Assignment 1 :

1)Create a Hash table containing Employee name and Salary. Display the details of the hash table. Also search for a specific Employee and display Salary of that Employee.

import java.util.\*;

public class EmployeeHashTable {

public static void main(String[] args) {

HashMap<String, Double> empTable = new HashMap<>();

empTable.put("Amit", 45000.0);

empTable.put("Neha", 52000.0);

empTable.put("Ravi", 60000.0);

empTable.put("Priya", 48000.0);

System.out.println("Employee Details:");

for (Map.Entry<String, Double> entry : empTable.entrySet()) {

System.out.println("Name: " + entry.getKey() + ", Salary: " + entry.getValue());

}

Scanner sc = new Scanner(System.in);

System.out.print("\nEnter employee name to search salary: ");

String name = sc.nextLine();

if (empTable.containsKey(name)) {

System.out.println(name + "'s Salary = " + empTable.get(name));

} else {

System.out.println("Employee not found!");

}

}

}

2)Write a Java program to accept the details of students (rno, sname, per) at least 5 Records, store it into database and display the details.

import java.sql.\*;

import java.util.Scanner;

public class StudentDatabase {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/testdb", "root", "password");

Statement st = con.createStatement();

String createTable = "CREATE TABLE IF NOT EXISTS Student ("

+ "RNo INT PRIMARY KEY, "

+ "SName VARCHAR(50), "

+ "Per FLOAT)";

st.execute(createTable);

PreparedStatement ps = con.prepareStatement("INSERT INTO Student VALUES (?, ?, ?)");

for (int i = 1; i <= 5; i++) {

System.out.print("Enter Roll No: ");

int rno = sc.nextInt();

sc.nextLine(); // consume newline

System.out.print("Enter Name: ");

String name = sc.nextLine();

System.out.print("Enter Percentage: ");

float per = sc.nextFloat();

ps.setInt(1, rno);

ps.setString(2, name);

ps.setFloat(3, per);

ps.executeUpdate();

}

ResultSet rs = st.executeQuery("SELECT \* FROM Student");

System.out.println("\nStudent Records:");

while (rs.next()) {

System.out.println("RNo: " + rs.getInt(1) + ", Name: " + rs.getString(2) + ", Percentage: " + rs.getFloat(3));

}

con.close();

} catch (Exception e) {

System.out.println("Error: " + e);

}

}

}

Assignment 2 :

1)Accept ‘n’ integers from the user. Store and display integers in sorted order having proper collection class. The collection should not accept duplicate elements.

import java.util.\*;

public class SortedUniqueIntegers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of integers: ");

int n = sc.nextInt();

TreeSet<Integer> numbers = new TreeSet<>();

System.out.println("Enter integers: ");

for(int i = 0; i < n; i++) {

numbers.add(sc.nextInt());

}

System.out.println("Sorted unique integers: " + numbers);

}

}

2)Write a java program to design a following GUI. Use appropriate Layout and Components.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class LoginForm extends JFrame {

JTextField userField;

JPasswordField passField;

LoginForm() {

setTitle("Login");

setSize(300,150);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new GridLayout(3,2,5,5));

JLabel userLabel = new JLabel("Username:");

JLabel passLabel = new JLabel("Password:");

userField = new JTextField();

passField = new JPasswordField();

JButton loginBtn = new JButton("Login");

JButton resetBtn = new JButton("Reset");

add(userLabel); add(userField);

add(passLabel); add(passField);

add(loginBtn); add(resetBtn);

loginBtn.addActionListener(e -> JOptionPane.showMessageDialog(this,

"Username: " + userField.getText() + "\nPassword: " + new String(passField.getPassword())));

resetBtn.addActionListener(e -> {

userField.setText("");

passField.setText("");

});

setVisible(true);

}

public static void main(String[] args) {

new LoginForm();

}

}

Assignment 3:

1)Create a HashMap containing Employee name and Salary. Display the details of the hash table. Also search for a specific Employee and display Salary of that Employee.

