Test cases 1a-1e will be created at any location and direction that would not cause a boat to a normal game board. 1f-1g will be done with any boat and at any location that would not cause the boat to leave a normal game board. 1h-1n will be done with any boat and direction that would not cause the boat to leave a normal game board. The process should not cause any exceptions.

1. Test Boat Constructor:
   1. Create Aircraft Carrier
   2. Create Battleship
   3. Create Cruiser
   4. Create Submarine
   5. Create Destroyer
   6. Create boat with direction set as vertical
   7. Create boat with direction set as horizontal
   8. Create boat with initial position at first column
   9. Create boat with initial position at last column
   10. Create boat with initial position at top row
   11. Create boat with initial position at bottom row
   12. Create boat with initial position at the center of the board

Test cases 2a-2e will be performed on boats previously created. The names should match

1. Testing name():
   1. Call on an Aircraft Carrier
   2. Call on a Battleship
   3. Call on a Cruiser
   4. Call on a Submarine
   5. Call on a Destroyer

Test cases 3a-3e will be performed also on boats previously created. The abbreviations should be the first letters of the names.

1. Testing abbreviation()
   1. Call on an Aircraft Carrier
   2. Call on a Battleship
   3. Call on a Cruiser
   4. Call on a Submarine
   5. Call on a Destroyer

All tests 4a-4e will be performed on boats previously created and results are compared to the intended sizes.

1. Testing size()
   1. Call on an Aircraft Carrier
   2. Call on a Battleship
   3. Call on a Cruiser
   4. Call on a Submarine
   5. Call on a Destroyer

All tests 5a-5e will be performed on boats previously created and checked against the initial test input.

1. Testing position()
   1. Call on a Boat on the top row
   2. Call on a Boat on the bottom row
   3. Call on a Boat on the left column
   4. Call on a Boat on the right column
   5. Call on a Boat on the center

All tests 6a-6b will be performed on boats previously created. The results will be compared against initial inputs.

1. Testing direction()
   1. Call on a vertical boat
   2. Call on a horizontal boat

All tests in 7 will be performed on the boats previously created. 7a-7e should return true, 5f-6g should return false.

1. Testing onBoat()
   1. Call on all occupied squares on a ship on the top row
   2. Call on all occupied squares on a ship on the bottom row
   3. Call on all occupied squares on a ship on the right column
   4. Call on all occupied squares on a ship on the left column
   5. Call on all occupied squares on a ship in the middle
   6. Call on 7 random unoccupied squares
   7. Call on a nonexistent square

All tests in 8a-8e will be performed on the boats previously created. All of them should return false, as the boats should not have been hit since. 8h-8l should return true as the squares are hit.

1. Testing isHit()
   1. Call on all occupied squares on a ship on the top row
   2. Call on all occupied squares on a ship on the bottom row
   3. Call on all occupied squares on a ship on the right column
   4. Call on all occupied squares on a ship on the left column
   5. Call on all occupied squares on a ship in the middle
   6. Call on 7 random unoccupied squares
   7. Call on a nonexistent square
   8. Hit one square on a boat on a ship on the top row and check on the square
   9. Hit one square on a boat on a ship on the bottom row and check on the square
   10. Hit one square on a boat on a ship on the left column and check on the square
   11. Hit one square on a boat on a ship on the right column and check on the square
   12. Hit one square on a boat on a ship in the center and check on the square

All tests in 9a-9e will be performed on the boats previously created.

1. Testing hit()
   1. Call on an occupied square on a boat on the top row, check with isHit()
   2. Call on an occupied square on a boat on the bottom row, check with isHit()
   3. Call on an occupied square on a boat on the left column, check with isHit()
   4. Call on an occupied square on a boat on the right column, check with isHit()
   5. Call on an occupied square on a boat in the center, check with isHit()

All tests in 10- will be performed on the boats previous created and are still afloat. All tests should return true.

1. Testing sunk()
   1. Hit every square on an Aircraft Carrier, call on boat
   2. Hit every square on a Battleship, call on boat
   3. Hit every square on a Cruiser, call on boat
   4. Hit every square on a Submarine, call on boat
   5. Hit every square on a Destroyer, call on boat