**dayType Class Project**

**Project Objectives**

* Class definition is contained in a header (.h file).
* Class contains appropriate constructors.
* Header file is appropriately documented (precondition/post condition comments are included).
* Class implementation is contained in a separate implementation file (.cpp).
* Modifier functions (including constructors) contain input validation as appropriate.
* A client program is included which adequately tests the class functionality.
* A Git repository has been created and the source code has been committed to this repository.

**Report Instructions**

Read the problem requirements in the project instructions section below.

Plan the project.

Go the class team on replit and write and test your code in the designated project folder.

When your project is ready to be graded write a project report with the following format:

* Name of program (title, number, or both)
* Your name
* Name of your partner (if you have one)
* Planning notes – UML diagrams, IPO charts for member functions, handwritten notes, etc.
* Answer the following reflection questions:

1. What did you find most challenging with this program?

2. What problems did you encounter and how did you solve them?

3. What did you learn from writing this program?

Your final project must be in .pdf format. Upload your project report(s) to the assignment in Canvas and submit.

**Project Instructions**

We will be working on a project to design a class calendarType, so that a client program can use this class to print a calendar for any month starting January 1, 1900.

An example of the calendar for September 2019 is:Table

Description automatically generated

**We will develop several classes which will work together to create a calendar.**

The first of these classes is called dayType which will manipulate a day of the week. In addition to the day (which is stored as a string), the class has a static array to hold the valid names for the day variable. Remember, static features belong to the class. Non-static class features (attributes or operations) belong to individual instances of the class (i.e., to an object). The class should be able to perform the following operations on an object of type dayType:

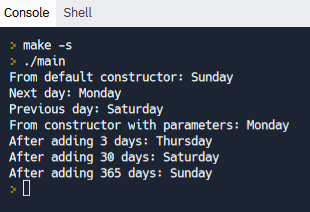
1. Set the day.
2. Print the day.
3. Return the day.
4. Return the next day.
5. Return the previous day.
6. Set the day by adding a certain number of days to the current day. For example, if the current day is Monday and we add 4 days, the new day is Friday. Similarly, if today is Tuesday and we add 13 days, the new day is Monday.
7. Include the appropriate constructors.

Design a UML diagram for the dayType class. Unless otherwise specified, features in a UML class diagram are non-static. The UML diagram denotes static features by underlining the feature. The write the header file and an implementation file. **Document the header file appropriately.**

Write a client program should do the following:

1. Create a day object using the default constructor and use the print function to display the instance variable. The output should be “Sunday”.
2. Display the previous day (output should be “Saturday”).
3. Display the next day (output should be “Monday”).
4. Create a day object using the constructor with parameters and initialize the object to “Monday”.
5. Use get function to display the value of the instance variable (should be “Monday”).
6. Using the second object, add 3 days to the current day and display the new day value (should be “Thursday”).
7. Use the second object to call add 30 days to the current day and display the new day value (should be “Saturday”).
8. Use the second object to add 365 days to the current day and display the new day value (should be “Sunday”).

Label your output so that the user (i.e., me, your instructor) can tell what your program is doing. This client program will serve as your test to ensure that the class is working as it is supposed to. Here is an example of the labeled output:



**Once you are satisfied with your class and test program, create a Git repository for this class and make sure to commit all your modifications. This will be your main branch.**