Q.1) Write an application to create a splash screen.

```
import javax.swing.*;
import java.awt.*;
import java.util.Timer;
import java.util.TimerTask;

public class SplashScreen extends JWindow {
    private JLabel label;

    public SplashScreen() {
        // Set the size and position of the splash screen
        setSize(400, 300);
        setLocationRelativeTo(null); // Center the splash screen on the screen

        // Set a background image for the splash screen
        JLabel background = new JLabel(new ImageIcon("splash_background.jpg"));
        background.setLayout(new BorderLayout());
        getContentPane().add(background);

        // Add a label to display a message on the splash screen
        label = new JLabel("Welcome to My App", SwingConstants.CENTER);
        label.setFont(new Font("Arial", Font.BOLD, 20));
        label.setForeground(Color.WHITE);
        background.add(label, BorderLayout.CENTER);

        // Simulate some loading time using a Timer
        Timer timer = new Timer();
        timer.schedule(new CloseSplashTask(), 3000);
    }

    private class CloseSplashTask extends TimerTask {
```

```
    @Override
    public void run() {
        dispose(); // Close the splash screen after 3 seconds
    }
}

public static void main(String[] args) {
    // Create and display the splash screen
    SwingUtilities.invokeLater(() -> {
        SplashScreen splash = new SplashScreen();
        splash.setVisible(true);
    });

    // Simulate loading of the main application
    // Replace this with the actual code to initialize your main application
    // For demonstration purposes, we'll just sleep for 3 seconds
    try {
        Thread.sleep(3000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // Once loading is done, launch the main application
    SwingUtilities.invokeLater(() -> {
        // Create and display the main application window
        JFrame frame = new JFrame("Main Application");
        frame.setSize(600, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Add your main application components here...
        frame.setVisible(true);
    });
```

```
    }
}
```

Q.2) Create table Student (roll_no, name, address, percentage). Create Application for performing the following operation on the table. (Using SQLite database). i] Insert record of 5 new student details. ii] Show all the student details.

```
import java.sql.*;


public class StudentDatabase {
    // JDBC URL, username, and password of SQLite database
    private static final String url = "jdbc:sqlite:student.db";


    public static void main(String[] args) {
        try {
            // Connect to the SQLite database
            Connection conn = DriverManager.getConnection(url);


            // SQL statement for creating a new table
            String createTableSQL = "CREATE TABLE IF NOT EXISTS Student (\n"
                                    + "    roll_no INTEGER PRIMARY KEY,\n"
                                    + "    name TEXT NOT NULL,\n"
                                    + "    address TEXT,\n"
                                    + "    percentage REAL\n"
                                    + ");";


            // Create table if it doesn't exist
            try (Statement stmt = conn.createStatement()) {
                stmt.execute(createTableSQL);
            } catch (SQLException e) {
                System.out.println(e.getMessage());
            }


            // Insert 5 new student records
```

```
            insertStudent(conn, 1, "John Doe", "123 Main St", 85.5);
            insertStudent(conn, 2, "Jane Smith", "456 Elm St", 92.3);
            insertStudent(conn, 3, "Alice Johnson", "789 Oak St", 78.9);
            insertStudent(conn, 4, "Bob Brown", "321 Pine St", 90.0);
            insertStudent(conn, 5, "Emily Davis", "654 Maple St", 88.7);

            // Close the connection
            conn.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    private static void insertStudent(Connection conn, int rollNo, String name, String address, double
percentage) {
        String insertSQL = "INSERT INTO Student(roll_no, name, address, percentage) VALUES(?,?,?,?)";

        try (PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {
            pstmt.setInt(1, rollNo);
            pstmt.setString(2, name);
            pstmt.setString(3, address);
            pstmt.setDouble(4, percentage);
            pstmt.executeUpdate();
            System.out.println("Inserted record for roll number " + rollNo);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Q.1) Create an application that allows the user to enter a number in the textbox. Check whether the number in the textbox is perfect number or not. Print the message using Toast control. [10 Marks]

