Unit 5

Contents

- Applet Fundamentals
- Applet Tag
- Applet life-cycle
- Passing parameters to applets
- Graphics Programming
- JDBC drivers
- JDBC connection

Applets

- Small Java programs that can be downloaded and executed in a Java enabled Web browser.
- Transported via internet.
- Perform arithmetic operations, display graphics, play sounds, create animations, play interactive games etc.
- Two types of applets

1. Local applet
2. Remote applet

Applets

- Applet developed locally and stored in a local system is known as local applet.
- Does not require any internet connection.
- Remote applet is developed by someone else and stored on a remote computer.
- If our computer is connected to internet, we can download a remote applet via internet and runs it.

Building applet code

- Two classes Applet and Graphics needs to be imported.
- The Applet class is contained in the java.applet package.

Example Program

Applet Tag

- <applet> tag is used to start an applet from an HTML document and from an appletviewer.

```
<Applet>
[CODE BASE=codebase URL]
CODE = appletfile
[ALT=alternate Text]
WIDTH=pixels HEIGHT=pixels
[ALIGN=alignment]
[<PARAM NAME=attributename1 VALUE=value1>]
[<PARAM NAME=attributename2 VALUE=value2>]
</Applet>
```

Applet Tag

[CODE BASE=codebase URL] – This is an optional attribute which specifies the directory that will be searched for the applet’s executable class file.

CODE = appletfile – This is a required attribute that gives the name of the file that contains the applet compiled subclass.

Applet Tag

[ALT=alternate Text] – This is an optional attribute which specifies a short text message that should be displayed if the browser understands the applet tag, but cannot run Java applets.

WIDTH=pixels HEIGHT=pixels – These are required attributes that give the initial size in pixels of the applet display area.

Applet Tag

[ALIGN=alignment] - This is an optional attribute that specifies the alignment of the attribute.

[<PARAM NAME=attributename1 VALUE=value1>] – This attribute is used to pass parameter to applets.

Passing Parameters to Applets

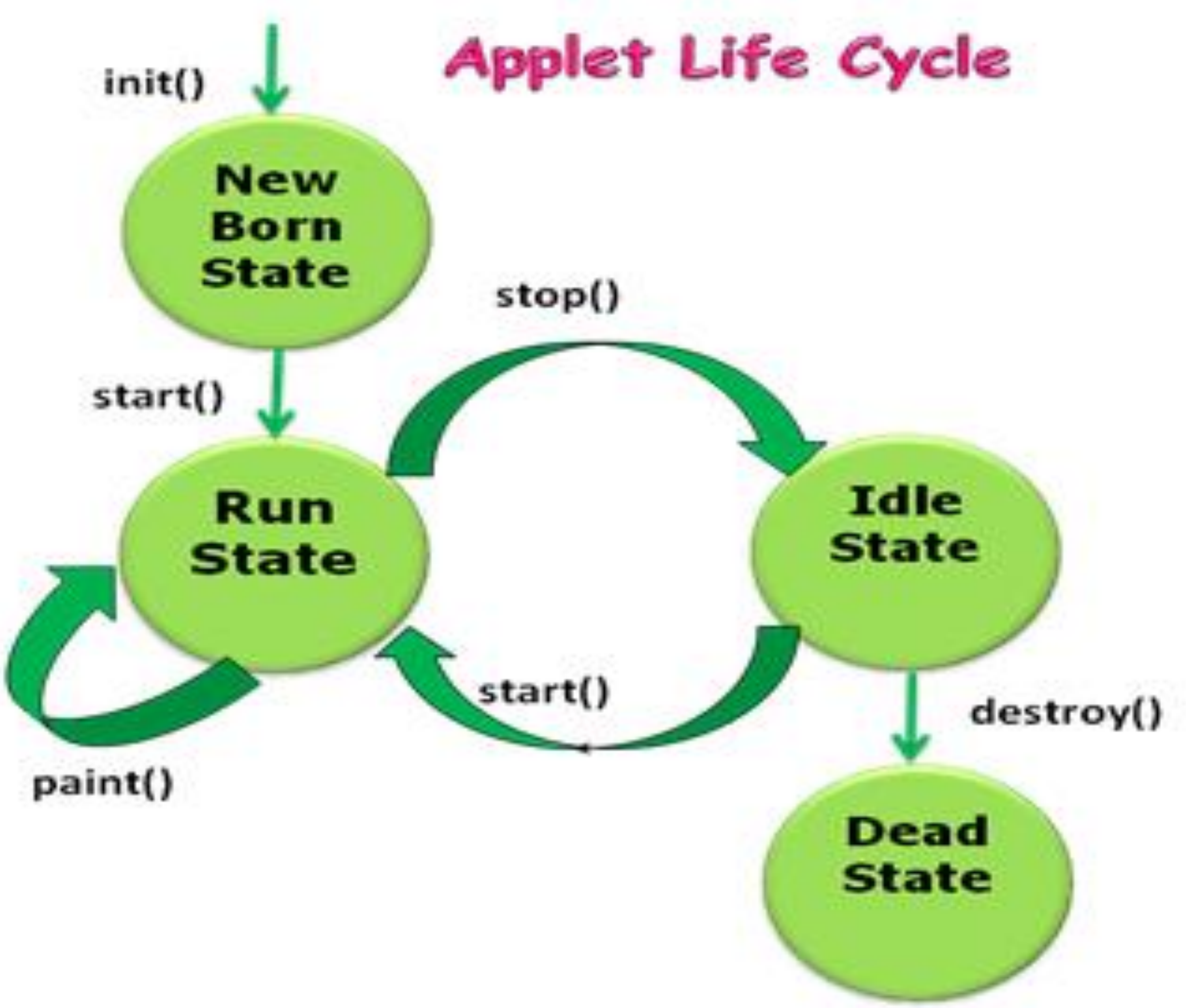
[<PARAM NAME=attributename1 VALUE=value1>]

