**Assignment 5**

Dr. Matthews

CS 210

Bradley Justice

**Question 1**

public static double sum(double[] a) {

double sum = 0;

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

sum = sum + a[i];

}

return sum;

}

public static double[] fill(int n, double v) {

double[] resArray = new double[n];

for (int i = 0; i < n ; i++) {

resArray[i] = v;

}

return resArray;

}

public static double[] random (int n) {

double[] array = new double [n];

for (int i = 0; i<n ; i++ ) {

array[i] = Math.random()\*100;

}

return array;

}

public static double min (double[]a){

double min = a[0];

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

if (a[i] < min) {

min = a[i];

}

}

return min;

}

public static double max (double[] a) {

double max = a[0];

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

if (a[i] > max) {

max = a[i];

}

}

return max;

}

public static double average(double[] a) {

double sum = 0;

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

sum = sum + a[i];

}

double avg = sum/a.length;

return avg;

}

public static double variance (double[] a) {

double mean = average(a);

double sum = 0;

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

double variance = a[i] - mean;

sum = sum + (variance\*variance);

}

double avgVar = sum/a.length-1;

return avgVar;

}

public static double stdDev (double[] a) {

double stnDev = Math.sqrt(variance(a));

return stnDev;

}

public static double[] read() {

Scanner enter = new Scanner(System.in);

System.out.print("Input the number of elements: ");

int n = enter.nextInt();

double[] array = new double[n];

for (int i = 0 ; i < n ; i ++) {

System.out.print("Enter a value for element "+ (i+1) +": ");

array[i] = enter.nextDouble();

}

enter.close();

return array;

}

public static void print (double[] a){

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

System.out.print(a[i] + ", ");

}

}

public static void main(String[] args) {

double[] testArray = genTestArray();

System.out.println(Arrays.toString(testArray));

System.out.println(sum(testArray));

System.out.println(Arrays.toString(fill(5, 10)));

System.out.println(Arrays.toString(random(8)));

System.out.println(min(testArray));

System.out.println(max(testArray));

System.out.println(average(testArray));

System.out.println(variance(testArray));

System.out.println(stdDev(testArray));

System.out.println(Arrays.toString(read()));

print (testArray);

}

public static double[] genTestArray () {

int n = (int)(Math.random()\*10);

double[] ranTestArray = new double[n];

System.out.println(n);

for (int i = 0; i < n ; i++) {

ranTestArray[i] = Math.random()\*100;

}

return ranTestArray;

}

**Sample output/input:**

[46.96285269446132, 12.675180532529263, 85.5275784372573, 92.91165508917987, 83.14278262840965, 38.914060680233185]

360.1341100620706

[10.0, 10.0, 10.0, 10.0, 10.0]

[52.987384428463436, 22.927628333976223, 74.50667421710723, 24.615794281184357, 33.82275662352336, 11.427452154691832, 87.41101066384255, 57.27964813967561]

12.675180532529263

92.91165508917987

60.02235167701176

853.1070460453985

29.207996268922635

Input the number of elements: 4

Enter a value for element 1: 3

Enter a value for element 2: 1

Enter a value for element 3: 8

Enter a value for element 4: 1

[3.0, 1.0, 8.0, 1.0]

46.96285269446132, 12.675180532529263, 85.5275784372573, 92.91165508917987, 83.14278262840965, 38.914060680233185

**Question 2**

The output for this code is; 10, 11, 11. When a[0] is first printed, it has a value of 10, as initialized. Then, the next line, int[] b = inc(a), increments all of the elements of a. The reason for this is that inc (a) increments a[0], and returns it, therefore, no new array is made, all of these calls and adjustments are done to the same array located at address x in the computer’s memory.

**Question 3**

The output for this code is true, then false. A == b comes out to be true, and a == c comes out to be false. When int[] b is initialized, it is pointed to the same array in the same location, therefore both int[]a and int[]b are variables that point to the same array in the computers memory. When int[] c is initialized, it is initialized with the same elements and values as int[] a, but is assigned a new address in memory, therefore, a new array is created and stored. The check a == c does not check each value for equal values in both, it just compares the array’s themselves in memory, and as they are not the same array, as they point to two different arrays in memory, it comes out to be false.