Anthony Snow

CS-216

11/7/2014

Program 3

## Program Design 3 Documentation

PROBLEM DEFINITION

Your teacher Robert Chirwa wants you to understand the importance of Object Oriented Programming, and has asked you to create a Program that takes a date from the user and is able to be manipulated and compared to other dates. Write this program so you get an A.

PROBLEM ANALYSIS

A date has three parts. The first part being a year, second being a month, and the third being a day. Each month has a different number of days.

INPUT: we want the user to give us a date

OUTPUT: we want to give the user the option of choosing to display the date as a string which uses the string library. And whether or not they want to display “mm/dd/yyyy” or “month, day year”.

LIBRARY

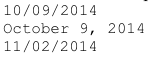
|  |  |
| --- | --- |
| NAME | PURPOSE |
| Iostream | Output objects such as cout and “<<” |
| String | Use string type for names |
| Cstdlib | Convert string to int |
| Sstream | Convert int to string |
| Stdio.h | xcode |
| Ctime | Get current time |

VARIBALES

|  |  |  |
| --- | --- | --- |
| NAME | TYPE | PURPOSE |
| Year | Int | Hold a year |
| Month | Int | Hold a month |
| Day | Int | Hold a day |
| Date | Class | Hold a date |
| shortDisplay | Bool | Holds true or false |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

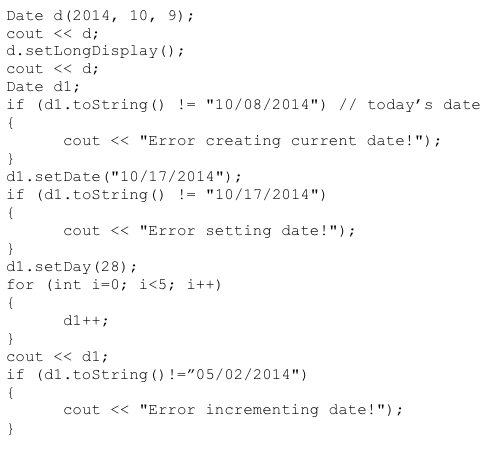
DESIGN

Output Design



Pretty simple. I wish it was that simple to make it.

Input Design



Data Structure Design

The program will be organized in a class called **Date**



Table - These are the required classes\*

I added some functions on my own. They are as follows:

+Boundary(): int

+operator !=(Date& otherDate, Date& otherDate2);

Algorithm Design

The first tasked performed by the program is to accept a date from them. If everything checks out okay, it will be stored in the respect variables for year, month, and day.

And from there, it can be manipulated in multiple ways that will be explained below.

Function main

Here is a list of tasks that the program will perform

1. Read a date from the user.
2. Validate that date.
3. Process that date
4. Be able to be compared and incremented
5. Output the results

Main function psuedocode

1. Declare the variables (including class, example Date d();)
2. Output the value of d.
3. Set to long display
4. Output the value again
5. Set a new declaration, Date d2, empty, giving it the default value of today’s date
6. Compare it to the same date, if they are not equal, tell the user so
7. Set the day to the 28th.
8. Now increment that by five.
9. Convert to string and compare once again. If they aren’t equal tell the user so.

Member Function operator<<

This function overloads the “<<” so it can be used with the **Date** class so depending on the form used, will output the date on the screen. The parameters are the ostream& and Date& otherDate.

Operator << pseudocode

Parameters: ostream& output, Date& otherDate

If short display is true:

Print the date in the mm/dd/yyyy form

If short display is false

Print the date in the month day, year form.

Return output

Member Function operator >>

Used to get direct access from user.

Operator >> pseudocode:

Ask for year

Ask for month

Ask for day

Return input

Member Function Date::Date();

First constructor, this is the default one. If nothing is got back from the user, it gives you the current date. Sets the short display to true as well (this is true for all constructors).

