Marco Seman

CIS 285 Software Engineering Tools

Professor Tommy Xu

April 4, 2019

Lab 9-10

A) **Source Code for SelectionSort.java class**

**package** Lab10;

**public** **class** SelectionSort {

**private** **int** temp;

**public** SelectionSort() {

}

**public** **int**[] basicSelectionSort(**int**[] x) {

**for** (**int** i = 0; i < x.length; ++i) {

**for** (**int** j= i+1; j < x.length; ++j) {

**if** (x[i] > x[j]) {

temp = x[i];

x[i] = x[j];

temp = x[j];

}

} // end of inner for loop

} // end of outer for loop

**return** x;

} // end of basicSelectionSort method

}

B) **Source Code of unit test created**

**package** Lab10;

**import** **static** org.junit.Assert.\*;

//import org.junit.Assert;

**import** org.junit.Test;

**public** **class** testSelectionSort {

@Test

**public** **void** test() {

testPositive();

//testNegative();

//testDuplicates();

//testMixed();

//testZeros();

}

**public** testSelectionSort() {

}

**public** **void** testPositive(){

**int**[] arr = **new** **int**[5];

arr[0] = 8;

arr[1] = 9;

arr[2] = 7;

arr[3] = 10;

arr[4] = 2;

**int**[] Sortedarr = **new** **int**[5];

Sortedarr[0] = 2;

Sortedarr[1] = 7;

Sortedarr[2] = 8;

Sortedarr[3] = 9;

Sortedarr[4] = 10;

SelectionSort temp = **new** SelectionSort();

arr = temp.basicSelectionSort(arr);

*assertArrayEquals*("Positive Test Fails", Sortedarr, arr);

/\*\* add tests to check for this unit test \*\*/

}

**public** **void** testNegative(){

**int**[] arr = **new** **int**[5];

arr[0] = -2;

arr[1] = -5;

arr[2] = -7;

arr[3] = -3;

arr[4] = -9;

**int**[] Sortedarr = **new** **int**[5];

Sortedarr[0] = -9;

Sortedarr[1] = -7;

Sortedarr[2] = -5;

Sortedarr[3] = -3;

Sortedarr[4] = -2;

SelectionSort temp = **new** SelectionSort();

arr = temp.basicSelectionSort(arr);

*assertArrayEquals*("Negative Test Fails", Sortedarr, arr);

/\*\* Test data contains negative values only \*\*/

}

**public** **void** testMixed(){

**int**[] arr = **new** **int**[5];

arr[0] = -8;

arr[1] = 9;

arr[2] = 7;

arr[3] = -10;

arr[4] = 0;

**int**[] Sortedarr = **new** **int**[5];

Sortedarr[0] = -10;

Sortedarr[1] = -8;

Sortedarr[2] = 0;

Sortedarr[3] = 7;

Sortedarr[4] = 9;

SelectionSort temp = **new** SelectionSort();

arr = temp.basicSelectionSort(arr);

*assertArrayEquals*("Mixed Test Fails", Sortedarr, arr);

/\*\* Test data contains with both positive, negative and zeros \*\*/

}

**public** **void** testDuplicates(){

**int**[] arr = **new** **int**[5];

arr[0] = -8;

arr[1] = 9;

arr[2] = -8;

arr[3] = 9;

arr[4] = -5;

**int**[] Sortedarr = **new** **int**[5];

Sortedarr[0] = -8;

Sortedarr[1] = -8;

Sortedarr[2] = -5;

Sortedarr[3] = 9;

Sortedarr[4] = 9;

SelectionSort temp = **new** SelectionSort();

arr = temp.basicSelectionSort(arr);

