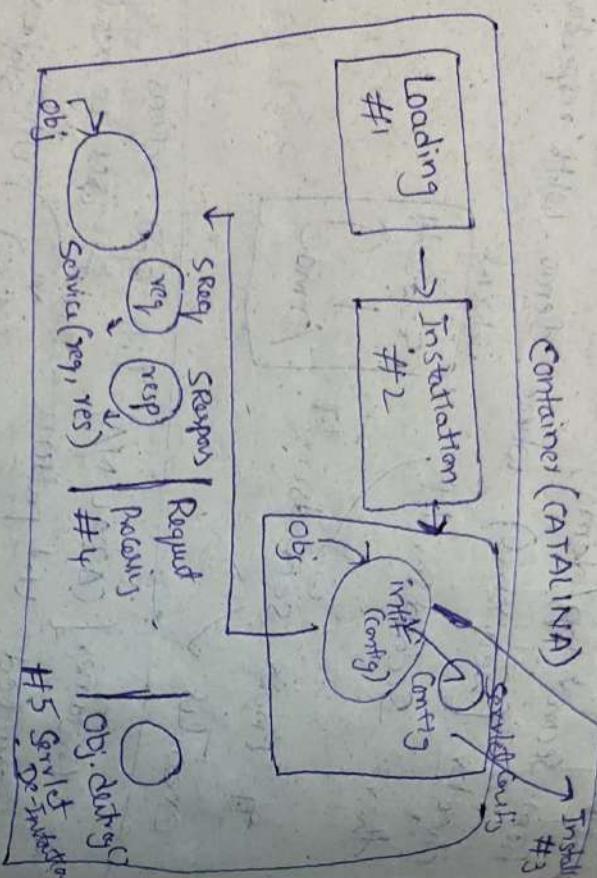
without writing no class is no
without this limitation is
constructed to be in public
constructor with zero argument

Object inject the required values

To get the inputs supplied by the user to Note - the server, we need to use "ServletRequest" object.

obj.init(ServletConfig Config)

Note:- One object injected to one more method it should be called "dependency injection"



4) Request processing (for client request this method call is called)

obj.service(ServletRequest request, ServletResponse res)

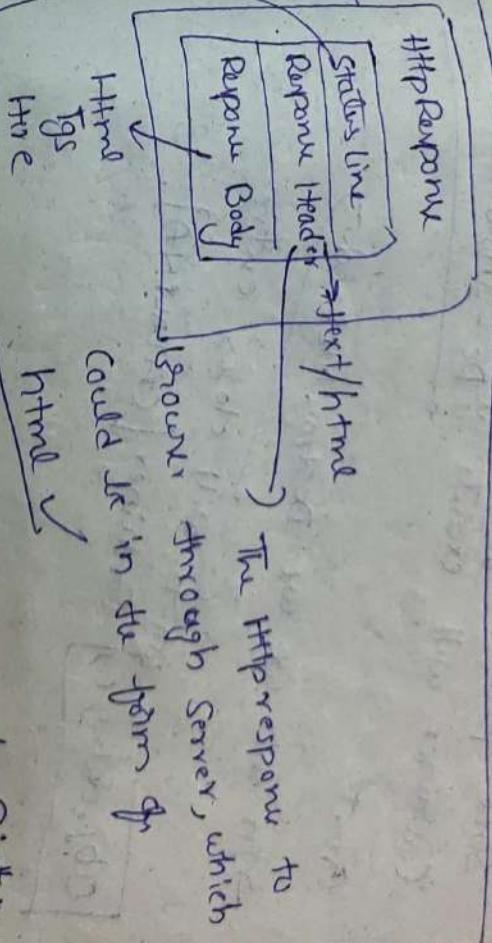
Then After complete protocol call carried to Browser.

Note:- If we want to use System.out.println in requesting processing logic, then output will print on the available on console of tomcat engine.

→ If we want to print to something on browser → Then we use "PrintWriter" obj

To write the output from the application → To write the browser, we need to use "ServletResponse" to the browser, we need to use "ServletRequest" object.

