

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**  
**Κ10: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

**Εξετάσεις 20 Σεπτεμβρίου 2016**

1. (α') Γράψτε μια παράγραφο περίπου 30 λέξεων που να περιέχει τις λέξεις “αντικείμενο”, “κλάση”, “κληρονομικότητα” και “αρχικοποίηση” στο νοηματικό πλαίσιο του αντικειμενοστραφούς προγραμματισμού.

- (β') Δίδεται το παρακάτω πρόγραμμα C++:

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

class A{ int data;
public:
A(int i=10):data(i) { cout << "I am constructing an A with " << data << endl; }
~A(){ cout << "I am deleting an A with " << data << endl; }

// virtual void bla() {cout << "This is A::bla" << endl; data++; }
// void bla() { cout << "This is A::bla" << endl; data++; }
// virtual void bloo(){ cout << "This is A:bloo()" << endl; bla(); }
// void bloo(){ cout << "This is A::bloo()" << endl; bla(); }
};

class B : public A { int data;
public:
B(int i=100):data(i){ cout << "I am constructing a B with " << data << endl; }
~B(){ cout << "I am deleting a B with " << data << endl; }

void bla() { cout << "This is B::bla" << endl; data++; }
void bloo(){ cout << "This is B:bloo()" << endl; bla(); } };

int main(){
A a; B b; A* pa;
pa = &b;
pa->bla(); pa->bloo(); }
```

Αποσχοιάζοντας δύο από τις παραπάνω σχολιασμένες γραμμές τη φορά ώστε το πρόγραμμα να μεταγλωττίζεται, δώστε το αποτέλεσμα της εκτέλεσης, αιτιολογώντας την απάντησή σας. Αυτό να γίνει για όλους τους δυνατούς συνδυασμούς αποσχολιασμού.

2. Αιτιολογώντας την απάντησή σας, δώστε το αποτέλεσμα της εκτέλεσης του παρακάτω προγράμματος C++:

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

class A{ int data;
public:
    A(int di=10): data(di)
        { cout << "I just created an A with " << data << endl; }
    A(const A& a): data(a.data)
        { cout << "I just created an A by Copying" << endl; }
    ~A(){ cout << "I shall delete an A with " << data << endl; }

    void inc() { data++; } };

class B{ A data1; A* data2;
public:
    B(const A& a): data1(a), data2(NULL)
        { cout << "I just created a B " << endl; }
    B(const B& b): data1(b.data1), data2(b.data2)
        { cout << "I just created a B by Copying" << endl; }
    ~B(){ cout << "I shall delete a B " << endl; }

    void inc() { data1.inc(); if (data2 != NULL) data2->inc(); }
    void set(A* pa) { data2 = pa; } };

class C{ B data1; B& data2;
public:
    C(B& b): data1(b), data2(b)
        { cout << "I just created a C " << endl; }
    C(const C& c): data1(c.data1), data2(c.data2)
        { cout << "I just created a C by Copying" << endl; }
    ~C(){ cout << "I shall delete a C " << endl; }

    void inc() { data1.inc(); data2.inc(); } };

class D{ C& data1; C& data2;
public:
    D(C& c1, C& c2): data1(c1), data2(c2)
        { cout << "I just created a D " << endl; }
    ~D(){ cout << "I shall delete a D " << endl; }

    void inc() { data1.inc(); data2.inc(); } };

int main(){
    A a; B b1(a); B b2(b1); C c1(b1); C c2(c1); D d(c1,c2);
    a.inc(); b1.inc(); b2.inc(); c1.inc(); c2.inc(); d.inc();
    b1.set(&a);
    a.inc(); b1.inc(); b2.inc(); c1.inc(); c2.inc(); d.inc(); }
```

3. Υλοποιήστε σε C++ την προσομοίωση της υπηρεσίας διαλογής ενός κέντρου διανομής Ταχυδρομείου από το οποίο διανέμονται γράμματα και δέματα.

Κάθε δέμα και κάθε γράμμα έχει μια διεύθυνση παραλήπτη. Αυτή αποτελείται από το δρόμο και τον αριθμό. Τα γράμματα ενδέχεται να είναι συστημένα ή επείγοντα. Τα δέματα ενδεχόμενα χαρακτηρίζονται ως ογκώδη οπότε αντί για το ίδιο το δέμα, διανέμεται ένα απλό γράμμα που έχει τη διεύθυνση του δέματος.

Κάθε τι που διανέμεται χαρακτηρίζεται από *μονάδες διανομής*. Κάθε απλό γράμμα έχει μία μονάδα διανομής. Ένα συστημένο γράμμα ή ένα επείγον γράμμα έχει δύο μονάδες διανομής. Ένα μη ογκώδες δέμα έχει τρεις μονάδες διανομής.

Τα γράμματα και τα δέματα που πρέπει να διανεμηθούν, καθώς αφικνούνται στο ταχυδρομείο, προστίθενται σε μια λίστα διανομής. Τα επείγοντα γράμματα, όμως, φυλάσσονται σε άλλη λίστα διανομής.

Το κέντρο αρχικά δεν έχει τίποτα να διανείμει. Στο κέντρο προστίθεται ένα γράμμα για διανομή (*arrive*), τοποθετώντας το στην κατάλληλη λίστα για διανομή. Στο κέντρο προστίθεται ένα δέμα για διανομή (*arrive*), προωθώντας το κατάλληλα για διανομή. Το κέντρο χαρακτηρίζεται από το πλήθος των μονάδων διανομής των αντικειμένων που οφείλει να διανέμει καθώς και το πλήθος των ογκωδών δεμάτων που φυλάσσει για παράδοση.

Ένα γράμμα αρχικοποιείται, αναθέτοντας τη διεύθυνση παραλήπτη.

Ένα δέμα αρχικοποιείται, αναθέτοντας τη διεύθυνση παραλήπτη.

Υλοποιήστε τα παραπάνω μέσω καταλλήλων κλάσεων, ορίζοντας μέλη-δεδομένα που χρειάζονται και συναρτήσεις-μέλη που υλοποιούν την παραπάνω συμπεριφορά.

Όπου θεωρείτε απαραίτητο, κάνετε παραδοχές στις προδιαγραφές, τεκμηριώνοντάς τις.

*Σημείωση:* Θεωρήστε δεδομένο έναν τύπο λίστας με τις λειτουργίες που χρειάζεστε.