Module Four Milestone Screenshot and Summary

CS 405 23EW6

Jared Smith

7/23/23

**Screenshot:**

A screenshot of a computer program

Description automatically generated

**Summary:**

There are 16 tests in this package, 3 of which were written before me.

The 13 tests I wrote are as follows:  
CanAddToEmptyVector: This test ensures that a vector that is initialized with no elements can be added to using the add\_entries method defined in the CollectionTest class. This is done by asserting that the collection is no longer empty and that the new size equals the expected size after adding a single entry.

CanAddFiveValuesToVector: This test ensures that an empty vector can have 5 values added, ensuring that the loop in our add\_entries method functions as intended. This is also done by asserting that the collection is no longer empty and matches the number of elements added using the method.

CollectionSizeIsLargeEnough: This test ensures that a vector with 0, 1, 5, and 10 elements remains within the memory limit for the C++ vector class type. This is done by asserting that the max size of a Vector is greater than or equal to the size of the collection at each number of elements.

CollectionCapacityIsGreaterThanSize: This test ensures that the capacity for the vector (the amount of reserved memory) is greater than or equal to the size (or actual number of elements) in the vector. This is tested by asserting that the collection capacity >= collection size at 0, 1, 5, and 10 elements.

CanResizeIncreaseCollectionSize: This test ensures that the vector method resize results in the expected number of elements. This is done by calling vector resize(5) and asserting that the collection is no longer empty, and has a new size of 5.

CanResizeDecreaseCollectionSize: This test does the same as the previous test, but ensures that calling resize on a vector larger than the call decreases the vector size as expected. This is done in the same way as the last, but by calling resize(1) on a vector of size 5, and asserting that the size is no longer 5 and is equal to 1.

CanResizeEmptycollection: This test ensures that if we call resize(0) the vector will be emptied. This is done by populating the vector and asserting that it is indeed populated, and then calling resize(0) and asserting that the new collection size is not 5 and is equal to 0.

DoesEraseEmptycollection: This test ensures that the vector method erase() is functioning as intended on our collection. This is done by populating the collection vector with 5 elements, asserting that the new size is correct, and calling vector erase. We then assert that the size is no longer 5, and that it is equal to 0.

DosReserveIncreaseCapcityNotSize: This test ensures that the vector method reserve is functioning as intended (by increasing the reserved memory for the vector, and not adding elements). This is done by ensuring that the vector is empty, asserting that the capacity is 0, that the size is equal to the capacity, and calling collection reserve(5). We then assert that the capacity is no longer 0, that the capacity is not equal to the size and that the capacity is now 5.

ExceptionThrowWhenCallingOutOfVectorRange: This test ensures that the vector method at() checks the vector range before returning the value. This is done by populating our empty collection vector with 5 elements, asserting that the new size is 5, and asserting that assigning an integer with the collection at index 6 throws an out-of-range error.

AddEntryValuesStayInRange: This test ensures that values added with the add\_values method stay within the predetermined range of 0-99. This is done by calling add\_entreis(1000) to add 1000 values to the collection, then asserting that each value is greater than or equal to 0 and less than 100.

CannotAddLessThanOneEntry: I struggled to implement another exception from the vector library since it only throws debug assertion fails, so I added exception handling to the add\_entries method so it throws an out of range exception when you try to add less than 1 entry. This test ensures that that exception handling functions as intended, by asserting that add\_entries(0) and add\_entries(-1) both throw the out of range exception.