- Each <PARAM> tab has a NAME attribute and a VALUE attribute.
- Example

```
<Applet>
<PARAM NAME=COLOR VALUE="Red">
</Applet>
```

Example Program

Life Cycle of an Applet



Life Cycle of an Applet

- init() – this method is called first. This is where we should initialize variables.
- start() – this method is called after init(). This is also called after an applet has been stopped. While init() is called once(first time an applet is loaded), start() is called each time an applet's HTML document is displayed on screen. For example, a user leaves a Web page and comes back.

Life Cycle of an Applet

- paint() – This method is called each time the applet is damaged. If any other window covers the applet, the window system calls paint() method.
- update() – The default update method in class Applet first fills an applet with default background color and then calls paint().

Life Cycle of an Applet

- stop() – This method is called when a Web browser leaves the HTML document containing the applet. When the user returns to the page applets are restarted using start() method.
- destroy() – This method is called when the runtime environment determines that the applet needs to be removed completely from memory.
- repaint() – Calling repaint() in turn calls update() method.

Life Cycle of an Applet

- Getting Input from the user – Example Program

Graphics Programming

- Every applet has its own area of the screen called canvas.
- Java’s co-ordinate system has the origin(0,0) in the upper left corner.
- Positive x-values are to the right and positive y-values are to the bottom.
- The value of co-ordinates x and y are in pixels.

Graphics Class - Methods

1. drawArc() – draws an arc.
2. drawLine() – draws a straight line.
3. drawOval() – draws an oval.
4. drawPolygon() – draws a polygon.
5. drawRect() – draws a hollow rectangle.
6. drawRoundRect() – draws a hollow rectangle with rounded corners.

Graphics Class - Methods

7. drawString() – displays a string.
8. fillArc() – draws a filled arc.
9. fillOval() – draws a filled oval.
10. fillPolygon() – draws a filled polygon.
11. fillRect() – draws a filled rectangle.
12. fillRoundRect() – draws a filled rectangle with rounded corners.
13. getColor() – retrieves the current drawing color.
14. setColor() – sets the drawing color.

Graphics Class - Methods

Example program for drawLine(), drawRect(), fillRect(),drawRoundRect(), fillRoundRect()

- drawLine() method takes 2 pairs of co-ordinates and draws a line between them.
- drawRect() method takes 4 arguments, first two represents the x and y co-ordinates of the top left corner of the rectangle and remaining 2 represents the width and height of the rectangle.

Graphics Class - Methods

- fillRect() method is similar to drawRect() except that it draws a solid rectangle.
- fillRoundRect() and drawRoundRect() method takes 6 arguments, the two extra arguments representing the width and height of the angle of corners.

Graphics Class - Methods

Example Program for drawOval()

- drawOval() method can be used to draw a circle.
- Takes 4 arguments. The first 2 represent the top left corner of the imaginary rectangle and the other 2 represents the width and height of the oval itself.
- If the width and height are same, the oval becomes a circle.

