

Programming in Java

Assignment – 3



Name : Sanjay R

Roll Number : 2016124

Department : Electronics and Instrumentation Engineering

Problem statement:

Create a Java applet that displays a calendar for the current month.

Description:

The Calendar Applet is a graphical user interface program that presents a calendar for the current month. It utilizes the `java.applet.Applet` class to create a Java applet, which can be embedded in a web page and run in a web browser.

The applet features include:

Displaying the current month and year at the top of the calendar.

Drawing the days of the week (Sunday to Saturday) as column headers.

Drawing the calendar days for the month, starting from the appropriate day of the week..

Concepts used:

1. **Java Applet:** The program is developed as a Java applet using the `java.applet.Applet` class. The applet provides a graphical interface that can be displayed in a web browser.
2. **Graphics:** The `java.awt.Graphics` class is used to draw the calendar on the applet's canvas. The `paint()` method is overridden to handle the drawing operations.
3. **Date and Time:** The `java.time.LocalDate` class is used to retrieve the current date and extract the month, year, and other date-related information. It provides convenient methods for working with dates.
4. **Graphical User Interface (GUI):** The applet's interface is designed using graphical elements such as fonts, colors, and drawing operations. The `java.awt.Font` class is used to set the font style and size for the text elements.
5. **Layout and Positioning:** The calendar layout is achieved by defining positions and sizes for the calendar cells and using loops to draw the days of the month in the appropriate locations.

Program Code:

```
import java.applet.Applet;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.time.LocalDate;

/*

<Applet code=CalendarApplet width=450 height=450>

</Applet>

*/

public class CalendarApplet extends Applet {

    private LocalDate currentDate;

    public void init() {

        currentDate = LocalDate.now();

        setSize(400, 300);

        setBackground(Color.WHITE);

    }

    public void paint(Graphics g) {

        int x = 50;

        int y = 50;

        int cellWidth = 50;

        int cellHeight = 30;

        // Draw month and year

        g.setFont(new Font("Arial", Font.BOLD, 16));

        g.drawString(currentDate.getMonth().toString() + " " + currentDate.getYear(), x, y);

        // Draw days of the week

        y += cellHeight;
```

```
String[] daysOfWeek = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

g.setFont(new Font("Arial", Font.BOLD, 14));

for (int i = 0; i < daysOfWeek.length; i++) {

    g.drawString(daysOfWeek[i], x + i * cellWidth, y);

}


// Draw calendar days

y += cellHeight;

int daysInMonth = currentDate.lengthOfMonth();

int firstDayOfWeek = currentDate.withDayOfMonth(1).getDayOfWeek().getValue();

int currentDay = 1;

g.setFont(new Font("Arial", Font.PLAIN, 14));

for (int row = 0; row < 6; row++) {

    for (int col = 0; col < 7; col++) {

        if (row == 0 && col < firstDayOfWeek - 1) {

            continue;

        }

        if (currentDay > daysInMonth) {

            break;

        }

        g.drawString(String.valueOf(currentDay), x + col * cellWidth, y + row * cellHeight);

        currentDay++;

    }

}

}
```

Screenshots:

