# Workflow

| | |
|---|---|
| **Name:** | 03.01 |
| **Points:** | 3 pts |
| **Deadline:** | 03/08 |
| **Prerequisite(s):** | none |

## Main

1. Create a header file named **w0301.h** that defines a generic class named *StaticSet* that inherits the interface *SetInterface*. It will represent a set with a finite capacity, where a set is an unordered collection of distinct objects. The class must contain

   □ a private generic *Array* field named *data*.

   □ a private int field named *size*.

   □ its special member functions that should be public and the default size of *data* is 100 and the default value of *size* is 0.

   □ a public overloaded constructor that takes an int parameter. If the parameter is at least 2, the function makes the parameter the size of *data*; otherwise, it makes 100 the size of *data*. Furthermore, it assigns 0 to *size*.

   □ a public overridden `Insert()` method. It adds the parameter to the set if the set is not at capacity and the parameter is not already in the set.

   □ a public overridden `Remove()` method. It removes the parameter from the set if the set is not empty and the parameter is in the set.

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

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

   □ a public bool constant method named `IsFull()` that takes no parameters. It returns true if the set is at capacity; otherwise, it returns false.

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

## Test

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