

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών
Αντικειμενοστραφής Προγραμματισμός Ι
Εργαστήριο 5

Στο 5^ο εργαστήριο θα ασχοληθούμε με την υπερφόρτωση

Στόχοι εργαστηρίου:

- Συναρτήσεις Friends
- Υπερφόρτωση τελεστών εισόδου, εξόδου
- Ο δείκτης this

Στο εργαστήριο αυτό έχετε να υλοποιήσετε μια κλάση με το όνομα **Date**. Τα δεδομένα της κλάσης είναι:

```
int day;  
int month;  
int year;
```

Για την κλάση αυτή έχετε να υλοποιήσετε τις ακόλουθες συναρτήσεις:

1. **2 constructors** (με και χωρίς ορίσματα) . Ο constructor χωρίς ορίσματα να αρχικοποιεί μια ημερομηνία στις 1/1/1900.
2. **bool leapYear();** Η συνάρτηση ελέγχει αν πρόκειται για δίσεκτο έτος . Ο υπολογισμός που πρέπει να γίνει είναι ο ακόλουθος.
$$\text{if (year \% 400 == 0 || (year \% 100 != 0 \&\& year \% 4 == 0))}$$
3. **bool endOfMonth();** Η συνάρτηση ελέγχει αν μια ημερομηνία είναι η τελευταία μέρα ενός μήνα (π.χ αν είναι ο Φεβρουάριος η τελευταία μέρα στα μη δίσεκτα έτη είναι 28 κ.τ.λ) και ανάλογα επιστρέφει true αν πρόκειται για το τέλος ενός μήνα η false σε αντίθετη περίπτωση. Αρχικά πρέπει να ελέγξετε αν πρόκειται για δίσεκτο έτος και για το μήνα Φεβρουάριο σε αυτή τη περίπτωση η τελευταία μέρα του Φεβρουαρίου είναι η 29^η διαφορετικά η 28^η.
4. **void helpIncrement();** Συνάρτηση η οποία αυξάνει την ημερομηνία κατά μια ημέρα. Βασικοί έλεγχοι που πρέπει να γίνουν, αν πρόκειται για το τέλος του μήνα, αν πρόκειται για το τέλος του χρόνου.
5. **Date &Date::operator++();** Υπερφόρτωση προθεματικού τελεστή ++
6. **Date &Date::operator++(int);** Υπερφόρτωση μεταθεματικού τελεστή ++

Υπερφόρτωση τελεστών εισόδου- εξόδου με φιλικές συναρτήσεις

7. `friend ostream &operator<<(ostream &output, const Date &d);`
8. `friend istream &operator>>(istream &input, Date &d);`

Στη `main` να δηλώσετε 2 ημερομηνίες και να τις διαβάσετε από το χρήστη χρησιμοποιώντας την υπερφόρτωση του `>>` και να τις τυπώσετε χρησιμοποιώντας την υπερφόρτωση του `<<`. Να αυξήσετε τις 2 ημερομηνίες κατά 1 ημέρα και να τυπώσετε το αποτέλεσμα. Να δοκιμάσετε τις οριακές τιμές, αλλαγή μήνα και αλλαγή έτους.

Ακολουθεί παράδειγμα υπερφόρτωσης μοναδιαίου τελεστή.

`//point.h`

```
#include <iostream>
using namespace std;
class Point
{
private:
    int _x, _y;
public:
    // Declare prefix and postfix increment operators.
    Point& operator++();    // Prefix increment operator.
    Point operator++(int);  // Postfix increment operator.

    // Declare prefix and postfix decrement operators.
    Point& operator--();    // Prefix decrement operator.
    Point operator--(int);  // Postfix decrement operator.

    // Define default constructor.
    Point() { _x = _y = 0; }
    void display(){cout<<_x<<" "<<_y<<endl;}
    // Define accessor functions.
    int x() { return _x; }
    int y() { return _y; }
};
```

Point.cpp

```
#include "point.h"

// Define prefix increment operator.
Point& Point::operator++()
{
    _x++;
    _y++;
    return *this;
}

// Define postfix increment operator.
Point Point::operator++(int)
{
    Point temp = *this;
    ++*this;
    return temp;
}

// Define prefix decrement operator.
Point& Point::operator--()
{
    _x--;
    _y--;
    return *this;
}

// Define postfix decrement operator.
Point Point::operator--(int)
{
    Point temp = *this;
    --*this;
    return temp;
}
```

Main.cpp

```
#include <iostream>
#include "point.h"
using namespace std;
```

```
/* run this program using the console pauser or add your own getch, system("pause") or input loop */
```

```
int main(int argc, char** argv) {  
    Point a, b, c, d, e, f;  
  
    cout << "Before using the operator ++()\n";  
    cout << "a = ";  
    a.display();  
    cout << "b = ";  
    b.display();  
  
    ++a;  
    b++;  
  
    cout << "After using the operator ++()\n";  
    cout << "a = ";  
    a.display();  
    cout << "b = ";  
    b.display();  
  
    c = ++a;  
    d = b++;  
  
    cout << "Result prefix (on a) and postfix (on b)\n";  
    cout << "c = ";  
    c.display();  
    cout << "d = ";  
    d.display();  
  
    cout << "Before using the operator --()\n";  
    cout << "e = ";  
    e.display();  
    cout << "f = ";  
    f.display();  
  
    --e;  
    f--;  
  
    cout << "After using the operator --()\n";  
    cout << "e = ";  
    e.display();  
    cout << "f = ";  
    f.display();  
  
}
```

```
    return 0;  
}
```