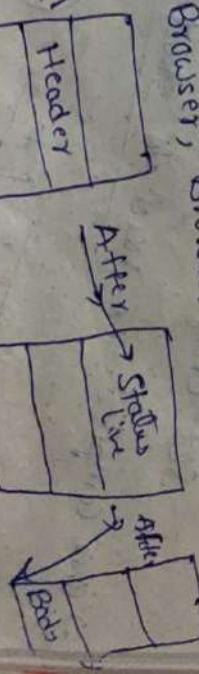
In this internally Printwriter object



) Then After Status Line Come to Picture

) Then After complete protocol call carried

* After handed to Browser, Browser do ~~process~~ first come to



* channel (ut to the Server i.e., Network)

↳ Because HTTP Protocol is "stateless" in nature

Note: HTTP Protocol

5) Servlet De-instantiation -

↳ Deployment

b) once server is stopped (x for the same

resource if the request won't come for

some period of time automatically

Container will execute "De-Instantiation"

event

→ In this we will Destroy Method

↳ which will close entire

↳ Container [CATALINA].

obj.destroy()

Note:-

→ Before sending the request, ~~we need to start server.~~ we need to keep the compiled code in WEB-INF / classes folder.

Set Path =
Set classpath = location of servlet file.

→ Once compiled we need to start server.

↳ Process (creation of output for first request,

→ Servlet loaded ✓

→ Servlet instantiation ✓

→ Servlet initialization ✓

→ Servlet request processing ✓

→ Output for Second Request, only ✓

→ Servlet request processing only ✓

Note: The processing time low for second request as compared for first request because in second

Compared for first request processing is hopped ✓.

Output only servlet request processing is configured in 2 ways

[Tomcat engine can configure in 2 ways]

① XML ② Annotation

Note:- To maintain the uniform response time we

need to use <load-on-startup> tag

↳ write in

↳ XML

e.g:-

<web-app>

< servlet >

< servlet-name > DemoServlet </servlet-name >

< servlet-class > SecondServlet </servlet-class >

< load-on-startup > 10 </load-on-startup >

</servlet >

< servlet-mapping >

< servlet-name > DemoServlet </servlet-name >

Annotation:-

@ webServletPatterns - "1 test of load startup = 10)

which you instead of
write instead of
write file
XML file
with protocol with
XML in code
give these

which want
you write or
to write or
not of servlet class

Limitation of implementing Servlet interface:-

- * If we create a servlet using servlet (T), it is mandatory for us to give the implementation of all the methods of the interface whether it is required or not.

} Because of this length of code increases and also readability decreases.

To overcome this problem we need to use "GenericServlet".

- * GenericServlet has already implemented Servlet interface and it gives body for all the methods of

* If we need to give body only for service, then we can result of which the "body of code" will be less, which increase the readability of application.

Note:-

- * GenericServlet is an best example of "Adapter design pattern".
- * "Adapter" class overloaded in GenericServlet.

* init() is overloaded to solve the problems of direct implementation of Adapter class:- If it is a design pattern allowed to solve the problems of interface method.

- * I know about implementation go for Abstract class
- * Nothing I know about implementation go for Adapter class
- * Partially I know about

Note:-

- * By default response type / content type in "text/html" and also no need to give content directly
- * No Need of writing this line [give content directly goes like this]
- * By writing with GenericServlet it is easy and also codes of line it is reduced.

When a user enters...

* Request Processing: Container will call

- * It will create 2 .class files, **loading ✓**
 - a) fourthServlet.class
 - b) GenericServlet.class

- * Container will load **fourthServlet.class file** for the URL pattern ("disp")

- * Container will create an object for **fourthServlet.class**

- * **[Note]:** for GenericServlet it is **Abstract class** & it can't create an object because it is an **abstract class**.

- * **Installation:** Container will call **init()**, first it will check in **fourthServlet.class** if not, it would check in **GenericServlet**.

- * In **GenericServlet** 2 **init** methods.
- i.e., **init(SC config)**
- init()**
- Container will call init(SC config) which internally makes a call to init() ✓**

- * Can we override **init logic?**

- (A) Yes, we can override, but good practice we can override only **init()** but not **init(SC config)** because In **init(SC config)**, config is local variable and the config variable memory would be gone over the system.

- * **Request (req, resp)** to provide response to the service.

