

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΞΕΤΑΣΗ ΜΑΘΗΜΑΤΟΣ «Εργαστήριο Προγραμματισμού»
Διδάσκων: Δρ. Σιούτας Σπύρος

ΘΕΜΑΤΑ

- ~~(α)~~ Ποιά τα πλεονεκτήματα και μειονεκτήματα των *inline* συναρτήσεων; ~~(β)~~ Τι είναι τα πρότυπα συναρτήσεων (**Template Functions**); (γ) Πώς θα δηλώσουμε έναν πίνακα $f[i]$ τριών θέσεων, όπου κάθε θέση i , $0 \leq i \leq 2$, είναι δείκτης σε μία συνάρτηση *void function_k(int, char)*, $1 \leq k \leq 3$ και $k=i+1$; ~~(δ)~~ Δώστε τον ορισμό των συναρτήσεων *friend* μέσα σε μία κλάση; Δώστε ένα παράδειγμα. ~~(ε)~~ Τι είναι ο δείκτης *This*; Δώστε ένα παράδειγμα. (2 Μονάδες)
- Ορίστε και Υλοποιήστε την κλάση *Employee* για τα δεδομένα ενός εργαζόμενου (*firstname, lastname*) μιας επιχείρησης. Ο *constructor* δεσμεύει τον κατάλληλο χώρο για τα *private* δεδομένα (*firstname* και *lastname* αντίστοιχα) και καταχωρεί σε αυτά τις κατάλληλες τιμές. Ο *destructor* αποδεσμεύει τον προηγούμενο χώρο. Ορίστε συναρτήσεις *get()* οι οποίες θα επιστρέφουν τις καταχωρηθείσες τιμές των *private* δεδομένων. Διαχωρίστε τη διασύνδεση από την υλοποίηση. (2.5 Μονάδες)
- Ορίστε και υλοποιήστε κλάση για τη δομή της σωρού (**LIFO**) ώστε να μπορούμε να διαχειριζόμαστε τις ενέργειες *pop()* και *push()* ακέραιων αριθμών. (2.5 Μονάδες)
- Ορίστε και υλοποιήστε μία κλάση *point* (δηλαδή, *constructor, set ()* και *get()* για τα *private* δεδομένα x και y αντίστοιχα). Εν συνεχεία την κλάση *circle* (δηλαδή, *constructor, set ()* και *get()* για τα *private* δεδομένα *center(x,y)* και *radius* αντίστοιχα) η οποία να παράγεται από την κλάση βάσης *point* σύμφωνα με την ιδιότητα της κληρονομικότητας. Επίσης ορίστε συναρτήσεις *print()* και στις δύο παραπάνω κλάσεις. Η πρώτη θα τυπώνει μήνυμα της μορφής $[x,y]$ και η δεύτερη $[\text{center}(x,y), \text{Radius}]$. Και για τις δύο παραπάνω κλάσεις διαχωρίστε τη διασύνδεση από την υλοποίηση. Στη συνέχεια θεωρήστε το παρακάτω κομμάτι κώδικα :

```
int main( )
{
    Point point (30,60);
    Point *pointptr=0;
    Circle circle (100,80, 2.5);
    Circle *circleptr=0;
    point.print();
    circle.print();
    pointptr=& point;
    pointptr-->print();
    circleptr=& circle;
    circleptr-->print();
    pointptr =& circle;
    pointptr-->print();
    return 0;
}
```

Τί θα τυπώσουν οι παραπάνω *print()* εντολές; Αν δηλώσω τις συναρτήσεις *print()* των δύο παραπάνω κλάσεων ως *virtual* τί θα τυπώσει τότε η τελευταία εντολή; Πώς ονομάζεται αυτό το φαινόμενο στη C++; (3 Μονάδες)