Graphics Class - Methods

Example Program for drawArc() and fillArc()

- drawArc() and fillArc() method takes 6 arguments. The first 4 is same as that of drawOval(), and the last 2 represent the start angle and sweep angle (no: of degrees) around the arc.
- Zero degrees is at 3'O Clock and degrees increase in a counter-clockwise direction.

Graphics Class - Methods

Example Program for drawPolygon() and fillPolygon()

- Both have the form drawPolygon(int[], int[],int) and fillPolygon(int[], int[], int).

Color Class

- We can use static variables in Color to specify a number of common colors. (Example: Color.blue)
- Color(int,int,int) – takes red green and blue integers between 0 and 255.
- Color(float,float,float) – takes 3 float values between 0.0 and 1.0 for red, green and blue.

Java Data Base Connectivity-JDBC

4 Types of JDBC Drivers

1. JDBC-ODBC bridge driver
2. Native-API driver (partially Java driver)
3. Network Protocol driver (fully Java driver)
4. Thin driver (fully Java driver)

JDBC-ODBC bridge driver

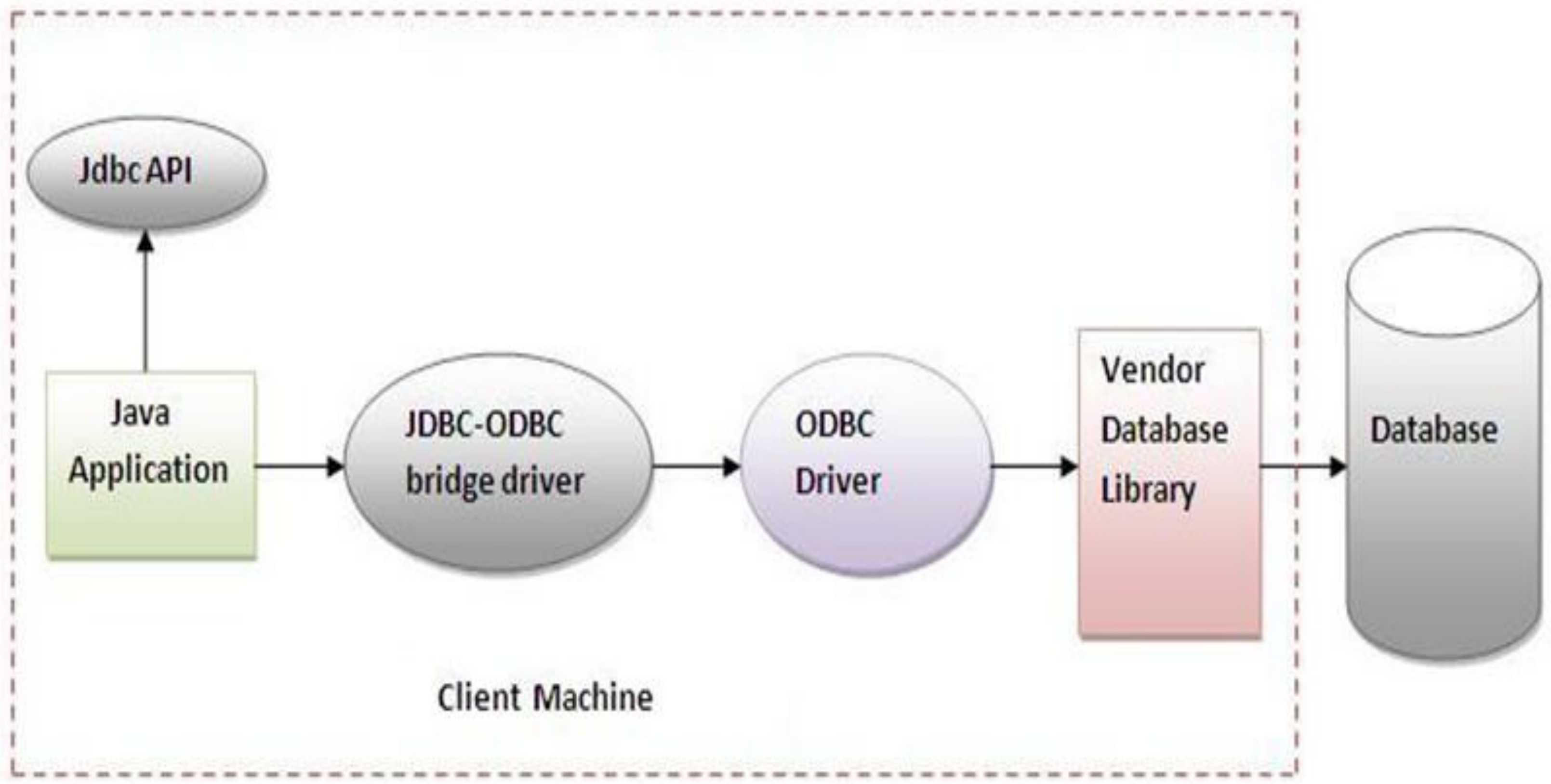


Figure- JDBC-ODBC Bridge Driver

Native-API driver (partially Java driver)

