ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΟΝΤΟΚΕΝΤΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι(C++)

ΔΙΔΑΣΚΩΝ : Ι. ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ,Χ. ΜΑΚΡΗΣ,Μ. ΡΗΓΚΟΥ

ΦΟΙΤΗΤΡΙΑ : ΚΑΛΑΜΑΤΙΑΝΟΥ ΔΗΜΗΤΡΑ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ : 1054406
ΕΤΟΣ : 1ο
ΕΞΑΜΗΝΟ : 2ο

3ο ΣΕΤ ΑΣΚΗΣΕΩΝ

ΑΣΚΗΣΗ 1

Ο κώδικας της άσκησης έχει την παρακάτω μορφή :

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


using namespace std;

class student
{
    private:
    string name,surname;
    int AM;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in superclass Student
\n";
        }
};
class undergrad : public student
{
    private:
    int entrance_order;
    int passed_courses;
    public:
    void message()
        {
```

```
        cout<<"Hello, I am message() defined in  class undergraduate
student \n";
        }
};

class MSc: public student
{
    private:
    string dipl_dept;
    string thesis;
    int dipl_grade;
    public:
    int  get_dipl_grade();
};

class PhD: public student
{
    private: private:string name,surname;int AM;
    string PhD_title,professor;
    int start_month,start_yearQ;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in  class Phd
student\n";
        }

};


int  main()
{
    student S1;
    undergrad U1 ;
    MSc M1;
    PhD Ph1;

    S1.message(); //grammh 1
    U1.message(); //grammh 2
    M1.message(); //grammh 3
    Ph1.message(); //grammh 4

    system("pause");
    return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Hello, I am message() defined in superclass Student

Hello, I am message() defined in class undergraduate student
Hello, I am message() defined in superclass Student
Hello, I am message() defined in class Phd student

α . Με την εντολή « S1.message() » καλούμε την συνάρτηση message πάνω στο αντικείμενο S1 το οποίο είναι τύπου student. Οπότε και εμφανίζεται το πρώτο μήνυμα « Hello, I am message() defined in superclass Student » της κλάσης student.

Με την εντολή « U1.message() »καλούμε την συνάρτηση message πάνω στο αντικείμενο U1 το οποίο είναι τύπου undergrad. Οπότε και εμφανίζεται το δεύτερο μήνυμα « Hello, I am message() defined in class undergraduate student » της κλάσης undergrad.

Με την εντολή «M1.message()» καλούμε την συνάρτηση message πάνω στο αντικείμενο M1 το οποίο είναι τύπου MSc. Όμως αυτή η κλάση δεν έχει δικιά της συνάρτηση οπότε εμφανίζει το τρίτο μήνυμα « Hello, I am message() defined in superclass Student » της κλάσης student., η οποία είναι υπερκλάση της MSc.

Με την εντολή « Ph1.message() »καλούμε την συνάρτηση message πάνω στο αντικείμενο PH1 το οποίο είναι τύπου PhD . Οπότε και εμφανίζεται το τέταρτο μήνυμα « Hello, I am message() defined in class Phd student » της κλάσης undergrad.

β . Με την εντολή « S1 = U1 » εμφανίστηκε ένα καινούργιο μήνυμα : « Hello, I am message() defined in superclass Student » Εκτελέστηκε η υλοποίηση της message στην κλάση student. Με την εντολή « S1 = U1 » το S1 γίνεται U1 όμως παραμένει αντικείμενο τύπου student.

γ . Ο κώδικας είναι της μορφής :

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


using namespace std;

class student
{
```

```
    private:
    string name,surname;
    int AM;
     public:
    virtual void message()
        {
        cout<<"Hello, I am message() defined in superclass Student
\n";
        }
};
class undergrad : public student
{
    private:
    int entrance_order;
    int passed_courses;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in  class undergraduate
student \n";
        }
};

class MSc: public student
{
    private:
    string dipl_dept;
    string thesis;
    int dipl_grade;
    public:
    int  get_dipl_grade();
};

class PhD: public student
{
    private:
    string PhD_title,professor;
    int start_month,start_yearQ;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in  class Phd
student\n";
        }

};


int  main()
{
    student *S1;
    student S2;
    S1 = & S2;
    S1 -> message();
    undergrad U1 ;
```