Pseudocode:

Short display is set to true

Set time\_t = to 0 so we get current time

Setyear to value + 1900

Setmonth to value + 1

Setday to value

Member Function Date::Date(int yy, int mm, int dd)

If the user enters variable for all of the parameters (year month and day) it goes to this one. Which uses the set(year, month, day) function to the respective variable.

Pseudocode

Set shortdisplay to true

Get the input from user

Set them to respective variables

Member Function Date::Date(int yy, int mm)

Used if the user gives to parametes, the day is set to the default variable of 1 and year and month are set in respective variables.

Pseudocode

Set shortdisplay to true

Get input from user

Set them respectively

Day will equal one.

Member Function Date::Date(int yy)

if only one parameter is given. Will set the day and month equal to one and year whatever the user chooses.

Pseudocode

Set shortdisplay to true

Get year from user

set it.

Month and day both equal one.

Member Function

Date::Date(Date &otherDate)

Sets this class with class parameter to whatever the variables of it are.

Psuedocode

Shortdisplay is true

Get value

Set it

Member Function Date::Date(string)

A string variant of the constructor. Gives a default value of January 1, 1900

Pseudocode :

Get value, if it doesn’t pass validation, pass default date.

Member Function setDate

okay there is five variants of the setDate function. The first three take 1-3 parameters (year, month, and day); if a parameter is missing, it is set to the default value of 1900,1,and 1 for year, month, and day respectively.

The class and string setDate member functions return January 1, 1900.

Psuedocode

Set the Variables

Validate they are within proper ranger, if not they return default values.

If a parameter is left empty, it is also set to default value.

\*string setDate function\*

searches for classes in the date

then depeneding on the location of the slashes looks for the numbers.

Member Function setMonth

has a parameter of mm. Checks to see if month is between 1 and 12.

Pseudocode

If year is greater than 1900.

Set it equal.

If not, give it value one

Member Function setYear

Has a parameter yy. If it is greater than 1900, returns value

Pseudocode

If year is greater than 1900, set it equal

If not, give it value 1900.

Member Function setDay

has parameter dd. If the parameter is between 1 and the number of days in that month, set it equal, if not set day to one.

Member Function get(Year, month, and day)

these three functions just return the value of the corresponding function.

Member Function monthString

takes one parameter month, and returns a string of that corresponding month.

Member Function setShortDisplay

set display to true, which gives the mm/dd/yyyy form when outputted.

Pseudocode

Set display to true

Member Function setLongDisplay

set display to false, which gives month day, year form.

Member Function Operator ==

Used to see if two class objects are equal. If they are, they print true.

Parameters of two referential date classes.

Psdueocode

Boolean called ans

If years are equal

if months are equal

and if days are equal, they are the equal.

If not, it is false

Return the answer

Member Function Operator !=

Parameters of two referential date classes. Checked to see if two classes are not equal

Pseudocode

Boolean called ans

If years aren’t equal. This is true

Else if the months aren’t equal, this is true

Then if the days are not equal, this is true

Else this is false;

Member Function Operator < used to see if a date is smaller than another date.

Taking in two referential parameter variables.

Pseudocode

If date 2 is bigger, its true

If the years are equal, check month

If months 2 is bigger its true

If months are equal we check day.

Else it is false.

Member Function operator++

increment the day by one.

Declare class.

If day goes over boundary of that month, set day to one and add a month. If month goes over 12, add 1 to year, if year goes over 9999, year = 1900. The += function uses this SAME exact algorithm but uses a for loop to increment depending on the number.

Member Function boundary()

Set the boundary for that month. Basically

Pseudocdoe

Parameter: month

Depending on the month, it returns the end month of that day. Default value of one.

Member Function Date::tostring()

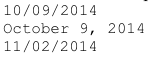
Converts the date to a string.

Takes the integers and convert them using string stream.

**Implementation**

The program was developed in the Xcode programming environment. The GNU g++ compiler was used to produce executable files.

Here is sample output:



Results are unpredictable when the files are in the wrong format. For example if characters are in the votes file, the program may go into an infinite loop or crash.