ICS 4U0 Java Review – Arrays

Arrays are *the* ticket to holding large data sets. Carefully review how to declare, initialize, and iterate through arrays. The following example is a good start (also available on the website).

Declaring Arrays

```
int myNumbers []; // declaration without initialization
int [] myDigits; // alternative declaration without initialization.
```

Initializing Arrays

```
myNumbers = new int[10]; // creates an array with 10 elements
myDigits = new int[5]; // creates an array with 5 elements
```

Note that you *can* combine both the array declaration and initialization. You might not want to if you first have to calculate or ask the user for the size of the array:

```
int n = 0;   // Variable used for number of elements in array
int grades [];      // Array declaration

Scanner sc = new Scanner(System.in);
System.out.println("How many courses have you taken?");
n = sc.nextInt();

// Creates an instance of the array where the user
// defines the size of the array making it dynamic.
grades = new int [n];
```

Looping through Arrays

Recall that arrays start at index 0 and finish at an index that is *one less* than the size of the array. For instance, an array of size 10 holds values at index 0 through 9. As a programmer, it's best to iterate through an array by using the *.length* attribute as follows:

```
for (int i = 0; i < grades.length; i++){
      System.out.print("Please enter a grade:");
      grades[i] = sc.nextInt();
}
```

You can also used an *enhanced for loop* to pass through an array.

```
int numPasses = 0;

for (int g : grades){      // Declares g as a temporary variable that will
                           // take on each value within grades.

      if (g >= 50) {       // Tests the current grade, g, for a pass.
            numPasses++;
      }
}
```

Problem Set 1

Instructions: Create a new Java project in Eclipse called *PS_ArrayReview* by clicking File -> New-> Java Project… Use the file naming conventions given below for each problem. To start a new problem, go to File -> New -> Class. Once your code is working, you will need to copy and paste it into the boxes. Remember to comment your code fully and be sure to include a header.

1. **(Random Number Stats)** Create a program that will randomly generate integer numbers 1 to 100 and store them in an array with size specified by the user (greater than or equal to 1).

Your program should then display:
a) The minimum integer value
b) The maximum integer value
c) The mean of all the values
d) The median value (BONUS – this will require a sorting algorithm)

Finally, have your program ask the user for a number between 1 and 100. Your program should then tell them how many times that number appears in the array.

Save your program as BasicStats.java.

Code (once your code is working, copy it into the box below):

```
        Scanner input = new Scanner (System.in);
                int size;
                int random;
                int min;;
                int max;
                double mean;
                double median;
                do {
                        System.out.println("Enter the size of the array:");
                        size =input.nextInt();
                        if (size<1) {
                                System.out.println("Invalid size!");
                                System.out.println("Please enter an array size greater or
equal than 1");
                        }
                }
                while (size<1);
                int [] array = new int [size];
                for (int x = 0; x < size; x++) {
                        random = (int) (1 + Math.random() * 100);
```

```
        array[x] = random;
}

//PART A: Display minimum integer value
min = array [0];
for (int x = 0; x < size; x++) {
        if (array[x] < min)
                min = array[x];
}
System.out.println("The minimum integer is " + min);

//PART B: Display maximum integer value
max = array [1];
for (int x = 0; x < size; x++) {
        if (array[x] > max)
                max = array[x];
}
System.out.println("The maximum integer is " + max);

//PART C: Display the mean of all the values
mean = 0;
for (int x = 0; x < size; x++) {
        mean = mean + array[x];
}
mean = mean/size;
System.out.println("The mean of all integers is " + mean);

//PART D: Display the median value
int temp;
for (int x = 0; x < size; x++) {
        for (int y = 0; y < array.length; y++) {
                if (array[x]<array[y]) {
                        temp = array[x];
                        array[x] = array[y];
                        array[y]= temp;
                }
        }
}

if (size%2 == 0) {
        median = (array[(size/2)-1] + array[size/2])/2.0;
        System.out.println("The median is " + median);
}
else {
        median = array[(size/2)];
        System.out.println("The median is " + median);
}
```

2. **(Shifting values)**
Write a program that fills an array with ten random integers between 1 and 100. Your program should then move the first integer to the last position in the array, and *slide* all other integers to the left by one position.
Save your program as FirstBecomesLast.java.

