Dt : 26/10/2023(day-1)

define Application?

=>Set-of-programs collected together to perform defined action is known as Application.

Types of Applications:

=>Applications are categorized into four types:

1.Stand-Alone-Applications

2.Web Applications

3.Enterprise Applications

4.Mobile Applications

1.Stand-Alone-Applications:

=>The Applications which are installed in one computer and performs actions in the same Computer are knonw as Stand-Alone-Applications or DeskTop Applications or Windows Applications

=>Based on user interaction the Stand-Alone-Applications are categorized into two types:

(i)CUI Applications

(ii)GUI Applications

(i)CUI Applications:

=>The Applications in which the user interacts through Console (CommandPrompt) are known as CUI Applications.

(CUI - Console User Interface)

(ii)GUI Applications:

=>The Applications in which the used interacts through GUI Components are known as GUI Applications.

(GUI - Graphical User Interface)

=>we use the following to construct GUI Components:

AWT - Abstract Window Toolkit

Swing

JavaFX

**imp*

2.Web Applications:

=>The applications which are executed in Web Environment or Internet Environment are known as WebApplications or Internet Applications.

=>we use the following three technologies to construct WebApplications

(i)JDBC

(ii)Servlet

(iii)JSP

3.Enterprise Applications:

=>The applications which are executing in distributed environment and depending on the features like "Security","Load Balancing" and "Clustering" are known as Enterprise Applications or Enterprise Distribute Applications.

=>To develop Enterprise Applications we use Java Frameworks and Java Tools.

4.Mobile Applications:

=>The appliactions which are executed in Mobile environment are known as Mobile Applications.

==========

****imp***

define Storage?(Syllabus)

=>The location where the data is available for access,is known as Storage.

Types of Storages:

=>According to Java Application development the storages are categorized into four types:

1.Field Storage

2.Object Storage

3.File Storage

4.Database Storage

Dt : 27/10/2023(day-2)

1.Field Storage:

=>The memory generated to hold 'single data value' is known as Field Storage.

=>The primitive datatypes(byte,short,int,long,float,double,char,boolean) will generate Field Storages.

**imp*

2.Object Storage:

=>The memory generated to hold 'group members' is known as Object Storage.

=>The NonPrimitive datatype(Class,Interface,Array,Enum) will generate Object Storage.

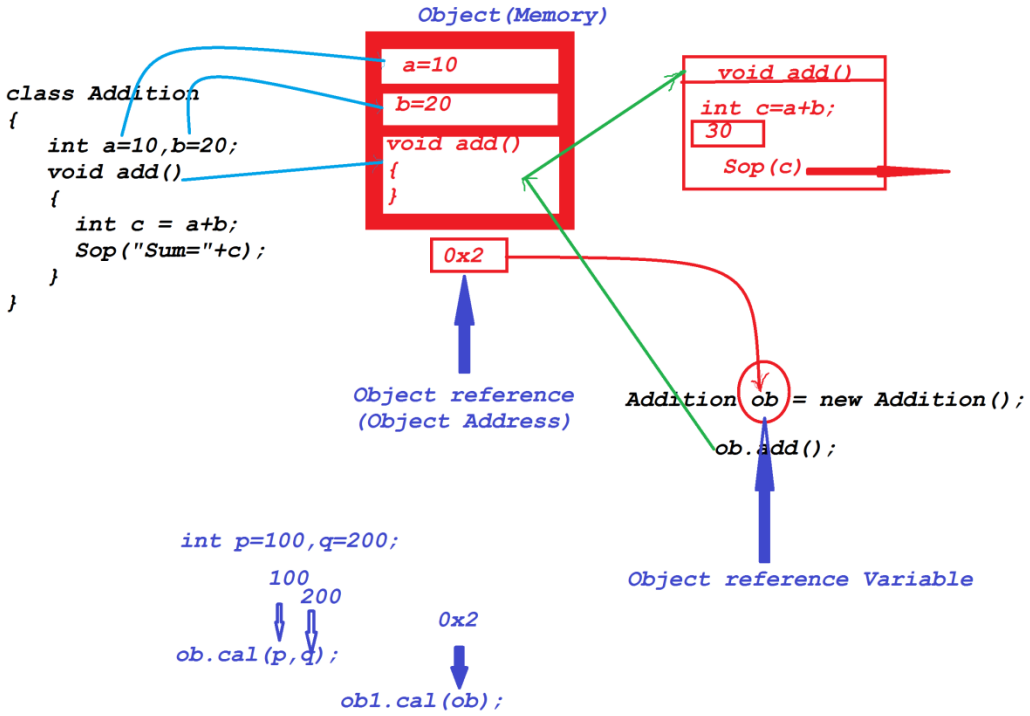
Ex:

```
class Addition
{
```

```
  int a=10,b=20;
  void add()
  {
    int c = a+b;
    Systenm.out.println("Sum="+c);
  }
}
```

```
Addition ob = new Addition();
```