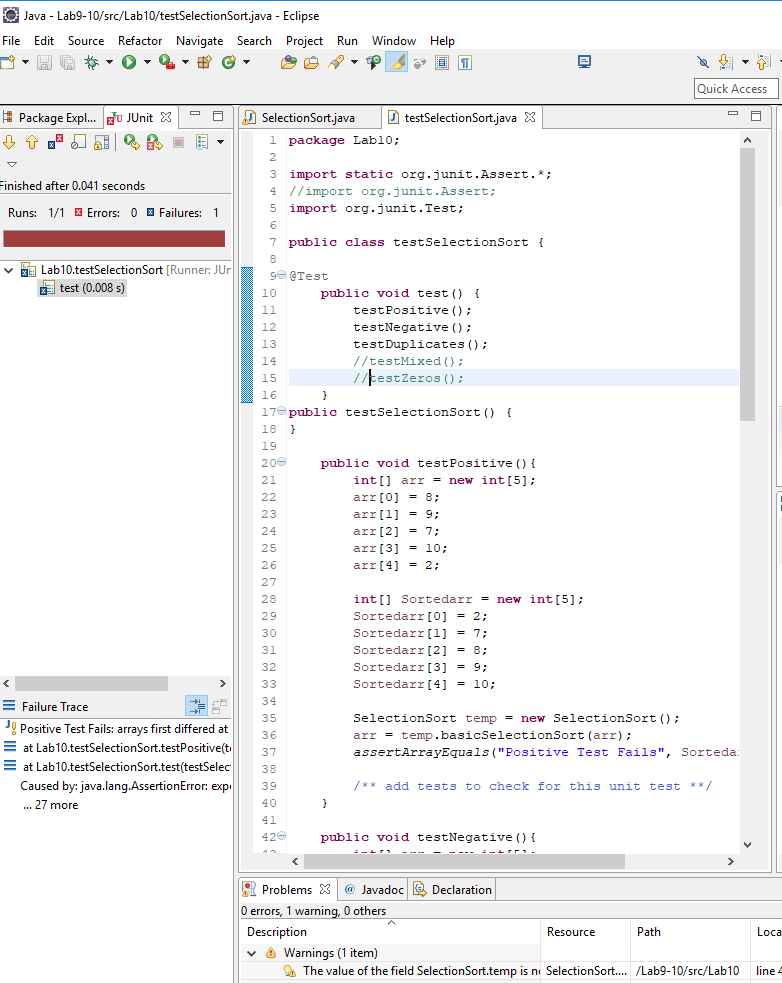
*assertArrayEquals*("Duplicate Test Fails", Sortedarr, arr);