Sample Program Output:

Before transformation: 10, 87, 5, 46, 91, 67, 49, 20, 58, 88
After transformation: 87, 5, 46, 91, 67, 49, 20, 58, 88, 10

Save your Shift.java

Code:

```
int [] array = new int [10];
int [] shift = new int [10];
int random;

for (int x = 0; x<array.length; x++) {
        random = (int) (1 + Math.random() * 100);
        array[x] = random;
}

for (int x = 0; x+1<array.length; x++) {
        shift [x] = array[x+1];
}
shift [9] = array[0];

System.out.print("Before transformation: ");
for (int x = 0; x<array.length; x++) {
        System.out.print(array[x] + " ");
}

System.out.println();

System.out.print("After transformation: ");
for (int x = 0; x<array.length; x++) {
        System.out.print(shift[x] + " ");
}
```

3. Write a program that asks the user for an integer n > 0. Your program should then fill an array with *n* random integers between 1 and 20. Afterwards, move all the even numbers into a secondary array and the odd numbers into a third array. These arrays should be declared dynamically once you know how many elements should go in each (that is, don't just declare two more arrays of size *n*!).

Save your program as OddEven.java.

Sample Program Output (user input shown in ***bold italics***):

How many integers? ***5***
Your array: 20, 5, 1, 17, 13
Odd elements array: 5, 1, 17, 13
Even elements array: 20

Code:

```
        Scanner input = new Scanner (System.in); //Creates scanner

                int integer = 0;                                        //Stores user
input for array size
                int random;
//Generates random integers (1-20) to store into the array
                int evenCount = 0;                                      //Keeps track of
even integers in the array
                int oddCount = 0;                                       //Keeps track of
odd integers in the array

                //User input
                do {                                                            //Runs
until user enters an integer greater than 0
                        System.out.println("Enter an integer");//Prompts user for
integer
                        integer = input.nextInt();                  //Stores user integer
for array size
                        if (integer<0)                                  //Runs only if
user enters an invalid integer (less than or equal to 0)
                                System.out.println("Please enter an integer greater than
0");
                }
                while (integer<0);                                      //Checks if
integer is greater than 0
```

```
		int [] array = new int [integer];		//Declares array to integer size
		for (int x=0; x<integer; x++) {			//Initializes array to random integers
			random = (int) (1 + Math.random() * 20); //Generates a random integer
			array[x] = random;				//Stores random integer into array at index x
		}

		//Counts how many EVEN and ODD integers are in the array
		for (int x=0; x<integer; x++) {
			if (array[x]%2 ==0) {
				evenCount++;
			}
			else {
				oddCount++;
			}
		}

		//Declares even and odd arrays
		int [] evenArray = new int [evenCount];
		int [] oddArray = new int [oddCount];
		int e=0;
//Counter variable for even integers array
		int o=0;
//Counter variable for odd integers array

		for (int x=0; x<integer; x++) {			//For loop that runs until all integers are read
			if (array[x]%2 == 0) {				//Runs only if array contains an even integer
				while (e<evenCount) {			//Runs until even counter variable is less than the size of evenArray or evenCount
					evenArray[e] = array[x];		//Stores even integers from array to evenArray
					e++;
//Increments e by 1
					break;
//Exits while loop
				}
			}
			else {
//Runs only if array contains an odd integer
				while (o<oddCount) {			//Runs until odd counter variable is less than the size of oddArray or oddCount
```

```
                                        oddArray[o] = array[x];             //Stores odd
integers from array to oddArray
                                        o++;
//Increments o by 1
                                        break;
//Exits while loop
                                }
                        }
                }

                //Outputs the initial array
                System.out.print("Your array: ");
                for (int x=0; x<integer; x++)
                        System.out.print(array[x] + " ");
                System.out.println();

                //Outputs the odd array
                System.out.print("Odd elements array: ");
                for (int x=0; x<oddArray.length; x++)
                        System.out.print(oddArray[x] + " ");
                System.out.println();

                //Outputs the even array
                System.out.print("Even elements array: ");
                for (int x=0; x<evenArray.length; x++)
                        System.out.print(evenArray[x] + " ");
```