```
    S1 = & U1;
    S1 -> message();
    MSc M1;
    M1.message();
    PhD Ph1;
    S1 = & Ph1;
    S1 -> message();



    system("pause");
    return 0;
      }
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Hello, I am message() defined in superclass Student
  Hello, I am message() defined in class undergraduate student
  Hello, I am message() defined in class Phd student
  Hello, I am message() defined in superclass Student

Έπρεπε να βάλω virtual void message() στην κλάση student. Όταν δεν υπήρχε η λέξη virtual η έξοδος ήταν λάθος. Ο λόγος για την εσφαλμένη έξοδο είναι ότι η κλήση της message () ορίζεται μία φορά από τον compiler στην κλάση student. Αυτό ονομάζεται **early binding** επειδή η συνάρτηση message () έχει οριστεί κατά τη διάρκεια της σύνταξης του προγράμματος. Αλλά με μια μικρή τροποποίηση στο πρόγραμμά μας βάζοντας virtual void message() στην κλάση student ο μεταγλωττιστής εξετάζει τα περιεχόμενα του δείκτη αντί του τύπου του. Επομένως, επειδή οι διευθύνσεις των αντικειμένων των κλάσεων S2,U1 και Ph1 αποθηκεύονται σε μορφή *, η αντίστοιχη συνάρτηση message() καλείται.

δ . Ο κώδικας είναι της μορφής :

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


using namespace std;

class student
{
    private:
    string name,surname;
    int AM;
    public:
    virtual void message()
```

```
        {
        cout<<"Hello, I am message() defined in superclass Student
\n";
        }
};
class undergrad : public student
{
    private:
    int entrance_order;
    int passed_courses;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in  class undergraduate
student \n";
        }
};

class MSc: public student
{
    private:
    string dipl_dept;
    string thesis;
    int dipl_grade;
    public:
    int  get_dipl_grade();
};

class PhD: public student
{
    private:
    string PhD_title,professor;
    int start_month,start_yearQ;
    public:
    void message()
        {
        cout<<"Hello, I am message() defined in  class Phd
student\n";
        }

};


int  main()
{
    student *S1;
    student S2;
    S1 = & S2;
    undergrad U1 ;
    S1 = & U1;
    MSc M1;
    PhD Ph1;
    S1 = & Ph1;
```

```
student * Message[4] = {&S2,&U1,&M1,&Ph1};
for(int i = 0; i <4; i++)
{
    Message[i] -> message();

}

system("pause");
return 0;
   }
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Hello, I am message() defined in superclass Student
  Hello, I am message() defined in class undergraduate student
  Hello, I am message() defined in class Phd student
  Hello, I am message() defined in superclass Student

ΑΣΚΗΣΗ 2

α . Η λειτουργία του κώδικα είναι να δίνονται το όνομα, η τιμή και το score ενός προϊόντος και να εμφανίζει το όνομα του καλύτερου προϊόντος , της καλύτερης τιμής και του καλύτερου score.

β. Ο κώδικας είναι της μορφής :

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

class Product {
 private :
string name = "";
double price = 1;
int score = 0;
public:
void read()
{
cout << "Please enter the model name: ";
getline(cin, name);
cout << "Please enter the price: ";
cin >> price;
cout << "Please enter the score: ";
cin >> score;
            string remainder; /* read remainder of line */
            getline(cin, remainder);
        }
        double comparison()
        {
```

```
            return score / price;
        }
        bool is_better_than(Product p)
        {
          if (comparison() > p.comparison())
            {
            return true;
            }
            return false;
        }
        void print()
        {
            cout << "The best value is " << name
            << " (Score: " << score
            << " Price: " << price << ")\n";
        }
};

int main()
{
Product best;
bool more = true;
while (more)
{
    Product next;
    next.read();
    if (next.is_better_than(best)) best = next;
    cout << "More data? (y/n) ";
    string answer;
    getline(cin, answer);
    if (answer != "y") more = false;
}
cout << "The best value is ";
best.print();
return 0;
}
```

γ . Ο κώδικας είναι της μορφής :

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

class Product {
    private :
        string name = "";
        double price = 1;
        int score = 0;
    public:
        void read()
        {
            cout << "Please enter the model name: ";
            getline(cin, name);
            cout << "Please enter the price: ";
            cin >> price;
```

```
            cout << "Please enter the score: ";
            cin >> score;
            string remainder; /* read remainder of line */
            getline(cin, remainder);
        }
        double comparison()
        {
            return score / price;
        }
        bool is_better_than(Product p)
        {
          if (comparison() > p.comparison())
            {
            return true;
            }
            return false;
        }
        friend void print(Product p);
};

int main()
{
Product best;
bool more = true;
while (more)
{
    Product next;
    next.read();
    if (next.is_better_than(best)) best = next;
    cout << "More data? (y/n) ";
    string answer;
    getline(cin, answer);
    if (answer != "y") more = false;
}
cout << "The best value is ";
print(best);
return 0;
}
void print(Product p)
{
           cout << "The best value is " << p.name
            << " (Score: " << p.score
            << " Price: " << p.price << ")\n";
}
```

