

ΑΣΚΗΣΗ 1

Χρησιμοποιήστε το παράδειγμα `book.cpp`

1. Να προσθεθεί ένα νέο πεδίο(data member) `copies` που θα αντιστοιχεί στον αριθμό των αντιτύπων του βιβλίου που υπάρχουν
2. Να δημιουργηθούν `getters` και `setters` για το πεδίο `copies`
3. Να συμπληρωθεί η συνάρτηση `read`. Η συνάρτηση `read` θα επιτρέπει στον χρήστη να εισάγει στοιχεία ενός βιβλίου από το πληκτρολόγιο
4. Να συμπληρωθεί η συνάρτηση `discount`, η οποία για βιβλία με αντίτυπα λιγότερα από 10 θα πραγματοποιεί έκπτωση 30% επί της τιμής, για βιβλία με αντίτυπα λιγότερα από 50 έκπτωση 15%, για βιβλία με αντίτυπα λιγότερα από 100 έκπτωση 10% και για βιβλία με παραπάνω αντίτυπα έκπτωση 7%.

Στην κύρια συνάρτηση ο χρήστης να εισάγει 5 βιβλία και στην συνέχεια για όλα τα αποθηκευμένα βιβλία να εφαρμοστεί η έκπτωση και να εμφανιστούν οι πληροφορίες για τα βιβλία πριν και μετά την έκπτωση

ΑΣΚΗΣΗ 2

1. Κατασκευάστε μια κλάση με όνομα `cargo` (φορτίο) με μέλη δεδομένων τα ακόλουθα:
 - `id` (αναγνωριστικός αριθμός φορτίου) τύπου `int`.
 - `destination` (προορισμός) τύπου `string`.
 - `weight` (βάρος) τύπου `double`.
 - `fragile` (εύθραυστο) τύπου `bool`.
2. Δημιουργήστε προκαθορισμένο κατασκευαστή που να δίνει τις εξής τιμές: 0, "Undefined", 0, false.
3. Δημιουργήστε έναν κατασκευαστή που να θέτει όλα τα μέλη δεδομένων σε τιμές που θα δέχεται ως παραμέτρους.
4. Δημιουργήστε `getters` και `setters` μόνο το πεδίο `weight`.
5. Δημιουργήστε μια συνάρτηση `show()` που να εμφανίζει τις τιμές όλων των πεδίων του αντικειμένου.
6. Δημιουργήστε μια συνάρτηση μέλος `make_safe()` που εφόσον ένα φορτίο δεν είναι `fragile` να το κάνει και να αυξάνει το βάρος του κατά 5 κιλά (για υλικά συσκευασίας).

Στη `main`:

- ❖ Εισάγετε σε έναν πίνακα 5 θέσεων αντικείμενα cargo ως εξής:
 - 1, Patras, 1000, false
 - 2, Athens, 200, true
 - 3, Kalamata, 2000, false
 - 4, Patras, 1500, true
 - 5, Athens, 1100, false
- ❖ Με μια επανάληψη πραγματοποιήστε κλήση της `make_safe` για όλα τα αντικείμενα.
- ❖ Εμφανίστε όλα τα αντικείμενα

ΑΣΚΗΣΗ 3

Να κατασκευαστεί κλάση `employee` που θα περιέχει τα ακόλουθα μέλη δεδομένα

- ID: Τυχαιός αριθμός
- Name: όνομα υπαλλήλου
- Lastname: επώνυμο υπαλλήλου
- Salary: μισθός υπαλλήλου
- child_number

1. Να δημιουργηθεί κατασκευαστής κατασκευαστής που θα αρχικοποιεί τα μέλη της κλάσης με βάση τις παραμέτρους που θα δέχεται
2. Να υλοποιηθούν `setters` και `getters` για όλα τα μέλη δεδομένα της κλάσης
3. Να υλοποιηθεί συνάρτηση `increase_salary(percent)` που θα αυξάνει τον μισθό ενός υπαλλήλου με βάση την παράμετρο ποσοστό.
4. Να υλοποιηθεί συνάρτηση `annual_salary` που θα υπολογίζει τον ετήσιο μισθό ενός υπαλλήλου.

Ο μισθός υπολογίζεται ως εξής:

- Τους Μονούς μήνες(1,3,5,...) υπάρχει παρακράτηση φόρου 5%
- Τους ζυγούς μήνες(2,4,6) υπάρχει παρακράτηση φόρου 6%
- Αν ένας υπάλληλος έχει 1 παιδί η ετήσια επιστροφή φόρου είναι 1%
- Αν ένας υπάλληλος έχει 2 παιδί η ετήσια επιστροφή φόρου είναι 2%

Αντικειμενοστραφής Προγραμματισμός

- Αν ένας υπάλληλος έχει 3 παιδιά η ετήσια επιστροφή φόρου είναι 4%

5. Να κατασκευαστεί συνάρτηση display που θα εμφανίζει τα στοιχεία ενός υπαλλήλου καθώς και τον συνολικό μισθό

A. Στην κύρια συνάρτηση, Να δημιουργηθεί δυναμικός πίνακας με 10 υπαλλήλους

B. Όσοι υπάλληλοι έχουν ετήσιο μισθό κάτω από 10000 να υπάρχει αύξηση μισθού κατά 10%

Γ. Να εμφανιστούν τα στοιχεία όλων των υπαλλήλων.

ΠΡΟΕΡΕΤΙΚΑ: Να κατασκευαστεί συνάρτηση η οποία θα αποθηκεύει τα δεδομένα των υπαλλήλων σε ένα αρχείο.