

1. Για την παρακάτω τάξη SH που αναπαριστά ένα σχήμα (πχ. ορθογώνιο) με διαστάσεις x και y (σε μ.) να γραφούν οι μέθοδοι (συναρτήσεις)

```
class SH {
    int x;
    int y;
public:
    float z;
    //
};
```

-- Τρεις δομητές ώστε να είναι δυνατή η αρχικοποίηση των αντικειμένων τύπου SH (πχ. αντικείμενα ob2, ob1, obj) με εξής τρόπους (πλήρη δομητή, δομητή με μια παράμετρο, δομητή χωρίς παράμετρους

```
SH ob2(4,7), // x = 4 και y = 7
    ob1(9),   // x = 9 και y = 9
    obj;      // x = 0 και y = 0
```

-- Απονομής και ανάκτησης τιμών

```
void setxy(int a,int b); // δίνει τιμές στα μελή x και y
void getxy(int& a,int& b); // ανακτά τις τιμες των x και y
```

-- Μέθοδο υπολογισμού του εμβαδού

2. Για την παραπάνω τάξη SH να γράψατε

-- Μια μέθοδο (συνάρτηση μέλος – member function) που προσθέτει έναν αριθμό k στις διαστάσεις x, και y (αυξάνει τα x, και y κατά k μ.)

```
void pros(int k);
```

-- Μια φίλη μέθοδο (friend function) που πολλαπλασιάζει έναν αριθμό k μόνο την διάσταση x ενός σχήματος ob και επιστρέφει την νέα τιμή του x.

```
int pol(SH& ob,int k);
```

-- Μια μέθοδο μη-μέλος (non-member function) που προσθέτει δυο σχήματα ob1 και ob2 και επιστρέφει το αποτέλεσμα δηλ. αντικείμενο τύπου SH. Το x του ob1 προστίθεται με το x του ob2 και το y του ob1 προστίθεται με το y του ob2.

```
SH addobs(SH &ob1, SH &ob2);
```

3.

-- Ελέγξτε την ορθότητα των παραπάνω μεθόδων (συναρτήσεων) με το παρακάτω main πρόγραμμα.

```
int main()
{
    SH ob2(4,7), ob1(9), obj;

    obj.setxy(3,4);

    int a, b;
    obj.getxy(a,b);
    cout << "obj=" << a << " " << b << endl; // 3 4

    ob1.pros(5); //
```

```

ob1.getxy(a,b);
cout << "ob1=" << a << " " << b << endl; // 14 14

pol(obj,7); //
obj.getxy(a,b);
cout << "obj=" << a << " " << b << endl; // 21 4

SH obX;
obX = addobs(ob2,ob1); //
obX.getxy(a,b);
cout << "obX=" << a << " " << b << endl; // 18 21

system("pause");
return 0;
}

```

-- **Αλλάξτε** τις κεφαλίδες συναρτήσεων

```

void pros(int k)
friend int pol(SH& ob,int k)
SH addobs(SH& obA,SH& obB)

```

με τις αντίστοιχες κεφαλίδες

```

void operator+(int k)
friend int operator*(SH& ob,int k)
SH operator+(SH& obA,SH& obB)

```

**Αλλάξτε** επίσης και τις κλήσεις των συναρτήσεων

```

ob1.pros(5);
pol(obj,8);
obX = addobs(ob2,ob1);

```

με τις αντίστοιχες κλήσεις και «τρέξετε το πρόγραμμα»

```

ob1.operator+(5);
operator*(obj,8);
obX = operator+(ob2,ob1);

```

-- **Αλλάξτε** ακόμη μια φορά τις κλήσεις των συναρτήσεων και «τρέξετε το πρόγραμμα»

```

ob1+5;
int X = obj*8;
obX = ob2 + ob1;