δ . Ο κώδικας είναι της μορφής :

```
#include <iostream>

      #include <string>
      using namespace std;

      class Product {
```

```
    private :
        string name = "";
        double price = 1;
        int score = 0;
    public:
        void read()
        {
            cout << "Please enter the model name: ";
            getline(cin, name);
            cout << "Please enter the price: ";
            cin >> price;
            cout << "Please enter the score: ";
            cin >> score;
            string remainder; /* read remainder of line */
            getline(cin, remainder);
        }
        double comparison()
        {
            return score / price;
        }
        friend void print(Product p);
        bool operator> (Product &p1)
        {
            return comparison() > p1.comparison();

        }
};


int main()
{
Product best;
bool more = true;
while (more)
{
    Product next;
    next.read();
    if  (next > best) best = next;
    cout << "More data? (y/n) ";
    string answer;
    getline(cin, answer);
    if (answer != "y") more = false;
}
cout << "The best value is ";
print(best);
return 0;
}
void print(Product p)
{
             cout << "The best value is " << p.name
            << " (Score: " << p.score
            << " Price: " << p.price << ")\n";
}
```

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

ΑΣΚΗΣΗ 1

1. Ο τροποποιημένος κώδικας είναι της μορφής :

```
#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";
		}
	else if (value > 10)
		{
		    cout << "This is a big enough number!";
		}
	return 0;
}
```

2. Ο κώδικας :

```
#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;
}
```

Λειτουργεί ως εξής : ζητά τις τιμές τριών μεταβλητών και στην συνέχεια τις εμφανίζει με την σειρά που δόθηκαν. Για να εμφανίζει αυτές τις τιμές με ανάστροφη σειρά αρκεί να αλλάξω την σειρά των ονομάτων των μεταβλητών στο cout. Ο κώδικας θα έχει τώρα την παρακάτω μορφή :

```
#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<<val3<<"\n"<<val2<<"\n"<<val1<<"\n";
system("PAUSE");
return EXIT_SUCCESS;
}
```

3. Ο κώδικας :

```
#include<iostream>
using namespace std;
int main(){
    cout<<" give me the size of the arrays"<<endl;
    int x;
    cin>>x;
    const int a=x;
    const int b=x;
cout<<"First set of numbers:"<<endl;
int first[a][b], second[a][b];
int i,j;
for( i=0;i<a;i++){
cout<<"Enter two integers: "<<i+1<<endl;
for( j=0;j<b;j++){
cin>>first[i][j];
}
}
cout<<"\n\n\nSecond set of numbers:"<<endl;
for( i=0;i<a;i++){
cout<<"Enter two more integers: "<<i+1<<endl;
for( j=0;j<b;j++){
cin>>second[i][j];
}
}
for( i=0;i<a;i++){
        for( j=0;j<b;j++){
first[i][j]=first[i][j]+second[i][j];
}
}
cout<<"The results are:"<<endl;
for( i=0;i<a;i++){
        for( j=0;j<b;j++){

            cout<<first[i][j]<<endl;

        }
}
return 0;
}
```

4. i) Ο κώδικας είναι της μορφής :

```
#include <iostream>

using namespace std;

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

int main()
{
    double ogkos = 0.0;
    Kouti KoutiA;
    Kouti KoutiB;

    KoutiA.length = 2.0;
    KoutiA.breadth = 3.2;
    KoutiA.height = 6.0;

    KoutiB.length = 2.5;
    KoutiB.breadth = 4.0;
    KoutiB.height = 5.0;

    ogkos =  KoutiA.length * KoutiA.breadth * KoutiA.height;
    cout << "Volume of KoutiA : " << ogkos << endl;

    ogkos =  KoutiB.length * KoutiB.breadth * KoutiB.height;
    cout << "Volume of KoutiB : " << ogkos << endl;
    return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Volume of KoutiA : 38.4
  Volume of KoutiB : 50

ii. Ο νέος κώδικας έχει την μορφή :