import java.util.\*;

public class EmployeeHashMap {

public static void main(String[] args) {

HashMap<String, Double> employees = new HashMap<>();

employees.put("Alice", 50000.0);

employees.put("Bob", 60000.0);

employees.put("Charlie", 55000.0);

System.out.println("Employee details:");

for(Map.Entry<String, Double> entry : employees.entrySet()) {

System.out.println(entry.getKey() + " : " + entry.getValue());

}

Scanner sc = new Scanner(System.in);

System.out.print("Enter employee name to search: ");

String name = sc.nextLine();

if(employees.containsKey(name)) {

System.out.println(name + "'s Salary: " + employees.get(name));

} else {

System.out.println("Employee not found.");

}

}

}

2)Write a Java program to accept the details of Student (RNo, SName, Per, Gender, Class) and store into the database. (Use appropriate Swing Components and PreparedStatement Interface).

import java.sql.\*;

import java.util.Scanner;

public class StudentConsoleDB {

Class.forName() ;

private static final String DB\_URL = "jdbc::students.db";

public static void main(String[] args) {

createTableIfNotExists();

try (Scanner sc = new Scanner(System.in)) {

System.out.println("Enter details of 5 students:");

for (int i = 1; i <= 5; i++) {

System.out.println("\nStudent " + i + ":");

System.out.print("Roll Number (int): ");

int rno = sc.nextInt();

sc.nextLine(); // consume newline

System.out.print("Student Name: ");

String sname = sc.nextLine();

System.out.print("Percentage (double): ");

double per = sc.nextDouble();

sc.nextLine();

System.out.print("Gender (Male/Female/Other): ");

String gender = sc.nextLine();

System.out.print("Class: ");

String className = sc.nextLine();

insertStudent(rno, sname, per, gender, className);

}

System.out.println("\nAll students added successfully!\n");

displayAllStudents();

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

private static void createTableIfNotExists() {

String createTableSQL = """

CREATE TABLE IF NOT EXISTS students (

rno INTEGER PRIMARY KEY,

sname TEXT NOT NULL,

per REAL NOT NULL,

gender TEXT NOT NULL,

class TEXT NOT NULL

); """;

try (Connection conn = DriverManager.getConnection(DB\_URL);

Statement stmt = conn.createStatement()) {

stmt.execute(createTableSQL);

} catch (SQLException e) {

System.out.println("DB Error (creating table): " + e.getMessage());

}

}

private static void insertStudent(int rno, String sname, double per, String gender, String className) {

String insertSQL = "INSERT INTO students (rno, sname, per, gender, class) VALUES (?, ?, ?, ?, ?)";

try (Connection conn = DriverManager.getConnection(DB\_URL);

PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {

pstmt.setInt(1, rno);

pstmt.setString(2, sname);

pstmt.setDouble(3, per);

pstmt.setString(4, gender);

pstmt.setString(5, className);

pstmt.executeUpdate();

} catch (SQLException e) {

System.out.println("DB Error (inserting student): " + e.getMessage());

}

}

private static void displayAllStudents() {

String selectSQL = "SELECT \* FROM students";

try (Connection conn = DriverManager.getConnection(DB\_URL);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(selectSQL)) {

System.out.println("RNo\tName\tPercentage\tGender\tClass");

System.out.println("--------------------------------------------------");

while (rs.next()) {

int rno = rs.getInt("rno");

String sname = rs.getString("sname");

double per = rs.getDouble("per");

String gender = rs.getString("gender");

String className = rs.getString("class");

System.out.printf("%d\t%s\t%.2f\t\t%s\t%s\n", rno, sname, per, gender, className);

}

} catch (SQLException e) {

System.out.println("DB Error (fetching data): " + e.getMessage());

}

}

}

Assignment 4:

1)Write Java program to demonstrate any five different HashSet operations.

import java.util.HashSet;

public class HashSetDemo {

public static void main(String[] args) {

HashSet<String> set = new HashSet<>();

set.add("Apple");

set.add("Banana");

set.add("Cherry");

set.add("Date");

set.add("Elderberry");