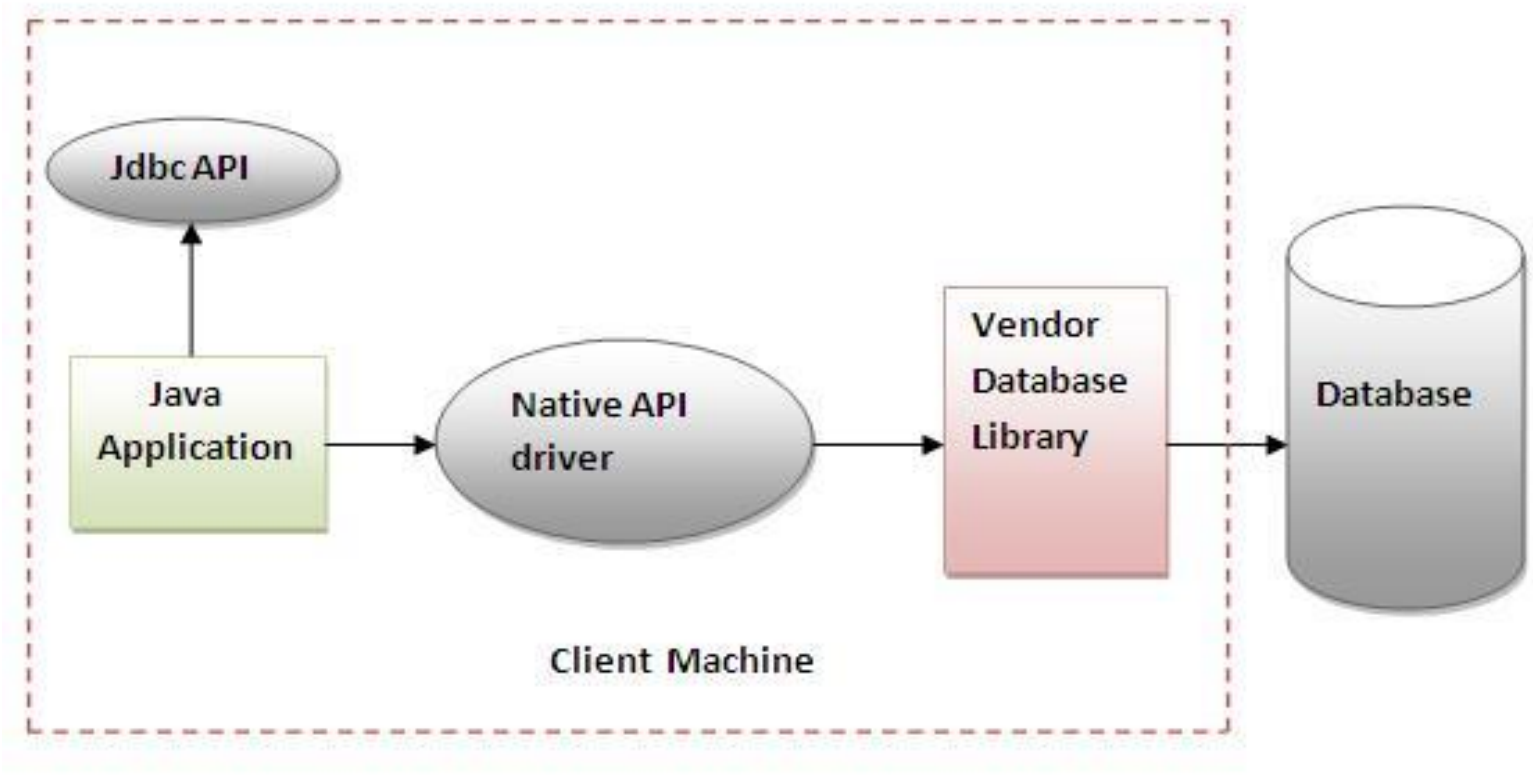


Figure- Native API Driver

Network Protocol driver (fully Java driver)

