

## Αντικειμενοστρεφής Προγραμματισμός I - C++

### 3ο Φυλλάδιο εργαστηρίου

#### Τετάρτη 16/03/2022

**Σκοπός:** Προγραμματισμός C++ με αντικειμενοστρεφή σχεδίαση. Δημιουργία κλάσεων και αντικειμένων κλάσεων, δημόσια (public) και ιδιωτικά (private) μέλη κλάσεων, συναρτήσεις δόμησης (constructor) και αποδόμησης (destructor) κλάσεων.

#### Άσκηση 1η

Παρακάτω ακολουθεί ένα πρόγραμμα σε C++ το οποίο υλοποιεί μια κλάση υπαλλήλου (Employee) με δεδομένα μέλη το ονοματεπώνυμό του/της και τον μισθό καθώς και συνάρτηση δομητή (constructor) και 2 συναρτήσεις μέλη. Να εντοπίσετε λάθη και να τα διορθώσετε.

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

class Employee {
    string fullname;
    float salary;
public:
    Employee(string p_fullname, float p_salary):fullname(p_fullname), salary(p_salary) {
        cout << "Employee " << fullname <<
            " just created having a salary of " << salary << endl;
    }
    void set_salary(float new_sal) {
        salary = new_sal;
    }
    void print_details();
};

void print_details() {
    cout << fullname << " has now a salary of " << salary << endl;
}

int main() {
    Employee e;
    e.salary = 1500;
    cout << e.print_details();
    return 0;
}
```

#### Απάντηση

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

class Employee {
    string fullname;
    float salary;
public:
    Employee(string p_fullname, float p_salary):fullname(p_fullname),
    salary(p_salary) {
        cout << "Employee " << fullname <<
            " just created having a salary of " << salary << endl;
    }

    void set_salary(float new_sal) {
        salary = new_sal;
    }
}
```

```

        void print_details();
};

// Πρέπει να υπάρχει το όνομα της κλάσης και ο τελεστής εμβέλειας ::
void Employee::print_details() {
    cout << fullname << " has now a salary of " << salary << endl;
}

int main() {

    // Πρέπει να δίνονται και όσα ορίσματα ορίζονται στον δομητή της κλάσης
    Employee e("Nick", 1000);

    // Το μέλος salary έχει πρόσβαση ιδιωτική (private).
    // Πρέπει να χρησιμοποιήσουμε την δημόσια (public) συνάρτηση μέλος
    // set_salary()
    e.set_salary(1500);

    // Η συνάρτηση μέλος print_details() δεν επιστρέφει κάτι και συνεπώς δεν
    // μπορεί να συμμετέχει σε ένα cout. Από μόνη της η συνάρτηση εκτυπώνει
    // στην οθόνη και αρκεί η κλήση της στο αντικείμενο e
    e.print_details();

    return 0;
}

```

## Άσκηση 2η

Ζητείται να ορίσετε μια κλάση **Person**, κάθε αντικείμενο της οποίας θα περιέχει το ονοματεπώνυμο ενός ανθρώπου και την ηλικία του. Η κλάση Person πρέπει να μπορεί να χρησιμοποιηθεί ως εξής:

```

int main() {
    Person p("John", "Pappas", 30);

    //Τυπώνει στην οθόνη: "Ονοματεπώνυμο: John Pappas, Ηλικία : 30"
    p.print();

    Person q;
    //Τυπώνει στην οθόνη: " Ονοματεπώνυμο: Unknown Unknown, Ηλικία : 0"
    q.print();

    return 0;
}

```

Γράψτε τον πλήρη ορισμό της κλάσης Person. Όλες οι ιδιότητες της κλάσης θα πρέπει να οριστούν ως ιδιωτικές. Ορίστε αποκλειστικά και μόνο τις μεθόδους που απαιτούνται.