Diagram:



---

faq:

wt is the diff b/w

(i)Object

(ii)Object reference

(iii)Object reference Variable

(i)Object:

=>The memory which is generated to hold instance members of Class is known as Object.

(ii)Object reference

=>The address where object is generated is known as Object reference.

(iii)Object reference Variable:

=>The NonPrimitive datatype variable which is holding Object reference is known as Object reference Variable or Object Name.

==========

Dt : 30/10/2023

**imp*

Summary of Objects created from CoreJava:

1.User defined Class Objects

2.String-Objects

3.WrapperClass-Objects

4.Array-Objects

5.Collection<E>-Objects

6.Map<K,V>-Objects

7.Enum<E>-Objects

==========

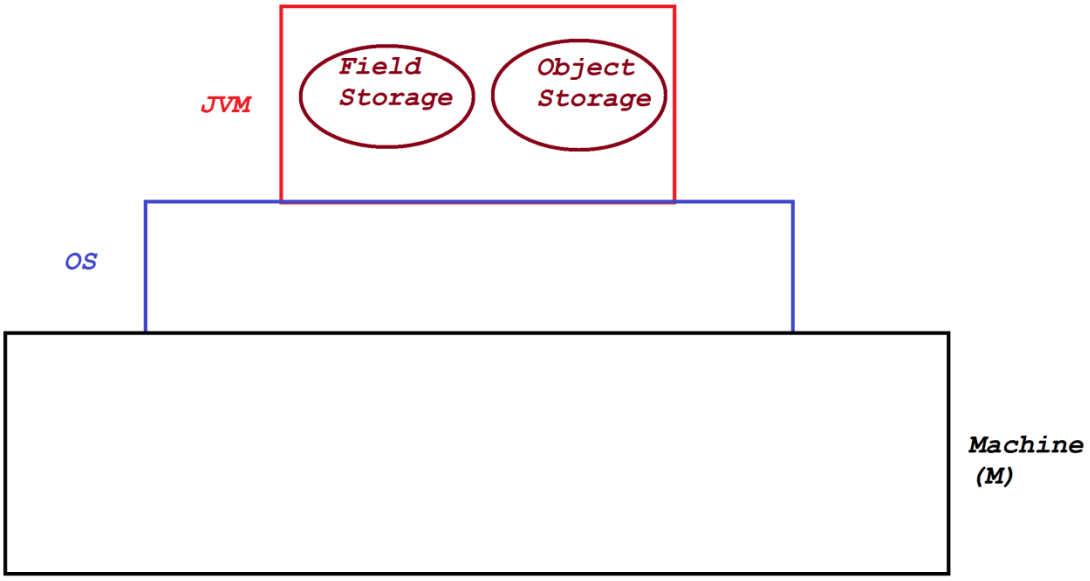
Note:

=>The Field-Storages and Object-Storages which are generated part of JVM will be destroyed automatically when JVM ShutDowns.