System.out.println("Initial HashSet: " + set);

String checkFruit = "Banana";

System.out.println("Does the set contain '" + checkFruit + "'? " + set.contains(checkFruit));

set.remove("Date");

System.out.println("After removing 'Date': " + set);

System.out.print("Iterating over HashSet elements: ");

for (String fruit : set) {

System.out.print(fruit + " ");

}

System.out.println();

set.clear();

System.out.println("After clearing, is the set empty? " + set.isEmpty());

}

}

2)Write a JDBC program to count number of records in a Emp table.

import java.sql.\*;

public class EmpRecordCount {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database"; // Example for MySQL

private static final String USER = "your\_username";

private static final String PASS = "your\_password";

public static void main(String[] args) {

String query = "SELECT COUNT(\*) AS total FROM Emp";

try (

Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query);

) {

if (rs.next()) {

int count = rs.getInt("total");

System.out.println("Total records in Emp table: " + count);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

Assignment 5:

1)Write Java program to demonstrate any five different LinkedList operations.

public class LinkedListDemo {

public static void main(String[] args) {

LinkedList<String> list = new LinkedList<>();

list.add("Red");

list.add("Green");

list.add("Blue");

list.add("Yellow");

list.add("Purple");

System.out.println("Initial LinkedList: " + list);

list.addFirst("Black");

System.out.println("After addFirst('Black'): " + list);

list.remove("Green");

System.out.println("After removing 'Green': " + list);

String elementAtIndex2 = list.get(2);

System.out.println("Element at index 2: " + elementAtIndex2);

System.out.print("Iterating over LinkedList: ");

for (String color : list) {

System.out.print(color + " ");

}

System.out.println();

}}

2)Write a java program to design a following GUI. Use appropriate Layout and Components(Vaccination).

import javax.swing.\*;

import javax.swing.event.\*;

import java.awt.\*;

public class VaccinationFormCompact extends JFrame {

private final JTextField txtName;

private final JCheckBox chkFirstDose, chkSecondDose;

private final JRadioButton rbCovishield, rbCovaxin, rbSputnik;

private final JLabel valOutputName, valOutputFirstDose, valOutputSecondDose, valOutputVaccine;

public VaccinationFormCompact() {

setTitle("Vaccination Details");

setSize(550, 450);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout(10, 10));

JPanel inputPanel = new JPanel(null);

addComp(new JLabel("Vaccination Details"){{setFont(new Font("Arial", Font.BOLD, 20));}}, inputPanel, 180, 20, 200, 30);

addComp(new JLabel("Name:"), inputPanel, 50, 80, 50, 25);

txtName = (JTextField) addComp(new JTextField(), inputPanel, 110, 80, 350, 25);

addComp(new JLabel("Dose"){{setFont(new Font("Arial", Font.BOLD, 14));}}, inputPanel, 90, 140, 100, 25);

chkFirstDose = (JCheckBox) addComp(new JCheckBox("1st Dose"), inputPanel, 90, 170, 100, 25);

chkSecondDose = (JCheckBox) addComp(new JCheckBox("2nd Dose"), inputPanel, 90, 200, 100, 25);

addComp(new JLabel("Vaccine"){{setFont(new Font("Arial", Font.BOLD, 14));}}, inputPanel, 300, 140, 100, 25);

ButtonGroup bg = new ButtonGroup();

rbCovishield = (JRadioButton) addComp(new JRadioButton("Covishield"), inputPanel, 300, 170, 100, 25, bg);

rbCovaxin = (JRadioButton) addComp(new JRadioButton("Covaxin"), inputPanel, 300, 200, 100, 25, bg);

rbSputnik = (JRadioButton) addComp(new JRadioButton("Sputnik V"), inputPanel, 300, 230, 100, 25, bg);

add(inputPanel, BorderLayout.CENTER);