```
#include <iostream>

using namespace std;

class Kouti
{
private:
    double length;
```

```
        double breadth;
        double height;
    public :
        double calculateOgkos(double length,double breadth,double
    height)
        {
           return length * breadth * height;
        }
    };

    int main()
    {
        Kouti KoutiA;
        Kouti KoutiB;

        KoutiA.calculateOgkos(2.0,3.2,6.0);
        cout << "Volume of KoutiA : " <<
    KoutiA.calculateOgkos(2.0,3.2,6.0) << endl;

        KoutiB.calculateOgkos(2.5,4.0,5.0);
        cout << "Volume of KoutiB : " <<
    KoutiB.calculateOgkos(2.5,4.0,5.0) << endl;
        return 0;
    }
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Volume of KoutiA : 38.4
  Volume of KoutiB : 50

5.Ο κώδικας είναι της μορφής :

```
#include <iostream>

using namespace std;

class Kouti
{
private:
    double length;
    double breadth;
    double height;
public :
    double calculateOgkos()
    {
       return length * breadth * height;
    }
    void setMikos(double mikos)
    {
        length = mikos;
    }
    void setPlatos(double platos)
    {
        breadth = platos;
    }
    void setYpsos(double ypsos)
    {
```

```
        height = ypsos;
    }
    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;
    }
};

int main()
{
    Kouti KoutiA;
    Kouti KoutiB;
    Kouti KoutiC;

    KoutiA.setMikos(2.0);
    KoutiA.setPlatos(3.2);
    KoutiA.setYpsos(6.0);

    KoutiB.setMikos(2.5);
    KoutiB.setPlatos(4.0);
    KoutiB.setYpsos(5.0);

    KoutiA.calculateOgkos();
    cout << "Volume of KoutiA : " << KoutiA.calculateOgkos() << endl;

    KoutiB.calculateOgkos();
    cout << "Volume of KoutiB : " << KoutiB.calculateOgkos() << endl;

    KoutiC = KoutiA + KoutiB;

    KoutiC.calculateOgkos();
     cout << "Volume of KoutiC : " << KoutiC.calculateOgkos();
    return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Volume of KoutiA : 38.4
  Volume of KoutiB : 50
  Volume of KoutiC : 356.4

ΑΣΚΗΣΗ 2

1. Ο κώδικας είναι ο εξής :

```
#include <iostream>
#include <vector>
#include <stdlib.h>

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(10 + rand() % 90);

}
cout << "extended vector size = " << vec.size() << endl;
for(i = 0; i < vec.size(); i++)
{
    cout << "Vector [" << i << "] = " << vec[i] << endl;
}
vector<int>::iterator v = vec.begin();
 while( v != vec.end()) {
cout << "value of v = " << *v << endl;
v++;
}
 vec.resize(5);
 cout << "reduced vector size = " << vec.size() << endl;
 for(i = 0; i < 5; i++)
{
             vec.push_back(10 + rand() % 90);

}
for(i = 0; i < 5; i++)
{
    cout << "Vector [" << i << "] = " << vec[i] << endl;
}
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*

  vector size = 0
  extended vector size = 7
  Vector [0] = 51
  Vector [1] = 27
  Vector [2] = 44
  Vector [3] = 50
  Vector [4] = 99
  Vector [5] = 74
  Vector [6] = 58
  value of v = 51
  value of v = 27
  value of v = 44
  value of v = 50
  value of v = 99
  value of v = 74
  value of v = 58
  reduced vector size = 5

Vector [0] = 51
Vector [1] = 27
Vector [2] = 44
Vector [3] = 50
Vector [4] = 99

Αν η resize() κληθεί ως εξής: resize(10,5) τότε τα αποτελέσματα αφού τρέξω τον κώδικα είναι :

vector size = 0
extended vector size = 7
Vector [0] = 51
Vector [1] = 27
Vector [2] = 44
Vector [3] = 50
Vector [4] = 99
Vector [5] = 74
Vector [6] = 58
value of v = 51
value of v = 27
value of v = 44
value of v = 50
value of v = 99
value of v = 74
value of v = 58
reduced vector size = 10
Vector [0] = 51
Vector [1] = 27
Vector [2] = 44
Vector [3] = 50
Vector [4] = 99

2. *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
   Ogkos gia Kouti Small: 4
   Ogkos gia Kouti Big: 1560

Οι συναρτήσεις getIpsos() και setIpsos() έχουν οριστεί έξω από την κλάση Kouti. Αυτός είναι ένας άλλος τρόπος να τις ορίσουμε. Είναι σωστά ορισμένες.

