Test cases

|  |  |  |
| --- | --- | --- |
| Operation Purpose | Object State | Expected Result |
| anyObj.toString()  Make sure that the toString is properly overridden | var1 = 8;  var2 = “Hand”;  var3 = ‘g’; | “var1: 8,  var2: “Hand”,  var3: ‘g’” |
| obj1.equals(obj2)  Make sure equals is properly overridden | obj1 = {  var1 = 8,  var2 = “Hand”,  var3 = ‘g’}  obj2 = {  var1 = 8,  var2 = “Hand”,  var3 = ‘g’} | true |
| obj1.equals(obj2)  Make sure equals is properly overridden | Same as previous cell except obj2.var1 = 9 | false |
| guessLetter(‘p’);  Make sure that all occurrences of the letter are selected | word = a>p>p>l>e  guessed =  wrong = | guessed = p>p  wrong = |
| guessLetter(‘z’);  Make sure wrong letters are properly appended | word = a>p>p>l>e  guessed =  wrong = g | wrong = g>z  guessed = |
| getHint();  Make sure getHint works properly |  | The letter should be a random letter that has not yet been guessed and is in the word. |
| restartGame()  Make sure the read list of words has the last used word removed from it | word = a>p>p>l>e  words = (c>h>e>e>s>e)> (a>p>p>l>e)> (g>r>e>e>n) | words = (c>h>e>e>s>e)> (g>r>e>e>n) |
| addScore((Entry for Kyle))  Make sure the scoreboard is sorted alphabetically and adds properly | scoreboard = (entry for Alex)-(Entry for Ben)-(Entry for Michael)-(Entry for Sarah)-(Entry for Zoe) | scoreboard = (entry for Alex)-(Entry for Ben)-(Entry for Kyle)-(Entry for Michael)-(Entry for Sarah)-(Entry for Zoe) |

Notes:

* > indicates the next item of a singly linked list. – indicates the linked items in a doubly linked list.
* No test cases for file reading/writing since serialization is an unknown concept at this point.