Java

```
class Date {
    private int month;
    private int day;
    private int year;

    // Constructor 1: Initializes all three instance variables
    public Date(int month, int day, int year) {
        this.month = month;
        this.day = day;
        this.year = year;
    }

    // Constructor 2: Initializes month and day, sets year to 2023
    public Date(int month, int day) {
        this.month = month;
        this.day = day;
        this.year = 2023;
    }

    // Overloaded displayDate() method to display date in different formats
    public void displayDate(String format) {
        if (format.equals("/")) {
            System.out.println(day + "/" + month + "/" + year);
        } else if (format.equals(".")) {
            System.out.println(month + "." + day + "." + year);
        } else {
            System.out.println("Invalid format. Please use '/' or '.'.");
        }
    }
}

public class DateTest {
    public static void main(String[] args) {
        // Create a Date object with all three parameters
        Date date1 = new Date(10, 25, 2023);

        // Create a Date object with only month and day
        Date date2 = new Date(12, 31);

        // Display dates in different formats
        date1.displayDate("/");
        date1.displayDate(".");

        date2.displayDate("/");
        date2.displayDate(".");
    }
}
```

Use code with caution.

Explanation:

1. **Date Class:**
   - Defines three instance variables: `month`, `day`, and `year`.
   - Provides two constructors:
     - One takes all three arguments and initializes the variables.
     - Another takes only month and day, setting the year to 2023 by default.
   - Overloads the `displayDate()` method to display the date in two formats:
     - With forward slashes `/`
     - With full stops `.`
2. **DateTest Class:**
   - Creates two `Date` objects, one with all three parameters and another with only month and day.
   - Calls the `displayDate()` method on both objects with different formats to demonstrate their usage.