```
Script started on Thu 19 May 2016 03:44:07 PM CDT
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ pwd
/home/students/g_butler4/csc122/ops
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ cat point.cpp\033[K\033[K\033
[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\007\007\007\007\007\007\007\007
\007\007\007\007\007\007\007\007\007cat point\033[K_info.cpp\033[K\033[K\033[Ktxt\033[K\033
[Kxt
Name: Gary Butler
For: CSC122-002
Project:"Operate on this!"(lab)
Levels Attempted:2
Description:This program adds overloaded operators to a given point class
to output,input, add, subtract, determine midpoint, and check for equality\033]0;g_but
ler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ cat point_info.txt\033[K\033[K\033[K\033[K
\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\033[K\007\007\007
\007\007\007\007\007\007\007\007\007\007\007\007\007cat point.cpp \033[Kpoint.t\033[Kp
\033[Kh
#include <string>
#include <iostream>
#include "point.h"

using namespace std;

int main()
{

        Point A;
        Point B;
        Point C;

        A.set_x(5);
        A.set_y(10);

        B.set_x(25);
        B.set_y(25);

        cout << "Point A: " << A;
                cout<<"\n";
                cout << "Point B: " << B;
                cout<< "\n";

                cout << "Midpoint (A/B): " << (A / B);
                cout<< "\n";
                cout << "A+B: " << (A + B);
                cout<< "\n";
                cout << "A-B: " << (A - B);
                cout<< "\n";
                cout << "B-A: " << (B - A);
                cout<< "\n";
        cout << "B=A?: " << (A == B?"True":"False") << "\n";
        cout << "B!=C? " << (A != B ? "True" : "False") << "\n";

        B = A;
        cout << "B is now equal to A:" << B;

        if (C.get_y() == 0.0)
        {
                cout << "\nInput Point C in (x,y) format\n";
                cin >> C;
                while ((C.get_x() == 0.0) || (C.get_y() == 0.0))
                {
                        cout << "\nTry again: ";
                        cin.clear();
                        cin >> C;
                }
                cout << "\n";
        }
        cout << "Your Point C: " << C;
        cout<< "\n";
        cout << "Midpoint (B/C): " << (B / C);
        cout<< "\n";
        cout << "B+C: " << (B + C);
        cout<< "\n";
        cout << "B-C: " << (B - C);
        cout<< "\n";
        cout << "C-B: " << (C - B);
        cout<< "\n";
        cout << "B=C?: " << (B == C ? "True" : "False") << "\n";
        cout << "B!=C? " << (B != C ? "True" : "False") << "\n";


        return 0;
}#ifndef POINT_H
#define POINT_H
#include <cmath>
#include <iostream>

// A 2D point class

class Point
{
        double x, // x coordinate of point
        y; // y coordinate of point

public:
        Point(void)
                :x(0.0), y(0.0){}

        Point(double new_x, double new_y)
                :x(new_x), y(new_y){}

        std::ostream & Output(std::ostream & out) const  // output this point
        {
                out << "( " << x << ", " << y << " )";//(x,y)
                return out;
        }
        std::istream & Input(std::istream & in)    // input this point
        {
                                in.ignore();
                                in>>x;
                                in.ignore();
                                in >> y;
                                in.ignore();
                                        return in;
        }

        Point distance(const Point & b)   // distance between this point and other
        {
                Point temp;