## Απάντηση

```

#include <iostream>
#include <string>

```

```

using namespace std;

class Person {
public:
    // Ορίζουμε τους constructor της κλάσης
    Person();
    Person(string name, string surname, int age);

    // Ορίζουμε τον destructor της κλάσης
    ~Person();

    // Ορίζουμε την συνάρτηση print για την εμφάνιση των στοιχείων
    void print();

private:
    string onoma, epwnymo;
    int ilikia;
};

// Δημιουργούμε τον default constructor
Person::Person() {
    onoma = "Unknown";
    epwnymo = "Unknown";
    ilikia = 0;
}

// Δημιουργούμε τον constructor με τις τρεις παραμέτρους
Person::Person(string name, string surname, int age) {
    onoma = name;
    epwnymo = surname;
    ilikia = age;
}

// Δημιουργούμε τον destructor της κλάσης
Person::~~Person() {
    cout << "Egine i diagرافي του antikeimenou" << endl;
}

// Συνάρτηση print
void Person::print() {
    cout << "Onomatepwnymo:" << onoma << " " << epwnymo << ", Hlikia : "
    << ilikia << endl;
}

int main() {
    Person p("John", "Pappas", 30);

    //Τυπώνει στην οθόνη: "Onomatepwnymo: John Pappas, Hlikia : 30"
    p.print();

    Person q;
    //Τυπώνει στην οθόνη: " Onomatepwnymo: Unknown Unknown, Ηλικία : 0"
    q.print();
}

```

```
return 0;
}
```

### Άσκηση 3η

Υλοποιήστε ένα πρόγραμμα σε C++ το οποίο να δημιουργεί μια κλάση *Book* που αναπαριστά την οντότητα βιβλίο. Για κάθε αντικείμενο *Book* καταχωρούμε τον τίτλο του, τον αριθμό σελίδων του και την τιμή του. Η κλάση να υλοποιηθεί με τον ακόλουθο τρόπο:

1. Όλες οι μεταβλητές μέλη να είναι δημόσιες εκτός από την *τιμή* που θα πρέπει να την ορίσετε ως ιδιωτική.
2. Να υπάρχει η δυνατότητα δημιουργίας αντικειμένου χωρίς όρισμα που να δημιουργεί ένα αντικείμενο βιβλίου με κενό τίτλο, 0 σελίδες και μηδενική τιμή.
3. Να υπάρχει η δυνατότητα δημιουργίας αντικειμένου με ένα όρισμα, τον τίτλο, που να δημιουργεί αντικείμενο με τίτλο αυτόν του ορίσματος και τα υπόλοιπα 0.
4. Να παρέχεται η δυνατότητα δημιουργίας αντικειμένου με 2 ορίσματα, τον τίτλο και τον αριθμό σελίδων. Να δημιουργεί αντικείμενα με τις αντίστοιχες τιμές των ορισμάτων και την τιμή 0.
5. Να περιέχει συνάρτηση μέλος που να εμφανίζει στην οθόνη τα στοιχεία ενός βιβλίου.
6. Να περιέχει συνάρτηση που να θέτει την τιμή του βιβλίου.

Στο κυρίως πρόγραμμα (συνάρτηση *main()*), να δημιουργήσετε 3 αντικείμενα χρησιμοποιώντας τους 3 τρόπους δημιουργίας που αναφέρθηκαν στις προδιαγραφές παραπάνω. Αναθέστε τιμές για όλα τα μέλη τους και στο τέλος εμφανίστε τα στοιχεία και των τριών βιβλίων.

