# Java Week One

a)

**protected** **void** randomise()

{

//start timer

setUp();

//create an int array and fill it

**int**[] copy = getArray();

//create int j

**int** j;

/\*

\* this for loop will swap position i and a random position

\* \*/

//for every int in copy

**for**(**int** i : copy)

{

//store a random int in the array

**int** randomPos = getRandomIndex();

//make j equal pos i

j = copy[i];

//make position i of the array equal the one from a random pos

copy[i] = copy[randomPos];

//make a randompos in the array equal the the one in pos i

copy[randomPos] = j;

}

//set the array to equal the new randomised copy

**for** (**int** index = 0; index < getArray().length; index ++)

{

getArray()[index] = copy[index];

}

//end timer

tearDown();

}

b)

//get time at the start

**protected** **void** setUp()

{

*testStart* = System.*nanoTime*();

}

//get time at the end

**protected** **void** tearDown()

{

*testEnd* = System.*nanoTime*();

}

//return the the time in microseconds

**protected** String printTime()

{

String time = "\n" + "Test " + "took " + (*testEnd*-*testStart*)/1000 + " microseconds";

**return** time;

}

Tests:

**public** **class** CleverRandomListingTest **extends** ListingTest

{

**private** **static** **long** *testStart*, *testEnd*;

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Rule **public** TestName testName = **new** TestName();

@Before

**public** **void** setUp() **throws** Exception {

*testStart* = System.*nanoTime*();

}

@After

**public** **void** tearDown() **throws** Exception {

*testEnd* = System.*nanoTime*();

System.***out***.println("Test \"" + testName.getMethodName() + "\" took " + (*testEnd*-*testStart*)/1000 + " microseconds");

}

@Test

**public** **void** testOneSize() {

testSize(1,**new** CleverRandomListing(1));

}

@Test

**public** **void** testOneContents() {

testContents(1,**new** CleverRandomListing(1));

}

@Test

**public** **void** testTwoSize() {

testSize(2,**new** CleverRandomListing(2));

}

@Test

**public** **void** testTwoContents() {

testContents(2,**new** CleverRandomListing(2));

}

@Test

**public** **void** testFourSize() {

testSize(4,**new** CleverRandomListing(4));

}

@Test

**public** **void** testFourContents() {

testContents(4,**new** CleverRandomListing(4));

}

@Test

**public** **void** testHundredSize() {

testSize(100,**new** CleverRandomListing(100));

}

@Test

**public** **void** testHundredContents() {

testContents(100,**new** CleverRandomListing(100));

}

@Test

**public** **void** testThousandContents() {

testContents(1000,**new** CleverRandomListing(1000));

}

@Test

**public** **void** testMillionSize() {

testSize(1000000,**new** CleverRandomListing(1000000));

}

}

Test "testHundredSize" took 37993 microseconds

Test "testOneSize" took 34 microseconds

Test "testHundredContents" took 720 microseconds

Test "testTwoSize" took 27 microseconds

Test "testOneContents" took 26 microseconds

Test "testTwoContents" took 37 microseconds

Test "testThousandContents" took 15724 microseconds

Test "testMillionSize" took 128112 microseconds

Test "testFourSize" took 101 microseconds

Test "testFourContents" took 20 microseconds