alpha.h

```
#ifndef ALPHA_H
#define ALPHA_H

#include <iostream>
using namespace std;
class alpha
{
private:
	int data;
public:
	alpha();
	alpha(int n);
	~alpha() {};
	void display();
	alpha(alpha& a);
	alpha& operator=(alpha& a);
	void setData(int n); //testing purpose
};
#endif ALPHA_H
```

alpha.cpp

```
#include "alpha.h"

alpha::alpha() {
	data = 0;
}

alpha::alpha(int n) {
	data = n;
}

void alpha::display() {
	cout << data;
}

alpha::alpha(alpha& a) {
	data = a.data;
}

alpha& alpha::operator=(alpha& a) {
	data = a.data;
	return *this;
}
void alpha::setData(int n) {
	data = n;
}
```

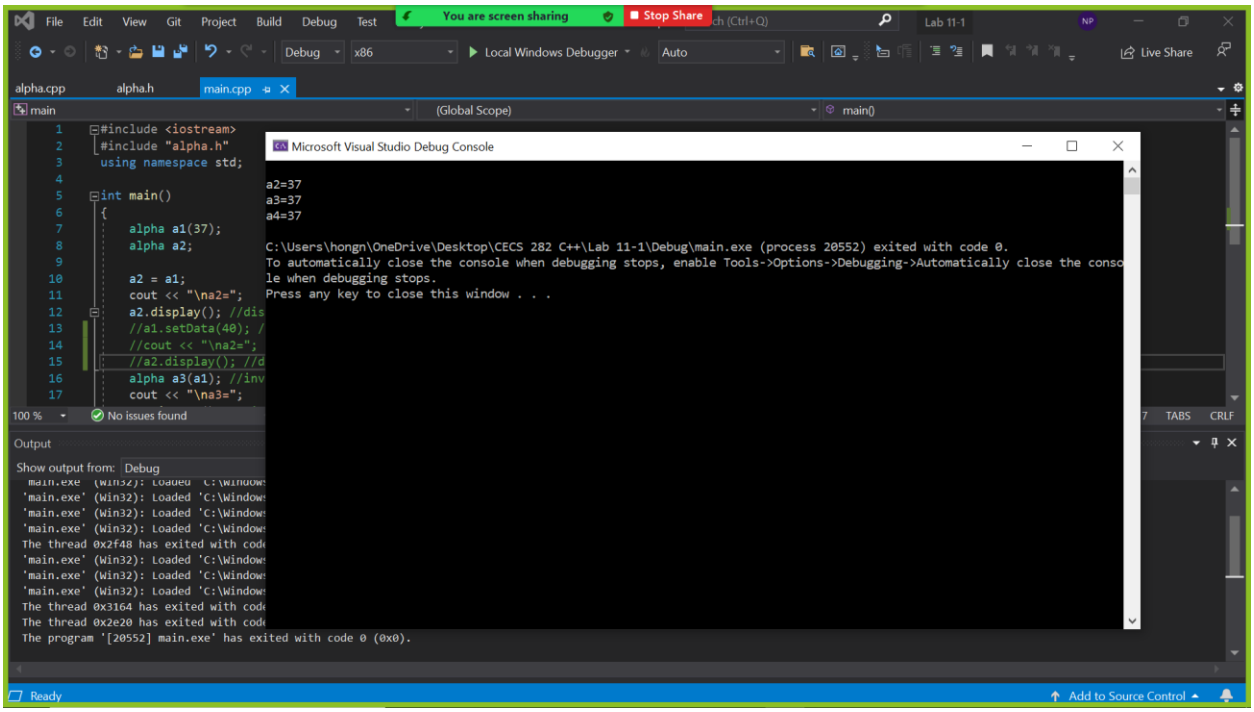
main.cpp

```
#include <iostream>
#include "alpha.h"
```

```cpp
using namespace std;

int main()
{
	alpha a1(37);
	alpha a2;

	a2 = a1;
	cout << "\na2=";
	a2.display(); //display a2
	//a1.setData(40); //testing
	//cout << "\na2=";
	//a2.display(); //display a2 again
	alpha a3(a1); //invoke copy constructor
	cout << "\na3=";
	a3.display(); //display a3
	alpha a4 = a1;
	cout << "\na4=";
	a4.display();
	cout << endl;
	return 0;
}
```



Demonstrated at 11:14 am on 10/26/2021