Αφού αλλάξω τις ιδιότητες σε private ο νέος κώδικας είναι της μορφής :

```
#include <iostream>
using namespace std;
class Kouti
{
private:
	double mikos = 2.0;
	double platos;
	double ipsos;
public:
	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.setPlatos(1.0);
	Small.setIpsos (2.0);
	ogkos = Small.getPlatos() * Small.getIpsos() *
	Small.getMikos();
	cout << "Ogkos gia Kouti Small: " << ogkos <<endl;
	Big.setMikos(12.0);
	Big.setPlatos(13.0);
	Big.setIpsos (10.0);
	ogkos = Big.getPlatos() * Big.getIpsos() * Big.getMikos();
```

```
        cout << "Ogkos gia Kouti Big: " << ogkos <<endl;
        return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Ogkos gia Kouti Small: 4
  Ogkos gia Kouti Big: 1560

3. Ο κώδικας είναι ο εξής :

```
#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; }
};
class Triangle : public Polygon {
    public :
        int area()
        {
            return (width * height)/2;
        }
};
int main () {
    Rectangle rect;
    rect.set_values (5,8);
    cout << "Emvadon orthogoniou: " << rect.area() << '\n';
    Triangle trig;
    trig.set_values(6,2);
    cout << "Embadon trigonou: " << trig.area() << endl ;
    return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Emvadon orthogoniou: 40
  Embadon trigonou: 6

4. Ο κώδικας είναι ο εξής :

```
#include <iostream>
```

```
using namespace std;
class Polygon {
    protected:
        int width,
        height;
    public:
        void set_values (int a, int b)
        {
        width=a; height=b;
        }
};
class PaintCost{
    public:
        int getCost(int area)
        {
        return area * 70;
        }
};
class Rectangle: public Polygon, public PaintCost {
    public:
        int area ()
        { return width * height; }
};
class Triangle : public Polygon {
    public :
        int area()
        {
            return (width * height)/2;
        }
};

int main () {
    Rectangle rect;
    rect.set_values (5,8);
    cout << "Emvadon orthogoniou: " << rect.area() << '\n';
    cout << "Synoliko kostos xrwmatos: " <<
rect.getCost(rect.area()) << " euro" << '\n';
    Triangle trig;
    trig.set_values(6,2);
    cout << "Embadon trigonou: " << trig.area() << endl ;
    return 0;
}
```

- *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*
  Emvadon orthogoniou: 40
  Synoliko kostos xrwmatos: 2800 euro
  Embadon trigonou: 6

5. *Αφού τρέξω το πρόγραμμα τα αποτελέσματα είναι:*

   This is area as computed by the Polygon class

Αν βάλω virtual πριν το int area () θα εμφανίσει αυτό το μήνυμα :

This is area as computed by the Rectangle class

Αυτό συμβαίνει εξαιτίας της χρήσης της δεσμευμένης λέξης virtual. Όταν δεν υπήρχε η λέξη virtual η έξοδος ήταν λάθος. Ο λόγος για την εσφαλμένη έξοδο είναι ότι η κλήση της area () ορίζεται μία φορά από τον compiler στην κλάση Polygon. Αυτό ονομάζεται **early binding** επειδή η συνάρτηση area () έχει οριστεί κατά τη διάρκεια της σύνταξης του προγράμματος. Αλλά με μια μικρή τροποποίηση στο πρόγραμμά μας βάζοντας virtual int area() στην κλάση student ο μεταγλωττιστής εξετάζει τα περιεχόμενα του δείκτη αντί του τύπου του. Επομένως, από την εντολή polygon = &rec; Ο δείκτης αντικείμενο της κλάσης Polygon δείχνει σε αντικείμενο κλάσης Rectangle οπότε και καλείται η συνάρτηση area () της κλάσης Rectangle.