

Άσκηση 1

Με δεδομένη την κλάση **kafetiera** του παρακάτω κώδικα, να υπερφορτωθεί ο τελεστής κλήσης συνάρτησης () ώστε όταν χρησιμοποιείται σε συνδυασμό με κάποιο αντικείμενο αυτής της κλάσης, να καταχωρίζει τις τιμές των ορισμάτων του στις μεταβλητές – μέλη **kafes**, **gala**, **zaxari** και **nero** αντίστοιχα

```
#include <iostream>
using namespace std;
class kafetiera
{
    int kafes;
    int gala;
    int zaxari;
    int nero;
public:
    kafetiera(int k, int g, int z, int n);
    kafetiera(){};
    kafetiera operator + (kafetiera op2);
    void status();
};
kafetiera::kafetiera(int k, int g, int z, int n)
{
    kafes=k;
    gala=g;
    zaxari=z;
    nero=n;
}
```

```

void kafetiera::status()
{
    cout<<"Καφές:"<<kafes<<endl;
    cout<<"Γάλα:"<<gala<<endl;
    cout<<"Ζάχαρη:"<<zaxari<<endl;
    cout<<"Νερό:"<<nero<<endl;
}

```

Μέθοδος υπερφόρτωσης του τελεστή
+

```

kafetiera kafetiera::operator + (kafetiera op2)

```

```

{
    kafetiera temp;
    temp.kafes=kafes+op2.kafes;
    temp.gala=gala+op2.gala;
    temp.zaxari=zaxari+op2.zaxari;
    temp.nero=nero+op2.nero;
    return temp;
}

```

Η μέθοδος επιστρέφει ως τιμή το προσωρινό αντικείμενο – καφετιέρα temp, το οποίο περιέχει το άθροισμα των υλικών των δυο μελών του τελεστή

```

int main()

```



```

{
    kafetiera kaf1(100,50,200,1000), kaf2(10,20,30,40), kaf3;
    kaf3=kaf1+kaf2;
    kaf3.status();
    return 0;
}

```

Λύση 1

```
kafetiera kafetiera::operator() (int k, int g, int z, int n)
{
    kafes=k;
    gala=g;
    zaxari=z;
    nero=n;
    return *this;
}
```

-  Η μέθοδος υπερφόρτωσης του τελεστή () έχει ως αποτέλεσμα την ανάθεση τιμών στις μεταβλητές-μέλη του αντικειμένου. Για παράδειγμα, η πρόταση: `kaf1(10,20,100,200)` θα έχει ως αποτέλεσμα την καταχώριση των τιμών 10,20,100 και 200 στις μεταβλητές-μέλη `kafes`, `gala`, `zaxari` και `nero` ενός αντικειμένου `kaf1` της κλάσης `kafetiera`.
-  Ο δείκτης `this` δείχνει στο αντικείμενο που καλεί τη μέθοδο υπερφόρτωσης. Η μέθοδος επιστρέφει ως τιμή το ίδιο το αντικείμενο (`*this`).

Άσκηση 2

- Με δεδομένη την κλάση **kafetiera** της προηγούμενης άσκησης, να υπερφορτωθεί ο τελεστής ισότητας (**==**) ώστε, όταν χρησιμοποιείται για να συγκρίνει δυο αντικείμενα αυτής της κλάσης, να επιστρέφει τιμή **true** όταν τα δυο αντικείμενα περιέχουν ακριβώς τις ίδιες ποσότητες υλικών, διαφορετικά να επιστρέφει τιμή **false**

Λύση 2

```
bool kafetiera::operator==(kafetiera op2)
{
    if (kafes==op2.kafes && gala==op2.gala && zaxari==op2.zaxari && nero==op2.nero)
        return true;
    else
        return false;
}
```



Η μέθοδος υπερφόρτωσης του τελεστή `==` επιστρέφει τιμή `bool`. Η μέθοδος ελέγχει αν όλες οι τιμές των μεταβλητών-μελών του αριστερού μέλους του τελεστή ισούνται με τις αντίστοιχες τιμές των μελών του δεξιού μέλους του τελεστή.