=>when we want to have permanent storage for application,then we must use any one of the following:

=>File Storage

=>Database Storage



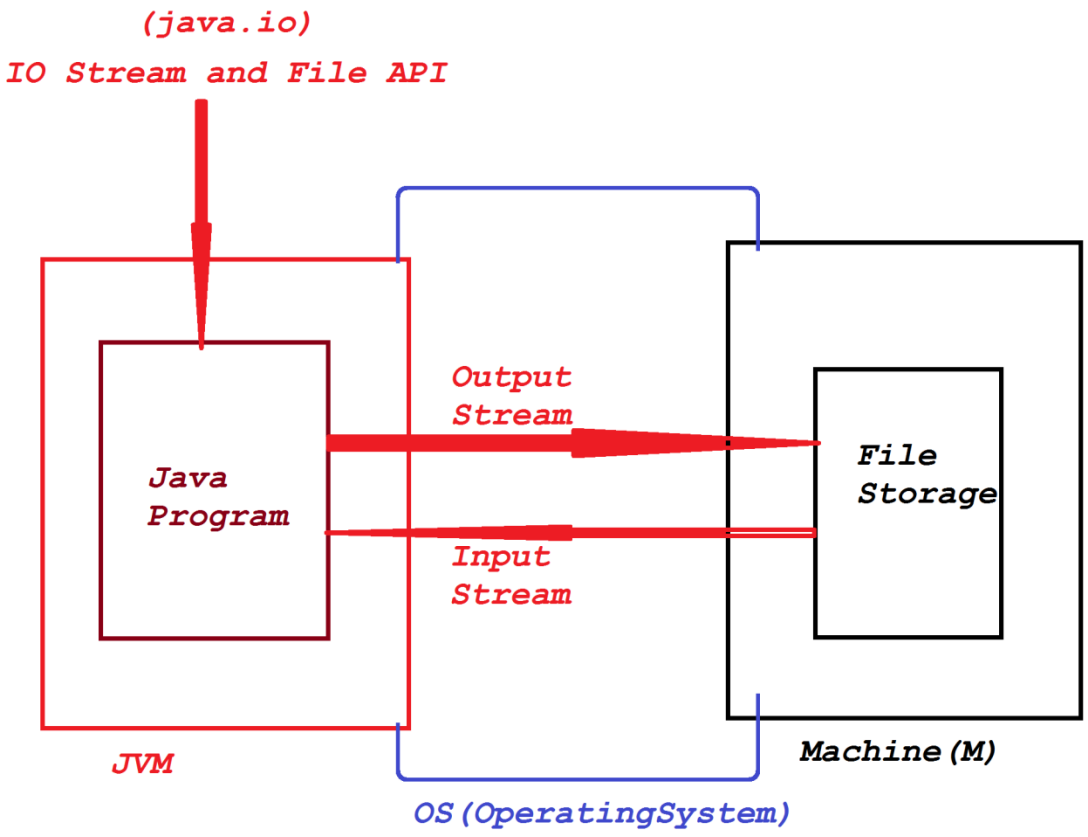
==========

3.File Storage:

=>The smallest permanent storage of ComputerSystem which is controlled and managed by the OperatingSystem is known as File Storage.

=>In the process of establishing communication b/w JavaProgram and File-Storage,the JavaProgram must be constructed using 'classes and Interfaces' available from 'java.io' package(IO Stream and FIle API).

Diagram:



==========
===

Dis-Advantages of File-Storage:

1.Data Redundancy

2.Data Inconsistency

3.Data Integrity problem

4.Data Sharing Problem

5.Data Security Problem

1.Data Redundancy:

=>The process in which File-Storage will store duplicate data,known as Data Redundancy or Data Replication.

2.Data Inconsistency:

=>The process in which File-storage will different data fields,known as Data Inconsistency.

3.Data Integrity problem:

