Purpose:

Track the name of a day. Return yesterday and/or tomorrow. Return what day it will be after some number of days have passed.

Members:

-Private Members:

*string dayName*: name of the day of the week

*int dayId:* numeric identifier for day of the week (0 - 6)

A. 0 = Sunday, 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday

*static string dayArray[7]:* “SUN”, “MON”, “TUE”, “WED”, “THU”, “FRI”, “SAT”

Static because this array will be the same across all objects; uppercase intentionally

Methods

*DayType( ):* Default constructor; dayName = “UNSET”, dayId = -1

These are default values for testing. Ease of detecting problems when implementing

*DayType(string):* Constructor with name of the day sent in and set; dayName = *string*

*DayType(int):* Constructor with dayId = *int*

*string getDay( ) const:* Accessor; return the name of the day of the week. Constant method

*void setDay(const int& new):* Mutator; change the object’s day. Iterate through an day array

*string tomorrow( )*: return the name of tomorrow

*string yesterday( ):* return the name of tomorrow

*string futureDay(const int& future):* return the name of the day for some given int value in the future

Algorithms

*DayType( ):*

dayName = “UNSET”

dayId = -1

*DayType(string):*

*string* to uppercase (for comparison)

for(i = 0; i < dayArray.size( ); i++)

if(dayArray[i][0] == *string*[0] && dayArray[i][1] == *string*[i])

dayName = dayArray[i]

*DayType(int):*

if(*int* > 6 || *int* < 0)

cout “Invalid number. Calling default constructor”

DayType( )

else

for(i = 0; i < dayArray.size( ); i++)

if(*int* == i) dayName = dayArray[i]

*string getDay( ) const:*

return dayName

*void setDay(new):*

/\*Note: (new - 1) assumes user input will be sent as 1 to 7, not 0 to 6\*/

dayName = dayArray[new - 1]

*string tomorrow( ):*

int tomorrow

if(dayId == 6) tomorrow = 0

else tomorrow = dayId + 1

return dayArray[tomorrow]

*string yesterday( ):*

int yesterday

if(dayId == 0) yesterday = 6

else yesterday = dayId - 1

return dayArray[yesterday]

*string futureDay(const int& future):*

result = dayId

for(i = 0; i < future; i++)

/\*result must stay within bounds of dayArray\*/

result++

if(result > 6) result = 0

return dayArray[result]