Problem 2 Test Plan

Part a: Y

Part b: Y

The expected output and the actual output match up.

|  |  |  |
| --- | --- | --- |
| Test input | Expected result | Actual result |
| cout << "Entering main\n";  cout << "SharedString a = \"Fred\";\n";  SharedString a = "Fred";  cout << "SharedString b = \"Alice\";\n";  SharedString b = "Alice";  cout << "SharedString c;\n";  SharedString c;  cout << "c = a; // Fred\n";  c = a; // Fred  cout << "a = b; // Alice\n";  a = b; // Alice  cout << "g = b; // Alice\n";  g = b; // Alice  cout << "Exiting main\n";    cout << endl << endl << "Assignment 2 Problem 3: " << endl;  cout << "Test a == b :";  bool test1 = a == b;  cout << test1 << endl;  cout << "Test c == a: ";  bool test2 = c == a;  cout << test2 << endl;  cout << "Test a < c: ";  bool test3 = a < c;  cout << test3 << endl;  cout << "Test changing a letter: " << endl;  cout << "a and b are Alice" << endl;  cout << "check if a and b are equal: " << test1 << endl;  cout << "Changing b to blice" << endl;  b[0] = 'B';  bool afterChange = a == b;  cout << "check if a and b are equal: " << afterChange << endl; | Entering default constructor SharedString().  Cell pointer is NULL. There is no memory buffer.  Entering main  SharedString a = "Fred";  Entering constructor SharedString(const char\*).  Reassign: Incremented count of Fred to 1.  After reassign, memory cell is Fred, count: 1.  SharedString b = "Alice";  Entering constructor SharedString(const char\*).  Reassign: Incremented count of Alice to 1.  After reassign, memory cell is Alice, count: 1.  SharedString c;  Entering default constructor SharedString().  Cell pointer is NULL. There is no memory buffer.  c = a; // Fred  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Fred to 2.  After reassign, memory cell is Fred, count: 2.  a = b; // Alice  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Alice to 2.  Reassign. Assignment or destructor. Old memory cell: Fred. Decremented count to: 1.  After reassign, memory cell is Alice, count: 2.  g = b; // Alice  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Alice to 3.  After reassign, memory cell is Alice, count: 3.  Exiting main  Assignment 2 Problem 3:  Test a == b :1  Test c == a: 0  Test a < c: 1  Test changing a letter:  a and b are Alice  check if a and b are equal: 1  Changing b to blice  changing letter, new copy creaeted  check if a and b are equal: 0  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Fred, count: 1.  Reassign. Assignment or destructor. Old memory cell: Fred. Decremented count to: 0.  Reassign. Deallocating Fred.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Blice, count: 0.  Reassign. Assignment or destructor. Old memory cell: Blice. Decremented count to: -1.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Alice, count: 2.  Reassign. Assignment or destructor. Old memory cell: Alice. Decremented count to: 1.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Alice, count: 1.  Reassign. Assignment or destructor. Old memory cell: Alice. Decremented count to: 0.  Reassign. Deallocating Alice. | Entering default constructor SharedString().  Cell pointer is NULL. There is no memory buffer.  Entering main  SharedString a = "Fred";  Entering constructor SharedString(const char\*).  Reassign: Incremented count of Fred to 1.  After reassign, memory cell is Fred, count: 1.  SharedString b = "Alice";  Entering constructor SharedString(const char\*).  Reassign: Incremented count of Alice to 1.  After reassign, memory cell is Alice, count: 1.  SharedString c;  Entering default constructor SharedString().  Cell pointer is NULL. There is no memory buffer.  c = a; // Fred  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Fred to 2.  After reassign, memory cell is Fred, count: 2.  a = b; // Alice  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Alice to 2.  Reassign. Assignment or destructor. Old memory cell: Fred. Decremented count to: 1.  After reassign, memory cell is Alice, count: 2.  g = b; // Alice  Entering assignent operator=(const SharedString&).  Reassign: Incremented count of Alice to 3.  After reassign, memory cell is Alice, count: 3.  Exiting main  Assignment 2 Problem 3:  Test a == b :1  Test c == a: 0  Test a < c: 1  Test changing a letter:  a and b are Alice  check if a and b are equal: 1  Changing b to blice  changing letter, new copy creaeted  check if a and b are equal: 0  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Fred, count: 1.  Reassign. Assignment or destructor. Old memory cell: Fred. Decremented count to: 0.  Reassign. Deallocating Fred.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Blice, count: 0.  Reassign. Assignment or destructor. Old memory cell: Blice. Decremented count to: -1.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Alice, count: 2.  Reassign. Assignment or destructor. Old memory cell: Alice. Decremented count to: 1.  Entering destructor ~SharedString().  Reassign. Destructor. Memory cell before change: Alice, count: 1.  Reassign. Assignment or destructor. Old memory cell: Alice. Decremented count to: 0.  Reassign. Deallocating Alice. |