**Polymorphism**

**Pointers to base class**

One of the greater advantages of deriving classes is that **a pointer to a derived class is typecompatible with a pointer to its base class**. This section is fully dedicated to taking advantage of this powerful C++ feature. For example, we are going to rewrite our program about the rectangle and the

triangle of the previous section considering this property:

*// pointers to base class*

#include <iostream.h>

class CPolygon {

protected:

int width, height;

public:

void set\_values (int a, int b)

{ width=a; height=b; }

};

class CRectangle: public CPolygon {

public:

int area (void)

{ return (width \* height); }

};

class CTriangle: public CPolygon {

public:

int area (void)

{ return (width \* height / 2); }

};

int main () {

CRectangle rect;

CTriangle trgl;

CPolygon \* ppoly1 = &rect;

CPolygon \* ppoly2 = &trgl;

ppoly1->set\_values (4,5);

ppoly2->set\_values (4,5);

cout << rect.area() << endl;

cout << trgl.area() << endl;

return 0;

}

Ouput: **20**

**10**