

Computer Science - Programming 2 - Object Oriented
Laboratory class #5

Define and implement Date class representing information concerning dates belonging to range from 01 Jan 1901 to 31 Dec 2099.

The parts should be realized in the order of their numbers.

For parts I-IV assume that there is no leap years (there is always 28 days in February).

General hint: Desired output is presented in output.txt file (it is the output of the complete program - for parts I-V there is no days of the week names)

Part I 1.5 point

- implement (in Date.cpp file) isValid method (it is already declared in Date.h file), which should return true if the values of year, month and day fields specified the valid date from the considered range and false otherwise, you can use this method in implementation of other methods
- uncomment Part I in main function (Lab05_Main.cpp file)
- add all needed declarations (Date.h file) and implementations (Date.cpp file)
- if parameters values are incorrect, the constructor should create the date with year, month and day fields equal to 1
- parameterless constructor creates date with year, month and day fields equal to 1

Hint 1: use monthsLengths array for data validation

Hint 2: use monthsNames array for output

Part II 1 point

- uncomment Part II in main function (Lab05_Main.cpp file)
- add all needed declarations (Date.h file) and implementations (Date.cpp file)

Note: you can skip this part (and return to it later) if you want

Part III 2 points

- uncomment Part III in main function (Lab05_Main.cpp file)
- add all needed declarations (Date.h file) and implementations (Date.cpp file)
- the parameter of the single-parameter constructor describes the number of the days in considered range (01 Jan 1901 to 31 Dec 2099) e.g.
 - parameter equals 1 means 01 Jan 1901,
 - parameter equals 2 means 02 Jan 1901,
 - parameter equals 31 means 31 Jan 1901,
 - parameter equals 32 means 01 Feb 1901,
 - parameter equals 366 means 01 Jan 1902,
 - and so on
- if the parameter isn't positive or exceeds the number of the days in the considered periods, the constructor should create the date with year, month and day fields equal to 1
- DayFromStart method for the given dates returns its number in the considered period (it does reverse computation than the constructor described above)
 - for date 01 Jan 0001 DayFromStart method should return -1

Hint: use monthsLengths array in computations

Part IV 1.5 point

- uncomment Part IV in main function (Lab05_Main.cpp file)
- add all needed declarations (Date.h file) and implementations (Date.cpp file)
- if any dates subtraction argument is 01 Jan 0001, the operator should return INT_MIN value (defined in climits standard header file)
- for other operators if their argument is 01 Jan 0001 they should return 01 Jan 0001

Hint1: use members implemented in part III (then the implementation of each element of part IV will be very short and easy)

Hint2: remember about references (&)

Part V 1 point

- uncomment Part V in main function (Lab05_Main.cpp file)
- correct implementations (Date.cpp file) of the members defined so far taking leap years into account

Hint: data validation and implementation of part III need corrections

Part VI 1 point

- uncomment Part VI in main function (Lab05_Main.cpp file)
- add all needed declarations (Date.h file) and implementations (Date.cpp file)
- add information about a day of the week in the output
- the day of the week for 01 Jan 0001 is "****"

Note: 01 Jan 1901 is Tuesday

Hint1: use daysOfWeek array for the output

Hint2: consider adding const modifier to some previous defined members