**Overloading operators**

Here is a list of all the operators that can be overloaded:

+ - \* / = < > += -= \*= /

= << >>

<<= >>= == != <= >= ++ -- % &

^ ! |

~ &= ^= |= && || %= [] () new

Delete

To overload an operator we only need to write a class member function whose name is

**operator** followed by the operator sign that we want to overload, following this

prototype:

*type* **operator** *sign* **(***parameters***);**

*// vectors: overloading operators example*

#include <iostream.h>

class CVector {

public:

int x,y;

CVector () {};

CVector (int,int);

CVector operator + (CVector);

};

CVector::CVector (int a, int b) {

x = a;

y = b;

}

CVector CVector::operator+ (CVector param) {

CVector temp;

temp.x = x + param.x;

temp.y = y + param.y;

return (temp);

}

int main () {

CVector a (3,1);

CVector b (1,2);

CVector c;

c = a + b;

cout << c.x << "," << c.y;

return 0;

}

Output:

**4,3**