/\*\* Test data contains duplicates \*\*/

}

}

**Executed unit test:**

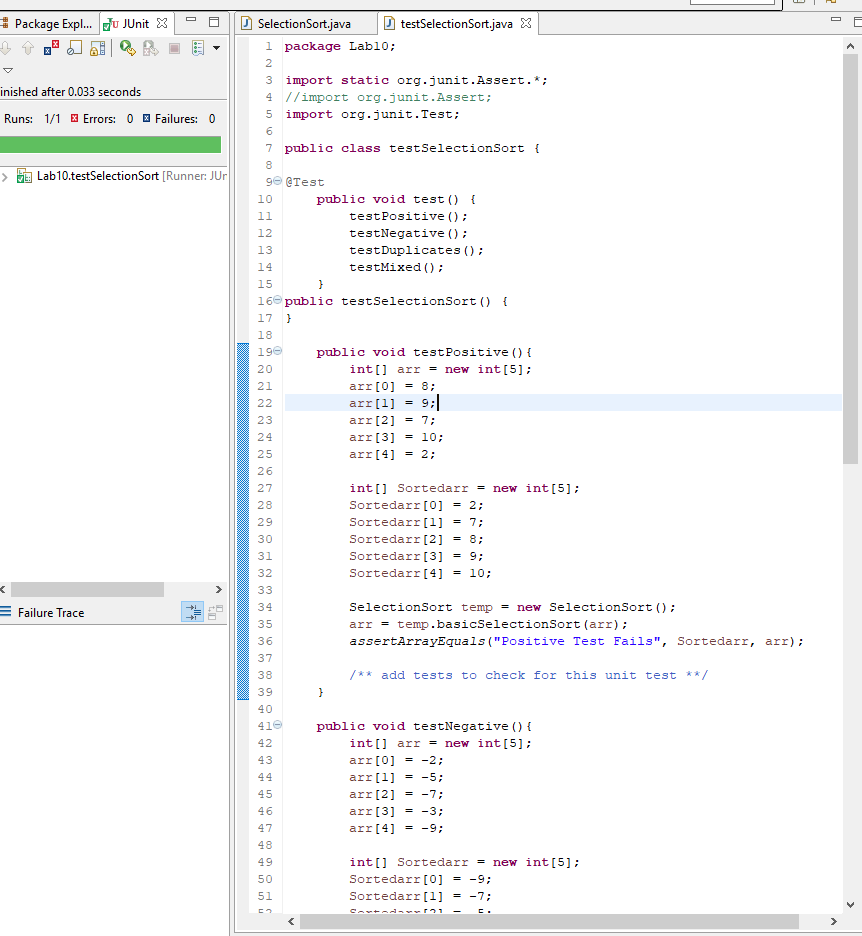


C) **Outputs of unit tests**

**What was wrong and and what was done to fix the problem**

testZeros was not included in the test implementation inside the public testSelectionSort() method. Also, testMixed was also never called, so when executing the program, the function was never run. To fix this issue, I renamed testZero in the the public void test() function to testMixed(). In SelectionSort, one of the arrays was wrong. Temp = x[j]; was incorrectly defined making the sort array never to update. To fix the issue, I changed that last line in the if statement to x[j] = temp so that the array is updated.

D) **Output of all unit tests showing successful pass for all**



E) **Source code of final SelectionSort.java class**

**public** **class** SelectionSort {

**private** **int** temp;

**public** SelectionSort() {

}

**public** **int**[] basicSelectionSort(**int**[] x) {

**for** (**int** i = 0; i < x.length; ++i) {

**for** (**int** j= i+1; j < x.length; ++j) {

**if** (x[j] < x[i]) {

temp = x[i];

x[i] = x[j];

x[j] = temp;

}

} // end of inner for loop

} // end of outer for loop

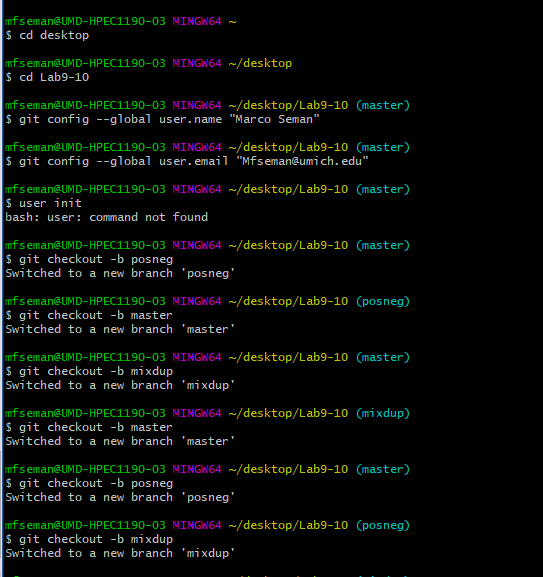
**return** x;

