ICS4U0 – Arrays

Section 1:

1. **An array to store marks for twenty students has been declared as follows:**

```
int[] marks = new int[20];
```

(a) What is the array identifier?

`marks` is the arrays identifier.

(b) What is the identifier of the first element in the array?

The identified of the first element in the array is `marks[0];`

(c) What value is stored in each element by the declaration?

By declaration, each element stores a 0.

(d) What is the value of `marks.length`?

The value of `marks.length` is 20.

(e) What are the indices of the array?

There are 19 indices of the array

2. **How much space (in bits) would be required to store the elements of each array?**

(a) `int[] a = new int[20];`

Approximately 1280 bits.

(b) `double[] b = new double[100];`

Approximately 6400 bits

(c) `float [] c = new float[50];`

Approximately 1600 bits

(d) `boolean[] d = new boolean[1000];`

Approximately 1000 bits.

3. **Write declarations to create arrays that would be appropriate for storing the indicated data.**
(a) the numbers of votes cast for five candidates in an election.

```
int [] voteCast = new int [5];
```

(b) the answers to a twenty-question true/false quiz

```
char [] quiz = new char [20];
```

(c) average family size in the years 1900, 1910, ... , 2000

```
int [] familysize = new int [100];
```

4. **(a) Write a statement that creates and initializes an array `terms` of `double` values to store the terms of the sequence**

$$t_1 = \frac{1}{2}, t_2 = \frac{2}{3}, \ldots, t_6 = \frac{6}{7}$$

(b) What is the value of `terms.length`?

5. **The table gives atomic masses of the eight lightest elements listed according to atomic number.**

Atomic Number	1	2	3	4	5	6	7	8
Atomic Mass	1.0	4.0	6.9	9.0	10.8	12.0	14.0	16.0

Suppose that the data in this table were to be represented by the array mass declared by the statement.

```
double[] mass = {0,1,4,6.9,9,10.8,12,14,16};
```

(a) What is the value of `mass[2]`?

4.0

(b) What is the value of `mass[5]`?

10.8

(c) What are the possible values of the indices of the array?

The atomic masses seem to increase approximately by 2 as the atomic number increases by 1. Hence possible values of the indices of the array include 18,20,22..

(d) What is the identifier of the element whose value is 6.9?

`mass[3]`

(e) Of what type are the elements?

The elements are type of **`double`**

(f) What is the value of `mass.length`?

9

Section 2:

1. **What would be printed by the following program fragment?**

```
int[] list = new int[4];
for (int i = 0; i < list.length; i++)
list[i] = 3 - i;
System.out.println(list[1]+2);
System.out.println(list[1+2]);
System.out.println(list[1]+list[2]);
```

4 0 3

2. **Suppose that an array `sample` has been declared as follows:**

```
int[] sample = new int[SIZE];
```

Write one or more statements to perform each task.
(a) Initialize all elements of the array to one.

```
for (int x = 0; x < SIZE; x++) {
	sample[x] = 1; }
```

(b) Switch the values at either end of the array.

```
sample[0] = 5;
```

(c) Change any negative values to positive values (of the same magnitude).

```
for (int x = 0; x < SIZE; x++) {
	sample[x] = Math.abs(sample[x]); }
```

(d) Set the variable `sampleSum` to the sum of the values of all the elements.

```
int sampleSum = 0;
	for (int x = 0; x < SIZE; x++) {
		sampleSum = sampleSum + sample[x]; }
```

(e) Print the contents of the odd-numbered locations.

```
for (int x = 0; x < SIZE; x++) {
	if (x % 2 != 0)
		System.out.println(sample[x]);}
```

3. Write a method `max` that has one `double` array parameter. The method should return the value of the largest element in the array.

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		double [] array = new double [3];
		array[0] = 10;
		array[1] = 100;
		array[2] = 1000;
		System.out.println(max(array) + " is the largest value in
   the array");

	}

	public static double max (double [] array) {
		double max = 0;
		for (int x = 0; x < array.length; x++) {
			if (max <= array[x])
				max = array[x];
		}
		return max;
	}
```

```
    }
```

4. Complete the definition of the method `equals` so that it returns true if and only if its two array parameters contain equal elements.

```
   public static boolean equals (double[] a, double[] b)
```

```
public class Question4 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//Test Case 1 - When both arrays equal to each other
		double [] a = {1,2,3,4,5};
		double [] b = {1,2,3,4,5};
		System.out.println(equals(a,b));

		//Test Case 2 - When both arrays are NOT equal to each
other
		double [] c = {1,2,3,4,5};
		double [] d = {6,7,8,9,10};
		System.out.println(equals(c,d));

	}

	public static boolean equals (double[] a, double[] b) {

		if (Arrays.equals(a,b))
			return true;
		else
			return false;
	}

}
```

5. Write a program that repeatedly prompts the user to supply scores (out of 10) on a test. The program should continue to ask the user for marks until a negative value is supplied. Any values greater than ten should be ignored. Once the program has read all the scores, it should produce a table with the following headings:

```
Score          # of Occurrences
```

The program should then calculate the mean score, rounded to one decimal place.

```
package Section2;

import java.util.Scanner;

public class Question5 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
```

```
		Scanner input = new Scanner (System.in);
		int [] scores = new int [0];
		int temp;
		int mark;
		int mean = 0;
		int count = 0;
		do {
			System.out.println("Supply a score (out of 10) on a
test:");
			mark = input.nextInt();
			scores = addMark(mark, scores);
		}
		while (mark > 0);

		for (int x = 0; x < scores.length; x++) {
			for (int y = 0; y < scores.length; y++){
				if (scores[x] < scores[y]) {
					temp = scores[x];
					scores[x] = scores[y];
					scores[y] = temp;
				}
			}
		}


		System.out.println("Score" + "\t\t" + "# of Occurences");
		for (int x = 0; x < scores.length; x = x + count) {
			count = 0;
			for (int y = x; y < scores.length; y++) {
				if (scores[x] == scores[y])
					count++;
			}
			System.out.println(scores[x] + "\t\t" + count);
		}

		for (int x = 0; x < scores.length; x++)
			mean = mean + scores[x];
		mean = mean/scores.length;
		System.out.println("The MEAN score is " + mean);

	}

	/*
	 *This method adds the students score into the array
	 *pre : Mark & old array of scores
	 *post: New array of scores
	*/
	public static int [] addMark (int mark, int [] scores) {
		if (mark <= 10 && mark >= 0) {
			int [] temp = new int [scores.length + 1];
			for (int x = 0; x < scores.length; x++) {
				temp[x] = scores[x];
			}
			temp [temp.length-1] = mark;
			return temp;
```

```
            }
            else
                  return scores;
      }
}
```