Αντικειμενοστραφής Προγραμματισμός 2016-2017

Εργασία 2 (16/11/2016)

1. Ορίστε μια κλάση **Account** που αναπαριστά έναν λογαριασμό τράπεζας. Κάθε αντικείμενο χρειάζεται να έχει όνομα κατόχου, αριθμό λογαριασμού και υπόλοιπο λογαριασμού σε ευρώ. Υλοποιήστε inline έναν ή περισσότερους constructors (όσους χρειάζονται) για αρχικοποίηση των λογαριασμών, ώστε πχ να δουλεύει η δήλωση:

```
Account a(“Giorgos”,2, 5), b(“Giorgos”);
```

Ο λογαριασμός b να αρχικοποιηθεί στην προεπιλογή (“Giorgos”,0,0).

Ορίστε μια *public* συνάρτηση **Account::print()** τύπου **void**, η οποία τυπώνει το εξής μήνυμα : **X account Y has Z euro**, όπου **X**, **Y** και **Z** το όνομα του κατόχου, ο αριθμός λογαριασμού και το υπόλοιπο αντίστοιχα. Έστω το *Account a(“Giorgos”,2,5)*. Η κλήση του *a.print()* επιστρέφει το: **Giorgos account 2 has 5 euro** (Χωρίς αλλαγή γραμμής).

2. Ορίστε μια *public* συνάρτηση **Account::get_id()**, τύπου **int** η οποία επιστρέφει τον αριθμό λογαριασμού. Έστω *Account a(“Giorgos”,2,5)*. Η κλήση του *a.get_id()* επιστρέφει την τιμή *2*.

3. Ορίστε μια *public* συνάρτηση **Account::get_balance()**, τύπου **double** η οποία επιστρέφει το υπόλοιπο του λογαριασμού. Έστω *Account a(“Giorgos”,2,5).* Η κλήση του *a.get_balance()* επιστρέφει την τιμή *5*.

4. Ορίστε μια *public* συνάρτηση **Account::withdrawal(double p)** τύπου **void**, η οποία θα κάνει ανάληψη από το λογαριασμό το ποσό **p**, αφαιρεί δηλαδή από το υπόλοιπο το ποσό **p**. Θα πρέπει να ελέγχει αν το υπόλοιπο επαρκεί και επιπλέον η ανάληψη δε θα πρέπει να υπερβαίνει τα 420 ευρώ.

5. Ορίστε μια *public* συνάρτηση **Account::transfer(Account,double)**, τύπου **void**, η οποία μεταφέρει το ποσό που δηλώνετε με τον double, από τον έναν λογαριασμό στον άλλο. (Υποθέτουμε ότι το υπόλοιπο επαρκεί)Π.χ.

   Account a("Giorgos",2,5), b("Avi",1,12);

   b.transfer(a,3);

   a.print();

   cout << “\n”;

   b.print();

Giorgos account 2 has 8 euro

Avi account 1 has 9 euro

!!! Η κλάση Account θεωρείται blackbox. Μπορείτε δηλαδή να δημιουργήσετε επιπλέον μεταβλητές, συναρτήσεις ή ότι άλλο νομίζετε. Χρησιμοποιώντας όμως μόνο τις βιβλιοθήκες που αναφέρθηκαν στο μάθημα.

6. Ορίστε τον destructor της κλάσης **Account**, ώστε:
   - Αν οι αναλήψεις και οι μεταφορές συνολικά υπερβαίνουν τα 420 ευρώ να τυπώνει:
     X pockets are full!
   - Αν δεν έχει γίνει καμία ανάληψη και μεταφορά να τυπώνει:
     X pockets are empty!
   - Σε κάθε άλλη περίπτωση να τυπώνει:
     Bye bye!

Όπου **X** το όνομα του κατόχου.
!!! Για το μήνυμα που θα τυπώνει ο destructor θα λαμβάνει υπόψιν μόνο ποσά που αφαιρέθηκαν από το υπόλοιπο και όχι ποσά προστέθηκαν.

ΠΡΟΣΟΧΗ! Το όνομα του κατόχου να μπορεί να περιέχει τουλάχιστον 20 χαρακτήρες. Ο αριθμός λογαριασμού να είναι τύπου **int** και του υπολοίπου τύπου **double**.

Διευκρινίσεις/Παρατηρήσεις

- Όλα τα παραπάνω μηνύματα τυπώνονται στο ρεύμα εξόδου *cout* . 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 κώδικα 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