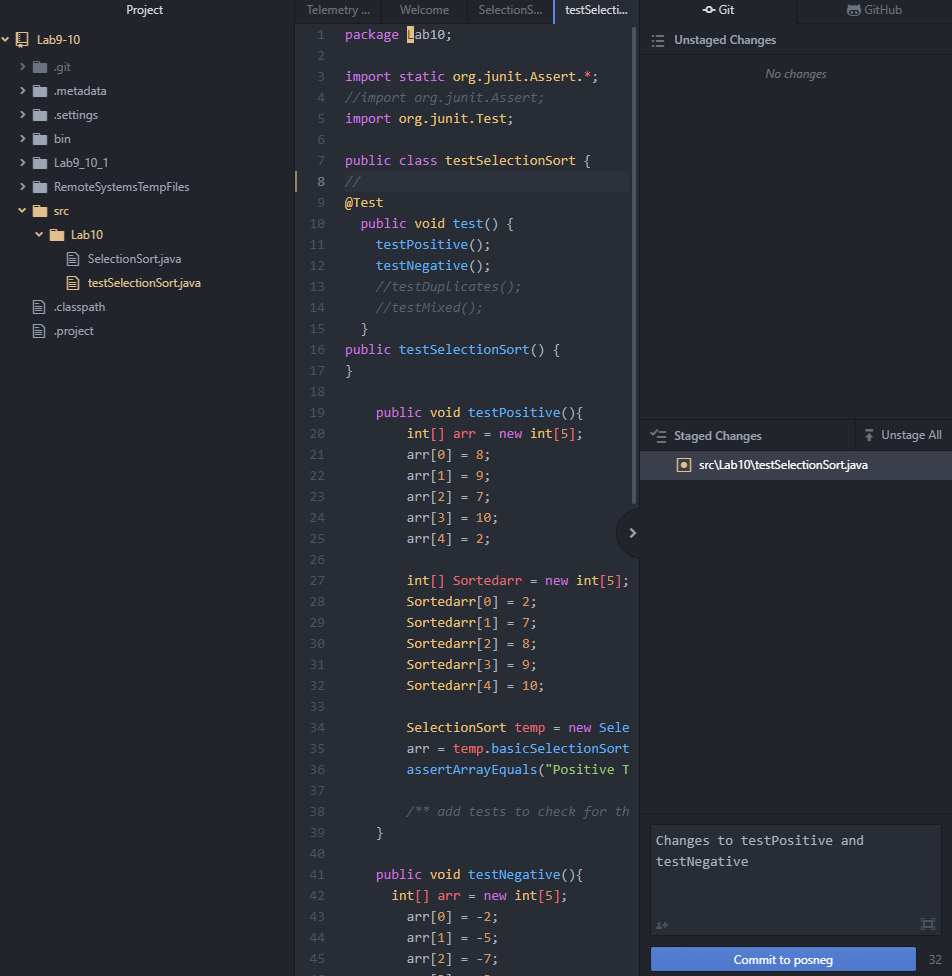
} // end of basicSelectionSort method

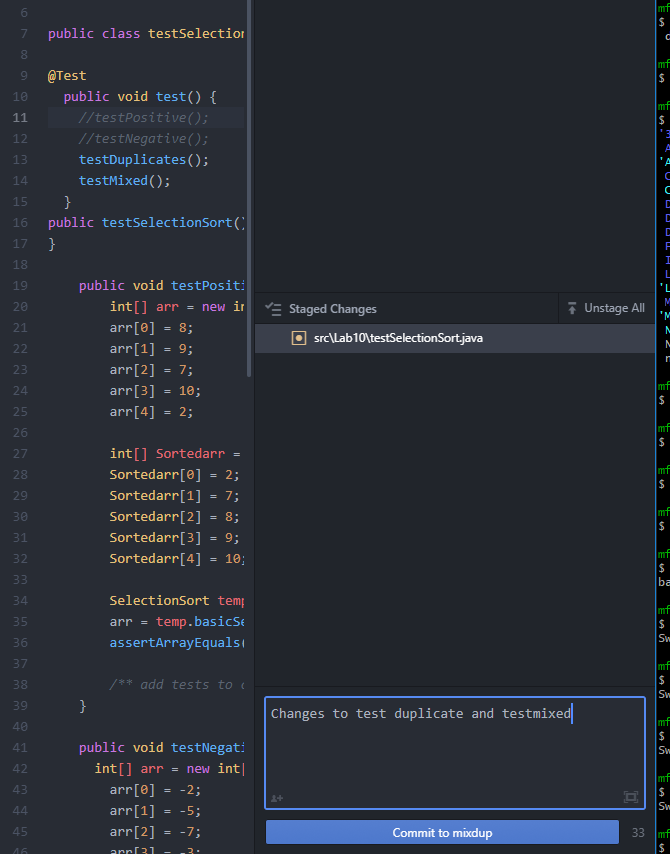
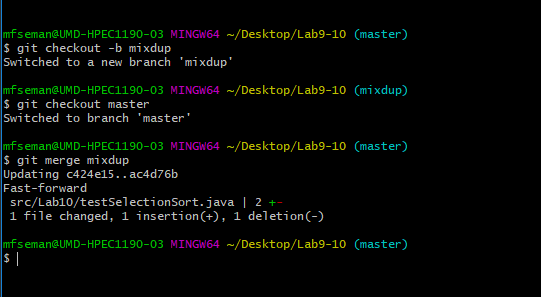
**}**

