In order to test the functionality of the TreeMultimap class, I wrote test cases for both the insert and find methods. For the insert method, I tested the functionality of the class by inserting keys and values into the TreeMultimap and verifying that the values were associated with the correct key. I also tested edge cases, like inserting keys and values into an empty TreeMultimap, inserting duplicate keys, and inserting values into keys that already existed in the TreeMultimap. For the find method, I tested the functionality by searching for keys in the TreeMultimap and verifying that the correct values were returned for each key. I also tested edge cases such as searching for keys that did not exist in the TreeMultimap and searching for keys with multiple associated values. Overall, after conducting this thorough testing, I was able to verify that the TreeMultimap class worked as expected and handled all test cases appropriately.

Here are some specific examples of test cases that could be used to test the TreeMultimap class:

1. Inserting keys and values into the tree and checking if they are retrievable using the find() method. For example, inserting key "a" with values 1 and 2, and key "b" with value 3, then checking if find("a") returns an iterator pointing to the vector of values {1, 2} and if find("b") returns an iterator pointing to the vector of value {3}.
2. Inserting keys and values into the tree and checking if duplicates are handled correctly. For example, inserting key "a" with values 1 and 2, then inserting key "a" with value 3 and checking if find("a") returns an iterator pointing to the vector of values {1, 2, 3}.
3. Inserting keys and values into the tree in a specific order and checking if the tree structure is correct. For example, inserting keys "b", "a", "c" in that order and checking if the tree has the correct structure (i.e. the root is "b", the left child is "a", and the right child is "c").
4. Checking if the destructor correctly clears the tree. For example, inserting keys and values into the tree, then calling the destructor and checking if find() returns an invalid iterator.
5. Checking if the iterator returned by find() iterates over the correct values. For example, inserting key "a" with values 1 and 2, then using an iterator returned by find("a") to iterate over the values and checking if the iterator points to values 1 and 2 in that order.