



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Οντοκεντρικός Προγραμματισμός

Ενότητα: Εργαστηριακές Ασκήσεις

Ιωάννης Χατζηλυγερούδης, Χρήστος Μακρής

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα **ΠΠ**

Εργαστηριακές Ασκήσεις C++

ΕΡΓΑΣΤΗΡΙΟ ΟΝΤΟΚΕΝΤΡΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ (C++)

ΑΣΚΗΣΗ 1

Σκοπός της άσκησης αυτής είναι:

- να δημιουργήσετε ένα νέο αρχείο C++, να εισάγετε κάποιο κώδικα, να τον μεταγλωττίσετε και να δείτε τα αποτελέσματα με χρήση του περιβάλλοντος Code::Blocks
- να εξασκηθείτε με την είσοδο και έξοδο δεδομένων από και προς την οθόνη
- να εξοικειωθείτε με βασικές δομές επανάληψης όπως το for loop
- να πειραματιστείτε με τη δημιουργία κλάσεων και την υπερφόρτωση τελεστών

1) Εκτελέσετε τον παρακάτω κώδικα δίνοντας σαν είσοδο τις τιμές 3 και 12. Στη συνέχεια τροποποιήστε τον ώστε εάν η είσοδος είναι μεγαλύτερη του 10 να εμφανίζεται μήνυμα «This is a big enough number!»

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter a number: " << endl;
    int value;
    cin >> value;
    if(value < 10)
    {
        cout << "This value is too small";
    }
    return 0;
}
```

2) Δοκιμάστε τον παρακάτω κώδικα. Πώς λειτουργεί και τι εμφανίζει στην οθόνη;

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int val1;
    int val2;
    int val3;
    cout<<"Please enter your 3 numbers:";
    cin>>val1>>val2>>val3;
    cout<<val1<<"\n"<<val2<<"\n"<<val3<<"\n";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Κάντε τις απαραίτητες αλλαγές ώστε πριν τερματίσει να εμφανίζει τους αριθμούς που δόθηκαν σαν είσοδος με ανάστροφη σειρά.

3) Μελετήστε τον παρακάτω κώδικα και εξηγήστε τι κάνει:

```

#include<iostream>

using namespace std;

int main(){
cout<<"First set of numbers:"<<endl;
int first[2][2], second[2][2];
int i,j;
for( i=0;i<2;i++){
    cout<<"Enter two integers: "<<i+1<<endl;
    for( j=0;j<2;j++){
        cin>>first[i][j];
    }
}
cout<<"\n\n\nSecond set of numbers:"<<endl;
for( i=0;i<2;i++){
    cout<<"Enter two more integers: "<<i+1<<endl;
    for( j=0;j<2;j++){
        cin>>second[i][j];
    }
}
for( i=0;i<2;i++){
    for( j=0;j<2;j++){
        first[i][j]=first[i][j]+second[i][j];
    }
}
return 0;
}

```

Προσθέστε κατάλληλο for loop που να εμφανίζει το αποτέλεσμα στην οθόνη.

Στη συνέχεια χρησιμοποιήστε 2 σταθερές που να καθορίζουν αντίστοιχα τις αντίστοιχες κοινές διαστάσεις των δύο πινάκων και κάντε τις κατάλληλες αλλαγές στα σημεία του κώδικα.

Τέλος, αλλάξτε τον κώδικα ώστε οι διαστάσεις των πινάκων να δίνονται από το χρήστη μετά από κατάλληλο μήνυμα που θα εμφανίζεται την οθόνη.

- 4) i. Χρησιμοποιώντας τον παρακάτω ορισμό μιας κλάσης (τάξης) σε C++ γράψτε τον κώδικα της main() όπου
- θα δημιουργούνται δύο αντικείμενα τύπου Kouti, τα KoutiA και KoutiB
 - θα αρχικοποιεί μια μεταβλητή ογκος τύπου double στην τιμή 0.0 (στην μεταβλητή αυτή θα αποθηκεύεται ο όγκος του κάθε κουτιού που θα υπολογιστεί στη συνέχεια
 - θα αρχικοποιείται το KoutiA με τιμές 2.0, 3.2 και 6.0 στις αντίστοιχες μεταβλητές
 - θα αρχικοποιείται το KoutiB με τιμές 2.5, 4.0 και 5.0 στις αντίστοιχες μεταβλητές
 - θα υπολογίζεται ο όγκος KoutiA στη μεταβλητή ογκος και θα εμφανίζεται στην οθόνη με σχετικό επεξηγηματικό μήνυμα
 - θα υπολογίζεται ο όγκος του KoutiB στη μεταβλητή ογκος και θα εμφανίζεται στην οθόνη με σχετικό επεξηγηματικό μήνυμα

```

class Kouti
{
public:
    double length;
    double breadth;
    double height;
};

```

ii. Μετατρέψτε τις ιδιότητες της κλάσης Kouti σε private και εφοδιάστε την κλάση με κατάλληλη public μέθοδο calculateOgkos() που θα υπολογίζει τον όγκο κάθε αντικειμένου. Κάντε στη main() τις κατάλληλες αλλαγές ώστε να λειτουργεί όπως και στο (i).

5) Προσθέστε στην παραπάνω κλάση Kouti τον εξής κώδικα ο οποίος πραγματοποιεί υπερφόρτωση του τελεστή +.