```
import javax.swing.*;
```

```
import java.awt.*;
import java.awt.event.*;

public class PerfectNumberChecker extends JFrame {
    private JTextField numberField;
    private JButton checkButton;

    public PerfectNumberChecker() {
        setTitle("Perfect Number Checker");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new BorderLayout());

        JLabel label = new JLabel("Enter a number:");
        panel.add(label, BorderLayout.NORTH);

        numberField = new JTextField();
        panel.add(numberField, BorderLayout.CENTER);

        checkButton = new JButton("Check");
        checkButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkPerfectNumber();
            }
        });
        panel.add(checkButton, BorderLayout.SOUTH);

        add(panel);
    }
```

```
private void checkPerfectNumber() {
    String input = numberField.getText();
    if (!input.isEmpty()) {
        try {
            int number = Integer.parseInt(input);
            boolean isPerfect = isPerfectNumber(number);
            if (isPerfect) {
                showToast("The number " + number + " is a perfect number.");
            } else {
                showToast("The number " + number + " is not a perfect number.");
            }
        } catch (NumberFormatException ex) {
            showToast("Please enter a valid integer number.");
        }
    } else {
        showToast("Please enter a number.");
    }
}

private boolean isPerfectNumber(int number) {
    int sum = 0;
    for (int i = 1; i <= number / 2; i++) {
        if (number % i == 0) {
            sum += i;
        }
    }
    return sum == number;
}

private void showToast(String message) {
```

```
        JOptionPane.showMessageDialog(this, message, "Result",
JOptionPane.INFORMATION_MESSAGE);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new PerfectNumberChecker().setVisible(true);
            }
        });
    }
}
```

Q.2) Java Android Program to perform all arithmetic Operations using Calculator. [20 marks].

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private TextView resultTextView;
    private double num1 = 0, num2 = 0;
    private String operator = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        resultTextView = findViewById(R.id.result_text_view);
```

```
// Initialize buttons
Button button0 = findViewById(R.id.button_0);
Button button1 = findViewById(R.id.button_1);
Button button2 = findViewById(R.id.button_2);
Button button3 = findViewById(R.id.button_3);
Button button4 = findViewById(R.id.button_4);
Button button5 = findViewById(R.id.button_5);
Button button6 = findViewById(R.id.button_6);
Button button7 = findViewById(R.id.button_7);
Button button8 = findViewById(R.id.button_8);
Button button9 = findViewById(R.id.button_9);
Button buttonPlus = findViewById(R.id.button_plus);
Button buttonMinus = findViewById(R.id.button_minus);
Button buttonMultiply = findViewById(R.id.button_multiply);
Button buttonDivide = findViewById(R.id.button_divide);
Button buttonEqual = findViewById(R.id.button_equal);
Button buttonClear = findViewById(R.id.button_clear);

// Set onClickListener for buttons
button0.setOnClickListener(this);
button1.setOnClickListener(this);
button2.setOnClickListener(this);
button3.setOnClickListener(this);
button4.setOnClickListener(this);
button5.setOnClickListener(this);
button6.setOnClickListener(this);
button7.setOnClickListener(this);
button8.setOnClickListener(this);
button9.setOnClickListener(this);
buttonPlus.setOnClickListener(this);
buttonMinus.setOnClickListener(this);
```

```
    buttonMultiply.setOnClickListener(this);
    buttonDivide.setOnClickListener(this);
    buttonEqual.setOnClickListener(this);
    buttonClear.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.button_0:
        case R.id.button_1:
        case R.id.button_2:
        case R.id.button_3:
        case R.id.button_4:
        case R.id.button_5:
        case R.id.button_6:
        case R.id.button_7:
        case R.id.button_8:
        case R.id.button_9:
            // Append clicked number to the resultTextView
            String buttonText = ((Button) v).getText().toString();
            resultTextView.append(buttonText);
            break;
        case R.id.button_plus:
        case R.id.button_minus:
        case R.id.button_multiply:
        case R.id.button_divide:
            // Store the first number and selected operator
            num1 = Double.parseDouble(resultTextView.getText().toString());
            operator = ((Button) v).getText().toString();
            resultTextView.setText("");
```

```
            break;
        case R.id.button_equal:
            // Perform calculation based on operator
            num2 = Double.parseDouble(resultTextView.getText().toString());
            double result = calculateResult(num1, num2, operator);
            resultTextView.setText(String.valueOf(result));
            break;
        case R.id.button_clear:
            // Clear the resultTextView
            resultTextView.setText("");
            num1 = num2 = 0;
            operator = "";
            break;
    }
}

private double calculateResult(double num1, double num2, String operator) {
    switch (operator) {
        case "+":
            return num1 + num2;
        case "-":
            return num1 - num2;
        case "×":
            return num1 * num2;
        case "÷":
            if (num2 != 0) {
                return num1 / num2;
            } else {
                return 0; // Division by zero error
            }
        default:
```

```
            return 0;
        }
    }
}
```