```

#### 4.

-- Σε πολλές γλώσσες προγραμματισμού ο τελεστής  $\wedge$  χρησιμοποιείται ως εξής  $A^k$  και υπολογίζει το  $A^k$ , πχ. αν  $A=2$  και  $k=3$  τότε υπολογίζει το  $2^3$  ( $2*2*2=8$ ). Γράψετε μια συνάρτηση υπερφόρτωσης του τελεστή  $\wedge$  με φίλη συνάρτηση ή με συνάρτηση μέλος ώστε να υπολογίζει το  $x^k$  και το  $y^k$  ενός αντικειμένου και επιστρέφει σε ένα αντικείμενο τύπου SH το αποτέλεσμα δηλ. το επιστρεφόμενο αντικείμενο να έχει το x ίσο με  $x^k$  και y ίσο με  $y^k$ . Το k περνά ως παράμετρο στην συνάρτηση υπερφόρτωσης.

-- Υπερφόρτωση του τελεστή  $\ll$  με φίλη συνάρτηση ώστε να είναι δυνατή η εμφάνιση στοιχείων στην οθόνη αντικειμένων του τύπου SH

```

friend ostream& operator<<(ostream& os,const SH& ob)
{
os << ob.x << " --- " << ob.y << endl;
}

```

```

}

// κληση
SH a;
cout << a;

```

**5.** Η παρακάτω τάξη κληρονομεί από την SH

```

class DerW : public SH {
    float w;
public:
    //
};

```

-- Γράψετε τις μεθόδους setw, getw ανάθεσης και ανάκτησης τιμής w

```

// Τεστ με τον ακόλουθο κώδικα
DerW G;
G.setxy(2,3);
G.setw(7.56);
// Εδώ να εμφανίσετε τα στοιχεία x,y,w του αντικείμενου G.

```

-- Για την προηγούμενη τάξη **DerW** να γράψετε έναν δομητή χωρίς παραμέτρους και έναν ακόμη δομητή ώστε να είναι δυνατή η ανάθεση τιμών στα μέλη x,y,w (Θεωρία: Λίστα αρχικοποιήσεων (initialiser list))

```

DerW(int a, int b, float c) : SH(a,b)
{
    //
}

```

```

// Τεστ με τον ακολουθο κωδικα
DerW H(4, 6, 16.72);
// Εδώ να εμφανισετε τα στοιχεια x,y,w του αντικειμενου H.

```

**6.** Η παρακάτω τάξη κληρονομεί από την SH (Ο δείκτης s δείχνει στην καταληφθείσα μνήμη [δημιουργία δυναμικού στρινγκ])

```

class DS : public SH {
    char *s;    // δυναμικό στρινγκ
    int len;    // μηκος στρινγκ
public:
    ...
};

```

-- Γράψετε δυο μεθόδους **δομητών**

α) ώστε να είναι δυνατή η αρχικοποίηση των αντικειμένων τύπου DS πχ. DS a;  
(s = 0, len = 0, x = 0, y = 0)

β) DS a(4,5,"hahaha"); // οι ακέραιοι θα δωσουν τιμες στα  
// μελη x και y.

-- Γράψετε μια μέθοδο **αποδομητή** που ελευθερώνει την καταληφθείσα μνήμη από ένα αντικείμενο.

```

-- Γράψετε τις μεθόδους απονομής και ανάκτησης τιμής
void set(char *a); // Αντιγράφει το string a στο string s
char * get(); // Επιστρέφει έναν δείκτη στο string s

-- Γράψετε μια μέθοδο void emf(); // Εμφανίζει τις s, x, y

(Βοήθημα Κάνετε χρήση των συναρτήσεων)
int strlen(char *a) // μήκος του string
void strcpy(char *b, char *a) // αντιγραφή του string a στο b

// Έλεγχος με τον ακόλουθο κώδικα
DS o1;
o1.set("Forrest Knight");
o1.emf();
DS o2(4,5," The sting");
o2.emf();
//

```

E07\_E14-15