```
Kouti operator+(const Kouti& b)
{
    Kouti kouti;
    kouti.length = this->length + b.length;
    kouti.breadth = this->breadth + b.breadth;
    kouti.height = this->height + b.height;
    return kouti;
}
```

Προσθέστε στην Kouti μεθόδους setMikos(), setPlatos() και setYpsos() και τροποποιήστε τη main() ώστε
-να δημιουργεί ένα επιπλέον αντικείμενο τύπου Kouti με όνομα KoutiC
-αφού εμφανίσει και τον όγκο του δεύτερου κουτιού να εφαρμόζει τον υπερφορτωμένο τελεστή στο πρώτο και το δεύτερο κουτί και να αποθηκεύει το αποτέλεσμα στο KoutiC
-να καλεί την calculateOgkos() επί του KoutiC και να εμφανίζει το αποτέλεσμα στην οθόνη με κατάλληλο επεξηγηματικό μήνυμα.

ΕΡΓΑΣΤΗΡΙΟ ΟΝΤΟΚΕΝΤΡΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ (C++)

ΑΣΚΗΣΗ 2

Σκοπός της άσκησης αυτής είναι:

- να εξοικειωθείτε με τη χρήση του vector της STL
- να πειραματιστείτε με την υλοποίηση ιεραρχίας κλάσεων μέσω κληρονομικότητας
- να εξασκηθείτε στη χρήση των προσδιοριστικών ορατότητας
- να πειραματιστείτε με τον πολυμορφισμό και να αντιληφθείτε τις έννοιες early και late binding

1). Μελετήστε τον κώδικα που ακολουθεί και εξηγήστε τη λειτουργία του. Μετατρέψτε τον ώστε στο διάνυσμα vec να τοποθετούνται τυχαίες τιμές χρησιμοποιώντας τη συνάρτηση rand(). Μην ξεχάσετε να προσθέσετε το `#include <stdlib.h>`

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

int main()
{
    vector<int> vec;
    int i;

    cout << "vector size = " << vec.size() << endl;

    for(i = 0; i < 7; i++){
        vec.push_back(i);
    }

    cout << "extended vector size = " << vec.size() << endl;

    for(i = 0; i < vec.size(); i++){
        cout << "Vector [" << i << "] = " << vec[i] << endl;
    }

    return 0;
}
```

Κάντε τις απαραίτητες αλλαγές ώστε τα περιεχόμενα του διανύσματος να εμφανίζονται χρησιμοποιώντας iterator όπως στον κώδικα που ακολουθεί.

```
vector<int>::iterator v = vec.begin();
while( v != vec.end()) {
    cout << "value of v = " << *v << endl;
    v++;
}
```

Προσθέστε τον κατάλληλο κώδικα ώστε να μειωθεί το μέγεθος του vector σε 5 χρησιμοποιώντας την `resize()`, στη συνέχεια να εκτυπωθεί το μήνυμα "reduced vector size = " ακολουθούμενο από το νέο μέγεθος του vector και να εμφανιστούν τα νέα περιεχόμενά του.

Τι θα συμβεί στα περιεχόμενα του vector αν η `resize()` κληθεί ως εξής: `resize(10,5)`;

2) Εκτελέστε τον παρακάτω κώδικα και παρατηρήστε τον τρόπο που έχουν οριστεί οι συναρτήσεις getIpsos() και setIpsos(). Λειτουργούν σωστά; Πώς μπορούμε να ορίζουμε και να χρησιμοποιούμε συναρτήσεις κλάσεων με αυτό τον τρόπο;

```
#include <iostream>

using namespace std;

class Kouti
{
public:
    double mikos = 2.0;
    double platos;
    double ipsos;

    void setMikos( double mik ) {
        mikos = mik;
    };
    void setPlatos( double pla ) {
        platos = pla;
    };
    double getPlatos (void){
        return platos;
    };
    double getMikos (void) {
        return mikos;
    };
    void setIpsos( double ips );
    double getIpsos( void );
};

double Kouti::getIpsos( void )
{
    return ipsos;
}

void Kouti::setIpsos ( double ips )
{
    ipsos = ips;
}

int main( )
{
    Kouti Small;
    Kouti Big;
    double ogkos = 0.0;

    Small.platos= 1.0;
    Small.ipsos = 2.0;
    ogkos = Small.platos * Small.ipsos * Small.mikos;
    cout << "Ogkos gia Kouti Small: " << ogkos <<endl;

    Big.mikos = 12.0;
    Big.platos = 13.0;
```

```

Big.ipsos = 10.0;

ogkos = Big.platos * Big.ipsos * Big.mikos;
cout << "Ogkos gia Kouti Big: " << ogkos <<endl;
return 0;
}