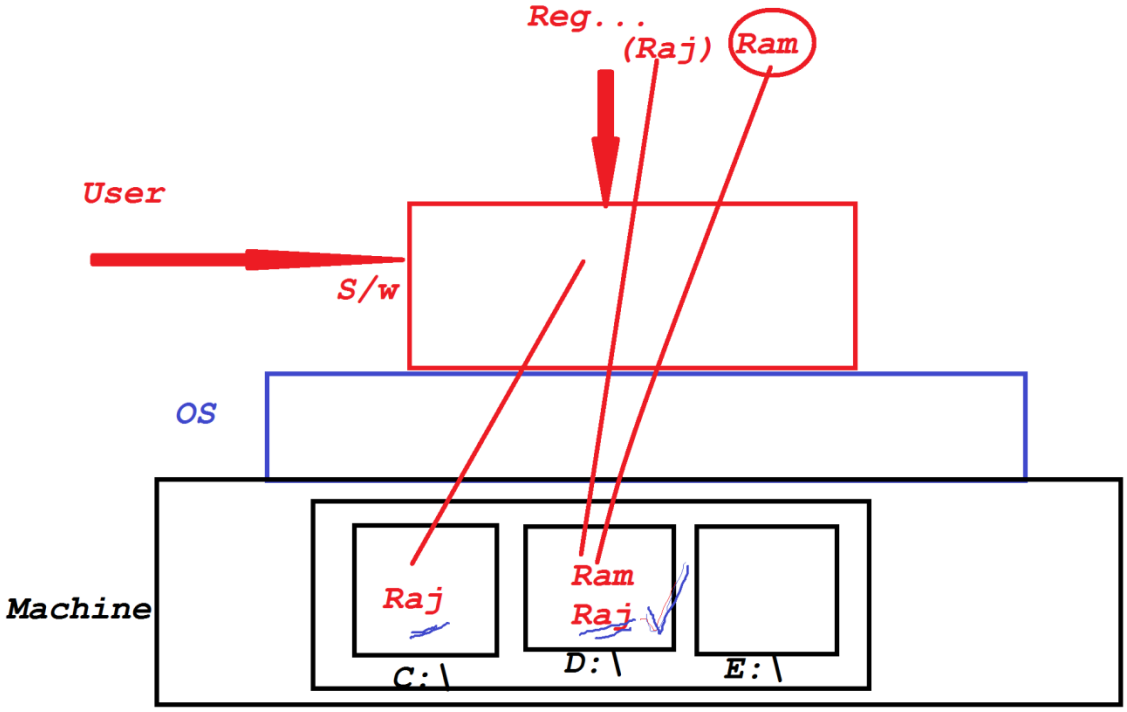
=>File-Storage will not support Data Integrity,which means combining data available in different locations.

4.Data Sharing Problem:

=>File-Storage will not Support File Sharing,which means File-Storage not available to multiple users at-a-time

5.Data Security Problem:

=>Data is not Secure in File-Storage



==========

Note:

=>Because of DisAdvantages of File-Storage,we are not prefering to take File-storage as Back-end for applications.

==========

Dt : 31/10/2023

**imp*

4.Database Storage:

=>The largest permanent storage of ComputerSystem,which is 'downloaded and installed' from externally is known as Database.

=>In the process of establishing communication b/w JavaProgram and Database product,the JavaProgram must be constructed using 'Classes and Interface' available from 'java.sql' package(JDBC API) and the JavaProgram must take the support of JDBC-driver

Diagram:

==========

faq:

define driver?

=>The small s/w program used by OperatingSystem to establish connection b/w two end-points

Ex:

Audio driver

Video driver

N/W driver

....

faq:

define JDBC driver?

=>The driver which is used to establish connection b/w JavaProgram and Database product is known as JDBC driver.

(Java DataBase Connectivity driver)

Types of JDBC drivers:

=>JDBC drivers are categorized into four types:

1.JDBC-ODBC bridge driver(Type-1)

2.Native API driver(Type-2)

3.Network Protocol driver(Type-3)

4.Thin driver(Type-4)

Note:

=>In realtime application development we use Type-4(Thin) driver.

==========

faq:

define API?

=>API stands for 'Application Programming Interface' and which is a PlatForm for programmers to develop applications using Language or Technology or Frameworks

=>According to JavaLanguage,API means package.

