# Workflow

| | |
|---|---|
| **Name:** | 04.02 |
| **Points:** | 3 pts |
| **Deadline:** | 03/15 |
| **Prerequisite(s):** | none |

## Main

1. Create a header file named **w0402.h** that defines a generic class named *Multiset* that inherits the interface *SetInterface*. It will represent a multiset with an infinite capacity, where a multiset is an unordered collection of objects. The class must contain

   □ a private generic *Node* pointer field named *data*.

   □ its special member functions that should be public and the default value of *data* is null.

   □ a public overridden `Insert()` method. It adds the parameter to the multiset.

   □ a public overridden `Remove()` method. It removes an instance of the parameter from the multiset if the multiset is not empty.

   □ a public overridden `IsEmpty()` method. It returns true if the multiset does not have any members; otherwise, it returns false.

   □ a public overridden `Contains()` method. It returns true if the parameter is in the multiset; otherwise, it returns false.

   □ a public int constant method named `Size()` that takes no parameters. It returns the size of the multiset.

   □ a public int constant method named `Count()` that takes a constant generic reference parameter. It returns the number of occurrences of the parameter in the multiset.

   □ a public string constant method named `ToString()` that takes no parameters. It returns a string of the elements of the multiset all enclosed in curly braces with each element separated by a comma.

## Test

2. Create a cpp file named **main.cpp** that creates a *Multiset* object and test each of its methods. The outputs of the methods that are not void must be displayed.