**Pointers to classes**

CRectangle \* prect;

*// pointer to classes example*

#include <iostream.h>

class CRectangle {

int width, height;

public:

void set\_values (int, int);

int area (void) {return (width \* height);}

};

void CRectangle::set\_values (int a, int b) {

width = a;

height = b;

}

int main () {

CRectangle a, \*b, \*c;

CRectangle \* d = new CRectangle[2];

b= new CRectangle;

c= &a;

a.set\_values (1,2);

b->set\_values (3,4);

d->set\_values (5,6);

d[1].set\_values (7,8);

cout << "a area: " << a.area() << endl;

cout << "\*b area: " << b->area() << endl;

cout << "\*c area: " << c->area() << endl;

cout << "d[0] area: " << d[0].area() << endl;

cout << "d[1] area: " << d[1].area() << endl;

return 0;

}

Output:

**a area: 2**

**\*b area: 12**

**\*c area: 2**

**d[0] area:**

**30**

**d[1] area:**

**56**

Next you have a summary on how can you read some pointer and class operators (**\*,**

**&, ., ->, [ ]**) that appear in the previous example:

**\*x** *can be read:* pointed by **x**

**&x** *can be read:* address of **x**

**x.y** *can be read:* member **y** of object **x**

**(\*x).y** *can be read:* member **y** of object pointed by **x**

**x->y** *can be read:* member **y** of object pointed by **x**

(equivalent to the previous one)

**x[0]** *can be read:* first object pointed by **x**

**x[1]** *can be read:* second object pointed by **x**

**x[n]** *can be read:* (n+1)th object pointed by **x**