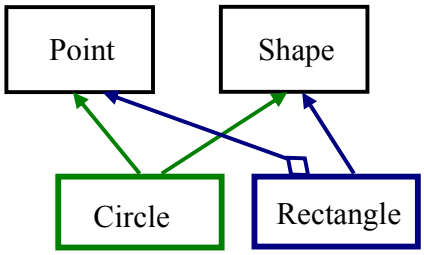
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ Α.Π.Θ.
ΕΡΓΑΣΤΗΡΙΟ C++
ΕΞΑΜΗΝΟ Γ – Ακαδημαϊκό Έτος 2006-2007

ΑΣΚΗΣΕΙΣ

1) Θέλουμε να κατασκευάσουμε την ακόλουθη ιεραρχία τάξεων:



όπου:

- η τάξη Shape έχει τη μορφή:

```
class Shape {
public:
   //.....
   virtual double Area() =0;
   virtual double Perifereia() =0;
   virtual void print() const = 0;
};
```

Σημείωση: οι συναρτήσεις "Area" και "Perifereia" υπολογίζουν αντίστοιχα το εμβαδό και την περιφέρεια ενός σχήματος.

- Η τάξη Rectangle μέσω της οποίας μπορούμε να κατασκευάσουμε αντικείμενα ορθογώνια έχει τη μορφή:

```
class Rectangle : public Shape {
public:
    // .......

   int GetWidth() const;  // Υπολογίζει το μήκος
   int GetHeight() const;  // Υπολογίζει το πλάτος
   virtual void print() const;
private:
   Point *myUpperLeft;
   Point *myLowerRight;
```

```
};
```

Σημείωση: Ένα ορθογώνιο ορίζεται από τα δυο σημεία που ανήκουν στην ίδια διαγώνιο (π.χ. το άνω αριστερά και το κάτω δεξιά σημείο).
Η συνάρτηση "print" της "Rectangle" τυπώνει πληροφορίες στην παρακάτω μορφή:
"**Rectangle : [0, 0]-[20, 30] has width=20, height=30**"

- Η τάξη Circle μέσω της οποίας μπορούμε να κατασκευάσουμε αντικείμενα κύκλους έχει τη μορφή:

```
class Circle : public Shape, public Point {
public:
   // .......

private:
   double radius;
};
```

- η τάξη Point έχει τη μορφή:

```
class Point {
public:
   // .......

   void setX( int );
   int getX() const;

   void setY( int );
   int getY() const;

   void print() const;
private:
   int x;
   int y;
};
```

I) Να συμπληρώσετε τον παραπάνω κώδικα ώστε το ακόλουθο πρόγραμμα να δουλεύει.
II) Ποια είναι η σειρά κλήσης των δομητών και αποδομητών για α) αντικείμενο circle, β) αντικείμενο Rectangle.

```
int main()
{
   Point * pUL = new Point(0,0);
   Point * pLR = new Point(20,30);

   cout << "myRectangle(pUL,pLR)\n";
   Rectangle myRectangle(pUL, pLR);
   myRectangle.print();

   Point p1(10,12);
   Point p2(21,24);
   cout << "Rectangle0(p1,p2)\n";
   Rectangle Rectangle0(p1, p2);
   Rectangle0.print();
```

```
    cout << "Rectangle1(10,15,25,40)\n";
    Rectangle Rectangle1(10,15,25,40);
    Rectangle1.print();

    cout << "Rectangle2(Rectangle1)\n";
    Rectangle Rectangle2(Rectangle1);
    Rectangle2.print();

    cout << "Rectangle1 = myRectangle\n";
    Rectangle1 = myRectangle;
    Rectangle1.print();

////////////////////////////////////////////////////////////
    cout << "\n\n";

    Shape* shapes [ 4 ];
    shapes [0] = new Circle (5,10, 2.0);
    shapes [1] = new Rectangle(10,15,25,40);
    shapes [2] = new Circle (7,12, 4.0);
    shapes [3] = new Rectangle(pUL, pLR);
    double total_area = 0;
    double total_perifereia = 0;

    for (int i=0; i < 4; i++)
         total_area += shapes[i]->Area();
     cout << "Total area = " << total_area << endl;

     for (int i=0; i < 4; i++)
         total_perifereia += shapes[i]->Perifereia();
     cout << "Total perifereia = " << total_perifereia << endl;

     for (int i=0; i < 4; i++) {
         shapes[i]->print();
         cout << endl;
     }

    for (int i=0; i < 4; i++)
        delete shapes [i];

    delete pUL;
    delete pLR;
    cout << "\n\n";

    return 0;
}
```

2) Δίνεται το παρακάτω πρόγραμμα. Εξηγείστε τα αποτελέσματα που δίνει.

```
#include <iostream>
using std::cout;
using std::endl;

     class X {
     public:
         X() { }
         X( int xx ) : x( xx ) { }
         int getx() const { return x; }
     private:
         int x;
     };

     class Y : virtual public X {
     public:
         Y( int xx ) : X( xx ) { }
     };

     class Z : public Y {
     public:
         Z( int xx ) : Y( xx ) { }
     };

     int main()
     {
          Z zobj( 100 );
          cout << zobj.getx() << endl;
          return 0;
     }
```

Η υλοποίηση των ασκήσεων είναι υποχρεωτική!. Οι φοιτητές οφείλουν στο επόμενο εργαστήριο (18/12/2006) να έχουν τον κώδικα των ασκήσεων στον υπολογιστή που ασκούνται πάνω στον οποίο θα εξεταστούν.