                temp.x = (x + b.x) / 2;
                temp.y = (y + b.x) / 2;
                return temp;
        }
        double get_x(void) { return x; }
        double get_y(void) { return y; }

        void set_x(double new_x) { x = new_x; };
        void set_y(double new_y) { y = new_y; };
```

```
	Point flip_x(void)
	{
		return Point(x, -y);
	}

	Point flip_y(void)
	{
		return Point(-x, y);
	}

	Point shift_x(double move_by)
	{
		return Point(x + move_by, y);
	}

	Point shift_y(double move_by)
	{
		return Point(x, y + move_by);
	}

	Point operator-(const Point & b)
	{
		Point temp;
		temp.x = x - b.x;
		temp.y = y - b.y;
		return temp;
	}

	Point operator+(const Point & b)
	{
		Point temp;
		temp.x =(x + b.x);
		temp.y =(y + b.y);
		return temp;
	}

	Point & operator=(const Point & b)
	{

		x = b.x;
		y = b.y;
		return *this;
	}

	bool operator==(const Point & b)
	{
		if (((x) == b.x) && ((y) == b.y))
			return true;
		else
			return false;
	}

	bool operator!=(const Point & b)
	{
		if (((x) != b.x) || ((y) != b.y))
			return true;
		else
			return false;
	}

	Point operator/(const Point & b)
	{
		return distance(b);
	}

		}

		friend std::istream& operator>>(std::istream & in,Point & b);
		friend std::ostream& operator<<(std::ostream & out,const Point & b);
};

inline std::istream& operator>>(std::istream & in,Point & b)//input
{
	b.Input(in);
	return (in);
}

inline std::ostream& operator<<(std::ostream & out,const Point & b)//output
{
	b.Output(out);
	return out;
}

#endif
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ ./point.out
Point A: ( 5, 10 )
Point B: ( 25, 25 )
Midpoint (A/B): ( 15, 17.5 )
A+B: ( 30, 35 )
A-B: ( -20, -15 )
B-A: ( 20, 15 )
B=A?: False
B!=C? True
B is now equal to A:( 5, 10 )
Input Point C in (x,y) format
(1,1^H

Your Point C: ( 1, 1 )
Midpoint (B/C): ( 3, 5.5 )
B+C: ( 6, 11 )
B-C: ( 4, 9 )
C-B: ( -4, -9 )
B=C?: False
B!=C? True
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ vi point.h point.cpp point.out
Point A: ( 5, 10 )
Point B: ( 25, 25 )
Midpoint (A/B): ( 15, 17.5 )
A+B: ( 30, 35 )
A-B: ( -20, -15 )
B-A: ( 20, 15 )
B=A?: False
B!=C? True
B is now equal to A:( 5, 10 )
Input Point C in (x,y) format
(5,5)

Your Point C: ( 5, 5 )
Midpoint (B/C): ( 5, 7.5 )
B+C: ( 10, 15 )
B-C: ( 0, 5 )
C-B: ( 0, -5 )
B=C?: False
B!=C? True
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ ./point.out
Point A: ( 5, 10 )
Point B: ( 25, 25 )
Midpoint (A/B): ( 15, 17.5 )
A+B: ( 30, 35 )
```

```
A-B: ( -20, -15 )
B-A: ( 20, 15 )
B=A?: False
B!=C? True
B is now equal to A:( 5, 10 )
Input Point C in (x,y) format
(5,10)

Your Point C: ( 5, 10 )
Midpoint (B/C): ( 5, 7.5 )
B+C: ( 10, 20 )
B-C: ( 0, 0 )
C-B: ( 0, 0 )
B=C?: True
B!=C? False
\033]0;g_butler4@mars:~/csc122/ops\007[g_butler4@mars ops]$ cat point_tpq.txt
Q's and A's:
1.All but the input and output operators are members.
2.Anyhting that doesnt change the original values should be const, or anything that is
 void.
3.Equality returns a bool, input and output return their respective streams, and assig
nment returns a new object.
4.No, it would be better off as a funtion to find midpoint because the / operator can
be used for its intened purpose.
5.Less than or greater than is hard to determine when it comes to points. If it were t
o be used, it would only be arithmetic and wouldnt serve a greater purpose in the prog
ram.
6.Operators are finite, its better to call flip and shift by functions rather than ope
rators.
7.No, the original methods are still useful when called directly.\033]0;g_butler4@mars
:~/csc122/ops\007[g_butler4@mars ops]$ exit
exit

Script done on Thu 19 May 2016 03:45:31 PM CDT
```