- * **Client** will check in first **fourthServlet.class**, if not it would check in **GenericServlet**.

- * In **Generic servlet**, **service(req, resp)** is available and also it is **abstract** and we need to give the **Body** of this Method inside the **fourthServlet.java**.

- * **Servlet Installation:** Container will call **destroy()** **destroy()** to perform De-Installation action. first it will check in **fourthServlet.class**, if not see there in **fourthServlet.class**, Then it check in **GenericServlet**, there having **destroy()** and will execute it.

- * Why 2 **init()**, **init(SC config)** in **GenericServlet**?

- (A) **init(SC config) → Container**
- init() → Developer.**

- * Why method is used for developer?
init() → best suited for writing

Different ways of creating a servlet

- 1) `Servlet()` → 5 abstract methods
 - 2) `GenericServlet()` → 1 abstract method
 - 3) `HttpServlet(A.C.)`
- When we build web applications, internally HTTP protocol is used for what request type can be one
- a) GET request ✓
 - b) POST request ✓

Q) we can write `GenericServlet` in anyway, then why we need `HttpServlet(A.C.)`?

- (A) → when we are one writing `GenericServlet(A.C.)`, we have only method which is generic for any type of request like `GET, POST` → ~~POST~~ for one method writing, debugging may difficult for any type of request(`GET, POST`)

→ when we deal with `HTTP`, we have a special approach for creating a servlet "HttpServlet".

In `HttpServlet`,

request method type is `GET` → `doGet(request, response)`

request method type is `POST` → `doPost(request, response)`

Note:- By default every it could on `GET` request? for a url in the address bar of browser and hit the request.

- a) type `Browser` and hit the request.
- b) By clicking the hyperlink in a webpage.
- c) Using `<form>` method = "GET" put in `HTML` page
- d) default `<form>` method attribute in `GET` only.

e.g.: `HTTP://localhost:8080/fifthApp/test`

queryString = `http://localhost:8080/fifthApp/test?username=Sashin`

question string = `http://localhost:8080/fifthApp/test?username=Sashin`

queryString for Post request →

which could write in `HTML` page where `form tag` we want to use.

queryString for Post request →

`http://localhost:8080/fifthApp/test`

By using `POST` method.

Note:- By default every it could on `GET` request

Servlet (I)

implements

GenericServlet(A.c)

extends

HttpServlet(A.c)

* Carefully observe that there is no init method in HttpServlet for performing Servlet Initialization,

then it takes from GenericServlet(A.c) because

for HttpServlet(A.c) parent is GenericServlet(A.c)

{ OBSERVE ABOVE DIAGRAM }.

Note:-

Hierarchy of calling the methods

a) public Service(SRreq, SRrep)

b) protected Service(HSreq, HSresp)

c) public void doXXX(HSReq, HSRep).

- Then click on src, right click - servlet
- Then click on src, right click - package
- Give package name
- Give class name
- Then next & select what you want
- Then edit w/ pattern
- Then next, by default select select what you want
- Then finish.

Eclipse setting for JEE

→ open perspective from window

→ open select JEE

→ Then right click new from after others

→ Then dynamic Web Project

→ project name give

→ Set Target runtime to "Tomcat v9.0"

→ web module version 3.0

→ Then click on Next

→ Next & finish

→ Now Below click on Servers

→ Select 9.0 ✓ Tomcat server

→ Then Next & select RequestHeaderApp [Project Name]

→ Then Add & finish.

Note :- User input will sent in the form of QueryString from browser to protocol Server and Container will store in request object.

Note :- When you want to pass to transport from local host to java program in our "file InputStream".

ServletRequest (I) → Just contains
↑ extends
HttpServletRequest (A.C)

Http Request Structure:

- a) Request Line → (request type, resource name, Protocol version)
- b) Request Header → (information about the Browser)
- c) Request Body → (actual data (QueryString))

↳ refer elip & RequestHeader

Working with HttpServletResponse:

Working with PrintWriter getWriter()

1. public abstract PrintWriter getWriter();
throws IOException;
2. To send character type of Response.

↳ To send binary information like videos, images etc response will be "Stream".

1. To send binary images etc response will be "audio, " "stream".
2. To send binary images etc response will be "audio, " "stream".