Q.1) Create an application that allows the user to enter a number in the textbox. Check whether the number in the textbox is Armstrong or not. Print the message accordingly in the label control.

```
import tkinter as tk


def is_armstrong(num):
    # Convert number to string to calculate its length
    num_str = str(num)
    num_digits = len(num_str)

    # Calculate the sum of each digit raised to the power of the number of digits
    armstrong_sum = sum(int(digit)**num_digits for digit in num_str)

    # Check if the sum is equal to the original number
    return armstrong_sum == num

def check_armstrong():
    # Get the number entered by the user
    num = int(entry.get())

    # Check if the number is Armstrong or not
    if is_armstrong(num):
        result_label.config(text=f"{num} is an Armstrong number.")
    else:
        result_label.config(text=f"{num} is not an Armstrong number.")

# Create the main application window
root = tk.Tk()
```

```
root.title("Armstrong Number Checker")

# Create a label for instructions
instruction_label = tk.Label(root, text="Enter a number:")
instruction_label.pack()

# Create a text entry box for the user to enter a number
entry = tk.Entry(root)
entry.pack()

# Create a button to check the number
check_button = tk.Button(root, text="Check", command=check_armstrong)
check_button.pack()

# Create a label to display the result
result_label = tk.Label(root, text="")
result_label.pack()

# Start the Tkinter event loop
root.mainloop()
```

Q.2) Create an Android application which examine a phone number entered by a user with the given format. • Area code should be one of the following: 040, 041, 050, 0400, 044 • There should 6 - 8 numbers in telephone number (+ area code). [20 Marks]

```
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {

    private EditText phoneNumberEditText;
    private TextView validationTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        phoneNumberEditText = findViewById(R.id.phone_number_edit_text);
        validationTextView = findViewById(R.id.validation_text_view);

        phoneNumberEditText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {}

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {}

            @Override
            public void afterTextChanged(Editable s) {
                validatePhoneNumber(s.toString());
            }
        });
    }

    private void validatePhoneNumber(String phoneNumber) {
        if (phoneNumber.length() < 6 || phoneNumber.length() > 8) {
            validationTextView.setText(getString(R.string.invalid_length));
            return;
```

```
        }

        String areaCode = phoneNumber.substring(0, 3);
        if (!areaCode.equals("040") && !areaCode.equals("041") && !areaCode.equals("050")
                && !areaCode.equals("0400") && !areaCode.equals("044")) {
            validationTextView.setText(getString(R.string.invalid_area_code));
            return;
        }

        validationTextView.setText(getString(R.string.valid_phone_number));
    }
}
```

Q.1) Construct image switcher using setFactory().

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ImageSwitcher extends JFrame {
    private JLabel imageLabel;
    private Timer timer;
    private int currentImageIndex;
    private ImageIcon[] images;

    public ImageSwitcher() {
        // Set up the JFrame
        setTitle("Image Switcher");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);
        setLocationRelativeTo(null);
```

```
        // Initialize images
        images = new ImageIcon[]{
                new ImageIcon("image1.jpg"),
                new ImageIcon("image2.jpg"),
                new ImageIcon("image3.jpg")
        };

        // Initialize components
        imageLabel = new JLabel(images[0]);
        add(imageLabel, BorderLayout.CENTER);

        // Set up timer to switch images every 2 seconds
        timer = new Timer(2000, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                currentImageIndex = (currentImageIndex + 1) % images.length;
                imageLabel.setIcon(images[currentImageIndex]);
            }
        });
        timer.start();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new ImageSwitcher().setVisible(true);
            }
        });
    }
}
```

Q.2) Write a program to search a specific location on Google Map.