=>The following are some important APIs(packages):

CoreJava:

java.lang - Language package(default package)

java.io - Input/Output Stream and Files package

java.util - Utility package

java.net - Networking package

AdvJava:

java.sql - Database package(JDBC API)

javax.servlet - Servlet programming package(Servlet API)

javax.servlet.jsp - JSP Programming package(JSP API)

==========

**imp*

Making ComputerSystem environment ready for executing JDBC Applications:

step-1 : Download and Install Database product(Oracle)

step-2 : Perform Login Process to Database project

step-3 : Create table with name Customer57

(cid,cname,ccity,cstate,pincode,mid,phno)

```
create table Customer57(cid varchar2(10),cname varchar2(15),
ccity varchar2(15),cstate varchar2(15),pincode number(10),
mid varchar2(25),phno number(15),primary key(cid));
```

step-4 : Insert min 5 Customer details

```
insert into Customer57 values('E111','Ram','Hyd','TS',612345,
                    'r@gmail.com',9898981234);
```

step-5 : Copy DB-Jar file into User defined folder(DeskTop)

Location of DB-Jar file,

C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib

ojdbc6 - Oracle11

faq:

define JAR file?

=>JAR stands for Java Archieve and which is compressed format of more class files.

step-6 : Find Database PortNo and ServiceName

Find PortNo and ServiceName from "tnsnames.ora" file

C:\oraclexe\app\oracle\product\11.2.0\server\network\ADMIN

PortNo : 1521

ServiceName : XE

==========

Dt : 1/11/2023

**imp*

JDBC API:

=>'java.sql' package is known as JDBC API and which provide 'Classes and Interfaces' to Construct JDBC Applications.

=>'java.sql.Connection' interface is the root of JDBC API.

=>The following are some important methods from 'Connection' interface:

1.createStatement()

2.prepareStatement()

3.prepareCall()

4.getAutoCommit()

5.setAutoCommit()

6.setSavepoint()

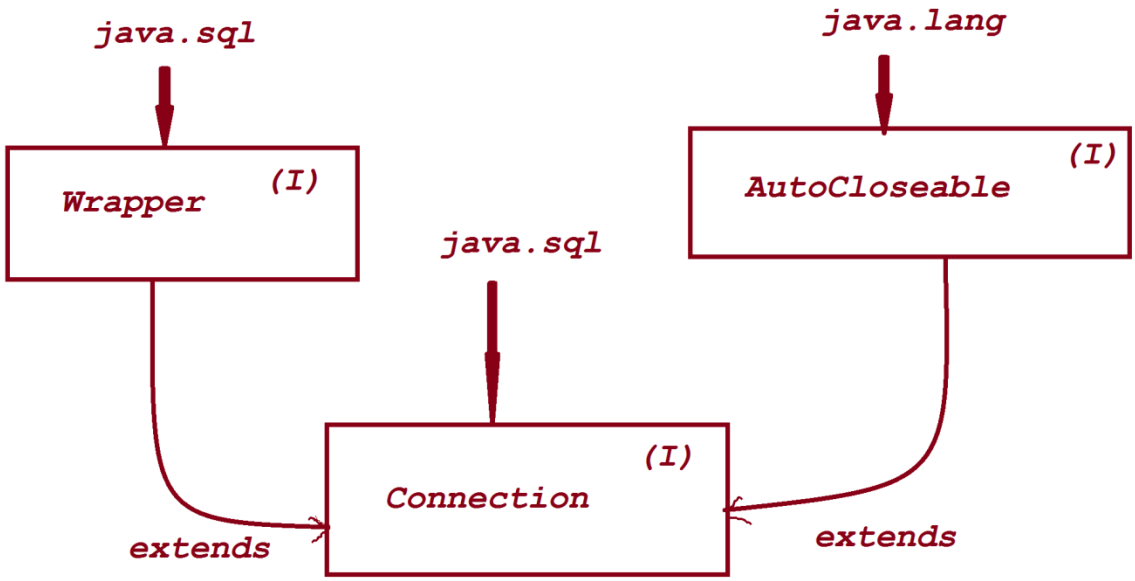
7.releaseSavepoint()

8.commit()

9.rollback()

10.close()

Hierarchy of 'Connection' Interface:



=>we use getConnection() method from 'java.sql.DriverManager' class to create implemention object for 'Connection' interface.

=>This getConnection() method internally holding Anonymous Local InnerClass as implementation class of 'Connection' interface.

