**1. //Design**



**//Code**

import java.util.Scanner;

import java.util.Calendar;

import java.util.GregorianCalendar;

public class mod4CalendarProgram1 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int month = getMonthFromUser(input);

int year = getYearFromUser(input);

printMonthCalendar(month, year);

input.close();

}

private static int getMonthFromUser(Scanner input) {

System.out.print("Enter a month (1-12): ");

return input.nextInt();

}

private static int getYearFromUser(Scanner input) {

System.out.print("Enter a year (e.g., 2023): ");

return input.nextInt();

}

private static void printMonthCalendar(int month, int year) {

printMonthHeader(month, year);

printMonthBody(month, year);

}

private static void printMonthHeader(int month, int year) {

System.out.println("\n" + "\t" + getMonthName(month) + " " + year);

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

System.out.println(" "+"Sun Mon Tue Wed Thu Fri Sat");

}

private static void printMonthBody(int month, int year) {

int startDay = getStartDay(month, year);

int numDaysInMonth = getNumDaysInMonth(month, year);

// Print leading spaces for the first week

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

System.out.print(" ");

}

// Print the days of the month

for (int day = 1; day <= numDaysInMonth; day++) {

System.out.printf("%3d ", day);

// Move to the next day

startDay++;

if (startDay > 7) {

startDay = 1;

System.out.println();

}

}

}

private static String getMonthName(int month) {

String[] monthNames = {

"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"

};

return monthNames[month - 1];

}

private static int getStartDay(int month, int year) {

// Adjust month number & year to fit Zeller's numbering system

if (month < 3) {

month += 12;

year -= 1;

}

int centuryYear = year % 100; // Calculate year within century

int centuryTerm = year / 100; // Calculate century term

int firstDayInMonth = 0; // Day number of first day in month 'm'

firstDayInMonth = (1 + // to shift index 0 to the 1-7 return range

(13 \* (month + 1) / 5)

+ centuryYear +

(centuryYear / 4) +

(centuryTerm / 4) +

(5 \* centuryTerm)) % 7;

// Convert Zeller's value to ISO value (1 = Mon, ... , 7 = Sun )

int dayNum = ((firstDayInMonth + 5) % 7) + 1;

return dayNum;

}

private static int getNumDaysInMonth(int month, int year) {

Calendar calendar = new GregorianCalendar(year, month - 1, 1);

return calendar.getActualMaximum(Calendar.DAY\_OF\_MONTH);

}

private static boolean isLeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}//end main

}//end class

**//output**



**2. //Design**



**//Code**

import java.util.Scanner;

import java.util.Calendar;

import java.util.GregorianCalendar;

public class mod4CalendarProgram2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a year (e.g., 2023): ");

int year = scanner.nextInt();

for (int month = 1; month <= 12; month++) {

printMonthCalendar(month, year);

}

}

private static void printMonthCalendar(int month, int year) {

printMonthHeader(month, year);

printMonthBody(month, year);

}

private static void printMonthHeader(int month, int year) {

System.out.println("\n" + "\t" + getMonthName(month) + " " + year);

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

System.out.println(" "+"Sun Mon Tue Wed Thu Fri Sat");

}

private static void printMonthBody(int month, int year) {

int startDay = getStartDay(month, year);

int numDaysInMonth = getNumDaysInMonth(month, year);

// Print leading spaces for the first week

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

System.out.print(" ");

}

// Print the days of the month

for (int day = 1; day <= numDaysInMonth; day++) {

System.out.printf("%3d ", day);

// Move to the next day

startDay++;

if (startDay > 7) {

startDay = 1;

System.out.println();

}

}

}

private static String getMonthName(int month) {

String[] monthNames = {

"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"

};

return monthNames[month - 1];

}

private static int getStartDay(int month, int year) {

// Adjust month number & year to fit Zeller's numbering system

if (month < 3) {

month += 12;

year -= 1;

}

int centuryYear = year % 100; // Calculate year within century

int centuryTerm = year / 100; // Calculate century term

int firstDayInMonth = 0; // Day number of first day in month 'm'

firstDayInMonth = (1 + // to shift index 0 to the 1-7 return range

(13 \* (month + 1) / 5)

+ centuryYear +

(centuryYear / 4) +

(centuryTerm / 4) +

(5 \* centuryTerm)) % 7;

// Convert Zeller's value to ISO value (1 = Mon, ... , 7 = Sun )

int dayNum = ((firstDayInMonth + 5) % 7) + 1;

return dayNum;

}

private static int getNumDaysInMonth(int month, int year) {

Calendar calendar = new GregorianCalendar(year, month - 1, 1);

return calendar.getActualMaximum(Calendar.DAY\_OF\_MONTH);

}

private static boolean isLeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}//end main

}//end class

**// output - continues on past september, but could not capture.**

