Student Name: Bair Suimaliev  
Student ID: 159350198  
Student Email: [bsuimaliev@myseneca.ca](mailto:bsuimaliev@myseneca.ca)

In this workshop first, I learned about what is dynamic memory, and how to allocate and deallocate it for an array. The reason why the dynamic memory is needed is to create a flexible array that will have the number of elements depending on the value of another variable. Nevertheless, allocating dynamic memory for an array takes up a lot of space, therefore a dynamic memory should be deallocated for an array.

The second concept I learned about while doing this workshop is overload functions. These functions have the same name in the same scope; however, they are differentiated by the number and type of parameters each function accepts or does not. In the workshop, the overloaded functions were read, load, and display functions. For read functions I used different types of arguments accepted by the functions to read information of different type from the file. Load functions have passed objects of different types. In DIY, one passed string of constant format, which was the name of the file passed to the load function, second passed a structure to another load function. And the display functions that printed the information read from the file to the output. First display function accepted no arguments, while the second accepted structure.

The last concept learned in the workshop 2 is references. The references are aliases for variables that are used to make the variable passed to the function more readable. In this workshop I passed a reference to a structure to modify it.

In C++ there are two types of parameters passed to functions: in and out. Variables passed as out parameter are usually passed by a non-constant pointer or reference. And variables passed as in parameter cannot be modified, so it is passed by value, or by constant pointer or constant reference. In this workshop, the example of a function accepting an in parameter would be load function that accepted a constant pointer to a string that is the name of the file to be read.