Άσκηση 3

- Με δεδομένη την κλάση **kafetiera** της άσκησης 1 να υπερφορτωθεί ο τελεστής πολλαπλασιασμού (*) ώστε να χρησιμοποιείται για να πολλαπλασιάσει ένα αντικείμενο αυτής της κλάσης με έναν ακέραιο αριθμό. Το αποτέλεσμα θα πρέπει να είναι ένα αντικείμενο του τύπου **kafetiera**, το οποίο θα περιέχει όσα υλικά περιέχει και το αρχικό αντικείμενο επί τον ακέραιο αριθμό. Για παράδειγμα, αν το αντικείμενο **kaf1** περιέχει καφέ=10, γάλα=20, ζάχαρη=30 και νερό=100, το αποτέλεσμα της πράξης **2*kaf1** (αλλά και της **kaf1*3**) θα είναι αντικείμενο τύπου **kafetiera** με καφέ=30, γάλα=60, ζάχαρη=90 και νερό=300

Λύση 3

```
kafetiera kafetiera::operator*(int value)
{
    kafetiera temp;
    temp.kafes=kafes*value;
    temp.gala=gala*value;
    temp.zaxari=zaxari*value;
    temp.nero=nero*value;
    return temp;
}
kafetiera operator*(int value, kafetiera op2)
{
    kafetiera temp;
    temp.kafes=op2.kafes*value;
    temp.gala=op2.gala*value;
    temp.zaxari=op2.zaxari*value;
    temp.nero=op2.nero*value;
    return temp;
}
```

- ☞ Η πρώτη μέθοδος υπερφόρτωσης του τελεστή `*` ορίζεται μέσα στην κλάση `kafetiera`. Η έκδοση αυτή καλείται όταν το αριστερό μέλος του τελεστή `*` είναι αντικείμενο της κλάσης `kafetiera` και το δεξιό μέλος είναι ακέραιος αριθμός. Π.χ. `kaf1*10`.
- ☞ Η δεύτερη συνάρτηση υπερφόρτωσης του τελεστή `*` ορίζεται εκτός της κλάσης `kafetiera`. Ο ορισμός της συνάρτησης εκτός της κλάσης είναι απαραίτητος όταν κατά την κλήση του διμελούς τελεστή `*` το αριστερό του μέλος δεν είναι αντικείμενο της κλάσης `kafetiera` αλλά ένας ακέραιος αριθμός. Η έκδοση αυτή καλείται όταν το αριστερό μέλος είναι ακέραιος αριθμός και το δεξιό μέλος είναι αντικείμενο της κλάσης `kafetiera`. Π.χ. `10*kaf1`. Η συνάρτηση αυτή πρέπει να δηλωθεί ως φίλια της κλάσης `kafetiera` για να μπορεί να έχει πρόσβαση στα ιδιωτικά μέλη της κλάσης

Άσκηση 4

- Το παρακάτω πρόγραμμα χρησιμοποιεί την κλάση **ergasia** για τη διαχείριση εργασιών φοιτητών. Κάθε εργασία μπορεί να ανατεθεί μέχρι το πολύ σε 5 φοιτητές. Η εργασία διαθέτει ημερομηνία παράδοσης του τύπου **my_date** που έχει δημιουργηθεί για τη διαχείριση ημερομηνιών. Και για τις δυο κλάσεις έχουν υπερφορτωθεί τελεστές που να κάνουν ευκολότερη τη διαχείριση των αντικειμένων τους. Αφού μελετήσετε τον κώδικα κάντε τις απαραίτητες τροποποιήσεις ώστε ο υπερφορτωμένος τελεστής **[]** να επιστρέφει μια αναφορά στη θέση που είναι καταχωρισμένο το όνομα του φοιτητή και όχι την τιμή του. Δηλαδή, μετά τις αλλαγές που θα κάνετε, μια πρόταση της μορφής **e1[2]="nikos"** θα πρέπει να καταχωρίζει το nikos ως το όνομα του δεύτερου φοιτητή της εργασίας **e1**.

```

#include <iostream>
#include <string>
using namespace std;

class my_date
{
    int hmera;
    int minas;
    int etos;
public:
    void set_hmer(int h, int m, int e);
    void show();
    my_date operator++();
};

void my_date::set_hmer(int h, int m, int e)
{
    hmera=h;
    minas=m;
    etos=e;
}

void my_date::show()
{
    cout<<hmera<<"/"<<minas<<"/"<<etos<<endl;
}

```

Η μέθοδος set_hmer () καταχωρίζει τις τιμές των παραμέτρων στις μεταβλητές hmera, minas, και etos αντίστοιχα

```

my_date my_date::operator++()
{
    hmera++;
    if (hmera>30)
    {
        hmera=1;
        minas++;
        if (minas>12)
        {
            minas=1;
            etos++;
        }
    }
    return *this;
}

