1. Write a Java program to connect to a MySQL database using JDBC.

ANSWER

```
import java.sql.*;


class Q1 {
   public static void main(String[] args) {
      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         System.out.println("Connected");
         con.close();
      } catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

2. Create a Java class to insert student records into a database table.

ANSWER

```
import java.sql.*;


class Q2 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         PreparedStatement ps = con.prepareStatement(
            "insert into students values(?,?,?,?)");
         ps.setInt(1,1);
```

```
        ps.setString(2,"Samarth");
        ps.setInt(3,21);
        ps.setString(4,"CS");
        int r = ps.executeUpdate();
        System.out.println(r+" row inserted");
        con.close();
      } catch(Exception e) {
        System.out.println(e);
      }
   }
}
```

3. Write a JDBC program to fetch and display all student records from the database.

ANSWER

```
import java.sql.*;

class Q3 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         Statement st = con.createStatement();
         ResultSet rs = st.executeQuery("select * from students");
         while(rs.next()) {
            System.out.println(rs.getInt(1)+" "+rs.getString(2)
               +" "+rs.getInt(3)+" "+rs.getString(4));
         }
         con.close();
      } catch(Exception e) {
```

```
        System.out.println(e);
      }
    }
}
```

4. Develop a program to search a student by ID using JDBC.

ANSWER

```
import java.sql.*;


class Q4 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         PreparedStatement ps = con.prepareStatement(
            "select * from students where id=?");
         ps.setInt(1,1);
         ResultSet rs = ps.executeQuery();
         if(rs.next()) {
            System.out.println(rs.getInt(1)+" "+rs.getString(2)
               +" "+rs.getInt(3)+" "+rs.getString(4));
         } else {
            System.out.println("Not found");
         }
         con.close();
      } catch(Exception e) {
         System.out.println(e);
      }
   }
```

```
}
```

5. Implement an update operation to modify student details in the database using JDBC.

ANSWER

```
import java.sql.*;


class Q5 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         PreparedStatement ps = con.prepareStatement(
            "update students set age=?,course=? where id=?");
         ps.setInt(1,22);
         ps.setString(2,"Data Science");
         ps.setInt(3,1);
         int r = ps.executeUpdate();
         System.out.println(r+" row updated");
         con.close();
      } catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

6. Write a Java program to delete a student record from the database using JDBC.

ANSWER

```
import java.sql.*;
```

```
class Q6 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/studentdb","root","password");
         PreparedStatement ps = con.prepareStatement(
            "delete from students where id=?");
         ps.setInt(1,1);
         int r = ps.executeUpdate();
         System.out.println(r+" row deleted");
         con.close();
      } catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

7. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.

ANSWER

```
import java.sql.*;
import java.util.*;

class Q7 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/testdb","root","password");
         Scanner sc = new Scanner(System.in);
```

```
System.out.println("1.Insert 2.Display 3.Update 4.Delete");
int ch = sc.nextInt();


if(ch==1) {
   PreparedStatement ps = con.prepareStatement(
      "insert into employee values(?,?,?)");
   ps.setInt(1,101);
   ps.setString(2,"Samarth");
   ps.setDouble(3,50000);
   ps.executeUpdate();
   System.out.println("Inserted");
}
else if(ch==2) {
   ResultSet rs = con.createStatement().executeQuery("select * from employee");
   while(rs.next()) {
      System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getDouble(3));
   }
}
else if(ch==3) {
   PreparedStatement ps = con.prepareStatement(
      "update employee set salary=? where id=?");
   ps.setDouble(1,60000);
   ps.setInt(2,101);
   ps.executeUpdate();
   System.out.println("Updated");
}
else if(ch==4) {
   PreparedStatement ps = con.prepareStatement(
      "delete from employee where id=?");
```

```
            ps.setInt(1,101);
            ps.executeUpdate();
            System.out.println("Deleted");
          }
          con.close();
        } catch(Exception e) {
          System.out.println(e);
        }
      }
    }
```

8. Create a JDBC-based program to count the total number of rows in a table.

ANSWER