```
<?php

function search_location_on_google_maps($location) {
    // Encode the location for the URL
    $encoded_location = urlencode($location);

    // Construct the Google Maps search URL
    $search_url = "https://www.google.com/maps/search/?api=1&query=" . $encoded_location;

    // Redirect the user to the search URL
    header("Location: " . $search_url);
    exit();
}

// Check if the location is provided through a form submission
if(isset($_POST['location'])) {
    $location = $_POST['location'];
    search_location_on_google_maps($location);
}

?>

<!DOCTYPE html>
<html>
<head>
    <title>Search Location on Google Maps</title>
</head>
<body>
    <h2>Enter a Location to Search on Google Maps</h2>
    <form method="post" action="">
```

```
    <input type="text" name="location" placeholder="Enter location">
    <button type="submit">Search</button>
  </form>
</body>
</html>
```

Q.1) Java Android Program to Demonstrate Alert Dialog Box.

```
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Button showAlertButton = findViewById(R.id.show_alert_button);
    showAlertButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        showAlert();
      }
    });
  }

  private void showAlert() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

```
        builder.setTitle("Alert Dialog")
                .setMessage("This is a simple alert dialog box.")
                .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // do something when OK button is clicked
                        dialog.dismiss(); // dismiss the dialog
                    }
                })
                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // do something when Cancel button is clicked
                        dialog.dismiss(); // dismiss the dialog
                    }
                });
        AlertDialog alertDialog = builder.create();
        alertDialog.show();
    }
}
```

Q.2) Create an Android application which will ask the user to input his / her name. A message should display the two items concatenated in a label. Change the format of the label using radio buttons and check boxes for selection. The user can make the label text bold, underlined or italic as well as change its color. Also include buttons to display the message in the label, clear the text boxes as well as label. Finally exit. [20 Marks]

```
public class MainActivity extends AppCompatActivity {
    private EditText nameInput;
    private TextView messageLabel;
    private RadioButton boldRadioButton, italicRadioButton, underlineRadioButton;
    private CheckBox colorCheckBox;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    nameInput = findViewById(R.id.nameInput);
    messageLabel = findViewById(R.id.messageLabel);
    boldRadioButton = findViewById(R.id.boldRadioButton);
    italicRadioButton = findViewById(R.id.italicRadioButton);
    underlineRadioButton = findViewById(R.id.underlineRadioButton);
    colorCheckBox = findViewById(R.id.colorCheckBox);

    Button displayButton = findViewById(R.id.displayButton);
    Button clearButton = findViewById(R.id.clearButton);
    Button exitButton = findViewById(R.id.exitButton);

    displayButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            displayMessage();
        }
    });

    clearButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            clearFields();
        }
    });

    exitButton.setOnClickListener(new View.OnClickListener() {
        @Override
```

```
        public void onClick(View v) {
            finish();
        }
    });
}

private void displayMessage() {
    String name = nameInput.getText().toString();
    StringBuilder message = new StringBuilder(name);

    if (boldRadioButton.isChecked()) {
        messageLabel.setTypeface(null, Typeface.BOLD);
    }
    if (italicRadioButton.isChecked()) {
        messageLabel.setTypeface(null, Typeface.ITALIC);
    }
    if (underlineRadioButton.isChecked()) {
        message.append(" (underlined)");
        messageLabel.setPaintFlags(messageLabel.getPaintFlags() | Paint.UNDERLINE_TEXT_FLAG);
    }

    if (colorCheckBox.isChecked()) {
        messageLabel.setTextColor(Color.RED);
    } else {
        messageLabel.setTextColor(Color.BLACK);
    }

    messageLabel.setText(message.toString());
}

private void clearFields() {
```

```
        nameInput.setText("");
        messageLabel.setText("");
        boldRadioButton.setChecked(false);
        italicRadioButton.setChecked(false);
        underlineRadioButton.setChecked(false);
        colorCheckBox.setChecked(false);
    }
}
```

