

Objectives:

Review classes, separate compilations, arrays, and operator overloading.

Implement the Bag class discussed in class. The bag code can be found on the class's website. Separate the class into its header and its implementation files. Add the following functions:

- A constructor that takes an array of integers and a size.
- A function called "**at**" that returns the value at given index. return -9999 if the index is invalid.
- A **toString** function that returns the bag in the format (a single space separates items):
[1, 2, 3]
- A function to sort the elements of the bag called **sort**
- A function to remove duplicates called **removeDuplicates**
- A function to overload the == operator. It should return true if two bags contain the same values regardless of the order of the elements.

To test your program, you may run the provided tests using the provided Makefile. Make sure you use the same function names as outlined above.

```
make run_tests
```

Grading:

Programs that contain syntax errors will earn zero points.

Programs that use global variables, other than constants, will earn zero points.

Programs that do not use separate files will earn zero points.

35 Points:

- (5 points) for each of the required functions
- (5 points) separate compilation and implementation of already provided functions

5 points: Programming Style & documentation.

- All files must have your name, date, and a description of their content
- All functions must be documented in the header file using the style outlined in the coding style below

Follow the coding style outline on GitHub:

<https://github.com/nasseef/cs/blob/master/docs/coding-style.md>

Submission:

Submit a link to your repository on Blackboard before the deadline.