

```

// weddingdates.cpp (c) 1999-2002 Kari Laitinen

#include <iostream.h>
#include <sstream> // class stringstream etc.
#include <string> // class string etc.

#include "class_current_date.h"

int main()
{
    Current_date date_to_increment ;

    int number_of_dates_printed = 0 ;

    cout << "\n These are easy-to-remember dates for weddings and"
         "\n other important events because the days and months"
         "\n consist of the digits used in the year: \n" ;

    while ( number_of_dates_printed < 60 )
    {
        stringstream date_as_stream ;

        date_as_stream << date_to_increment ;

        string day_as_string, month_as_string, year_as_string ;

        getline( date_as_stream, month_as_string, '/' ) ;
        getline( date_as_stream, day_as_string, '/' ) ;
        getline( date_as_stream, year_as_string ) ;

        if ( day_as_string.find_first_not_of( year_as_string ) ==
            string::npos &&
            month_as_string.find_first_not_of( year_as_string ) ==
            string::npos )
        {
            if ( number_of_dates_printed % 5 == 0 )
            {
                cout << "\n" ;
            }

            cout << " " << date_to_increment ;

            number_of_dates_printed ++ ;
        }

        date_to_increment.increment() ;
    }
}

```

Right after its declaration, object `date_to_increment` contains the current date of the computer. This date will be incremented hundreds of times when the program is executed.

The contents of `date_to_increment` is incremented to the next date. This program simply checks hundreds of dates to see which dates fulfil the criteria for a nice wedding date.

A valid wedding date has been found here. A newline is printed after every 5th date. The date object can be printed this way as operator `<<` is defined in class `Date`, the parent class of `Current_date`.

weddingdates.cpp - 1.+ Using stringstream and string objects to find the best wedding dates.

To read `string` objects, there exists a version of function `getline()` which takes (as its first argument) the stream from where the reading will take place. The second argument is the `string` object which receives the characters from the stream. The third argument specifies which is the character that terminates the reading activity. If there is no third argument, `getline()` reads until it encounters a new-line character or the end of the stream.

Since operator `<<` is defined in class `Date`, and it is inherited from there for `Current_date` objects, it is possible to output a `Current_date` object to `date_as_stream` in this way. `date_as_stream` is empty before this statement is executed.

```
while ( number_of_dates_printed < 60 )
{
    stringstream date_as_stream ;

    date_as_stream << date_to_increment ;

    string day_as_string, month_as_string, year_as_string ;

    getline( date_as_stream, month_as_string, '/' ) ;
    getline( date_as_stream, day_as_string, '/' ) ;
    getline( date_as_stream, year_as_string ) ;

    if ( day_as_string.find_first_not_of( year_as_string ) ==
        string::npos &&
        month_as_string.find_first_not_of( year_as_string ) ==
        string::npos )
    {
```

This boolean expression tests whether the date stored in `date_as_stream` is a nice wedding date. The beginning of the `if` construct could be put into words like: "If the day of the date and the month of the date contain only such digits that can be found in the year or the date, ...".

Here, function `find_first_not_of()` returns the code `string::npos` in situations when there is no such position in strings `day_as_string` or `month_as_string` which does not contain a character belonging to string `year_as_string`. The constant `string::npos` ("no such position") is declared in class `string`.

For example, date 02/06/2006 would be found to be an acceptable wedding date because strings "02" and "06" do not contain any characters that would not be found in string "2006".

`date_as_stream` contains the date in the American format MM/DD/YYYY because that is the default date format in class `Current_date`. Here function `getline()` reads the stream `date_as_stream` until it encounters character `'/'` which separates the numerical parts of the date.

```
D:\book3cpp>weddingdates
```

```
These are easy-to-remember dates for weddings and  
other important events because the days and months  
consist of the digits used in the year:
```

```
02/02/2003  02/03/2003  02/20/2003  02/22/2003  02/23/2003  
03/02/2003  03/03/2003  03/20/2003  03/22/2003  03/23/2003  
03/30/2003  02/02/2004  02/04/2004  02/20/2004  02/22/2004  
02/24/2004  04/02/2004  04/04/2004  04/20/2004  04/22/2004  
04/24/2004  02/02/2005  02/05/2005  02/20/2005  02/22/2005  
02/25/2005  05/02/2005  05/05/2005  05/20/2005  05/22/2005  
05/25/2005  02/02/2006  02/06/2006  02/20/2006  02/22/2006  
02/26/2006  06/02/2006  06/06/2006  06/20/2006  06/22/2006  
06/26/2006  02/02/2007  02/07/2007  02/20/2007  02/22/2007  
02/27/2007  07/02/2007  07/07/2007  07/20/2007  07/22/2007  
07/27/2007  02/02/2008  02/08/2008  02/20/2008  02/22/2008  
02/28/2008  08/02/2008  08/08/2008  08/20/2008  08/22/2008
```

weddingdates.cpp - X. The program prints 60 dates in the MM/DD/YYYY format.

These are sample pages from Kari Laitinen's book
"A Natural Introduction to Computer Programming with C++".
For more information, please visit
<http://www.naturalprogramming.com/cppbook.html>