**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Lab 4 | |
| **Topic** | Class, object and getter, setter, constructor, and destructor |
| **Objective** | **To practice implementing a class with getter and setter methods, constructors, and a destructor in** |

## Task 1: Define a `Student` class with the following private attributes, and public methods:

## Attributes:

## name (string): Represents the student's name.

## studentID (int): Represents the student's ID.

## grade (float): Represents the student's grade.

## Methods:

## Default constructor: Initializes all attributes to "0" or "Null".

## Parameterized constructor: Initializes all attributes to user-defined values.

## Getter and Setter methods for each attribute (i.e., getName(), setName(), getStudentID(), setStudentID(), getGrade(), setGrade() ).

## Display method: displayStudentInfo() to display the student's name, ID, and grade.

## Destructor: Implement a destructor in the `Student` class to release any resources allocated during the object's lifetime. You can add a message in the destructor to indicate when the destructor is called

## Main Program (Separate File):

## Create instances of the `Student` class using both the default and parameterized constructors.

## Use getter and setter methods to modify and retrieve student attributes.

## Display the information of the students created.

## Task 2: Create a `Calendar` Class. This class should represent a simple calendar and provide methods for manipulating and accessing dates.

## Attributes:

## year (int): Represents the current year.

## month (int): Represents the current month.

## day (int): Represents the current day.

## Methods:

## Constructor: Initialize the year, month, and day attributes.

## Setter methods: Provide setYear(int), setMonth(int), and setDay(int) methods to set the calendar date.

## Getter methods: Implement getYear(), getMonth(), and getDay() to return the calendar date.

## Other Method:

## printDate(): Print the current date in a human-readable format.

## advanceDays(int numDays): Calculate and return the date after advancing a specified number of days.

## isLeapYear(): Check if the current year is a leap year.

## Main (in a separate file):

## Create instances of the Calender class using both the default and parameterized constructors.

## Use getter and setter methods to modify and retrieve calendar attributes.

## Display the current date, display the date after a certain number of days.

## Tell if the specific year is a leap year