

## Αντικειμενοστραφής Προγραμματισμός 2016-2017

### Εργασία 3 (27/11/2015)

1. Ορίστε μια κλάση **Car** που αναπαριστά μια εγγραφή "αυτοκινήτου". Η κλάση αυτή θα πρέπει να έχει τουλάχιστον πεδία-μεταβλητές (variable members) τη **μάρκα** (brand) και τον **αριθμό κυκλοφορίας** (plate). Υλοποιήστε **inline** έναν **constructor** για αρχικοποίηση, ώστε πχ να δουλεύει η δήλωση:

Car c;

Η κλήση του παραπάνω αρχικοποιεί τη **μάρκα** στην τιμή **Tesla** και τον **αριθμό κυκλοφορίας** στην τιμή **AAA 0000**.

2. Ορίστε μια *public* συνάρτηση **Car::print()** τύπου **void**, η οποία κατά την κλήση του **c.print()** τυπώνει το εξής μήνυμα (όπου **c**, το αντικείμενο που ορίσαμε προηγουμένως):

**Brand: Tesla**

**Plate: AAA 0000**

**ΠΡΟΣΟΧΗ:** Το παραπάνω μήνυμα εμφανίζεται σε δύο γραμμές, πριν το χαρακτήρα : **δεν** υπάρχει κενό, ενώ μετά το χαρακτήρα : υπάρχει **ένα** κενό.

3. Ορίστε μια *public* συνάρτηση **Car::set(char \*newBrand, char \*newPlate)** τύπου **void**, η οποία θα αλλάζει τις τιμές της **μάρκας** και του **αριθμού κυκλοφορίας** της εγγραφής. Θα πρέπει να γίνεται έλεγχος αν ο **αριθμός κυκλοφορίας** που δόθηκε είναι έγκυρος. Για να είναι έγκυρος θα πρέπει να αποτελείται από 3 (τρία) γράμματα, 1 (ένα) κενό και 4 (τέσσερα) νούμερα. Τα γράμματα δεν έχει σημασία αν θα εισαχθούν ως κεφαλαία ή πεζά. Αν εισαχθούν ως πεζά η συνάρτηση θα πρέπει να τα μετατρέψει αυτόματα σε κεφαλαία.
4. Ορίστε μια *φίλια*, στην κλάση Car, συνάρτηση **trade(Car c1, Car c2)** τύπου **Car** η οποία επιστρέφει ένα αντικείμενο της κλάσης **Car**. Η συνάρτηση **trade** αφού ανταλλάξει τις πινακίδες των αυτοκινήτων, θα επιστρέφει το αντικείμενο **c1**.

5. Θα πρέπει να δουλεύει η δήλωση:

Car a, b;

Car c = trade(a,b);

**ΠΡΟΣΟΧΗ:** Να υλοποιήσετε με προσοχή τον copy constructor

### Διευκρινίσεις/Παρατηρήσεις

- Όλα τα παραπάνω μηνύματα τυπώνονται στο ρεύμα εξόδου *cout* και τερματίζονται με **'\n'**. Όλα τα κενά των *strings* αποτελούνται από **ένα** space
- Το παραδοτέο να ονομαστεί XXXX.cpp και στην πρώτη γραμμή του να περιέχει το `//XXXX όπου XXXX ο Α.Μ. σας.`
- Το XXXX.cpp να **ΜΗΝ** περιλαμβάνει συνάρτηση `main()`. (Κατά τη διάρκεια της υλοποίησης μπορείτε να γράψετε την δικιά σας `main()` σε ένα άλλο αρχείο, πχ `main.cpp`, ώστε να τεστάρετε τον κώδικά σας. Το `main.cpp` θα ενσωματώνει το XXXX.cpp με: `#include "XXXX.cpp"` )
- Οι τρεις παραπάνω διευκρινίσεις είναι πολύ σημαντικές γιατί οι εργασίες θα βαθμολογηθούν αυτόματα.
- Μη παραδώσετε εργασία που δεν κάνει `compile` σε τουλάχιστον έναν από τους `compilers`: `Dev C++` (σε Windows συστήματα), `g++` (σε linux/MacOS συστήματα). Αν έχετε την δυνατότητα, προτιμήστε τον `g++`. Εργασίες που δεν μεταγλωττίζονται βαθμολογούνται αυτόματα με μηδέν.
- Η εργασία είναι ατομική. Σε περίπτωση αντιγραφής, όλες οι ίδιες/παρόμοιες εργασίες θα μηδενιστούν και οι παραβάτες θα θεωρηθούν μετεξεταστέοι στο Εργαστήριο, δλδ δεν θα μπορέσουν να πάρουν μέρος στην γραπτή εξέταση για αυτή την ακαδημαϊκή χρονιά.

**Καλή επιτυχία**