```

Μέθοδος υπερφόρτωσης του προθεματικού τελεστή ++ για τη κλάση my_date. Αυξάνει την ημερομηνία κατά μια ημέρα. Υποθέτει ότι όλοι οι μήνες έχουν 30 ημέρες. Η μέθοδος επιστρέφει ως τιμή το ίδιο το αντικείμενο

```

class ergasia
{
    float bathmos;
    int plithos_foititon;
public:
    string titlos;
    my_date paradosi;
    string onomata[5];
    ergasia();
    void add_onom(string onom);
    void set_bathmos(float b);
    void display();
    int get_plithos_foititon();
    void operator=(string t);
    void operator++();
    string operator[](int i);
};

```

```

ergasia::ergasia()
{
    titlos="";
    bathmos=-1;
    plithos_foititon=0;
}

```

Μέθοδος δόμησης της κλάσης ergasia

```

void ergasia::add_onom(string onom)
{
    if (plithos_foititon<5)
    {
        onomata[plithos_foititon]=onom;
        plithos_foititon++;
    }
    else
        cout << "Όλες οι θέσεις είναι συμπληρωμένες"<<endl;
}

void ergasia::set_bathmos(float b)
{
    bathmos=b;
}

void ergasia::display()
{
    int i;
    cout<<endl<<"Τίτλος:"<<titlos<<endl;
    cout<<"======"<<endl;
    cout<<"Ημερομηνία παράδοσης:";
    paradosi.show();
    cout<<"Βαθμός:"<<bathmos<<endl;
    cout <<endl<< "Φοιτητές:"<<plithos_foititon<<endl;
    cout<<"-----"<<endl;
    for (i=0;i<plithos_foititon;i++) cout<<onomata[i]<<endl;
}

```

Η μέθοδος add_onom () προσθέτει το όνομα της παραμέτρου στα ονόματα φοιτητών της εργασίας

Μέθοδος εμφάνισης των στοιχείων και των φοιτητών της εργασίας

```

}
int ergasia::get_plithos_foititon()
{
    return plithos_foititon;
}
void ergasia::operator=(string t)
{
    titlos=t;
}
void ergasia::operator++()
{
    ++paradosi;
}
void operator>>(string onom,ergasia &e)
{
    e.add_onom(onom);
}
string ergasia::operator[](int i)
{
    if (i>=1 && i<=plithos_foititon)
        return onomata[i-1];
    else
        return "";
}

```

Μέθοδος υπερφόρτωσης του τελεστή ανάθεσης = για τη κλάση ergasia. Καταχωρίζει ως τίτλο της εργασίας την τιμή του δεύτερου μέλους

Μέθοδος υπερφόρτωσης του προθεματικού τελεστή ++ για τη κλάση ergasia. Αυξάνει την ημερομηνία παράδοσης κατά μια ημέρα

Συνάρτηση (εκτός κλάσης) υπερφόρτωσης του τελεστή εξαγωγής >>. Προσθέτει το όνομα του αριστερού μέλους στην εργασία του δεξιού μέλους. Το δεξί μέλος μεταβιβάζεται με αναφορά

Μέθοδος υπερφόρτωσης του τελεστή [] για τη κλάση ergasia. Επιστρέφει ως τιμή το όνομα του φοιτητή ανάλογα με την τιμή της παραμέτρου της

```

int main()
{
    ergasia e1;
    e1="Υπερφόρτωση τελεστών";
    e1.paradosi.set_hmer(30,5,2015);
    e1.add_onom("Νυχτόβιος Βασίλειος");
    "Παρλαπίπας Αναστάσιος">>e1;
    e1.display();
    "Τσιρίδα Μελπομένη">>e1;
    "Αργόστροφου Αγγελική">>e1;
    ++e1;
    e1.set_bathmos(9);
    e1.display();
    cout<<"===== "<<endl;
    cout<<e1[2]<<endl;
    return 0;
}

```

Καταχωρίζει ως τίτλο της εργασίας e1 το «Υπερφόρτωση τελεστών»
Χρησιμοποιεί τη μέθοδο υπερφόρτωσης του τελεστή =

Προσθέτει στους φοιτητές της εργασίας e1 τον Παρλαπίπα Αναστάσιο χρησιμοποιώντας τη συνάρτηση υπερφόρτωσης του τελεστή εξαγωγής >>

Με τη χρήση του υπερφορτωμένου τελεστή [] εμφανίζει το όνομα του δεύτερου φοιτητή της εργασίας e1

Με τη χρήση του υπερφορτωμένου τελεστή ++ αυξάνει την ημερομηνία παράδοσης της εργασίας e1 κατά μια ημέρα

Λύση 4

```
class ergasia
{
    .....
    string &operator[] (int i);
};

string &ergasia::operator[] (int i)
{
    if (i>=1 && i<=plithos_foititon)
        return onomata[i-1];
}

int main()
{
    .....
    e1[2]="Νίκος";
    e1.display();
    return 0;
}
```

Η μόνη αλλαγή που πρέπει να γίνει είναι να οριστεί η μέθοδος υπερφόρτωσης του τελεστή [] ως αναφορική με χρήση του προθέματος &. Τώρα η μέθοδος επιστρέφει μια αναφορά στη θέση μνήμης του πίνακα **onomata**.