JPanel outputPanel = new JPanel(null);

outputPanel.setPreferredSize(new Dimension(500, 100));

outputPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK));

addComp(new JLabel("Name :"), outputPanel, 20, 10, 50, 25);

valOutputName = (JLabel) addComp(new JLabel(), outputPanel, 75, 10, 200, 25);

addComp(new JLabel("1st Dose:"), outputPanel, 300, 10, 60, 25);

valOutputFirstDose = (JLabel) addComp(new JLabel(), outputPanel, 365, 10, 100, 25);

addComp(new JLabel("2nd Dose:"), outputPanel, 300, 40, 60, 25);

valOutputSecondDose = (JLabel) addComp(new JLabel(), outputPanel, 365, 40, 100, 25);

addComp(new JLabel("Vaccine:"), outputPanel, 20, 40, 60, 25);

valOutputVaccine = (JLabel) addComp(new JLabel(), outputPanel, 85, 40, 150, 25);

add(outputPanel, BorderLayout.SOUTH);

DocumentListener docListener = new DocumentListener() {

public void changedUpdate(DocumentEvent e) { updateOutput(); }

public void removeUpdate(DocumentEvent e) { updateOutput(); }

public void insertUpdate(DocumentEvent e) { updateOutput(); }

};

txtName.getDocument().addDocumentListener(docListener);

java.awt.event.ItemListener itemListener = e -> updateOutput();

chkFirstDose.addItemListener(itemListener);

chkSecondDose.addItemListener(itemListener);

rbCovishield.addItemListener(itemListener);

rbCovaxin.addItemListener(itemListener);

rbSputnik.addItemListener(itemListener);

setVisible(true);

updateOutput();

}

private Component addComp(Component c, Container p, int x, int y, int w, int h, ButtonGroup... bg) {

c.setBounds(x, y, w, h);

p.add(c);

if (bg.length > 0 && c instanceof AbstractButton) bg[0].add((AbstractButton)c);

return c;

}

private void updateOutput() {

valOutputName.setText(txtName.getText());

valOutputFirstDose.setText(chkFirstDose.isSelected() ? "Taken" : "Not Taken");

valOutputSecondDose.setText(chkSecondDose.isSelected() ? "Taken" : "Not Taken");

String vaccine = rbCovishield.isSelected() ? "Covishield" :

rbCovaxin.isSelected() ? "Covaxin" :

rbSputnik.isSelected() ? "Sputnik V" : "";

valOutputVaccine.setText(vaccine);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(VaccinationFormCompact::new);

}

}

Assignment 6:

1)Write Java program to demonstrate any five different ArrayList operations.

import java.util.ArrayList;

public class ArrayListDemo {

public static void main(String[] args) {

ArrayList<String> list = new ArrayList<>();

list.add("Java");

list.add("Python");

list.add("C++");

list.add("JavaScript");

list.add("Ruby");

System.out.println("Initial ArrayList: " + list);

list.add(2, "C#");

System.out.println("After adding 'C#' at index 2: " + list);

list.remove("Ruby");

System.out.println("After removing 'Ruby': " + list);

list.set(3, "TypeScript");

System.out.println("After updating element at index 3 to 'TypeScript': " + list);

System.out.print("Iterating over ArrayList: ");

for (String lang : list) {

System.out.print(lang + " ");

}

System.out.println();

}

}

2)Write a menu driven program in Java for the following: Assume Emp table with attributes ( ENo, EName, salary, Desg ) is already created. 1. Insert 2. Display 3. Exit.

import java.sql.\*;

import java.util.Scanner;

public class EmpMenuDriven {

private static final String DB\_URL = "jdbc:sqlite:emp.db"; // Change for your DB

// For MySQL, for example:

// private static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_db";

// private static final String USER = "your\_user";

// private static final String PASS = "your\_pass";

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int choice;

while (true) {

System.out.println("\n===== EMPLOYEE MENU =====");

System.out.println("1. Insert Employee");

System.out.println("2. Display Employees");

System.out.println("3. Exit");

System.out.print("Enter your choice: ");

choice = sc.nextInt();

sc.nextLine();

switch (choice) {

case 1:

insertEmployee(sc);

break;

case 2:

displayEmployees();

break;

case 3:

System.out.println("Exiting program...");

sc.close();

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

private static void insertEmployee(Scanner sc) {

try (Connection conn = DriverManager.getConnection(DB\_URL)) {

System.out.print("Enter Employee Number (ENo): ");

int eno = sc.nextInt();

sc.nextLine();

System.out.print("Enter Employee Name (EName): ");

String ename = sc.nextLine();

System.out.print("Enter Salary: ");

double salary = sc.nextDouble();

sc.nextLine();

System.out.print("Enter Designation (Desg): ");

String desg = sc.nextLine();

String insertSQL = "INSERT INTO Emp (ENo, EName, salary, Desg) VALUES (?, ?, ?, ?)";

try (PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {

pstmt.setInt(1, eno);

pstmt.setString(2, ename);

pstmt.setDouble(3, salary);

pstmt.setString(4, desg);

int rows = pstmt.executeUpdate();

if (rows > 0) {

System.out.println("Employee inserted successfully!");

} else {

System.out.println("Failed to insert employee.");

}

}

} catch (SQLException e) {

System.out.println("DB Error: " + e.getMessage());

}

}

private static void displayEmployees() {

try (Connection conn = DriverManager.getConnection(DB\_URL);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM Emp")) {

System.out.println("\nENo\tEName\tSalary\t\tDesg");

System.out.println("---------------------------------------------");

boolean empty = true;

while (rs.next()) {

empty = false;

int eno = rs.getInt("ENo");

String ename = rs.getString("EName");

double salary = rs.getDouble("salary");

String desg = rs.getString("Desg");

System.out.printf("%d\t%s\t%.2f\t\t%s\n", eno, ename, salary, desg);

}

if (empty) {

System.out.println("No employee records found.");

}

} catch (SQLException e) {

System.out.println("DB Error: " + e.getMessage());

}

}

}

Assignment 7:

1)Write Java program to demonstrate any five different HashSet operations.

import java.util.HashSet;

public class HashSetOperationsDemo {

public static void main(String[] args) {

HashSet<String> set = new HashSet<>();

set.add("Java");

set.add("Python");

set.add("C++");

set.add("JavaScript");

set.add("Ruby");

System.out.println("Initial HashSet: " + set);

String elementToCheck = "Python";

boolean contains = set.contains(elementToCheck);

System.out.println("Does set contain '" + elementToCheck + "'? " + contains);

set.remove("Ruby");

System.out.println("After removing 'Ruby': " + set);

System.out.print("Iterating over HashSet elements: ");

for (String language : set) {

System.out.print(language + " ");

}

System.out.println();

set.clear();

System.out.println("After clearing, is the set empty? " + set.isEmpty());

}

}

2)Write a menu driven program in Java for the following: Assume Emp table with attributes ( ENo, EName, salary, Desg ) is already created. 1. Update 2. Display 2. Exit.

import java.sql.\*;

import java.util.Scanner;

public class EmpMenuDrivenUpdateDisplay {

private static final String DB\_URL = "jdbc:sqlite:emp.db"; // Change this for your DB

// For MySQL or others, update URL, USER, PASS accordingly

// private static final String USER = "your\_user";

// private static final String PASS = "your\_pass";

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int choice;

while (true) {

System.out.println("\n===== EMPLOYEE MENU =====");

System.out.println("1. Update Employee");

System.out.println("2. Display Employees");

System.out.println("3. Exit");

System.out.print("Enter your choice: ");

choice = sc.nextInt();

sc.nextLine(); // consume newline

switch (choice) {

case 1:

updateEmployee(sc);

break;

case 2:

displayEmployees();

break;

case 3:

System.out.println("Exiting program...");

sc.close();

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

private static void updateEmployee(Scanner sc) {

System.out.print("Enter Employee Number (ENo) to update: ");

int eno = sc.nextInt();

sc.nextLine();

System.out.println("What do you want to update?");

System.out.println("1. Employee Name");

System.out.println("2. Salary");

System.out.println("3. Designation");