Q.1) Java Android Program to demonstrate login form with validation. [10 Marks]

```
public class LoginActivity extends AppCompatActivity {
    private EditText usernameEditText;
    private EditText passwordEditText;
    private Button loginButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        usernameEditText = findViewById(R.id.usernameEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        loginButton = findViewById(R.id.loginButton);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                validateLogin();
            }
        });
    }
```

```
private void validateLogin() {
    String username = usernameEditText.getText().toString().trim();
    String password = passwordEditText.getText().toString().trim();

    if (username.isEmpty()) {
        usernameEditText.setError("Username is required");
        return;
    }

    if (password.isEmpty()) {
        passwordEditText.setError("Password is required");
        return;
    }

    if (!isValidCredentials(username, password)) {
        Toast.makeText(this, "Invalid username or password", Toast.LENGTH_SHORT).show();
        return;
    }

    // If credentials are valid, you can proceed with login
    Toast.makeText(this, "Login successful", Toast.LENGTH_SHORT).show();
    // Add your code to proceed with the login process (e.g., starting a new activity)
}

private boolean isValidCredentials(String username, String password) {
    // You can implement your own logic to validate the username and password
    // For example, you might check against a hardcoded list of valid credentials,
    // or validate against a database or API.
    // For demonstration purposes, let's assume a hardcoded valid username and password.
    return username.equals("admin") && password.equals("password");
}
```

```
}
```

Q.2) Write a program to search a specific location on Google Map

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private EditText searchEditText;
    private Button searchButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        searchEditText = findViewById(R.id.search_edit_text);
        searchButton = findViewById(R.id.search_button);

        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
```

```
            .findFragmentById(R.id.map_fragment);
        mapFragment.getMapAsync(this);

        searchButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String location = searchEditText.getText().toString().trim();
                if (!location.isEmpty()) {
                    searchLocation(location);
                }
            }
        });
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
    }

    private void searchLocation(String location) {
        // Perform search and show the location on the map
        // For simplicity, let's just show a hardcoded location
        LatLng searchLatLng = new LatLng(37.7749, -122.4194); // San Francisco, CA
        mMap.addMarker(new MarkerOptions().position(searchLatLng).title(location));
        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(searchLatLng, 12.0f));
    }
}
```

Q.1] Java Android Program to Demonstrate ProgressBar.

```
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {

    private ProgressBar progressBar;
    private int progressStatus = 0;
    private Handler handler = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        progressBar = findViewById(R.id.progressBar);
        Button startButton = findViewById(R.id.startButton);

        startButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Reset progress status
                progressStatus = 0;
                // Start the progress bar
                progressBar.setVisibility(View.VISIBLE);
                // Start the background thread to update progress
                new Thread(new Runnable() {
                    public void run() {
                        while (progressStatus < 100) {
                            // Update the progress status
                            progressStatus += 1;
```

```
                // Try to sleep for 100 milliseconds to simulate a long operation
                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                // Update the progress bar on the UI thread
                handler.post(new Runnable() {
                    public void run() {
                        progressBar.setProgress(progressStatus);
                    }
                });
            }
            // After completing background task, hide the progress bar
            handler.post(new Runnable() {
                public void run() {
                    progressBar.setVisibility(View.GONE);
                }
            });
        }
    }).start();
  }
 });
 }
}
```

Q.2] Create table Employee (E_id, name, address, ph_no). Create Application for performing the following operation on the table. (Using SQLite database). i] Insert record of 5 new Employees. ii] Show all the details of Employee.