```
import java.sql.*;

class Q8 {
  public static void main(String[] args) {
    try {
      Connection con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/testdb","root","password");
      ResultSet rs = con.createStatement().executeQuery("select count(*) from students");
      if(rs.next()) {
        System.out.println("Total rows: "+rs.getInt(1));
      }
      con.close();
    } catch(Exception e) {
      System.out.println(e);
    }
  }
```

```
}
```

9. Develop a program to sort student data in ascending order by name using SQL in JDBC.

ANSWER

```
import java.sql.*;


class Q9 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/testdb","root","password");
         ResultSet rs = con.createStatement().executeQuery(
            "select * from students order by name asc");
         while(rs.next()) {
            System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getInt(3)+" "+rs.getString(4));
         }
         con.close();
      } catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

10. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.

ANSWER

```
import java.sql.*;


class Q10 {
   public static void main(String[] args) {
```

```
    try {
      Connection con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/testdb","root","password");
      ResultSet rs = con.createStatement().executeQuery(
        "select * from students where percentage>75");
      while(rs.next()) {
        System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getInt(3)+" "+rs.getString(4)+"
"+rs.getDouble("percentage"));
      }
      con.close();
    } catch(Exception e) {
      System.out.println(e);
    }
  }
}
```

11. Use **PreparedStatement** to insert multiple student records into the database.

ANSWER

```
import java.sql.*;

class Q11 {
  public static void main(String[] args) {
    try {
      Connection con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/testdb","root","password");
      PreparedStatement ps = con.prepareStatement(
        "insert into students values(?,?,?,?)");

      ps.setInt(1,2); ps.setString(2,"Aman"); ps.setInt(3,20); ps.setString(4,"CS");
      ps.addBatch();
```

```
        ps.setInt(1,3); ps.setString(2,"Riya"); ps.setInt(3,22); ps.setString(4,"IT");
        ps.addBatch();

        ps.setInt(1,4); ps.setString(2,"Neha"); ps.setInt(3,19); ps.setString(4,"Maths");
        ps.addBatch();

        int[] res = ps.executeBatch();
        System.out.println(res.length+" rows inserted");
        con.close();
      } catch(Exception e) {
        System.out.println(e);
      }
   }
}
```

12. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).

ANSWER

```
import java.sql.*;

class Q12 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/testdb","root","password");
         con.setAutoCommit(false);

         PreparedStatement ps1 = con.prepareStatement(
            "insert into students values(5,'Karan',21,'EE')");
         PreparedStatement ps2 = con.prepareStatement(
```

```
            "insert into students values(6,'Meena',20,'CS')");

        ps1.executeUpdate();
        ps2.executeUpdate();

        con.commit();
        System.out.println("Transaction committed");
        con.close();
      } catch(Exception e) {
        System.out.println("Error: "+e);
        try {
          DriverManager.getConnection(
          "jdbc:mysql://localhost:3306/testdb","root","password").rollback();
        } catch(Exception ex) { System.out.println(ex); }
      }
   }
}
```

13. Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.

ANSWER

```
import java.sql.*;

class Q13 {
   public static void main(String[] args) {
      try {
        Connection con = DriverManager.getConnection(
          "jdbc:mysql://localhost:3306/testdb","root","password");
        PreparedStatement ps = con.prepareStatement(
          "select * from students where id=?");
        ps.setInt(1,1000);
```

```
            ResultSet rs = ps.executeQuery();
            if(rs.next()) {
              System.out.println(rs.getString(2));
            } else {
              System.out.println("Invalid ID");
            }
            con.close();
          } catch(SQLException e) {
            System.out.println("Database error: "+e);
          }
        }
      }
```

14. Create a login system using JDBC where user credentials are verified from the database.

ANSWER

```
import java.sql.*;
import java.util.*;

class Q14 {
  public static void main(String[] args) {
    try {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter username: ");
      String u = sc.next();
      System.out.print("Enter password: ");
      String p = sc.next();

      Connection con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/testdb","root","password");
```

```
        PreparedStatement ps = con.prepareStatement(
            "select * from users where username=? and password=?");
        ps.setString(1,u);
        ps.setString(2,p);
        ResultSet rs = ps.executeQuery();

        if(rs.next()) System.out.println("Login success");
        else System.out.println("Invalid login");
        con.close();
      } catch(Exception e) {
        System.out.println(e);
      }
    }
}
```

15. Implement a Java application to take dynamic input from the user and perform insertion, search, or update using menu-driven logic.

