date.cpp

Class date

Variables

Private

- Int day
- Int month
- Int year

Methods

Public

- Constructors
  - Date()
    - Day - Initialize to 01
    - Month - Initialize to 01
    - Year - Initialize to 2023
  - Date(d: int, m:int, y:int)
    - If m > 12 or m < 1
      use default values for date (01/01/23)
      Else month = m
    - If month = any of these (1, 3,5,7,8,10,12)
      If d > 31 or d < 1
      Then use default values for date (01/01/23)
      Else day = d
    - If month = any of these (4,6,9,11)
      If d > 30 or d < 1
      use default values for date (01/01/23)
      Else day = d
    - If month = 2
      If d > 28 or d < 1
      use default values for date (01/01/23)
      Else day = d
    - If y > 2024 or y < 2023
      use default values for date (01/01/23)
      Else year = y

- Setters
  - setDay(d: int): bool
    - If d < 1 or d > 31
      Return false
    - If month = any of these (1, 3,5,7,8,10,12)
      If d > 31 or d < 1
      Return false
      Else If month = any of these (4,6,9,11)

If d > 30 or d < 1
Return false
Else if month = 2
If d > 28 or d < 1
Return false
Else
Day = d
Return true

- setMonth(m:int): bool
  - If m = any of these (1,3,5,7,8,10,12)
    If day > 31
    Return false
    Else If m = any of these (4,6,9,11)
    If day > 30
    Return false
    Else if m = 2
    If day > 28
    Return false
    Else
    Month = m
    Return true
- setYear(y:int): bool
  - If y > 2024 or y < 2023
    Return false
    Else
    year = y
    Return true
- addDays(days: int)
  - day += days
  - If m = any of these (1,3,5,7,8,10,12)
    If day > 31
    day = day - 31
    month++
    Else If m = any of these (4,6,9,11)
    If day > 30
    day = day - 30
    month++
    Else if m = 2
    If day > 28
    day = day -28
    month++

- Getters
  - getDay(): int
    - Return day

- getMonth(): int
    - Return month
- getYear(): int
    - Return year
- showDate(): string
    - Initialize a string fullDate as day + “/” + month + “/” + year
    - Return fullDate

calcDays.cpp

A function that takes two instances of date and returns the days between them

```
Int daysDiff1
Int daysDiff2
```

Create 2d array monthsList with 12 rows, 2 columns. Column 1 is months, column 2 is amount of days in that month

```
Find difference in date1 day
        If date1 day = 01
                daysDiff1 += 1
        Else
                daysDiff1 += (date1 day - 1)
If date1 year is 2024
        daysDiff1 += 365
If date1 month != 1
        Int monthsDiff = date1 month
        While (monthsDiff != 1)
                daysDiff1 += monthsList[monthsDiff] [2]
                monthsDiff - -
Find difference in date2 day
        If date2 day = 01
                daysDiff2 += 1
        Else
                daysDiff2 += (date2 day - 1)
If date2 year is 2024
        daysDiff2 += 365
If date2 month != 1
        Int monthsDiff = date2 month
        While (monthsDiff != 1)
                daysDiff2 += monthsList[monthsDiff] [2]
                monthsDiff - -
If daysDiff1 > daysDiff2
        return daysDiff1 - daysDiff2
Else
        return daysDiff2 - daysDiff1
```

Main.cpp

```
Create a string testResult
Create an int lengthOfIsolation
Create a string stringDate
Create an int day
Create an int month
Create an int year
Ask user for their test result
Store answer in testResult
If testResult is positive //case 1
        Ask user for date tested positive
        Store user response in stringDate
        month = stoi(stringDate.substr(0,2))
        day = stoi(stringDate.substr(3,2))
        year = stoi(stringDate.substr(6,4))
        create an instance of class date called datePositive using constructor Date(day, month,
        year)
        lengthOfIsolation = 7
        output testResult
        output datePositive.showDate()
        output lengthOfIsolation
        output datePositive.addDays(lengthOfIsolation)
if testResult is negative
        create a string exposedToPositive
        ask user if they were exposed to a positive case
        store answer in exposedToPositive
        if exposedToPositive = “No” //case 2
                lengthOfIsolation = 0
                output testResult
                output exposedToPositive
                output lengthOfIsolation
        else //exposedToPositive is yes here, so either case 3 or 4
                create a string hasSecondDose
                create a bool fullyVaccinated
                ask user when they were exposed to a positive case
                Store user response in stringDate
                month = stoi(stringDate.substr(0,2))
                day = stoi(stringDate.substr(3,2))
                year = stoi(stringDate.substr(6,4))
```

```
create an instance of class date called dateExposed using constructor Date(day,
month, year)
ask user if they have had a second dose
store answer in hasSecondDose
if hasSecondDose is Yes //checking to see if person is fully vaccinated
        create an int daysSinceSecond
        ask user when they received their second dose
        store answer in string date
        month = stoi(stringDate.substr(0,2))
        day = stoi(stringDate.substr(3,2))
        year = stoi(stringDate.substr(6,4))
        create an instance of class date called dateSecondDose using
        constructor Date(day, month, year)
        daysSinceSecond = calcDays(dateExposed,dateSecondDose)
        if daysSinceSecond > 14
                 fullyVaccinated = true
        else
                fullyVaccinated = false
else
        if fullyVaccinated is true //case 3
                lengthOfIsolation = 3
                output testResult
                output exposedToPositive
                output dateExposed.showDate()
                output hasSecondDose
                output dateSecondDose.showDate()
                output fullyVaccinated
                output lengthOfIsolation
                output datePositive.addDays(lengthOfIsolation)
        else //case 4
                lengthOfIsolation = 12
                lengthOfIsolation = 3
                output testResult
                output exposedToPositive
                output dateExposed.showDate()
                output hasSecondDose
                output fullyVaccinated
                output lengthOfIsolation
                output datePositive.addDays(lengthOfIsolation)
```