Η πρόταση αυτή θα καταχωρίσει στη δεύτερη θέση του πίνακα **onomata** του αντικειμένου **e1** το «Νίκος».

Άσκηση 5

- Κάντε τις απαραίτητες προσθήκες στη κλάση `my_date` της προηγούμενης άσκησης ώστε η εφαρμογή του προθεματικού τελεστή `--` σε αντικείμενα της κλάσης να έχει ως αποτέλεσμα τη μείωση της ημερομηνίας του αντικειμένου κατά μία ημέρα. Επίσης, όταν αφαιρούνται δυο αντικείμενα της κλάσης `my_date` με χρήση του τελεστή `-` να προκύπτει ένας ακέραιος αριθμός με τη διαφορά των ημερών μεταξύ των δυο ημερομηνιών. Θεωρήστε ότι όλοι οι μήνες έχουν 30 ημέρες. Να κατασκευαστεί ένα ολοκληρωμένο πρόγραμμα το οποίο θα χρησιμοποιεί την κλάση `my_date` και στο οποίο θα χρησιμοποιούνται οι δυο αυτοί τελεστές.

Λύση 5


```
#include <iostream>
#include <string>
using namespace std;
class my_date
{
    int hmera;
    int minas;
    int etos;
public:
    void set_hmer(int h, int m, int e);
    void show();
    my_date operator++();
    my_date operator--();
    int operator-(my_date op2);
};
void my_date::set_hmer(int h, int m, int e)
{
    hmera=h;
    minas=m;
    etos=e;
}
void my_date::show()
{
    cout<<hmera<<"/"<<minas<<"/"<<etos<<endl;
}

my_date my_date::operator++()
{
    hmera++;
    if (hmera>30)
```

```
{
    hmera=1;
    minas++;
    if (minas>12)
    {
        minas=1;
        etos++;
    }
}
return *this;
}
```

my_date my_date::operator--()

```
{
    hmera--;
    if (hmera==0)
    {
        hmera=30;
        minas--;
        if (minas==0)
        {
            minas=12;
            etos--;
        }
    }
}
return *this;
}
```



Μέθοδος υπερφόρτωσης του προθεματικού τελεστή --. Η εφαρμογή του τελεστή θα έχει ως αποτέλεσμα τη μείωση της ημερομηνίας κατά 1 ημέρα. Αν είναι η 1η του μήνα, η μείωση επηρεάζει και τον μήνα και το έτος. Η μέθοδος επιστρέφει ως τιμή το ίδιο το αντικείμενο.

```

int my_date::operator- (my_date op2)
{
    int hmeres1,hmeres2;
    hmeres1=etos*360+minas*30+hmera;
    hmeres2=op2.etos*360+op2.minas*30+op2.hmera;
    if (hmeres1>hmeres2)
        return hmeres1-hmeres2;
    else
        return hmeres2-hmeres1;
}

```

Μέθοδος υπερφόρτωσης του τελεστή -. Υπολογίζονται τα σύνολα των ημερών των ημερομηνιών του αριστερού και του δεξιού μέλους του τελεστή. Ως τιμή επιστρέφεται η απόλυτη τιμή της διαφοράς τους. Κάθε μήνας υπολογίζεται με 30 ημέρες.

```

int main()
{
    my_date d1,d2,d3;
    d1.set_hmer(1,1,2016);
    d2.set_hmer(3,7,1980);
    d3.set_hmer(8,3,1960);
    d1.show();
    --d1; // μειώνει την ημερομηνία d1 κατά μία ημέρα
    d1.show();
    cout<<d1-d2<<endl; // εμφανίζει τη διαφορά σε ημέρες των ημερομηνιών d1 και d2
    cout<<d3-d2<<endl; // εμφανίζει τη διαφορά σε ημέρες των ημερομηνιών d3 και d2
    return 0;
}

```

1/1/2016
30/12/2015
12777
7315