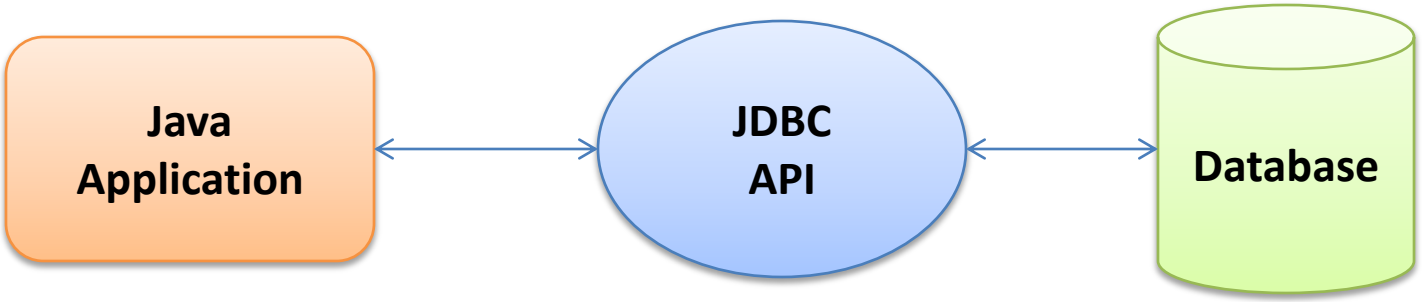
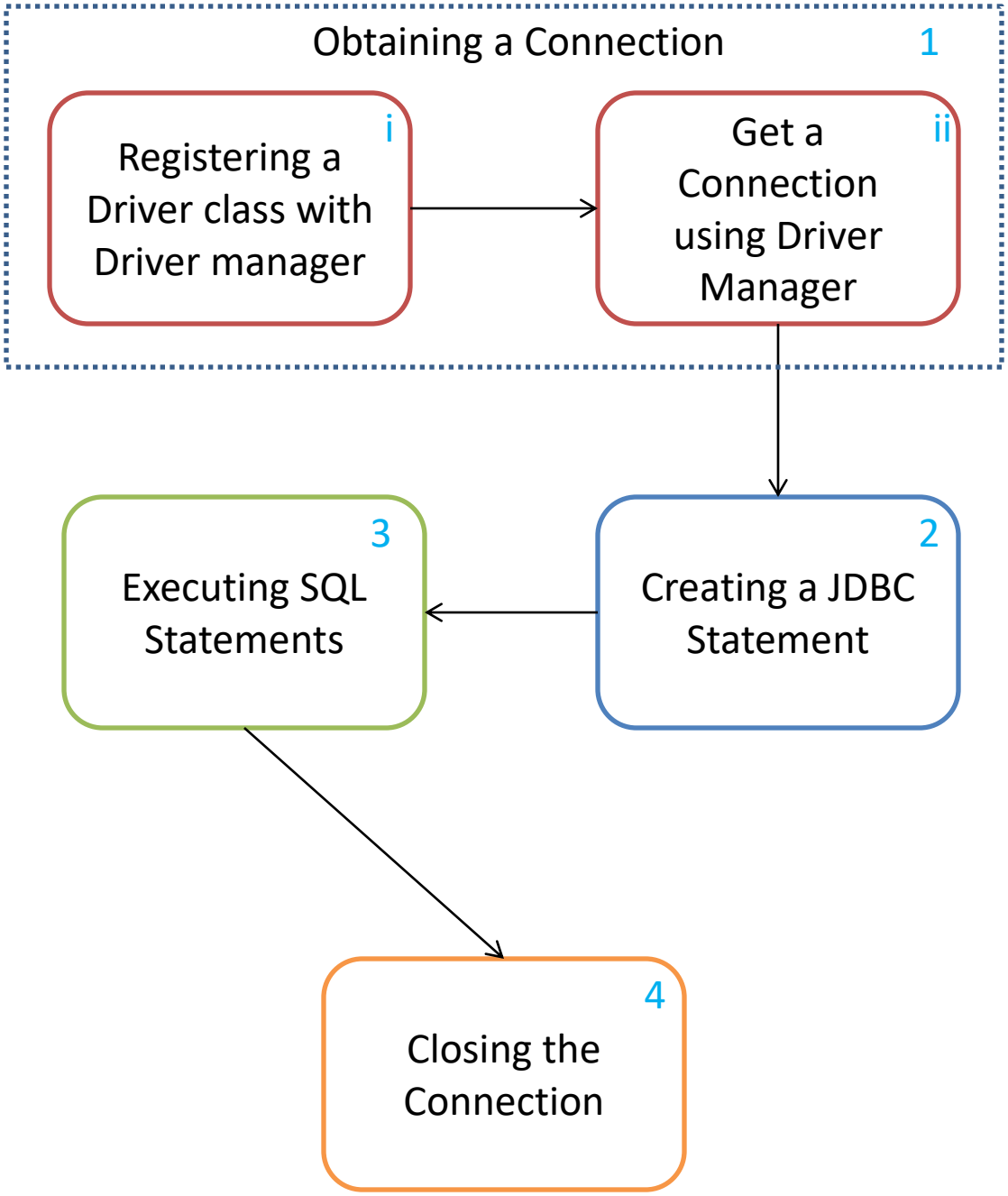
Database Programming using JDBC

- **JDBC** – Java DataBase Connectivity.
- Java Database Connectivity defines a set of classes that can execute SQL statements through applications.
- JDBC APIs are used by java applications to communicate with any database.
- In otherwords,we use JDBC to provide communication between application and database.



- There are some basic steps in JDBC database connectivity
  - **Step 1:** Obtaining a connection.
  - **Step 2 :** Creating a JDBC statement.
  - **Step 3:** Executing SQL statements.
  - **Step 4:** Closing the Connection.


Obtaining a Connection
1
i
Registering a Driver class with Driver manager
ii
Get a Connection using Driver Manager
2
Creating a JDBC Statement
3
Executing SQL Statements
4
Closing the Connection

Step 1: (Obtaining a Connection)

- In this step, we can obtain a connection with the database by registering a driver class with driver manager and establishing a connection using driver manager.

✓ Invoke '**class.forName("driver class name")**' method to load or register a driver class with driver manager.

Example:

```
class.forName("oracle.jdbc.driver.OracleDriver");
```

✓ Invoke '**getConnection("url","username","password")**' method to create/establish connection with the database.

Example:

```
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","password");
```

Step 2: (Creating a JDBC Statement)

- After the connection made, we need to create the JDBC statement object to execute SQL statements.
- To create JDBC statement object ,

✓ Invoke '**CreateStatement()**' method on the current connection object.

- The following code shows how to create a statement object.

Example:

```
Statement stmt=con.CreateStatement();
```

Step 3: (Execute SQL Statements)

- After the statement object created, it can be used to execute SQL statements by using **executeUpdate()** or **executeQuery()** method.

✓ The executeQuery() method is used to execute only SELECT SQL statements.

- ✓ The executeUpdate() method is used to execute all other SQL statements such as INSERT,UPDATE and DELETE except SELECT statement.

  Example:

```
//using executeQuery()
String query="select * from student";
resultset res=stmt.executeQuery(query);
//using executeUpdate()
 String query="insert into student values(501,'abc',19)";
 stmt.executeUpdate(query);
```

Step 4: (Closing the connection)

- After executing all the required SQL statements, we need to close the connection and release the session.
- Invoke '**close()**' method to close/release the connection.

  Example: con.close();