```cpp
// Cherno Copy Constructor Example

#include <iostream>
#include <string>
#include <cstring>
class String
{
private:
	char* m_Buffer;
	unsigned int m_Size;
public:
	// Const char pointer called string
	String (const char* string)
	{
		m_Size = strlen(string);
		// the size of the string + null terminating character
		m_Buffer = new char[m_Size + 1];
		memcpy(m_Buffer, string, m_Size);
	}
	// Copy Constructor
	String(const String& rhs)
		:m_Size{rhs.m_Size}
	{
		std::cout << "Copied String!" << std::endl;
		m_Buffer = new char[m_Size + 1];
		memcpy(m_Buffer, rhs.m_Buffer, m_Size  + 1);
	}

	friend std::ostream & operator<<(std::ostream& stream, const String& string);

	~String()
	{
		delete[] m_Buffer;

	}
};

// Left Shift Operator overload
std::ostream & operator<<(std::ostream& stream, const String& string)
{
	stream << string.m_Buffer;
	return stream;

}

void PrintString(const String& string)
{
	std::cout << string << std::endl;
}

int main()
{
	// one copy of string is made and put into second (DEEP COPY)
	String string = "Cherno";
	String second = string;
	//std::cout << string << std::endl;
	//std::cout << second << std::endl;
	PrintString(string);
	PrintString(second);

}
```