This homework gauges your ability to improve on a previous design, with minimal changes to the interface, while utilizing concepts of operator overloading and generic programming. Use concepts of good style and object-oriented design (as much as you know them) when you complete this assignment.

Your submission should have a `CMakeLists.txt` in the root of your `hw07-lastname` directory, from which all subdirectories containing code can be compiled. Failure to do this results in deductions for failures to compile/link, etc.

1 Templates, Shmemplates (100 points)

Improve the design of the `State` and `Input` vectors from `hw05`. In this portion of the assignment, create a templatized instance of a generic class that you define per the below requirements.

- Your template class should be called `vector` in the namespace `ece373`. Note that the template parameters for your vector class will necessarily be different from those of the `std::vector` class.

- Your vectors should have elements of type, with fixed length `n`.

- Elements of type must support the full set of operators provided below*. Elements must be able to take as their optional constructor argument an `int`.

- Other parameters, and operations, for your template are up to your design. Make sure these design decisions are justified!

- Your template class should provide at least the following class or global operator methods:
  - `=, ==, !=, +, +=, -, -=, [ ]`. Both binary, and unary, `operator-` should be provided.
  - In operations with scalar type values, the operators `*, /, *`, `/` can be used.
  - When operations between vectors of unequal size are performed, exceptions should be thrown, or the compile should fail. If you check for mismatched sizes, but it is impossible for the sizes to be mismatched based on strong typing (i.e., the compiler will fail), then you may lose up to 10 percent for this problem. Be careful in understanding whether you need to check for mismatched sizes or whether the compiler will do this for you.

You should instantiate tests to creates various instances of these template classes, perform operations on them, and compare outputs to known values. We suggest you make a class to perform these tests (similar to a JUnit class for Java). The following command will be executed to run your submission for this problem:

```bash
ctest -VV -R hw07-question1
```

The effectiveness and strengths of your tests are the sole basis for your grade of this problem.

--

*The `operator[]` is not among these members, as it provides access to individual type elements, and it not subsequently called on individual members.