```
<?php


// Connect to SQLite database
try {
```

```
    $db = new PDO('sqlite:employee.db');
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    echo 'Error: ' . $e->getMessage();
    exit();
}

// Create Employee table
try {
    $db->exec("CREATE TABLE IF NOT EXISTS Employee (
        E_id INTEGER PRIMARY KEY,
        name TEXT,
        address TEXT,
        ph_no TEXT
    )");
} catch(PDOException $e) {
    echo 'Error creating table: ' . $e->getMessage();
    exit();
}

// Insert records for 5 new employees
try {
    $stmt = $db->prepare("INSERT INTO Employee (name, address, ph_no) VALUES (:name, :address,
:ph_no)");

    $employees = [
        ['name' => 'John Doe', 'address' => '123 Main St, Anytown, USA', 'ph_no' => '123-456-7890'],
        ['name' => 'Jane Smith', 'address' => '456 Elm St, Othertown, USA', 'ph_no' => '987-654-3210'],
        ['name' => 'Alice Johnson', 'address' => '789 Oak St, Anycity, USA', 'ph_no' => '456-789-0123'],
        ['name' => 'Bob Brown', 'address' => '321 Pine St, Othercity, USA', 'ph_no' => '789-012-3456'],
        ['name' => 'Emily Davis', 'address' => '654 Maple St, Anystate, USA', 'ph_no' => '012-345-6789']
```

```
    ];

    foreach ($employees as $employee) {
        $stmt->execute($employee);
    }
} catch(PDOException $e) {
    echo 'Error inserting records: ' . $e->getMessage();
    exit();
}

// Show all details of employees
try {
    $stmt = $db->query("SELECT * FROM Employee");
    $employees = $stmt->fetchAll(PDO::FETCH_ASSOC);

    foreach ($employees as $employee) {
        echo "Employee ID: {$employee['E_id']}, Name: {$employee['name']}, Address: {$employee['address']}, Phone: {$employee['ph_no']}<br>";
    }
} catch(PDOException $e) {
    echo 'Error retrieving records: ' . $e->getMessage();
    exit();
}

// Close database connection
$db = null;

?>
```

Q.1] Create a Application which shows Life Cycle of Activity.

```
import android.os.Bundle;
import android.util.Log;
```

```
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate: ");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "onStart: ");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "onResume: ");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "onPause: ");
    }
```

```
    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop: ");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy: ");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "onRestart: ");
    }
}
```

Q.2] Create table Customer (id, name, address, ph_no). Create Application for performing the following operation on the table. (Using SQLite database). i] Insert new customer details (At least 5 records). ii] Show all the customer details. [20 Marks]

```
<?php

try {
    // Connect to SQLite database
    $db = new PDO('sqlite:customer.db');
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Create Customer table
    $db->exec("CREATE TABLE IF NOT EXISTS Customer (
        id INTEGER PRIMARY KEY,
        name TEXT,
```

```
        address TEXT,
        ph_no TEXT
    )");

    // Insert records for new customers
    $stmt = $db->prepare("INSERT INTO Customer (name, address, ph_no) VALUES (:name, :address, :ph_no)");

    $customers = [
        ['name' => 'Alice', 'address' => '123 Main St', 'ph_no' => '123-456-7890'],
        ['name' => 'Bob', 'address' => '456 Elm St', 'ph_no' => '987-654-3210'],
        ['name' => 'Charlie', 'address' => '789 Oak St', 'ph_no' => '456-789-0123'],
        ['name' => 'David', 'address' => '321 Pine St', 'ph_no' => '789-012-3456'],
        ['name' => 'Emily', 'address' => '654 Maple St', 'ph_no' => '012-345-6789']
    ];

    foreach ($customers as $customer) {
        $stmt->execute($customer);
    }

    // Show all customer details
    $stmt = $db->query("SELECT * FROM Customer");
    $customers = $stmt->fetchAll(PDO::FETCH_ASSOC);

    foreach ($customers as $customer) {
        echo "Customer ID: {$customer['id']}, Name: {$customer['name']}, Address: {$customer['address']}, Phone: {$customer['ph_no']}<br>";
    }