### Απάντηση

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

```
class Book {
    float price;
public:
    string title;
    int pages;

    Book();
    Book(string p_title);
    Book(string p_title, int p_pages);
    void display();
    void set_price(float p_price);
};
```

```
// Δομητές με αρχικοποίηση λίστας (List initialization)
```

```
Book::Book(): title(""), pages(0), price(0.0) {}
Book::Book(string p_title): title(p_title), pages(0), price(0.0) {}
Book::Book(string p_title, int p_pages): title(p_title), pages(p_pages),
price(0.0) {}
```

```
void Book::display() {
```

```

        cout << "Title: " << title << ", pages: " << pages << " price: " <<
price << "" << endl;
}

void Book::set_price(float p_price) {
    price=p_price;
}

int main() {
    Book b1;
    Book b2("Programming in C++");
    Book b3("Programming in Python", 200);

    b1.title = "Around the world in 80 days";
    b1.pages = 180;
    b1.set_price(12.80);

    b2.pages = 667;
    b2.set_price(48.50);

    b3.set_price(43.00);

    b1.display();
    b2.display();
    b3.display();
    return 0;
}

```

#### Άσκηση 4η

Η βιβλιοθήκη του Πανεπιστημίου Αιγαίου, θέλοντας να βελτιώσει τις υπηρεσίες της, σας ανέθεσε την δημιουργία ενός προγράμματος στο οποίο να μπορεί να καταχωρεί για κάθε φοιτητή/τρια το όνομα του/της, το επώνυμο του/της, τον αριθμό μητρώου του/της, και το πλήθος των υλικών της βιβλιοθήκης (βιβλία, περιοδικά, λογισμικό) που έχει δανειστεί. Επειδή η βιβλιοθήκη θέλει να διασφαλίσει την ευκολία συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης του κώδικα που θα παρέχετε, καθώς και να προστατέψει τα προσωπικά δεδομένα των φοιτητών, έχει ζητήσει η ανάπτυξη να γίνει αποκλειστικά με χρήση αντικειμενοστρεφούς προσέγγισης.

Υλοποιείτε ένα πρόγραμμα σε C++, το οποίο να περιλαμβάνει την κατάλληλη κλάση για την δημιουργία αντικειμένων φοιτητών. Τα στοιχεία του φοιτητή να μπορούν να οριστούν απευθείας με την χρήση κατάλληλου δομητή (constructor) ο οποίος να δέχεται ως ορίσματα όλα τα στοιχεία του φοιτητή εκτός από τον αριθμό των υλικών που έχει δανειστεί. Για αυτό το λόγο θα πρέπει να ορίσετε και κατάλληλη μέθοδο που θα επιτρέπει, μετά την δημιουργία των αντικειμένων, την ανάθεση τιμής στη μεταβλητή που αφορά το πλήθος των υλικών. Η μέθοδος αυτή θα πρέπει να αναθέτει την τιμή που ορίζεται μόνο αν είναι έγκυρη, δηλαδή θετικός αριθμός. Σε αντίθετη περίπτωση θα πρέπει να θέτει την τιμή των υλικών 0 και να επιστρέφει ψευδές.

Η κλάση θα πρέπει επίσης να περιλαμβάνει και κατάλληλη μέθοδο για την εκτύπωση των στοιχείων του κάθε φοιτητή. Κατά την εκτύπωση θα πρέπει να εμφανίζεται και κατάλληλο μήνυμα που να ενημερώνει τον χρήστη σε περίπτωση που το πλήθος των υλικών που έχει δανειστεί ο φοιτητής ξεπερνά το όριο των 10.

Ακολούθως υλοποιήστε main συνάρτηση στην οποία να δημιουργήσετε 2 φοιτητές με στοιχεία που επιθυμείτε. Στην συνέχεια εμφανίστε τα στοιχεία των φοιτητών με την κλήση της κατάλληλης λειτουργίας.

