```
//
// Class source file for A4
//
// Hal Bettle
//
// 5 September 2008
//

#include "CPTN230A4class_bettle.h"

Date::Date()
{
    temp_time = time(NULL);
    time_struct = localtime(&temp_time);
    year = (time_struct->tm_year + 1900);
    month = (time_struct->tm_mon + 1);
    day = (time_struct->tm_mday);
    validate_date(day, month, year);
}

Date::Date(int d)
{
    temp_time = time(NULL);
    time_struct = localtime(&temp_time);
    year = (time_struct->tm_year + 1900);
    month = (time_struct->tm_mon + 1);
    set_day(d);
    validate_date(day, month, year);
}

Date::Date(int d, int m)
{
    temp_time = time(NULL);
    time_struct = localtime(&temp_time);
    year = (time_struct->tm_year + 1900);
    set_month(m);
    set_day(d);
    validate_date(day, month, year);
}

Date::Date(int d, int m, int y)
{
    set_year(y);
    set_month(m);
    set_day(d);
    validate_date(day, month, year);
}

int Date::get_day(void)
{
    return day;
}

int Date::get_month(void)
{
    return month;
}

int Date::get_year(void)
{
    return year;
}

bool Date::operator==(const Date &source)
{
    if ( (year  == source.year)  &&
```

```cpp
          (month == source.month) &&
          (day   == source.day)
        )
    {
        compare_status = true;
    }
    else
    {
        compare_status = false;
    }
    return compare_status;
}

bool Date::operator!=(const Date &source)
{
    if ( (year  != source.year)  ||
          (month != source.month) ||
          (day   != source.day)
        )
    {
        compare_status = true;
    }
    else
    {
        compare_status = false;
    }
    return compare_status;
}

bool Date::operator<(const Date &source)
{
    if (year < source.year)
    {
        compare_status = 1;
    }
    else if ( (year == source.year) && (month < source.month) )
    {
        compare_status = 1;
    }
    else if ( (year == source.year) && (month == source.month) && (day < source.day) )
    {
        compare_status = 1;
    }
    else
    {
        compare_status = 0;
    }
    return compare_status;
}

bool Date::operator<=(const Date &source)
{
    if ( (*this < source) || (*this == source) )
    {
        compare_status = 1;
    }
    else
    {
        compare_status = 0;
    }
    return compare_status;
}

bool Date::operator>(const Date &source)
{
    if (year > source.year)
    {
        compare_status = 1;
    }
    else if ( (year == source.year) && (month > source.month) )
    {
        compare_status = 1;
    }
    else if ( (year == source.year) && (month == source.month) && (day > source.day) )
    {
        compare_status = 1;
    }
    else
    {
        compare_status = 0;
    }
    return compare_status;
}

bool Date::operator>=(const Date &source)
{
    if ( (*this > source) || (*this == source) )
    {
        compare_status = 1;
    }
    else
    {
        compare_status = 0;
    }
    return compare_status;
}

Date &Date::operator++()
{
    if ( (month == 12) && (day == 31) )
    {
        month = 1;
        day = 1;
        year++;
    }
    else if ( ( (num_days == 31) && (day == 31) ) ||
              ( (num_days == 30) && (day == 30) )
            )
    {
        day = 1;
        month++;
    }
    else if ( (  leap_year && (day == 29) ) ||
              ( !leap_year && (day == 28) )
            )
    {
        day = 1;
        month = 3;
    }
    else
    {
        day++;
    }
    return *this;
}

//  Private support member functions

void Date::set_day(int d)
{
    day = d;
    return;
```

```cpp
}

void Date::set_month(int m)
{
    month = m;
    return;
}

void Date::set_year(int y)
{
    year = y;
    return;
}

void Date::validate_date(int da, int mon, int ye)
{
    int error = 0;

    if ( ( ( (ye % 4) == 0) && ( !(ye % 100) == 0) ) ||
         ( (ye % 400) == 0)
       )
    {
        leap_year = true;
    }
    else
    {
        leap_year = false;
    }

    switch (mon)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: if ((da < 1) || (da > 31))
                 {
                     error = 1;
                 }
                 else
                 {
                     num_days = 31;
                 }
                 break;
        case 4:
        case 6:
        case 9:
        case 11: if ((da < 1) || (da > 30))
                 {
                     error = 1;
                 }
                 else
                 {
                     num_days = 30;
                 }
                 break;
        case 2:  if ( ( !leap_year && ( (da < 1) || (da > 28) ) ) ||
                      (  leap_year && ( (da < 1) || (da > 29) ) )
                    )
                 {
                     error = 1;
                 }
                 break;
        default: error = 1;
```

```
                break;
    }

    if (error)
    {
        cout << "Month or day invalid mon = " << mon << ", day = " << da << endl;
        num_days = 31;
        leap_year = true;
        set_year(2000);
        set_month(1);
        set_day(1);
    }

    return;
}
```