Short Return to Binary Files

This short assignment is to further explore how binary files, and in particular, the read and write operations, work.

Before, you defined a class with a int and a double data members, wrote an object of the class to a binary file, and read it back to a new object to see how by using read and write we can copy the whole object to the binary file and back. This time you are going to try the same exercise but see how it works when an object has string and character arrays as data member. We are also going to have 2 programs, one for writing into the file, and one for reading out.

First, define the class charStudent with 2 public data members: ID and name, where ID is an int and name is an array of characters of size 40 (will hold up to 39 characters plus the special ‘\0’ character signifying the end of the c-type string.) Just like before, declare an object of the class with some values, and write the object into a binary file. The second program should read from the binary file to an object and print out the values to see that the transfer worked correctly.

You can also take a look at the file and see what is looks like.

If you have not worked with character arrays very much, please read the lecture note “C-strings versus strings” to see how to work with them.

Next, define a class stringStudent with the same 2 public data members, ID and name, but make name be data type string. Try the same exercise as before, of writing the first object to a binary file in one program, and reading from the file to an object in a second program, and printing out the values.

Did the transfer work?

If it worked, try again, but enter a very long name with at least 30 characters. Did it still work? If not, why not? You can try looking at the contents of the file, once when you have a short name, and once when you have a long name.

Discuss with your instructor about what is going on.