```java
package database;

import datapack.DataNotFoundException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Scanner;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;


public class DatabaseConnection {
static Connection connection;

private static final Logger
logger=LogManager.getLogger(DatabaseConnection.class)
;
public static void main(String [] args) throws
DataNotFoundException {
String str = "Radar", reverseStr = "";
String ispalindrome;
   Scanner scanner=new Scanner(System.in);
   System.out.println("enter the input");
    str=scanner.nextLine();
    int strLength = str.length();

    for (int i = (strLength - 1); i >=0; --i) {
      reverseStr = reverseStr + str.charAt(i);

    }
   //System.out.println();
   if
(str.toLowerCase().equals(reverseStr.toLowerCase()))
{
      System.out.println(str + " is a Palindrome
String.");
      ispalindrome="yes";
    }
    else {
      System.out.println(str + " is not a Palindrome
String.");
      ispalindrome="no";
    }


    Properties properties=new Properties();
    try {
        FileInputStream input=new
FileInputStream("C:\\Users\\ADMIN\\eclipse-
workspace\\database\\src\\database.properties");
        properties.load(input);

        String
connectionurl=properties.getProperty("database.url");
        String
username=properties.getProperty("database.user");
        //System.out.println(username);
        logger.info("created");
        String
password=properties.getProperty("database.password");

    //connection=DriverManager.getConnection(username
);

    connection=DriverManager.getConnection(connection
url,username,password);
        ResultSet resultset;
        String inputstring="100";
        Statement
statement=connection.createStatement();
        statement.executeUpdate("insert into course
(Str,Ispalindrome) values
('"+str+"','"+ispalindrome+"');");
        resultset=statement.executeQuery("select *
from palandrom");
        if(!resultset.next()) {
             logger.error("Data not found");
            throw new DataNotFoundException("data
not found");
        }
        else{
            while(resultset.next()) {
```

```
            String str1=resultset.getString("str");
            String
ispalindrome1=resultset.getString("ispalindrome");
            System.out.println(str1 +
ispalindrome1);
        }
    }
        connection.close();
        logger.info("connection closed");
        }
    catch(SQLException |IOException e) {

    //System.out.println("exception"+e.getMessage());
        e.printStackTrace();
        logger.error("sql exception"+e);
    }
    finally {
        try {
            if(connection!=null) {
                connection.close();
            }
        }
            catch(SQLException e) {

    System.out.println("exception"+e.getMessage());
                logger.error("error while closing
connection"+e);
            }
        }
    }
}
```

Database.properties

```
databse.url=jdbc:sqlserver://172.16.51.64;databaseNam
e=231047012;encrypt=true;trustServerCertificate=true;
database.user=MANVITHA;
database.password:manvith@1;
```

DataNotFondException.java

```
package datapack;

public class DataNotFoundException extends Exception{
public DataNotFoundException (String message) {
    super(message);
}
}
-----------------------------------------------------
```

calcularor:

```
package example;
import java.util.Scanner;
public class calculator {
 public static void main(String []args) {
     char op;
     int n1,n2,r;
     Scanner scanner=new Scanner(System.in);
     op=scanner.next().charAt(0);
     n1=scanner.nextInt();
     n2=scanner.nextInt();
     switch(op) {
     case '+':
         r=n1+n2;
         System.out.println(r);
         break;
     case '/':
         if(n2==0) {
             throw new ArithmeticException("no is
divided by zero");
         }else {
         r=n1/n2;
         }
         break;
         default:
             System.out.println("invalid");
     }
 }
}
```

Student:

```
package example;

import datapack.DataNotFoundException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Scanner;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class Databasenew{

    static Connection connection;
    private static final Logger logger = 
LogManager.getLogger(Databasenew.class);

    public static void main(String[] args) throws 
DataNotFoundException {
        Scanner scanner = new Scanner(System.in);

        // Your palindrome checking code here...

        // Adding, displaying, and deleting students
        try {
            Properties properties = new Properties();
            FileInputStream input = new 
FileInputStream("C:\\Users\\ADMIN\\eclipse-
workspace\\database\\src\\database.properties");
            properties.load(input);

            String connectionurl = 
properties.getProperty("database.url");
            String username = 
properties.getProperty("database.user");
            String password = 
properties.getProperty("database.password");

            connection = 
DriverManager.getConnection(connectionurl, username, 
password);

            // Add a student
            System.out.println("Enter student 
name:");
            int studentID = scanner.nextInt();
            System.out.println("Enter student 
name:");
            String studentName = scanner.nextLine();

            System.out.println("Enter student age:");
            int studentAge = scanner.nextInt();
            scanner.nextLine(); // Consume the 
newline character left in the buffer

            System.out.println("Enter student 
grade:");
            String studentGrade = scanner.nextLine();

            addStudent(studentID,studentName, 
studentAge, studentGrade);

            // Display all students
            displayStudents();

            // Delete a student by ID
            System.out.println("Enter student ID to 
delete:");
            int studentIdToDelete = 
scanner.nextInt();
            deleteStudent(studentIdToDelete);

            connection.close();
            logger.info("connection closed");
        } catch (SQLException | IOException e) {
            e.printStackTrace();
            logger.error("sql exception" + e);
        } finally {
            try {
                if (connection != null) {
                    connection.close();
                }
            } catch (SQLException e) {
                System.out.println("exception" + 
e.getMessage());
                logger.error("error while closing 
connection" + e);
            }
        }
    }
    public static void addStudent(int id,String name,
int age, String grade) throws SQLException {
     //   String query = "INSERT INTO student
(id,name, age, grade) VALUES ('"+id+"' ,'" + name +
"', " + age + ", '" + grade + "')";

        Statement statement =
connection.createStatement();
            int rowsAffected =
statement.executeUpdate("INSERT INTO student
(id,name, age, grade) VALUES ('"+id+"' ,'" + name +
"', " + age + ", '" + grade + "');");

            if (rowsAffected > 0) {
                System.out.println("Student added
successfully");
            } else {
                System.out.println("Failed to add
student");
            }

    }

    public static void deleteStudent(int studentId)
throws SQLException {
      //  String query = "DELETE FROM student WHERE
id = " + studentId;

            Statement statement =
connection.createStatement();
            int rowsAffected =
statement.executeUpdate("DELETE FROM student WHERE id
= " + studentId);

            if (rowsAffected > 0) {
                System.out.println("Student deleted
successfully");
            } else {
                System.out.println("Failed to delete
student");

        }
    }
    public static void displayStudents() throws
SQLException {
        //String query = "SELECT * FROM student";

        Statement statement =
connection.createStatement();
             ResultSet resultSet =
statement.executeQuery("SELECT * FROM student;");

            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name =
resultSet.getString("name");
                int age = resultSet.getInt("age");
                String grade =
resultSet.getString("grade");

                System.out.println("ID: " + id + ",
Name: " + name + ", Age: " + age + ", Grade: " +
grade);
            }

    }
}
```