Abstract class: Servlet → ServletOutputStream through 10 methods

Note :- PrintWriter → only character type of data. ServletOutputStream → we can send both character type and binary type of data.

Q- If the constructor field in Instantiation giving value, then what purpose of having instantiation?

- (A) In Instantiation, there could be new instance which could be zero parametrised, then for this we can't assign value, that's the reason Instantiation comes to picture.

ServletConfig:—

This object will be used to initialize loaded servlet. The there could be servlet config object for every servlet. How you initialize the value of ServletConfig object? we need to configure the container.

① By

In 2 ways

1) XML

2) Annotations

Annotations

By giving value, In XML file simple code

or <init-param> name</param-name> or <param-name> name</param-name> on <param-value> value</param-value> or <param> value</param>

value & which don't understand value

which don't understand value

Note:-
ServletConfig object is unique w.r.t every servlet
→ servlet config object stores the data in the form
→ key, value pair.
Or key we want the db configuration details
→ Assume we want the db configuration details
for a servlet like for Jdbc, username, password

In web.xml, we tag <load-on-startup> which will load the server automatically

NOTE:-
ServletDBCommunicationApp [Project]

→ After adding jar(jar will available to jdk configuration for eclipse, not for tomcat)

→ To make those jar available for tomcat

Server (catalina container) & need to put

Server (catalina container) assembly (WEB/lib/jar)

in deployment assembly

How to do in eclipse

→ Right on project → select properties → click deployment tab

→ choose deployment folder

→ then add

→ build path entries & finish.

→ Then my sql lib & finish.

Note: when we process Servlet communication in different Browsers by request processing but it will happen once browser only fire, connection, loading --

↳ Only one creation when we trying in another browser

Why there is no main method in servlet?

- ④ Actually, JVM will execute based on the main Method class writing Java code.

But in web Application, container will

Perform Loading

Instantiation

Initialization

Request processing

De-Instantiation

So, Thus actions are done (Container), so, we don't need the JVM help.

That's the reason we don't need "main method" in servlet.

Note: Initialization parameter are key-value pair where both key and value are of type String.

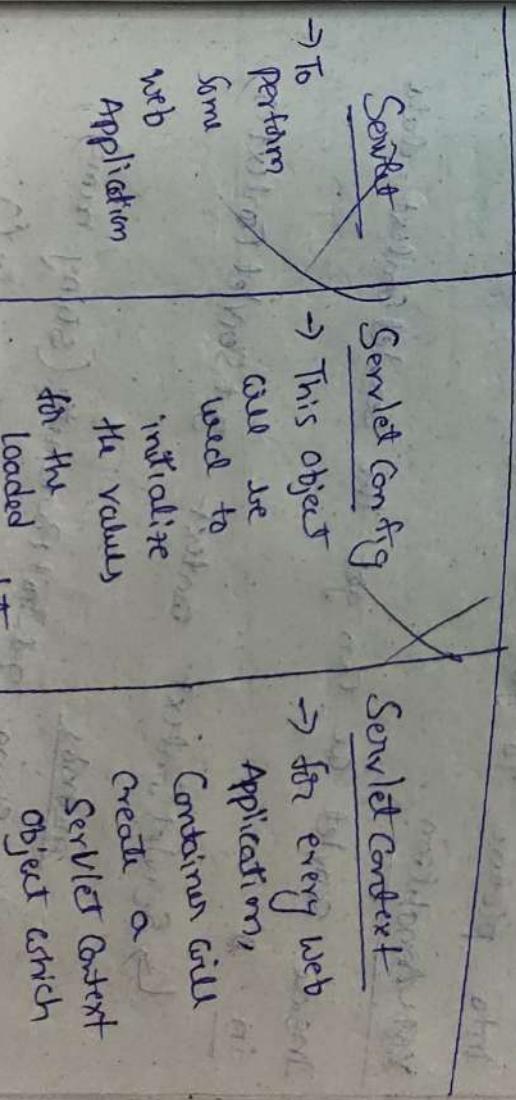
↳ from Servlet we can access these parameters but we can't modify [eg: writer / password]

↳ Since we can't modify we just have only getXXX() but not setXXX().

