@Override

public final void push(T x) {

assert x != null : "Violation of: x is not null";

// Create a new node

Node enterDigit = new Node();

enterDigit.data = x;

enterDigit.next = this.top;

this.top = enterDigit;

this.length++;

assert this.conventionHolds();

}

@Override

public final T pop() {

assert this.length() > 0 : "Violation of: this /= <>";

// Remove the top node

T result = this.top.data;

this.top = this.top.next;

this.length--;

assert this.conventionHolds();

return result;

}

@Override

public final int length() {

assert this.conventionHolds();

return this.length;

}

/\*

\* Push Tests --------------------------------------------------

\*/

/\*\*

\* Tests Push.

\*/

@Test

public final void testPushFromEmpty() {

Stack<String> s = this.createFromArgsTest();

Stack<String> sExpected = this.createFromArgsRef("Apple");

s.push("Apple");

assertEquals(sExpected, s);

}

/\*\*

\* Tests Push-non empty.

\*/

@Test

public final void testPushFromNonEmpty() {

Stack<String> s = this.createFromArgsTest("Banana");

Stack<String> sExpected = this.createFromArgsRef("Cherry", "Banana");

s.push("Cherry");

assertEquals(sExpected, s);

}

/\*

\* Pop Tests --------------------------------------------------

\*/

/\*\*

\* Tests Pop.

\*/

@Test

public final void testPopToEmpty() {

Stack<String> s = this.createFromArgsTest("Date");

Stack<String> sExpected = this.createFromArgsRef();

String ans = s.pop();

assertEquals(sExpected, s);

assertEquals("Date", ans);

}

/\*\*

\* Tests Pop-non empty.

\*/

@Test

public final void testPopToNonEmpty() {

Stack<String> s = this.createFromArgsTest("Elderberry", "Fig");

Stack<String> sExpected = this.createFromArgsRef("Fig");

String ans = s.pop();

assertEquals(sExpected, s);

assertEquals("Elderberry", ans);

}

/\*

\* Length Tests --------------------------------------------------

\*/

/\*\*

\* Tests Length.

\*/

@Test

public final void testLength() {

Stack<String> s = this.createFromArgsTest("Grape");

int l = s.length();

assertEquals(1, l);

}