**Name: Owen Godwin**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

Write down your answers according to their numbers on your lab instruction sheet.

**Part 1:**

1. What do you see?
   1. The issue I see after running the code is that the new values of n2 does not follow what the logic should say it should be.
2. Is this a problem and why?
   1. The problem is that the code is supposed to delete the last five values of n1 and add the four values of 5, 10, 15, and 20. This is an issue because there is a problem in the coed that needs to be fixed before it causes more issues later on.
3. What caused this to happen?
   1. This happens due to an error in the add function, in which the add function adds the values meant for the array n2 to the array n1.
4. What do you see?
   1. I see the code acting as the logic of the main function being followed and the n2 value having its own values separate form n1.
5. Is it different from what you saw before?
   1. It is different from what was seen beforehand as the new values being displayed are not being altered by the changes in the n2 values as what was intended from the code initially.
6. What caused this to happen – why is it different?
   1. What caused this change was overloading the ‘=’ operator, actually copying the values in the array of the n1 value into the n2 value. It is different because initially there were issues in merely saying that the two arrays are equal to one another.

**Part 2:**

1. Write down the five addresses that are output and the byte count output
   1. 0x7fffee9f0e40
   2. 0x7fffee9f1170
   3. 0x7fffee9f14a0
   4. 0x7fffee9f17d0
   5. 0x7fffee9f1b00
   6. Total bytes allocated = 4200
2. Can you tell how many bytes they are apart? Write down your best estimate. (Remember that these addresses are appearing in hexadecimal).
   1. I assume the addresses are approximately 800 bytes apart, given the total allocated for five addresses is 4200 bytes and 200 were already in use form step one.
3. What addresses did you see this time and how much were they apart from each other?
   1. The addresses I saw this time were all the same “0x7ffff5640e40”, with the total bytes being a total of 200. Those of which being the original values for the initial n1 and n2 values, meaning the addresses were all the same location.
4. What is the byte\_count?
   1. The byte\_count was at a value of 200, being the original values for the initial n1 and n2 values.
5. Explain why adding the destructor resulted in this different behavior
   1. Adding the destructor resulted in the byte\_count and addresses for the arrays being deleted nearly as soon as they were populated, resulting in it being empty after the code was finished running.