# Object Oriented Programming

## Assignment - 3

K. Varsha
B191437
R.NO: 27
E3-CSE-C1 (311)

1) JDBC and AWT components:

Describe JDBC, its components and the steps involved in JDBC connectivity. Discuss Abstract window toolkit (AWT) components & their significance in GUI Programming?

JDBC (Java DataBase Connectivity) is a java-based API that provides a standard interface for connecting & interacting with relational databases. It allows java programs to access databases, executes SQL queries & manipulate database data.

Components of JDBC:

1. Driver manager: Manages a list of database drivers it is used to establish a connection to the database.
2. Driver: A database-specific implementation that communicates with the database JDBC drivers are provided by database vendors.
3. Connection: Represents a connection to the database, it is used to create a statement object for executing SQL queries.
4. Statement: Represents an SQL statements, that is sent to the database. There are two types statement (used for the database) and Prepared statement.
5. ResultSet: Represents the result set of a query it is used to retrieve and manipulate the data.
6. SQL Exception: Handles exceptions related to database operations.

Steps in JDBC connectivity:

1. Import JDBC packages: Include the necessary JDBC packages in your Java program. import java.sql.*;
2. Load & Register the JDBC driver:

```
class.forName("com.mysql.jdbc.Driver");
```

3. Establish a connection: Use the drive manager to establish a connection to the database connection connect = DriverManager.getConnection ("jdbc:mysql://localhost:3306/database", "username", "password");

4. Create a Statement: create a statement (or) prepared statement object to execute SQL queries.

```
Statement statement = connection.createStatement ( );
ResultSet resultSet = statement.executeQuery ("Select * From Table");
```

5. Execute SQL queries:

6. Process the results: Iterate through the resultSet to retrieve & process the data.

```
while (resultSet.next()){
        // Process each row of data
}
```

7. close resources:

```
resultSet.close();
statement.close();
connection.close();
```

Abstract window Toolkit:

AWT is a set of classes & tools provided by java to create graphical user interfaces (GUI's). AWT components are the building blocks of GUI applications.

1. Frame: Represents a top-level window with a title & borders.
2. Panel: A container that holds are organizes other components.
3. Button: A clickable button that trigger can action when pressed.
4. label: Display a non-editable text
5. Text Area: Allows the user to input multiple lines of text.
6. choice: A drop-down list of choices

7. list: Displays a list of items

8. Scroll bar: Provides a Scroll bar for navigating through content

9. Canvas: A blank area for drawing custom graphics.

Significance of GUI programming

AWT components are essential in GUI programming because they provide a set of reusable and automizable building block for creating graphical user-interface for Java applications.

2) GUI programming and layout managers. Discuss GUI programming with Swing components containers (JFrame, JPane) and different layOut managers for organizing components

GUI programming in java involves creating graphical user Interface for your applications while AWT was the original GUI library in Java, Swing was introduced.

Swing in Java:

Swing is part of the java foundation classes (JFC) and built on the top of AWT. it offers classes (JFC) and is built on the top of AWT, it offers a rich set of components a customizable look.

Swing Components:

Swing provides a wide range of components, radio buttons, tables, trees and more. These components are light weight.

JFrame: It is the main window of a Swing.

JPanel: It is a generic container that can be used to group & organize other components.

LayOut Managers: components with containers Some common layOut managers include:

→ flow LayOut

→ Border LayOut

→ Grid LayOut
→ Box layOut
→ Grid Bag layOut

Limitations of AWT

While AWT was the original GUI library in java, it has some limitations:

1. platform dependence
2. Limited components
3. Look & feel
4. Event handling.

Advantages of Swing:

1. platform independence
2. RichSet of components
3. Consists of look & feel
4. Flexible layOut management

Event Handling in Java:

Event Handling in java Swing is a crucial aspect of GUI programming allowing developers to respond to user actions such as button clicks, mouse movements, keyboard inputs, - - -

Implementing Event handling in Swing:

1. Using Listeners: Swing components provide methods to add event Listeners to specific events.

Ex: addActionListener() for action events
addMouseListener() for mouse events

2. Using Adapter classes. Adapter classes provide default implementations for all methods of an event listener interface.

Menu in Java Swing:

Menu in Swing are components that allow user to interact with an application by selecting various options.

1. JMenu Bar: It represents the horizontal box at the top of a window containing menus.

2. JMenu: Represents a pull-down menu of a statement within a menubar.

3. JMenuItem: Represents an Individual item within a menu.

Significance in user Interaction:

1. Inhand user experience: Event handling allows developer to create interactive & responsive user interface.

2. User-Initiated Actions: Events represents user actions such as button clicks or menu selections.

3. Dynamic Application Behaviour: Event handling enables dynamic changes in application behaviours based on user input.

4. Customization: Developers can customize the behaviour of their applications by responding to specific events, tailoring the user experience.

5. Modularity:

Event-driven programming promotes Modularity by separating different aspects of the program's logic into distinct event handlers, making the code more maintainable & understandable.