

1.. Δίδεται η παρακάτω ιεραρχία τάξεων

```
class Bas {
    int ib;
    public:
    float z;
};
class Par1 : virtual public Bas {
    int ip1;
    public:
    float z1;
};
class Par2 : virtual public Bas {
    int ip2;
    public:
    float z2;
};
class Eng : public Par1, public Par2 {
    int ie ;
    public:
    float sum;
};
```

Υλοποιήσετε δομητές με για όλες τις παραπάνω τάξεις (να ΜΗ υλοποιήσετε default δομητές) ώστε να είναι δυνατή η δημιουργία αντικειμένων τύπου **Bas**, **Par1**, **Par2**, και **Eng**.

```
// Τεστ με τον ακόλουθο κώδικα
int ib(3), ip1=4, ip2=5, ie=6;

Bas b(ib);
Par1 op1(ib,ip1);
Par2 op2(ib,ip2);
Eng ob(ib,ip1,ip2,ie);

ob.z = 4.2;
ob.z1 = 3.14;
ob.z2 = 1.6;
ob.sum = ob.z + ob.z1 + ob.z2;
cout << ob.sum << endl;
```

Αφαιρέσετε τις λέξεις **virtual** από τις τάξεις **Par1** και **Par2**. Το προηγούμενο τεστ δίνει λάθος (:)

```
// Τεστ με τον ακόλουθο κώδικα
int ib(3), ip1=4, ip2=5, ie=6;
```

```

Bas b(ib);
Par1 op1(ib,ip1);
Par2 op2(ib,ip2);
Eng ob(ib,ip1,ip2,ie);
ob.Par1::z = 4.2;
ob.Par2::z = 2.73;
ob.z1 = 3.14;
ob.z2 = 1.6;
ob.sum = ob.Par1::z + ob.z1 + ob.z2;
cout << ob.sum << endl;
ob.sum = ob.Par2::z + ob.z1 + ob.z2;
cout << ob.sum << endl;

```

2. Δίδεται η παρακάτω ιεραρχία τάξεων

```

class AC {
    int x;
    public:
    AC() { x = 5; }
    int readx() { return x; }
    virtual float FN() { return(3.14*x) }
};

class DAC1 : public AC {
    public:
    virtual float FN()
    {
        return(readx()/4.0);
    }
};

```

Να ορίσετε μια τάξη **DAC2** που παράγεται από την **DAC1**. Στην **DAC2** ορίσετε μια **FN** η οποία υπερκαλύπτει “overrides” την **FN** που κληρονομεί και επεξεργάζεται μήλη της **DAC2**.

```

// Τεστ με τον ακολουθο κωδικα
AC v;
cout << v.readx() << endl;
cout << v.FN() << endl;

DAC1 w;
cout << w.readx() << endl;
cout << w.FN() << endl;

DAC2 q;
cout << q.FN() << endl;

AC oa, *p;

DAC1 oa1;
DAC2 oa2;

```

```

p = &oa;
cout << p->FN() << endl;

p = &oa1;
cout << p->FN() << endl;

p = &oa2;
cout << p->FN() << endl;
//

```

3. Δίδεται η παρακάτω ιεραρχία τάξεων

```

class Cx {
public:

    Cx(float a) { x = a; }

    void setxy(float a, float b) { x =a; y=b; }
    float readx() { return x; }
    float ready() { return y; }

    virtual float Func() = 0;    // pure virtual

private:
    float x;
    float y;
};

class DCx : public Cx {
public:

    DCx(float a) : Cx(a) {}

    float Func()
    {
        float a = readx();
        float y = a * 33.3; // Metatroph
        return(y);
    }
};

// Test
Cx *p;
DCx ob1(5);

p = &ob1;

cout << ob1.readx() << " Podia se cm " << p->FN() <<endl;

```

Να ορίσετε μια τάξη DCx2 που παράγεται από την Cx, και να ορίσετε μια Func η οποία μετατρέπει Km σε Miles (1Mil=1650m)

4. Δίδεται η παρακάτω τάξη με την οποία θα μπορούμε δημιουργούμε δυναμικούς πίνακες ακεραίων.

```
class MyAr {
    int *p;
    int len;
public:
    // Δομητες
    MyAr() ;
    MyAr(int a) ;
    MyAr(const MyAr& ob); // Δομητής αντιγράφου

    ~MyAr();

    Arr& operator=(const Arr& ox) // Τελεστης απονομής τιμής
    int& operator[](int i) { return p[i]; }

    int* begin();
    int* end();

    int* rbegin();
    int* rend();
}
```

A. Να υλοποιήσετε τους δομητες.

```
MyAr() ; //Τα ιδιωτικά μέλη παίρνουν τιμη μηδεν.
```

```
MyAr(int a) ; // το p δειχνει σε a ακεραιους (δυναμικος πίνακα). Χρηση του
// τελεστη new. Το len κρατα το πληθος των στοιχειων του πίνακα.
```

```
MyAr(const MyAr& ob) ; // Το περιεχόμενο του ob να αντιγραφεί στο *this.
```

B. Να υλοποιήσετε τον αποδομητή. Αποδεσμεύσετε την μνημη που κατέλαβε ο «πίνακας» p.

Γ. Να συμπληρωθεί η συναρτηση υπερφορτωσης του τελεστη= απονομης τιμής. Το περιεχόμενο του ox να αντιγραφεί στο *this.

Δ. Να υλοποιήσετε τις παρακατω συναρτησεις της τάξης Arr:

```
int length() ; // επιστρεφει το πληθος των στοιχειων του πινακα
```

```
int* begin() ; // επιστρεφει έναν δεικτη στο πρωτο στοιχειο του πινακα
```

```
int* end() ;// επιστρεφει έναν δεικτη στην επομενη θεση του τελευταίου στοιχειο του πινακα
```

```
int* rbegin() ;// επιστρεφει έναν δεικτη στο τελευταίο στοιχειο του πινακα
```

```
int* rend() ;// επιστρεφει έναν δεικτη στην προηγουμενη θεση του πρωτου στοιχειού του πινακα
```

Ελέγξτε την ορθότητα της τάξης και των μεθόδων πχ.

```
int main() {
    MyAr a;
    MyAr b(10);
    for(int i=0; i< b.length(); ++i)
        b[i] = i;
    //
    a = b;      // a.operator=(b); // κληση τελεστη=
    MyAr c(b); // κληση δομητη αντιγραφου
    //
}
```

E08_X10-11