ANSWER

```
import java.sql.*;
import java.util.*;

class Q15 {
    public static void main(String[] args) {
        try {
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/testdb","root","password");
            Scanner sc = new Scanner(System.in);
            System.out.println("1.Insert 2.Search 3.Update");
            int ch = sc.nextInt();
```

```
        if(ch==1) {
            PreparedStatement ps = con.prepareStatement("insert into students values(?,?,?,?)");
            System.out.print("ID: "); ps.setInt(1,sc.nextInt());
            System.out.print("Name: "); ps.setString(2,sc.next());
            System.out.print("Age: "); ps.setInt(3,sc.nextInt());
            System.out.print("Course: "); ps.setString(4,sc.next());
            ps.executeUpdate();
            System.out.println("Inserted");
        }
        else if(ch==2) {
            PreparedStatement ps = con.prepareStatement("select * from students where id=?");
            System.out.print("ID: "); ps.setInt(1,sc.nextInt());
            ResultSet rs = ps.executeQuery();
            if(rs.next()) System.out.println(rs.getString(2));
            else System.out.println("Not found");
        }
        else if(ch==3) {
            PreparedStatement ps = con.prepareStatement("update students set age=? where
id=?");
            System.out.print("Age: "); ps.setInt(1,sc.nextInt());
            System.out.print("ID: "); ps.setInt(2,sc.nextInt());
            ps.executeUpdate();
            System.out.println("Updated");
        }
        con.close();
    } catch(Exception e) {
        System.out.println(e);
    }
```

```
  }
}
```

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

ANSWER

```
import java.sql.*;


class Q16 {
  public static void main(String[] args) {
    try {
      Connection con = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/testdb","root","password");


      // Add book
      PreparedStatement add = con.prepareStatement("insert into books
values(101,'Java','Samarth')");
      add.executeUpdate();
      System.out.println("Book added");


      // View books
      ResultSet rs = con.createStatement().executeQuery("select * from books");
      while(rs.next()) {
        System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
      }
```

```
        // Issue book
        PreparedStatement issue = con.prepareStatement("insert into issue values(101,'Rahul')");
        issue.executeUpdate();
        System.out.println("Book issued");

        // Return book
        PreparedStatement ret = con.prepareStatement("delete from issue where bookid=101");
        ret.executeUpdate();
        System.out.println("Book returned");

        con.close();
      } catch(Exception e) {
        System.out.println(e);
      }
   }
}
```

17. Create a **Hospital Management System** database. Using JDBC, implement:
    Register new patient
    Assign doctor
    Generate billing

ANSWER

```
import java.sql.*;

class Q17 {
   public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/testdb","root","password");
```

```
        PreparedStatement p1 = con.prepareStatement("insert into patient values(1,'Aman',22)");
        p1.executeUpdate();
        System.out.println("Patient registered");

        PreparedStatement p2 = con.prepareStatement("insert into doctor values(1,'Dr. Mehta','Cardiology')");
        p2.executeUpdate();
        System.out.println("Doctor assigned");

        PreparedStatement p3 = con.prepareStatement("insert into bill values(1,2000)");
        p3.executeUpdate();
        System.out.println("Bill generated");

        con.close();
      } catch(Exception e) {
        System.out.println(e);
      }
   }
}
```

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

ANSWER

```
import java.sql.*;
import java.io.*;

class Q18 {
   public static void main(String[] args) {
```

```
        try {
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/testdb","root","password");
            ResultSet rs = con.createStatement().executeQuery("select * from students");

            FileWriter fw = new FileWriter("students.csv");
            while(rs.next()) {
                fw.write(rs.getInt(1)+","+rs.getString(2)+","+rs.getInt(3)+","+rs.getString(4)+"\n");
            }
            fw.close();
            System.out.println("Report generated");
            con.close();
        } catch(Exception e) {
            System.out.println(e);
        }
    }
}
```