```
// Author: Max Nelson

// Assignment: LAB 12

// Instructor: David Wagstaff

// Class: CS1410 001

// Date Written: 11/9/2016

// Description: pseudo code for the overloaded assignment operator. Well and
the code tbh.

// -----------

//I declare that the following source code was written solely by me.

//I understand that copying any source code, in whole or in part,

// constitutes cheating, and that I will receive a zero on this project

// if I am found in violation of this policy.

//class for vector object


                        // PSEUDO CODE //

// 1. test for self assignment

// 2. just return it bc it's the same thing. Saves time yo.

// 3. delete theArray of MyVector object getting assigned

// 4. gather the vectorCapacity of assigning MyVector object

// 5. dynamically allocate the new capacity, taken from assinging MyVector
theArray size

// 6. copy theArray data from the assigning MyVector Object theArray and Put
this old theArray data in the current object getting assigned's theArray
member

// 7. the current object being assigned has now succesfully replaced its own
theArray field with the assigning MyVector objects theArray data "ie: the
object getting passed in"

// 8. return the new theArray size, son. deal with it.
```

```
                                    // PSEUDO CODE + CODE //


MyVector& MyVector::operator=(const MyVector& assigning)

{

// test for self assignment

// just return it bc it's the same thing. Saves time yo.

if (this == &assigning)

return *this;

//delete theArray of MyVector object getting assigned

delete[] this->theArray;

//gather the vectorCapacity of assigning MyVector object

vectorCapacity = assigning.vectorCapacity;

//dynamically allocate the new capacity, taken from assinging MyVector
theArray size

this->theArray = new int[vectorCapacity];

//copy theArray data from the assigning MyVector Object theArray

//and Put this old theArray data

//in the current object getting assigned's theArray member

for (int i = 0; i < vectorSize; i++)

{

this->theArray[i] = assigning.theArray[i];

}

//the current object being assigned has now succesfully replaced its

//own theArray field with the assigning MyVector objects theArray data "ie:
the object getting passed in"

//return the new theArray size, son. deal with it.

return *this;

}
```