JDBC Statement :-

The JDBC *Statement, CallableStatement,* and *PreparedStatement* interfaces define the methods and properties that enable you to send SQL or PL/SQL commands and receive data from your database.

1. Interface Statement:- It is useful when we are using static SQL statements at runtime. The Statement interface cannot accept parameters.
2. PreparedStatement :- It is usefull when we plan to use the SQL statements many times. The PreparedStatement interface accepts input parameters at runtime.
3. CallableStatement :- It is use when we want to access the database stored procedures. The CallableStatement interface can also accept runtime input parameters.

Strucuture of Statement Interface:-

```
Connection conn=null;
Statement stmt = null;
try {
Class.forName("");

   stmt = conn.createStatement( );
   . . .
}
catch (SQLException e) {
   . . .
}
finally {
   stmt.Close();
}
```

Structure of PreparedStatement Objects:-

```
PreparedStatement pstmt = null;
try {
   String SQL = "Update Employees SET age = ? WHERE id = ?";
   pstmt = conn.prepareStatement(SQL);
   . . .
}
catch (SQLException e) {
   . . .
}
finally {
   pstmt.Close();
}
```

Note :- (?) Requires runtime value to update SQL command on database Connection.

CallableStatement Objects:-

We needs to create a procedure in oracle/mysql/sql-server.

Structure :-

Create procedure proc_name

@variable1 datatype,

@variable2 datatype

AS

Begin

//sql command

End

Structure of CallableStatement :-

```
CallableStatement cstmt = null;
try {
   String SQL = "{call proc_name (?, ?)}";
   cstmt = conn.prepareCall (SQL);
   . . .
}
catch (SQLException e) {
   . . .
}
finally {
   cstmt.Close();
}
```

Parameter Passing Methods:-

setInt():- Set Integer type value.

setFloat():- Set Float type value.

setDouble :- Passes the double type value.

setString():- Passes the value of String type.

setLong():- Passes the value of type long.