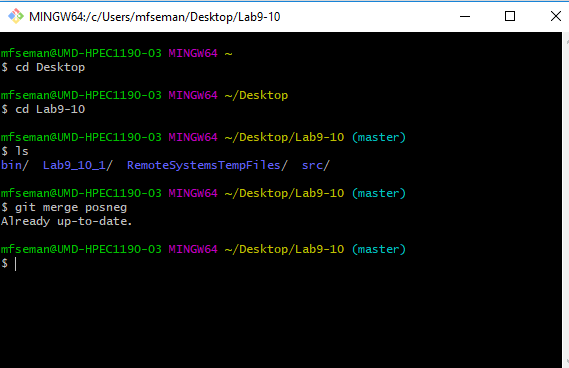
2) Git for Configuration Management

a) **Screenshot of github containing selectonsort.java and testselectionsort.java**





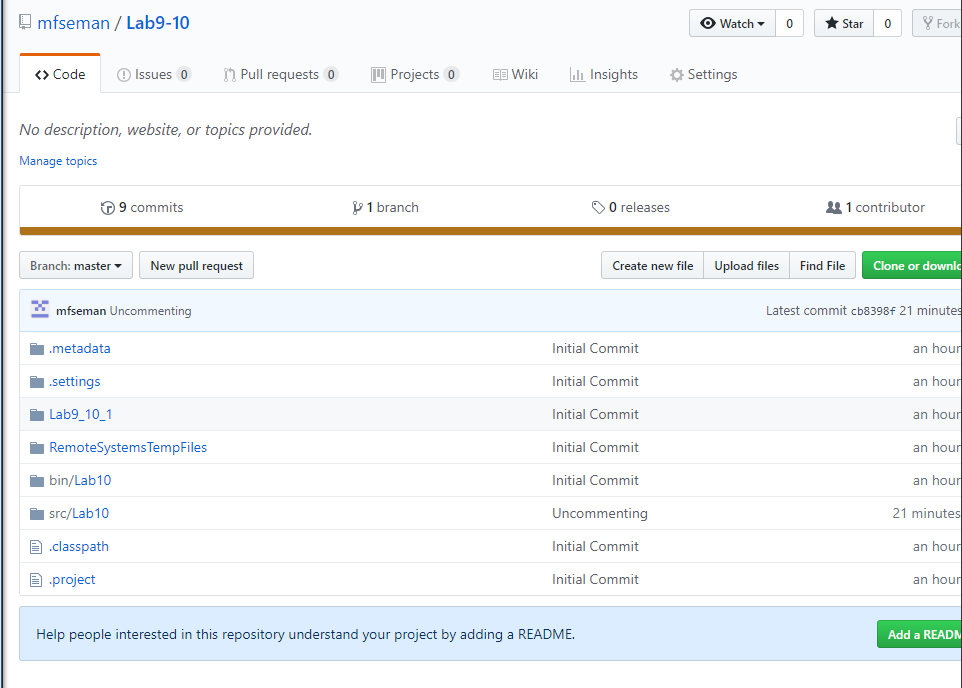
 



**(I changed it earlier where it showed the insertion and deletion. When I merged the branch, git bash crashed making me unable to see the changes)**

b) **local git folder**





Ignore the Lab9\_10\_1 folder