**/\*27. Create a clas box. Include following methods in it**

**a)calculate surface area**

**b)calculate volume**

**c)increment,overload ++operator**

**d)decrement,overload –operator**

**e) overload operator ==**

**f)overload assignment operator**

**g) check if it is a cube or cuboid \*/**

using namespace std;

#include<iostream>

class box

{

int length,breadth,height;

public:

box();

box(int x,int y,int z);

void surface\_area();

void volume();

box operator ++();

box operator --();

void operator ==(const box &b);

void operator =(const box &b);

void check();

void print();

};

box:: box()

{

length=0;

breadth=0;

height=0;

}

box::box(int x,int y,int z)

{

length=x;

breadth=y;

height=z;

}

box box:: operator ++()

{

length=length+1;

breadth=breadth+1;

height=height+1;

return \*this;

}

box box:: operator --()

{

length=length-1;

breadth=breadth-1;

height=height-1;

return \*this;

}

void box:: operator==(const box &b)

{

if(length==b.length && breadth==b.breadth && height==b.height)

cout<<"\nboxes are equal";

else

cout<<"\nboxes are not equal";

}

void box :: operator=(const box &b)

{

length=b.length;

breadth=b.breadth;

height=b.height;

}

void box:: print()

{cout<<"\n";

cout<<"Length:- "<<length<<" \nbreadth:- "<<breadth<<"\nheight:- "<<height;

}

void box:: surface\_area()

{

int l=length;

int b=breadth;

int h=height;

int s=2\*((l\*b)+(b\*h)+(h\*l));

cout<<"\nsurface area is:- "<<s;

}

void box:: volume()

{

int v=length\*breadth\*height;

cout<<"\n volume is:- "<<v;

}

void box::check()

{

if(length==breadth && breadth==height)

cout<<"\nit is a cube";

else

cout<<"\nit is a cuboid";

}

int main()

{

box b1(2,3,4),b2,b3,b4(5,5,5);

b1.print();

b1.surface\_area();

b1.volume();

cout<<"\nafter prefix ++ operator";

b2=++b1;

b2.print();

cout<<"\n after prefix-- operator";

b2=--b1;

b2.print();

cout<<"\n if the dimensions of 1st box are:-";

b1.print();

cout<<"\nand the dimensions of second box are:- ";

b4.print();

b1==b4;

cout<<"\nafter assignment operator";

b2=b4;

b2.print();

cout<<"\nif";

b1.print();

cout<<"\nthen";

b1.check();

cout<<"\nif";

b4.print();

cout<<"\nthen";

b4.check();

system("pause");

return 0;

}

Output:-

