

Operators

Use the Circle class given below as a starting point. You will need to implement the constructor and destructor and any other methods that you add.

```
class Circle
{
public:

    Circle( int radius,
            int xCoord = 0, int yCoord = 0,
            const char * name = NULL );

    ~Circle();

private:
    int    mXCoord;
    int    mYCoord;
    int    mRadius;
    char * mName;    // this stores the name of the circle
};
```

- Add the following operators to the Circle class:
 - *Operator +*: This should add two circles, which means create a new circle that has a radius which is the sum of the radius of the operands.
 - **Example:** `c3 = c1 + c2;` // c3 is a circle whose
// radius is
// c1.radius + c2.radius
 - *Operator ++ (both post and pre increment)*: This will increment the radius by one, and return the Circle object (by reference in case of pre increment, and by value in case of post increment).
 - **Example:** `c1 ++` or `++c1` increments the radius of circle c1 by 1 unit.
Let's say c1 and c2 both have a radius of 10 units.
`c3 = ++ c1;` // c3 has radius 11; c1 is also radius 11
`c3 = c2 ++;` // c3 has radius 10; c2 is radius 11
 - *Operator << (output operator)*: This should be able to output a circle to an output stream (cout or a file stream).
 - **Example:** `cout << c1 << endl;`
In the output operator, you can print some information about the circle like radius, coordinates, etc. One thing that you should print is the name of the circle.

Operators

- *Assignment operator:* Do you think you need to add your own assignment operator, or would the compiler generated assignment operator be sufficient?

You should be able to execute the following code sample on your Circle class, in addition to any other tests that you write for your class.

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

const int kMaxNameSize = 128;

void CreateTwoCircles()
{
    char * name = new char [kMaxNameSize];
    const char * kName1 = "Cir1";
    const char * kName2 = " CircleNumber2";

    strncpy_s( name, kMaxNameSize, kName1, _TRUNCATE );
    Circle c1( 2, 1,1, name );

    strncpy_s( name, kMaxNameSize, kName2, _TRUNCATE );
    Circle c2( 5, 1,1, name );

    delete [] name; // delete first before printing the circle below

    // Print the two below one after another
    cout << "c1 is " << c1 << endl; // Name printed: Cir1
    cout << "c2 is " << c2 << endl; // Name printed: CircleNumber2
}

void DoCircleAssignment( )
{
    char * name = new char [kMaxNameSize];
    const char * kName1 = "Cir1";
    const char * kName2 = "CircleNumber2";

    strncpy_s( name, kMaxNameSize, kName1, _TRUNCATE );
    Circle c1( 1, 0, 0, name );
    cout << "c1 is " << c1 << endl;

    strncpy_s( name, kMaxNameSize, kName2, _TRUNCATE );
    Circle c2( 2, 1, 1, name );

    delete [] name; // delete first before printing the circle below

    cout << "Before assignment, c2 is " << c2 << endl;

    c2 = c1;

    cout << "After assignment, c2 is " << c2 << endl;
```

Operators

```
    }  
  
    void CircleTest()  
    {  
        CreateTwoCircles( );  
  
        DoCircleAssignment( );  
    }
```