→ So we say Servlet Initialization Parameter as "Deploy time Constants"

→ for every Servlet we will have only one ServletConfig object to hold its Configuration information.

↳ for every web Application, Container will create a Servlet Context object which



ServletContext(I):

→ for every web application web container will create only one Servlet Context object to hold the configuration details.

→ By using Context object we can get

Configuration information like Context Parameters, requestDispatcher etc.

Parameters,

How to keep the data in ServletContext object?
① we can keep in only 1 way that is through

XML ✓

Annotation supports not available because

Container gets started only Servlet Context object is created and no java code is coming into picture to give the information through Annotation.

Inside servlet we can get the ServletContext data in getServletContext()

↳ ServletContext context = getServletContext();

Methods

```
public String getInitParameter(String name);
```

(Load-on-startup > 5 </load-on-startup>
(which will load the .class files for particular servlets)

</load-on-startup> 5 </load-on-startup>

Note:
* When 2 servlets have load-on-startup then

* When load-on-startup will get chance
a) lower load-on-startup first for execution.

* When 2 servlets have same load-on-startup
then
a) it depends on container (not in hands of the programmer).

* When you give any negative numbers in between load-on-startup then container will not load any of the servlet. So, you HAVE to give ONLY POSITIVE NUMBERS.
AND ZERO VALUE ALSO ALLOWED.

ServletContext

- for every web application, container will create only one ServletContext object to hold the data at application level.

- <context-param>
 - <param-name> </param-name>
 - <param-value> </param-value>
- <init-param>
 - <param-name> </param-name>
 - <param-value> </param-value>
- 1 way to get the context configuration in the XML annotation.
- 2 ways to get the Context object
 - ServletConfig config = getServletConfig();
 - ServletConfig config = getServletConfig();
 - ServletContext context = config.getServletContext();

ServletConfig

- servlet container will create only one ServletConfig object to hold the data at servlet level.

- <init-param>
 - <param-name> </param-name>
 - <param-value> </param-value>
- <init-param>
 - <param-name> </param-name>
 - <param-value> </param-value>
- It can be configured in
 - XML
 - * Annotation

- Approach to get ServletConfig object
 - ServletConfig config = getServletConfig();
 - ServletConfig config = getServletConfig();

(HTTP request, HTTP response)

Why it is loose coupling
because it is known by globally

Not only for particular server
(means tomcat server not only)

For Tomcat Server the implementation class name
for web application TestServlet App

One web application

In Servlet we have 3 types of Scope :-

- 1) request
- 2) session (HttpSession Tracking)
- 3) application / context

3 types of parameters (k,v) possible in Servlet:-

- 1) form parameters (Query String (k,v))
- 2) Servlet Initialization Parameters (ServletConfig (k,v))
- 3) Context Initialization Parameters (ServletContext (k,v))

The Above 3 parameter type are read only

from the servlet can we perform only read

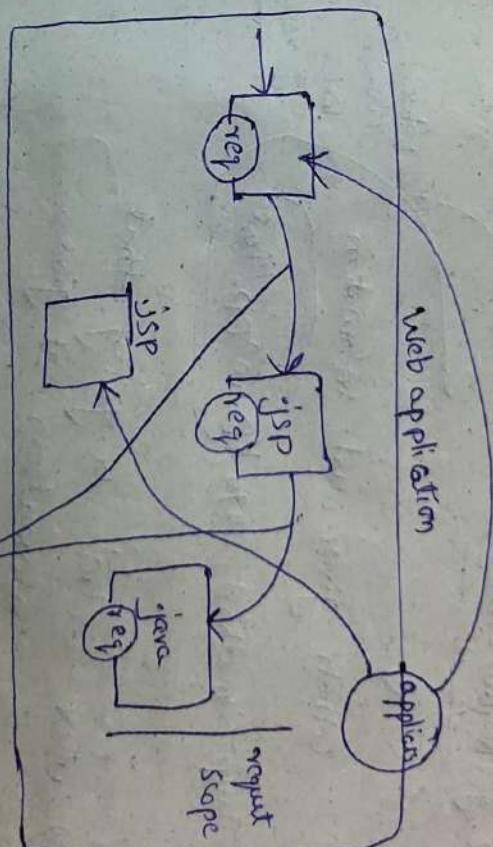
operation, we cannot modify remove values
based on our requirement.