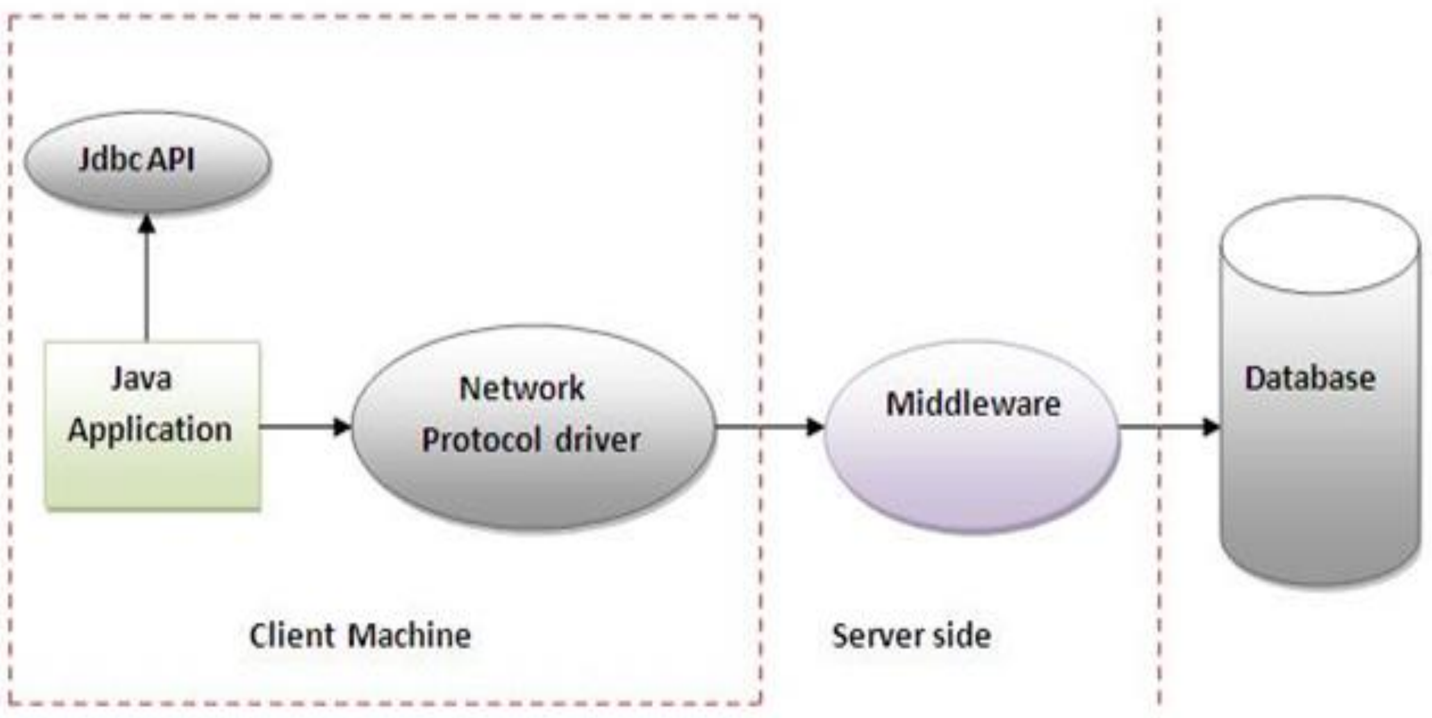


Figure- Network Protocol Driver

Thin driver (fully Java driver)