Method Signature of getConnection():

public static java.sql.Connection getConnection(java.lang.String,

java.lang.String, java.lang.String) throws java.sql.SQLException;

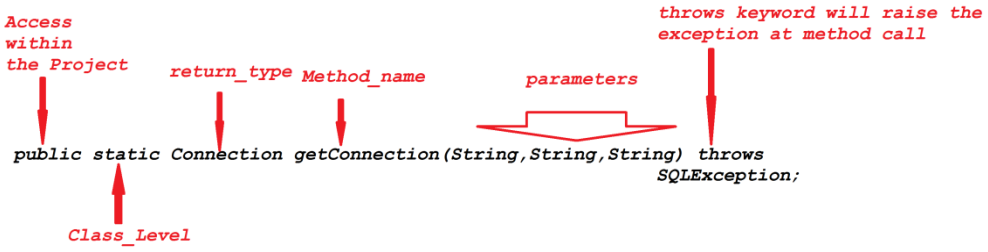
syntax:

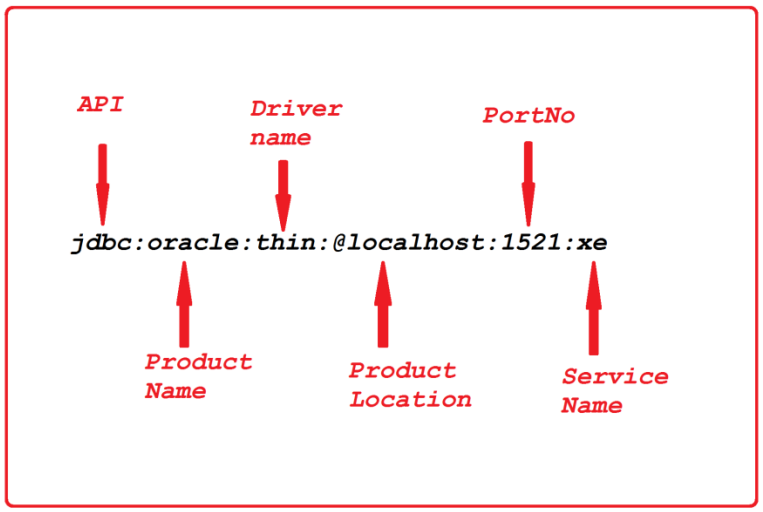
Connection con = DriverManager.getConnection("DBURL","UName","PWord");

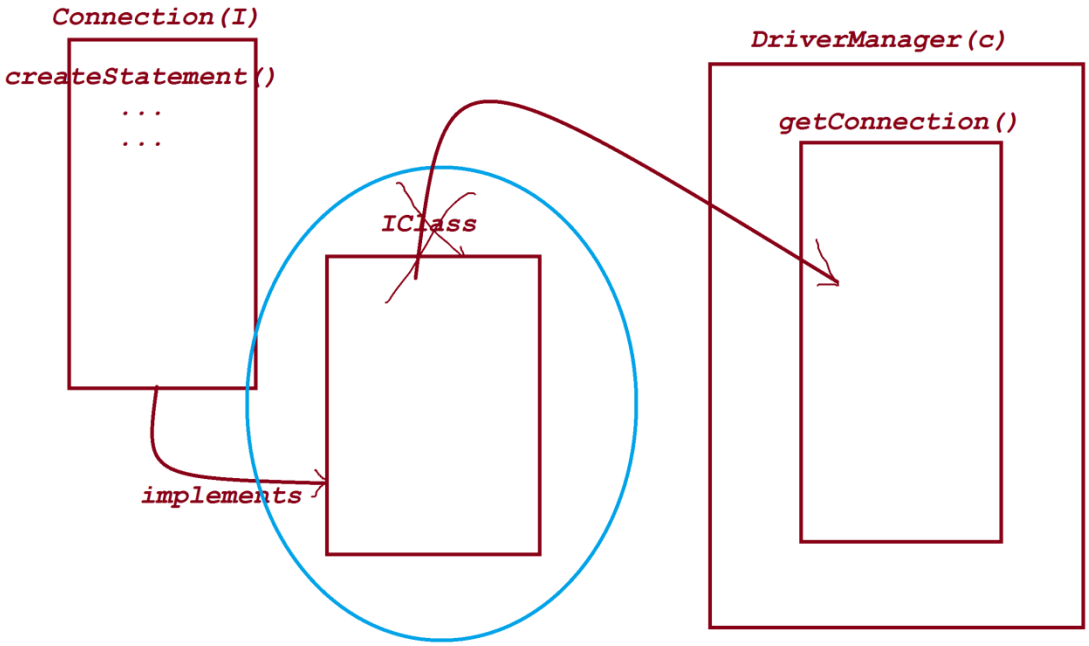
DBURL => jdbc:oracle:thin:@localhost:1521:xe