```

Αλλάξτε το προσδιοριστικό πρόσβασης public πριν τον ορισμό των ιδιοτήτων της κλάσης Kouti σε private ώστε να τηρείται η αρχή της απόκρυψης των πληροφοριών. Εκτελέστε τον κώδικα. Τι πρόβλημα προκύπτει και γιατί; Κάντε τις απαραίτητες αλλαγές στη main χρησιμοποιώντας μεθόδους αντί των ιδιωτικών ιδιοτήτων ώστε ο κώδικας να λειτουργεί σωστά.

3) Μελετήστε τον παρακάτω κώδικα:

```

#include <iostream>
using namespace std;

class Polygon {
protected:
    int width, height;
public:
    void set_values (int a, int b)
        { width=a; height=b;}
};

class Rectangle: public Polygon {
public:
    int area ()
        { return width * height; }
};

int main () {
    Rectangle rect;
    rect.set_values (5,8);
    cout << "Emvadon orthogoniou: " << rect.area() << '\n';
    return 0;
}

```

Προσθέστε στην κατάλληλη θέση της ιεραρχίας και μια κλάση Triangle που θα αναπαριστά αντικείμενα τύπου τρίγωνο για τα οποία διατηρούμε δύο ιδιότητες (μήκος και πλάτος), μια μέθοδο set_values () με δύο ακέραια ορίσματα για το μήκος και το πλάτος και μια μέθοδο area() που θα επιστρέφει το εμβαδό υπολογίζοντάς το κατά τα γνωστά, δηλ. $(\text{πλάτος} \times \text{ύψος}) / 2$

Στη συνέχεια προσθέστε στη main αντίστοιχες εντολές ώστε να δημιουργηθεί ένα αντικείμενο τρίγωνο με πλάτος 6 και μήκος 4 και να τυπωθεί στην οθόνη μήνυμα που θα εμφανίζει το εμβαδόν του.

4). Στον κώδικα που καταλήξατε στο ερώτημα 2 προσθέστε την κλάση PaintCost και κάντε τις απαραίτητες αλλαγές ώστε η κλάση Rectangle να είναι απόγονος και της PaintCost.

```

class PaintCost
{
public:
    int getCost(int area)

```



```

    {
        return area * 70;
    }
};

```

Συμπληρώστε στη main κατάλληλες εντολές ώστε να εμφανίζεται μετά το μήνυμα για το εμβαδό του ορθογωνίου και το εξής "Συνολικό κόστος χρωματος: "....."euro". Η κλάση Rectangle από πόσες κλάσεις κληρονομεί; Μπορεί κάτι τέτοιο να υλοποιηθεί σε Java; Εξηγήστε.

5). Ας υποθέσουμε ότι η μέθοδος area() πρέπει να έχει διαφορετική υλοποίηση στην Polygon από ότι στην Rectangle, όπως φαίνεται στον κώδικα που ακολουθεί:

```

#include <iostream>
using namespace std;

class Polygon {
protected:
    int width, height;
public:
    Polygon( int a=0, int b=0)
    {
        width = a;
        height = b;
    }
    int area()
    {
        cout << "This is area as computed by the Polygon class" <<endl;
        return 0;
    }
};

class Rectangle: public Polygon{
public:
    Rectangle( int a=0, int b=0):Polygon(a, b) { }
    int area ()
    {
        cout << "This is area as computed by the Rectangle class" <<endl;
        return (width * height);
    }
};

int main( )
{
    Polygon *polygon;
    Rectangle rec(10,7);

    polygon = &rec;
    polygon->area();

    return 0;
}

```

Εκτελέστε τον παραπάνω κώδικα. Τί αποτέλεσμα προκύπτει; Είναι αυτό που θα περιμέναμε καλώντας την area() επί αντικειμένου τύπου Rectangle; Αναζητήστε πληροφορίες σχετικά με το μηχανισμό early binding που εφαρμόζει η C++ και εξηγήστε γιατί ο παραπάνω κώδικας λειτουργεί όπως παρατηρήσατε.

Στο σημείο που ορίζεται η area() στην κλάση Polygon προσθέστε το χαρακτηριστικό virtual πριν το int area () υλοποιώντας με αυτό τον τρόπο το μηχανισμό late binding. Εκτελέστε ξανά τον κώδικα. Τί παρατηρείτε;

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0

Σημείωμα Αναφοράς

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015. «Οντοκεντρικός Προγραμματισμός».

Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Πίνακες

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