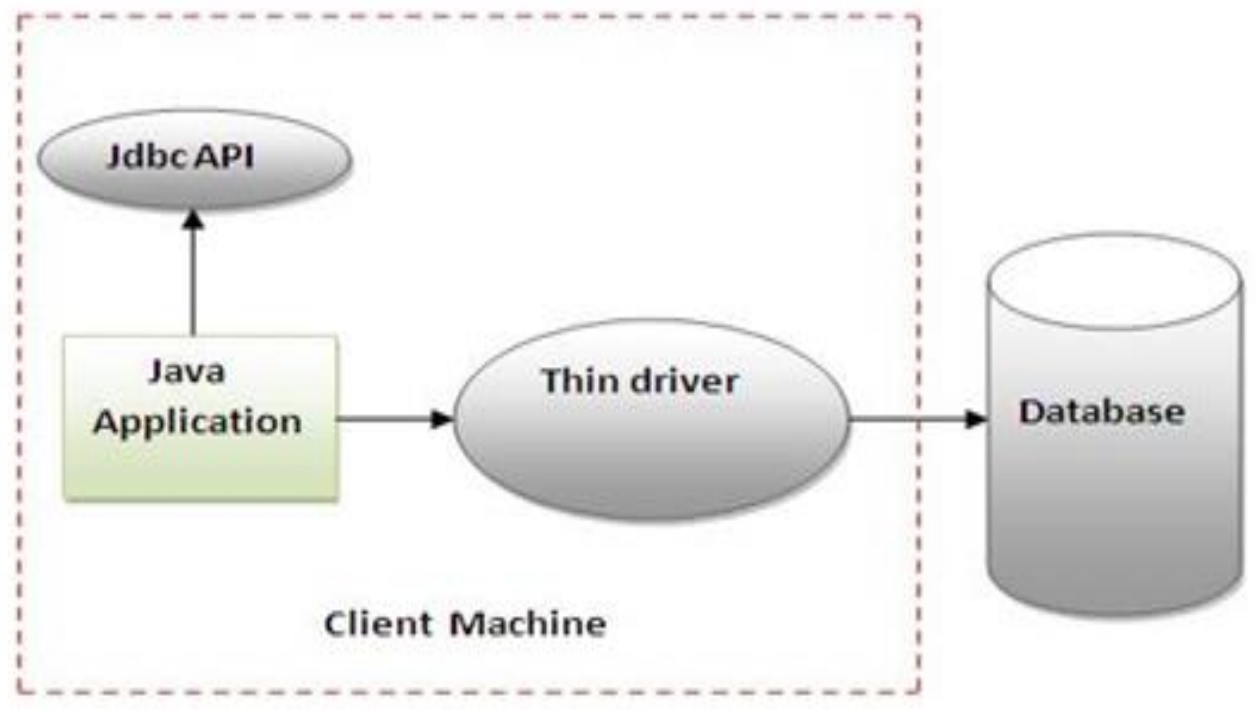


Figure- Thin Driver

Steps to connect to a Database

1. Register the driver class.
2. Create the connection object.
3. Create the Statement object.
4. Execute the query.
5. Close the connection object

1. Register the driver class

- The forName() method of Class is used to register the driver class.
- This method is used to dynamically load the driver class.
- Syntax of forName() method

```
Class.forName(“com.mysql.jdbc.Driver")
```

2. Create the connection object

- getConnection() method of DriverManager class is used to establish connection with the database.
- Syntax of getConnection() method.

```
Connection conn = DriverManager.getConnection(dbURL,
username, password);
```

Example

```
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost3306/mysql",“root","");
```

3. Create the Statement object

- createStatement() method of Connection interface is used to create statement.
- Example of createStatement() method.

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

4. Execute the query

- The executeQuery() method of Statement interface is used to execute queries to the database.
- This method returns the object of ResultSet that can be used to get all the records of a table.

Example

```
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next()){
System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
```

5 Close the connection object

- By closing connection object statement and ResultSet will be closed automatically.
- The close() method of Connection interface is used to close the connection.

Example

```
con.close();
```

DriverManager Class

- DriverManager class acts as an interface between user and drivers.
- It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver.

DriverManager Class - Methods

- public static void registerDriver(Driver driver): To register the given driver with DriverManager.
- public static void deregisterDriver(Driver driver):To deregister the given driver with DriverManager.
- public static Connection getConnection(String url):To establish the connection with the specified url.
- public static Connection getConnection(String url,String userName,String password): To establish the connection with the specified url, username and password.

Connection Interface

- Provides many methods for transaction management like commit(), rollback() etc.
- By default, connection commits the changes after executing queries.

Connection Interface - Methods

- public Statement createStatement(): creates a statement object that can be used to execute SQL queries.
- public void setAutoCommit(boolean status): is used to set the commit status. By default it is true.
- public void commit(): saves the changes made since the previous commit/rollback permanent.
- public void rollback(): Drops all changes made since the previous commit/rollback.
- public void close(): closes the connection and releases a JDBC resource immediately.

Statement Interface - Methods

- Provides methods to execute queries with the database.
- public ResultSet executeQuery(String sql): is used to execute SELECT query. It returns the object of ResultSet.
- public int executeUpdate(String sql): is used to execute specified query. It may be create, drop, insert, update, delete etc.

Programs

- Insert Records
- Update Records
- Delete Records

End of Unit 5