### Απάντηση

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

class StudentMember {
private:
    string name;
    string surname;
    string id;
    int numberItems;
public:
    StudentMember(string p_name, string p_surname, string p_id);
    bool setNumberItems(int p_numberItems);
    void display();
};

StudentMember::StudentMember(string p_name, string p_surname, string p_id)
{
    name = p_name;
    surname = p_surname;
    id = p_id;
}

bool StudentMember::setNumberItems(int p_numberItems) {
    if (p_numberItems > 0) {
        numberItems = p_numberItems;
        return true;
    }
    numberItems = 0;
    return false;
}

void StudentMember::display() {
    cout << "Name: " << name << ", Surname: " << surname << ", Member id: " << id << ", Number of Items: " << numberItems << endl;
    if (numberItems > 10)
        cout << "You have exceeded the limit. You can not loan other items!" << endl;
}

int main() {
    StudentMember stud1("George", "Iwannou", "001");
    stud1.setNumberItems(-6);
    stud1.display();
    stud1.setNumberItems(6);
    stud1.display();

    StudentMember stud2("Maria", "Douka", "011");
```

```

    stud2.setNumberItems(6);
    stud2.display();
    stud2.setNumberItems(12);
    stud2.display();
    return 0;
}

```

### Άσκηση 5η

Ένας φοιτητής/τρια έχει αριθμό μητρώου, ονοματεπώνυμο και βαθμούς για 5 από τα μαθήματά του/της. Να κατασκευάσετε ένα πρόγραμμα σε C++ χρησιμοποιώντας αντικειμενοστρεφή προγραμματισμό, ώστε να δημιουργείτε μαθητές (χρήση δομητή-constructor) και να εισάγετε βαθμολογίες για κάθε ένα από τα μαθήματά τους.

Για κάθε μαθητή/τρια που θα δημιουργείται στο κύριο πρόγραμμα, θα πρέπει να υπάρχουν κατάλληλες συναρτήσεις μέλη για να εμφανίζονται τα στοιχεία του, και αντίστοιχη συνάρτηση μέλος που να εμφανίζει το μέσο όρο των μαθημάτων του και αν πέρασε η όχι (πέρασε αν μέσος όρος  $\geq 5$ ).

### Απάντηση

```

#include <iostream>
using namespace std;

```

```

// Σε περίπτωση που στο μέλλον θέλουμε να αλλάξουμε τον αριθμό μαθημάτων
#define numGrades 5

```

```

class Student {
    int id;
    string fullName;
    float grade[numGrades];
    float average;

    // Υπολογίζει την μέση βαθμολογία
    void computeAvg() {
        float sum=0;
        int i;
        for (i=0; i<numGrades; i++)
            sum = sum + grade[i];
        average = sum / numGrades;
    }

public:
    Student(int id, string fullName):id(id),fullName(fullName) {};

    string getFullName() const {
        return fullName;
    }

    int getId() const {
        return id;
    }

    // Ανάλογα με τη μέση βαθμολογία εμφανίζει αν πέρασε ή όχι ο φοιτητής

```

```

void succeeded() const {
    if (average >= 5)
        cout << fullName << " Passed" << endl;
    else
        cout << fullName << " Failed" << endl;
}

// Θέτει τις βαθμολογίες στον πίνακα και υπολογίζει τον μέσο όρο
void setGrades(float g[]) {
    for (int i=0; i<numGrades; i++)
        grade[i]=g[i];
    computeAvg();
}

// Εμφανίζει όλα τα στοιχεία των φοιτητών
void display() const {
    cout << endl;
    cout << "Student's Name: " << fullName << endl;
    cout << "Average Grade: " << average << endl;
    succeeded();
    cout << endl;
}
};

int main() {
    // Δημιουργία 2 φοιτητών/ριών
    Student s1(1, "Student 1");
    Student s2(2, "Student 2");

    float grade[numGrades];

    cout << "Please insert grades for s1\n";
    for (int i=0; i<numGrades; i++) {
        cin >> grade[i];
    }

    s1.setGrades(grade);
    s1.display();

    // Εισαγωγή βαθμολογιών στον/την επόμενο/η
    cout << "Please insert grades for s2\n";
    for (int i=0; i<numGrades; i++) {
        cin >> grade[i];
    }

    s2.setGrades(grade);
    s2.display();
    return 0;
}

```