So, parameter type data is not best suited
for sharing the data between components of
the web application.

To resolve this problem we should go for
"attribute" type of data. Now we

/HttpServletRequest Object

→ This scope is maintained by Servlet Request
will be available for all components which
are processing that request.



3) application / context

→ This scope is maintained

by ServletContext object

which shows that
particular data
has been passed
through w.r.t
request scope.

→ The data stored in the
request context object will be available
for all components of the application,
irrespective of request and the user.

Parameter data → both key and value should be String

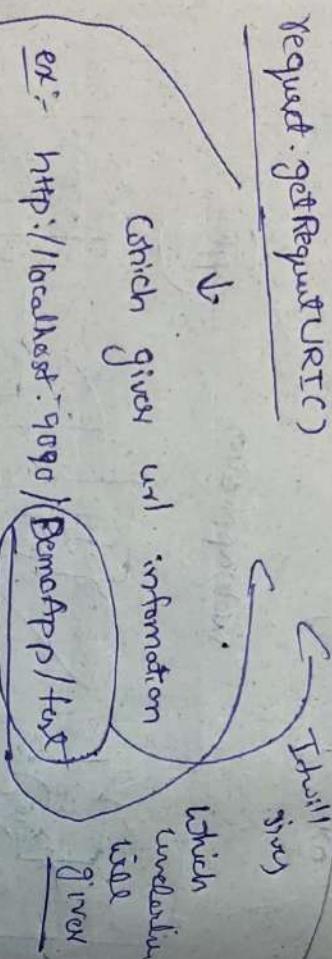
String

attribute data → key should be String,
value can be any object

Host ✓

request.getServletPort()

↳ which gives Port Number
ex:- 8080



↳ where control is going that
information is give.

request.getContextPath()

Hard deployment:-
Creating an web application inside webapps folder
of tomcat and starting the server manually
is called "hard deployment".

↳ which gives only context

Ex:- /DemoApp → only

NB → Host

Smooth deployment:-
Creating an web application outside webapps
folder of tomcat and starting the server
through some additional setup is called
"smooth deployment".

request.getQueryString()

↳ It will give attribute
i.e., name = sodium & power =

webapps(deployment folder) ✓

↳ which give particular resource path

Servlet projects starts from $18G \rightarrow$ visual

JEE \rightarrow Java Enterprise edition

↳ part of Servlet program

MVC design pattern

M - Model
V - View
C - Controller

NEVER WRITE A CODE WITHOUT
BLUE PRINT/DESIGN

Writer object
we cannot write
in HTML it is
not good approach

Note:- we will file name or index.html we don't
when we mapping its automatically loaded. So,
want to mapping all files in index.html.
make

Basic HTML file report Desktop

Refer Request Dispatcher in Eclipse
it shows complete Servlet, JSP, HTML

where there you have an picture of
refer login page

Request Dispatching mechanism:

(forward) \rightarrow By these methods using

✓ include

Request Dispatching

* when Servlet chaining picture
comes into mechanism

mean one one servlet

\rightarrow Servlet chaining
to another servlet

forward

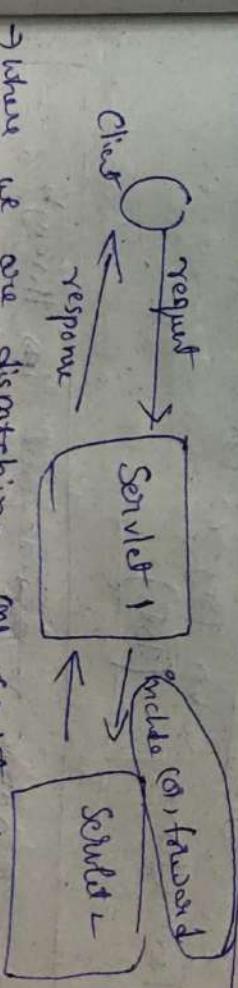
✓ redirecting to one
Controller to another Controller.

Controller

to another Controller

JSP \rightarrow is nothing but HTML file

\rightarrow it is also one Controller



forward Method:- This method forward the request objects to target servlet while getting error regarding path [when sending and response objects to target to the current and return to the current page]

The control doesn't return to the target servlet.