System.out.print("Enter your choice: ");

int updateChoice = sc.nextInt();

sc.nextLine();

String column = null;

Object newValue = null;

switch (updateChoice) {

case 1:

System.out.print("Enter new Employee Name: ");

newValue = sc.nextLine();

column = "EName";

break;

case 2:

System.out.print("Enter new Salary: ");

newValue = sc.nextDouble();

sc.nextLine();

column = "salary";

break;

case 3:

System.out.print("Enter new Designation: ");

newValue = sc.nextLine();

column = "Desg";

break;

default:

System.out.println("Invalid update choice.");

return;

}

String updateSQL = "UPDATE Emp SET " + column + " = ? WHERE ENo = ?";

try (Connection conn = DriverManager.getConnection(DB\_URL);

PreparedStatement pstmt = conn.prepareStatement(updateSQL)) {

if (newValue instanceof String) {

pstmt.setString(1, (String) newValue);

Assignment 8:

1)Write Java program to demonstrate any five different LinkedHashSet operations.

import java.util.LinkedHashSet;

public class LinkedHashSetDemo {

public static void main(String[] args) {

LinkedHashSet<String> set = new LinkedHashSet<>();

set.add("Apple");

set.add("Banana");

set.add("Cherry");

set.add("Date");

set.add("Elderberry");

System.out.println("Initial LinkedHashSet: " + set);

String checkFruit = "Cherry";

System.out.println("Does set contain '" + checkFruit + "'? " + set.contains(checkFruit));

set.remove("Date");

System.out.println("After removing 'Date': " + set);

System.out.print("Iterating over LinkedHashSet elements: ");

for (String fruit : set) {

System.out.print(fruit + " ");

}

System.out.println();

set.clear();

System.out.println("After clearing, is the set empty? " + set.isEmpty());

}

}

2)Write a JSP program to check whether a given number is even or not.

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Check Even or Odd</title>

</head>

<body>

<h2>Check if a Number is Even or Odd</h2>

<form method="post" action="">

Enter a number: <input type="number" name="number" required />

<input type="submit" value="Check" />

</form>

<%

String numStr = request.getParameter("number");

if (numStr != null) {

try {

int number = Integer.parseInt(numStr);

if (number % 2 == 0) {

out.println("<h3>" + number + " is Even.</h3>");

} else {

out.println("<h3>" + number + " is Odd.</h3>");

}

} catch (NumberFormatException e) {

out.println("<h3>Invalid input. Please enter a valid integer.</h3>");

}

}

%>

</body>

</html>

Assignment 9 :

1)Write Java program to demonstrate any five different TreeSet operations.

import java.util.\*;

public class TreeSetDemo {

public static void main(String[] args) {

TreeSet<String> treeSet = new TreeSet<>();

treeSet.add("Apple");

treeSet.add("Banana");

treeSet.add("Orange");

treeSet.add("Mango");

System.out.println("TreeSet after adding elements: " + treeSet);

treeSet.remove("Banana");

System.out.println("TreeSet after removing 'Banana': " + treeSet);

System.out.println("Contains 'Apple'? " + treeSet.contains("Apple"));

System.out.print("Iterating elements: ");

for(String fruit : treeSet) {

System.out.print(fruit + " ");

}

System.out.println();

System.out.println("First element: " + treeSet.first());

System.out.println("Last element: " + treeSet.last());

}

}

2)Write a java program to implement a simple arithmetic calculator using AWT / Swing.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class SimpleCalculator extends JFrame implements ActionListener {

private JTextField num1Field, num2Field, resultField;

private JButton addBtn, subBtn, mulBtn, divBtn, clearBtn;

public SimpleCalculator() {

setTitle("Simple Arithmetic Calculator");

setSize(350, 250);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null); // center the window

// Create components

JLabel num1Label = new JLabel("Number 1:");

JLabel num2Label = new JLabel("Number 2:");

