1. Write a Java program to connect to a MySQL database using JDBC.
2. Create a Java class to insert student records into a database table.
3. Write a JDBC program to fetch and display all student records from the database.
4. Develop a program to search a student by ID using JDBC.
5. Implement an update operation to modify student details in the database using JDBC.
6. Write a Java program to delete a student record from the database using JDBC.
7. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.
8. Create a JDBC-based program to count the total number of rows in a table.
9. Develop a program to sort student data in ascending order by name using SQL in JDBC.
10. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.
11. Use **PreparedStatement** to insert multiple student records into the database.
12. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).
13. Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.
14. Create a login system using JDBC where user credentials are verified from the database.
15. Implement a Java application to take dynamic input from the user and perform insertion, search, or update using menu-driven logic.
16. Design the schema for a **Library Management System** and write JDBC programs for:

* Adding a book
* Viewing all books
* Issuing a book to a member
* Returning a book

1. Create a **Hospital Management System** database. Using JDBC, implement:

* Register new patient
* Assign doctor
* Generate billing

1. Write a JDBC-based report generator that exports data from a MySQL table to a text or CSV file.