    // Close database connection
    $db = null;
} catch(PDOException $e) {
```

```
    echo 'Error: ' . $e->getMessage();
    exit();
}

?>
```

Q.1] Create an application that allows the user to enter a number in the textbox named „getnum". Check whether the number in the textbox „getnum" is Palindrome or not. Print the message accordingly in the label when the user clicks on the button „Check".

Q.2] Java android program to create simple calculator.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class PalindromeChecker extends JFrame {
    private JTextField textField;
    private JLabel resultLabel;

    public PalindromeChecker() {
        setTitle("Palindrome Checker");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        JLabel enterLabel = new JLabel("Enter a number:");
        add(enterLabel);

        textField = new JTextField(10);
        add(textField);

        JButton checkButton = new JButton("Check");
        checkButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
```

```
            checkPalindrome();
        }
    });
    add(checkButton);

    resultLabel = new JLabel();
    add(resultLabel);
}

private void checkPalindrome() {
    String input = textField.getText();
    try {
        int num = Integer.parseInt(input);
        if (isPalindrome(num)) {
            resultLabel.setText(num + " is a palindrome!");
        } else {
            resultLabel.setText(num + " is not a palindrome.");
        }
    } catch (NumberFormatException e) {
        resultLabel.setText("Invalid input. Please enter a valid number.");
    }
}

private boolean isPalindrome(int num) {
    String numStr = Integer.toString(num);
    StringBuilder reversed = new StringBuilder(numStr).reverse();
    return numStr.equals(reversed.toString());
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
```

```
            public void run() {
                new PalindromeChecker().setVisible(true);
            }
        });
    }
}
```

Q.1] Create an application that allows the user to enter a number in the textbox named getnum. Check whether the number in the textbox getnum is Armstrong or not. Print the message using Toast control when the user clicks on the button check.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class ArmstrongChecker extends JFrame {
    private JTextField textField;
    private JLabel resultLabel;

    public ArmstrongChecker() {
        setTitle("Armstrong Checker");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        JLabel enterLabel = new JLabel("Enter a number:");
        add(enterLabel);

        textField = new JTextField(10);
        add(textField);

        JButton checkButton = new JButton("Check");
        checkButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
```

```
            checkArmstrong();
        }
    });
    add(checkButton);

    resultLabel = new JLabel();
    add(resultLabel);
}

private void checkArmstrong() {
    String input = textField.getText();
    try {
        int num = Integer.parseInt(input);
        if (isArmstrong(num)) {
            displayToast(num + " is an Armstrong number!");
        } else {
            displayToast(num + " is not an Armstrong number.");
        }
    } catch (NumberFormatException e) {
        displayToast("Invalid input. Please enter a valid number.");
    }
}

private boolean isArmstrong(int num) {
    int originalNum = num;
    int sum = 0;
    int numDigits = String.valueOf(num).length();

    while (num > 0) {
        int digit = num % 10;
        sum += Math.pow(digit, numDigits);
```

```
            num /= 10;
        }

        return sum == originalNum;
    }

    private void displayToast(String message) {
        JOptionPane.showMessageDialog(this, message, "Result",
JOptionPane.INFORMATION_MESSAGE);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new ArmstrongChecker().setVisible(true);
            }
        });
    }
}
```

Q.2] Write a program to draw GUI by using Spinner, Buttons.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class GUIWithSpinnerAndButtons extends JFrame {
    private JSpinner spinner;
    private JButton increaseButton;
    private JButton decreaseButton;

    public GUIWithSpinnerAndButtons() {
        setTitle("Spinner and Buttons");
```

```
    setSize(300, 150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());

    // Create Spinner
    SpinnerModel spinnerModel = new SpinnerNumberModel(0, 0, 100, 1);
    spinner = new JSpinner(spinnerModel);
    add(spinner);

    // Create Increase Button
    increaseButton = new JButton("Increase");
    increaseButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            int value = (int) spinner.getValue();
            spinner.setValue(value + 1);
        }
    });
    add(increaseButton);

    // Create Decrease Button
    decreaseButton = new JButton("Decrease");
    decreaseButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            int value = (int) spinner.getValue();
            spinner.setValue(value - 1);
        }
    });
    add(decreaseButton);
}

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

```
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new GUIWithSpinnerAndButtons().setVisible(true);
            }
        });
    }
}
```