JLabel resultLabel = new JLabel("Result:");

num1Field = new JTextField(10);

num2Field = new JTextField(10);

resultField = new JTextField(15);

resultField.setEditable(false);

addBtn = new JButton("+");

subBtn = new JButton("-");

mulBtn = new JButton("\*");

divBtn = new JButton("/");

clearBtn = new JButton("Clear");

addBtn.addActionListener(this);

subBtn.addActionListener(this);

mulBtn.addActionListener(this);

divBtn.addActionListener(this);

clearBtn.addActionListener(this);

setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5,5,5,5);

gbc.gridx = 0;

gbc.gridy = 0;

add(num1Label, gbc);

gbc.gridx = 1;

add(num1Field, gbc);

gbc.gridx = 0;

gbc.gridy = 1;

add(num2Label, gbc);

gbc.gridx = 1;

add(num2Field, gbc);

JPanel buttonPanel = new JPanel();

buttonPanel.add(addBtn);

buttonPanel.add(subBtn);

buttonPanel.add(mulBtn);

buttonPanel.add(divBtn);

buttonPanel.add(clearBtn);

gbc.gridx = 0;

gbc.gridy = 2;

gbc.gridwidth = 2;

add(buttonPanel, gbc);

gbc.gridx = 0;

gbc.gridy = 3;

gbc.gridwidth = 1;

add(resultLabel, gbc);

gbc.gridx = 1;

add(resultField, gbc);

setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

try {

double num1 = Double.parseDouble(num1Field.getText());

double num2 = Double.parseDouble(num2Field.getText());

double result = 0;

if (e.getSource() == addBtn) {

result = num1 + num2;

} else if (e.getSource() == subBtn) {

result = num1 - num2;

} else if (e.getSource() == mulBtn) {

result = num1 \* num2;

} else if (e.getSource() == divBtn) {

if (num2 == 0) {

JOptionPane.showMessageDialog(this, "Cannot divide by zero!", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

result = num1 / num2;

} else if (e.getSource() == clearBtn) {

num1Field.setText("");

num2Field.setText("");

resultField.setText("");

return;

}

resultField.setText(String.valueOf(result));

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, "Please enter valid numbers!", "Input Error", JOptionPane.ERROR\_MESSAGE);

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(SimpleCalculator::new);

}

}

Assignment 10:

1)Write Java program to demonstrate any five different Queue operations.

import java.util.\*;

public class TreeMapDemo {

public static void main(String[] args) {

TreeMap<Integer, String> treeMap = new TreeMap<>();

treeMap.put(101, "Alice");

treeMap.put(103, "Charlie");

treeMap.put(102, "Bob");

treeMap.put(104, "David");

System.out.println("TreeMap after adding elements: " + treeMap);

treeMap.remove(103);

System.out.println("TreeMap after removing key 103: " + treeMap);

System.out.println("Contains key 102? " + treeMap.containsKey(102));

System.out.println("Iterating through TreeMap:");

for(Map.Entry<Integer, String> entry : treeMap.entrySet()) {

System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

System.out.println("First Key: " + treeMap.firstKey() + ", First Value: " + treeMap.get(treeMap.firstKey()));

System.out.println("Last Key: " + treeMap.lastKey() + ", Last Value: " + treeMap.get(treeMap.lastKey()));

}

}

2)Write a java program to find factorial of a number using AWT / Swing.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class FactorialCalculator {

public static void main(String[] args) {

JFrame frame = new JFrame("Factorial Calculator");

frame.setSize(400, 200);

frame.setLayout(new GridLayout(3, 2));

JLabel lblNumber = new JLabel("Enter a number:");

JTextField txtNumber = new JTextField();

JButton btnCalculate = new JButton("Calculate");

JLabel lblResult = new JLabel("Factorial:");

frame.add(lblNumber);

frame.add(txtNumber);

frame.add(btnCalculate);

frame.add(lblResult);

btnCalculate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

int num = Integer.parseInt(txtNumber.getText());

long factorial = 1;

for(int i = 1; i <= num; i++) {

factorial \*= i;

}

lblResult.setText("Factorial: " + factorial);

} catch(NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Please enter a valid integer.");

}

}

});

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}}