Servlet page after the target servlet is executed

Include Method:- The control return to the current servlet or after the target servlet is executed.

Example:- to include

In Bookingshow, are you due payment, it is redirected to Bookingshow homepage.

JSP → Java Server Pages.

Thymeleaf → New Technology

Note → for static only we perform request Dispatching by html.

By avoid static, for dynamic also

Then Java Server Pages comes into picture

(JSP)

By dynamic response request goes through

while getting error regarding path [when sending check that ("success.html") → ("..\\success.html")]

Session Tracking mechanism:-

As a part of web application, it is necessary to manage the client previous request data at the time of processing later request.

↓

Session usage of request object. ✓
→ request object gets created, when we send the request to the application and it gets destroyed at the time of sending the response, so we can't keep track of the client previous request data using "request object".

Session usage of Context object ✗

Session:-

→ It refers to the amount of time the client spend with the server.

* when we send request to application, the object

which is HttpServletRequest.

* If multiple users run the application, then we need to write a code to separate "HTTP Session" object.

* To manage the session objects
"Session Tracking Mechanism".

"Session Tracking"

(4) Session Tracking Mechanism ✓

a) HttpSession Tracking Mechanism ✓

b) Cookie Session Tracking Mechanism ✓ [Best]

c) URL-Rewriting Tracking Mechanism ✓

d) Hidden form field Session Tracking Mechanism

These 3 are Sun Application.

e) HttpSession Tracking mechanism ✓

Q Difference b/w request.getSession() &

request.getSession(false)

(A) Both the method will return HttpSession

Object only. Container will check whether Session object exists w.r.t the particular user or not.
if exist it will return same object,
otherwise it will create a new one.

getSession(true)

getSession(false)

Container will check whether Session object exists w.r.t the particular user or not. If existed

exists w.r.t the particular user then return the object.

→ To create a cookie.

* HttpSession objects are treated manually and unique ID in the form of Hexadecimal value called Session ID.

* Container will prepare Session id value in the form of key called "JSessionID". Container will create a cookie and attach id to send it in a response to the session every time when the interaction happens b/w client and server.

Note:- we can destroy the HttpSession object

manually using the method called

public void invalidate()

b) Cookie Session Tracking Mechanism ✓

→ we need to create for every request data

w.r.t every user.

After creating a cookie, we need to send

this cookie along with response object.
→ These cookies will be exchanged b/w client and server in the form "request-response" object

→ To add the cookie to response we

* public void addCookie(cookie c)

URL-Rewriting Tracking Mechanism!

The mechanism is same as "HttpSession Tracking".

Mechanism, it won't be stored inside cookie.

But session id will be appended
to the URL every time when the request-
response happens b/w client and server.

e.g:- <form method="get" action="response.encodeURL(
('1Second') +")"

which is not html file
written on html content encode
because frame content encode
understand in sync code
this mechanism
Important

So, use your
whole html
file

Note: In Realtime Application, we don't use direct
technologies. we use frameworks which have
support of URL-Rewriting Mechanism only.

URL Pattern Types:-

a) exact match URL pattern → eg:- /test
b) longest path URL pattern → eg:- /controller/controller/
c) URL pattern by extension → eg:- *.do

d) Default URL pattern → /.

NOTE: when none of the servlet is getting
mapped then default servlet will get

a chance once again

Q) In real time project which type of URL pattern means?
A URL pattern by extension [e.g:- *.do]

In Spring MVC to inherit servlet called
"DispatcherServlet (FC)". we config our it through
"URL-pattern by extension".

Configuring welcome page / landing pages after
submitting a request:-

* It increases the easiness of the use of the
web application for the end user.

* configuration can be done in XML only
for html files, JSP files

- * index.html & as Both Default files when execution
- * index.jsp

As per JEE specification, when we are configuring JSP pages inside <welcome-file>, we need to just specify the file name not with "/" instead go ↓

eg:-
<welcome-file>index.jsp</welcome-file>[val]
<welcome-file>/index.jsp</welcome-file>[val]

↓
refer welcome app in Eclipse

In web app, you want to get info by clicking simply go for doGET method