# CSI 2372 – Lab Task 8

## Abdorrahim Bahrami

## maps and sets in C++

uOttawa

Your task in this lab is to get yourself familiarized with the two important and powerful libraries in C++. In the test, we defined a dictionary of English words to French. Your task in this lab is to rewrite that class using maps and sets in STL. The difference here is that the dictionary can map anything to a set of something else. Namely, you should use templates for these. Make sure you have C++ installed, and you are familiar with the header files, and coding files. If you need help, ask your TA to help you with this.

Then, you should do the following programming task. Each programming task in the lab is a design based on the subjects you learned during lectures. There is a test code that you can use to test your design. If you have questions, ask your TAs.

Your task is to create a mapping that maps any type to a set of any other type. The one in the exam was a string to a set of string. If you use map <type1, set <type2> >, you will be able to design this class easily. Check the solution of the fourth question in the midterm.

| **Class Mapping** | |
|---|---|
| **Method** | **Description** |
| Mapping | The default constructor that creates an empty mapping |
| Mapping | The copy constructor |
| ~ Mapping | The destructor (you might not need it when you use STL) |
| add_item | Takes two arguments, the first one is of type 1 and the second one is of type 2, it adds the second given argument to the set of type 2. If the given first argument does not exist, it will be added to the mapping |
| remove_item | Takes two arguments, the first one is of type 1 and the second one is of type 2, it removes the second given argument from the set of the given first argument. Ignore, if it does not exist in the set. |
| clear_item | Takes one argument of type1 and remove it completely from the mapping (both the argument and its set) |
| find_item | Takes an argument of type 1 and returns its set |
| operator [ ] | Gets an element of type 1 as index and returns its set (it can be reference output or regular) |
| operator = | For assigning a mapping to another mapping |
| operator << | For printing all elements of the mapping in the following format Example: [ “Mike”: {19, 35, 60}, “Nadia”, {15, 17, 80} ] This is a mapping of string to a set of integers |