

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

Εξετάσεις 10 Σεπτεμβρίου 2019

1. (α') Τόσο στη C++ όσο και στη Java διακρίνουμε τα μέλη κλάσεων σε στατικά (static) και μη στατικά. Σε τι αναφέρεται αυτή η διάκριση; Είναι το ίδιο κρίσιμα τα στατικά μέλη και για τις δύο γλώσσες; Γιατί;
- (β') Δίδονται τα παρακάτω προγράμματα C++ και Java:

Πρόγραμμα C++:

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

class A{
    int data;
public:
    A(int i=10):data(i) {
        cout << "An A was just constructed with: " << data << endl;}
    void inc() {data++;}
    void print() {cout << data << endl;} };

class B : public A{
    int data;
public:
    B(int i=100):A(100),data(i) {
        cout << "A B was just constructed with: " << data << endl;}
    void inc() {data++;}
    void print() {A:: print(); cout << data << endl;} };

int main(){ B b; b.inc(); b.print();
            A* pab = new B(); pab->inc(); pab->print(); }
```

Πρόγραμμα Java:

```
public class Main{
    public static void main(String[] args){
        B.inc(); A.print_count(); B.print_count();
        B b = new B(); b.inc(); b.print_count();
        A a = new B(); a.inc(); a.print_count();}}

class A{
    private static int count;
    static {System.out.println("Initializing A ");}
    public static void inc() {count++;}
    public static void print_count() {System.out.println(count);} }

class B extends A{
    private static int count;
    static {System.out.println("Initializing B ");}
    public static void inc() {count++;}
    public static void print_count() {System.out.println(count);} }
```

Δώστε τα αποτελέσματα της εκτέλεσης των παραπάνω προγραμμάτων, αιτιολογώντας τα.

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 d=10) : data(d) {cout << "An A just constructed" << endl;}
    A(const A& a) : data(a.data) {cout << "An A just constructed by CP" << endl;}
    ~A(){cout << "An A to be destructed with: " << data << endl;}
    void inc() { data++;}
    int get() {return data;} };

class B{ A data1; A& data2; A* data3;
public:
    B(const A& d1, A& d2, const A& d3) : data1(d1), data2(d2){
        data3 = new A(d3); cout << "A B just constructed" << endl;}
    B(const B& b) : data1(b.data1), data2(b.data2){
        data3 = new A(*(b.data3)); cout << "A B just constructed by CP " << endl;}
    ~B(){cout << "A B to be destructed with: " << data1.get()
        << " and " << data2.get() << " and " << data3->get() << endl;
        delete data3; }
    B& operator=(const B& b)
        { cout<< "I am performing a B assignment " << endl;
        if (this != &b){
            delete data3;
            data1 = b.data1; data2 = b.data2; data3 = new A(*(b.data3));}
        return *this;}
    void inc() {data1.inc(); data2.inc(); data3->inc();} };

void fun(B b1, B& b2, B* pb3){
    b1.inc(); b2.inc(); pb3->inc();
    b2 = *pb3; b1 = b2;
    b1.inc(); b2.inc(); pb3->inc(); }

int main(){ A a1; A a2(100);
    B b1(a1, a2, a1); B b2(a2, a1, a2);
    fun(b1, b2, &b1); }
```

3. Ας υποθέσουμε ότι για την εκτίμηση της δυσκολίας (φόρτου) ενός ακαδημαϊκού εξαμήνου σε ένα πανεπιστημιακό τμήμα ισχύουν τα παρακάτω.

Το εξάμηνο αποτελείται από 5 (πέντε) μαθήματα. Το κάθε μάθημα περιλαμβάνει θεωρία (διδασκαλία με τη μορφή παραδόσεων και φροντιστηρίων) αλλά μπορεί να περιλαμβάνει και εργαστήριο ή/και ένα πρακτικό μέρος. Η θεωρία χαρακτηρίζεται από τον αριθμό των ωρών ανά εβδομάδα που διαρκεί η διδασκαλία. Το ίδιο και το εργαστήριο. Το εργαστήριο όμως μπορεί να έχει υποχρεωτική παρακολούθηση ή όχι. Το πρακτικό μέρος αποτελείται από μια σειρά από παραδοτέες ασκήσεις. Η κάθε άσκηση αποτελείται από ένα κείμενο που αντιστοιχεί στην εκφώνησή της.

Ο υπολογισμός της δυσκολίας (`evaluate_load`) του κάθε εξαμήνου προκύπτει από την συνολική δυσκολία των μαθημάτων του.

Ο υπολογισμός της δυσκολίας (`evaluate_load`) ενός μαθήματος προκύπτει από το άθροισμα της δυσκολίας της θεωρίας και της δυσκολίας του εργαστηρίου του πολλαπλασιασμένο με ένα συντελεστή (σταθερά του μαθήματος). Επιπλέον, προστίθεται και η δυσκολία του πρακτικού μέρους.

Η δυσκολία της θεωρίας (`evaluate_load`) προκύπτει από το πλήθος των ωρών της. Η δυσκολία του εργαστηρίου (`evaluate_load`) προκύπτει από το πλήθος των ωρών του. Αν το εργαστήριο δεν είναι υποχρεωτικό, μειώνεται στο μισό.

Ο υπολογισμός της δυσκολίας (`evaluate_load`) του πρακτικού μέρους προκύπτει από το άθροισμα της δυσκολίας των επί μέρους ασκήσεων. Η δυσκολία μιας άσκησης είναι σταθερά της άσκησης.

Το εξάμηνο αρχικοποιείται με την πληροφορία των μαθημάτων που περιλαμβάνει. Ένα μάθημα αρχικοποιείται με τις πληροφορίες για τη θεωρία του, για το εργαστήριο και για το πρακτικό μέρος. Η θεωρία αρχικοποιείται με το πλήθος των ωρών διδασκαλίας της. Το εργαστήριο αρχικοποιείται με το πλήθος των ωρών διδασκαλίας του και την ένδειξη αν είναι υποχρεωτικό ή όχι. Το πρακτικό μέρος αρχικοποιείται με τις ασκήσεις που το αποτελούν. Μια τέτοια άσκηση αρχικοποιείται με το κείμενο της εκφώνησής της. Κατά την αρχικοποίηση παρέχονται οι πληροφορίες και για τις σταθερές, όπου χρειάζεται.

Τόσο για το εξάμηνο αλλά και για τα μαθήματα, τη θεωρία, το εργαστήριο και το πρακτικό μέρος να παρέχονται συναρτήσεις εκτύπωσης (`print`) που να εκτυπώνουν τις πληροφορίες τους.

Να υλοποιήσετε την παραπάνω πληροφορία μέσω κλάσεων C++.