**Lab#4**

**Objective:** To illustrate the concepts of operator overloading through hands-on practice on binary operators overloading.

**Operator Overloading**:  is a specific case of [polymorphism](https://en.wikipedia.org/wiki/Polymorphism_(computer_science)), where different [operators](https://en.wikipedia.org/wiki/Operator_(programming)) have different implementations depending on their arguments. C++ allows you to specify more than one definition for a function name or an operator in the same scope, which is called function overloading and operator overloading respectively.

An overloaded declaration is a declaration that had been declared with the same name as a previously declared declaration in the same scope, except that both declarations have different arguments and obviously different definition (implementation).

When an overloaded function or operator is called, the compiler determines the most appropriate definition to use by comparing the argument types.

**Lab Tasks:**

* Write codes, compile and run.
* Write output of code in below given boxes.
* Write another code (program) that demonstrates concepts of operator overloading.
* Write the output of 2nd program.

**Program 1 (Code):**

#include<iostream>

using namespace std;

class Box {

public:

doublegetVolume(void){

return length \* breadth \* height;

}

voidsetLength (double Len) {

length=len;

}

voidsetBreadth(doublebre) {

breadth=bra;

}

voidsetHeight(doublehei) {

height=heir;

}

// Overload + operator to add two Box objects.

Box operator+(constBox& b){

Box box;

box. length=this->length +b. length;

box. Breadth=this->breadth +breadth;

box. Height=this->height +height;

return box;

}

private:

double length;// Length of a box

double breadth;// Breadth of a box

double height;// Height of a box

};

// Main function for the program

int main () {

BoxBox1;// Declare Box1 of type Box

BoxBox2;// Declare Box2 of type Box

BoxBox3;// Declare Box3 of type Box

double volume =0.0;// Store the volume of a box here

// box 1 specification

Box1.setLength(6.0);

Box1.setBreadth(7.0);

Box1.setHeight(5.0)

// box 2 specification

Box2.setLength(12.0);

Box2.setBreadth(13.0);

Box2.setHeight(10.0);

// volume of box 1

volume=Box1.getVolume();

cout<<"Volume of Box1: "<< volume <<endl;

// volume of box 2

volume=Box2.getVolume();

cout<<"Volume of Box2: "<< volume <<endl;

// Add two object as follows:

Box3=Box1+Box2;

// volume of box 3

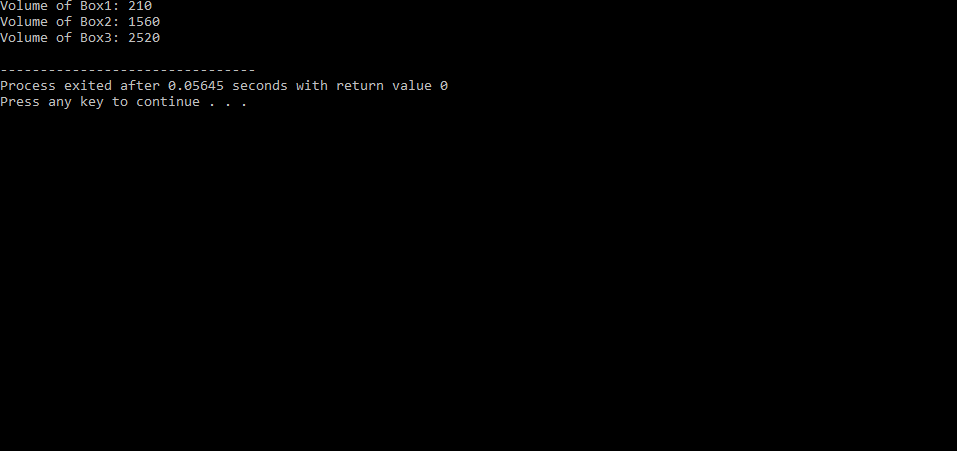
volume=Box3.getVolume();

cout<<"Volume of Box3: "<< volume <<endl;

return0;

}

**OUTPUT:**

****

#include<iostream>

using namespace std;

class Count

{

private:

int n;

public:

Count(){

n=0;

}

void show()

{

cout<<"n="<<n<<endl;

}

Count operator ++ ()//prefix

{

Count temp;

//n++

n=n+1;

tampon=n;

return temp;

}

Count operator ++(int)//postfix increment

{

Count temp;

//n=n+1;

n++;

tamp. n=n++;

return temp;

}

};

int main ()

{

Count x,y,z;

x.show();

z=++x;

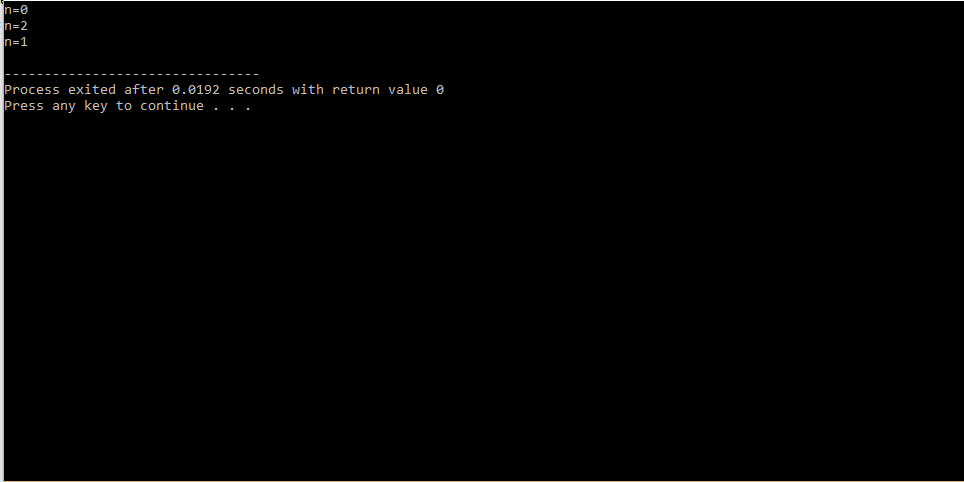
y=x++;

y.show();

z.show();

}

**OUTPUT:**

****