UName => system

PWord => manager




API
Driver name
PortNo
jdbc:oracle:thin:@localhost:1521:xe
Product Name
Product Location
Service Name


Connection(I)
createStatement()
...
...
DriverManager(c)
getConnection()
IClass
implements

Dt : 2/11/2023

*imp

JDBC Statements:

=>JDBC statements will specify the type of actions performed on database product.

=>These JDBC Statements are categorized into three types:

1.Statement

2.PreparedStatement

3.CallableStatement

1.Statement:

=>'Statement' is an interface from java.sql package and which is used to execute normal queries without IN Parameters.

(Normal Queries maens Create,Insert,Select,Update,Delete)

=>we use createStatement() method from 'Connection' interface to create implementation object for 'Statement' interface.

=>This createStatement() method internally holding Anonymous Local InnerClass as implementation class of 'Statement' interface.

Method signature of createStatement():

```
public abstract java.sql.Statement createStatement()
                                throws java.sql.SQLException;
```

syntax:

Statement stm = con.createStatement();

=>The following are some important methods from 'Statement':

(i)executeQuery()

(ii)executeUpdate()

(i)executeQuery():

=>executeQuery() method is used to execute select-queries.

Method Signature:

public abstract java.sql.ResultSet executeQuery(java.lang.String)

throws java.sql.SQLException;

syntax:

ResultSet rs = stm.executeQuery("select-query");

(ii)executeUpdate():

=>executeUpdate() method is used to execute NonSelect-queries.

Method Signature:

public abstract int executeUpdate(java.lang.String)

throws java.sql.SQLException;

syntax:

int k = stm.executeUpdate("NonSelect-query");

****imp***

we use the following steps to establish connection to Database Product:

step-1 : Loading driver

step-2 : Creating Connection

step-3 : Preparing Statement

step-4 : Executing query

step-5 : Closing the connection

==========

****imp***

Creating JDBC application using IDE Eclipse:

step-1 : Open IDE Eclipse,while opening name the workspace and click 'Launch'.

step-2 : Create Java Project

step-3 : Add DB-Jar file to Java Project

RightClick on JavaProject->Build Path->Configure Build Path->Libraries->

select 'Classpath' and click 'Add External Jars'->Browse and select DB-Jar

file from user defined folder->open->Apply->Apply and Close

step-4 : Create package in 'src'

step-5 : Create class in package to write JDBC code

Program : DBCon1.java

```
package test;
import java.sql.*;
public class DBCon1
{
     public static void main(String[] args)
     {
          try
          {
           //step-1 : Loading driver

Class.forName("oracle.jdbc.driver.OracleDriver");
           //step-2 : Creating Connection
           Connection con = DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1521:xe",
                    "system","manager");
           //step-3 : Preparing Statement
           Statement stm = con.createStatement();
           //step-4 : Executing query
           ResultSet rs = stm.executeQuery
                    ("select * from Customer57");
           System.out.println("****Customer-Details****");
           while(rs.next())
           {
                System.out.println(rs.getString(1)+"\t"+
                  rs.getString(2)+"\t"+rs.getString(3)+
```

```
                    "\t"+rs.getString(4)+"\t"+
                    rs.getInt(5)+"\t"+rs.getString(6)+"\t"+
                    rs.getLong(7));
             }//end of loop
             //step-5 : Closing the connection
             con.close();
            }//end of try
            catch(Exception e)
            {
             e.printStackTrace();
            }
      }
}
```

==========

Assignment:

Step-1 : Create table with name Product57

(code,name,price,qty)

step-2 : Insert Min 5 product details

step-3 : Construct JDBC Application to display all products

==========