> You shall submit a zipped, **and only zipped**, archive of your Exam 2 project. Do not change the files manually. **If the makefile does not work, you will not receive credit**.
> To prepare your project for submission, use
> `make archive`
> at the root of the project directory.

Rather than writing descriptions here, the problems are described in their respective headers, with test files provided for further clarification. As this is an exam, if you ask questions which can be answered by reading the test files or the project files, I will refer you to those. This is an exam, so late submissions are not accepted. You should start working sooner rather than later.

Academic integrity will be more strictly enforced on this assignment. Your instructors have been reminded to remind you that exams which are not group-based should be handled with greater care given the circumstances. FYI, there were submissions of HW3 similar enough that such similarities on this exam would be turned over to Academic Integrity.

NOTE: There are 18 points on this exam and that is a hard cap. If you earn 19, it is still a 100%. That means you need only solve 3 of the problems for 100% credit. The point allocation is as follows:

- CharMatrix:

  1. Compiles: 1 (make test_char_matrix)
  2. Lints: 1 (cpplint –root=./ */char_matrix.*)
  3. TestFillConstructor: 1 (./test_char_matrix 0)
  4. TestCopyConstructor: 1 (./test_char_matrix 1)
  5. TestAssignOperator: 1 (./test_char_matrix 2)
  6. test_char_matrix_memory: 1 (make test_char_matrix_memory)

- functional_array:

  1. Compiles: 1 (make test_functional_array)
  2. Lints: 1 (cpplint –root=./ */functional_array.*)
  3. TestAllocateRectangle: 0.5 (./test_functional_array 0)
  4. TestDeallocateRectangle: 0.5 (./test_functional_array 1)
  5. TestAllocateJagged: 1 (./test_functional_array 2)
  6. TestDeallocateJagged: 1 (./test_functional_array 3)
  7. test_functional_array_memory: 1 (make test_functional_array_memory)

- TightVector:

  1. Compiles: 1 (make test_tight_vector)
  2. Lints: 1 (cpplint */tight_vector.*)
  3. TestAppendEmpty: 0.5 (./test_tight_vector 0)
  4. TestAppendNotEmpty: 1 (./test_tight_vector 1)
  5. TestAppendEmpty: 0.5 (./test_tight_vector 2)
  6. TestAppendNotEmpty: 1 (./test_tight_vector 3)
  7. test_tight_vector_memory: 1 (make test_tight_vector_memory)

- TimeSpan:

  1. Compiles: 1 (make test_time_span)
  2. Lints: 1 (cpplint --root=./ */time_span.*)
  3. TestPlusTimeSpan: 1 (./test_time_span 0)
  4. TestPlusInt: 1 (./test_time_span 1)
  5. TestPlusExtract: 1 (./test_time_span 2)
  6. TestPlusInsert: 1 (./test_time_span 3)

The End.