JAVA PROGRAMMING – QUIZ 1 (and Exercise 5)

WARNING. This test is to be treated as if taken under exam conditions. Any plagiarism or collusion will result in immediate disqualification and zero marks awarded, and will be pursued as a disciplinary matter.

As explained previously, part of the coursework consists of five multiple-choice quizzes, that are designed to test your understanding of Java. They appear in weeks 5, 6, 7, 8 and 11. Each one is worth 2% of the course unit mark, must be undertaken during a lab session, that is between 2.00-4.00pm on the day you attend. Each quiz can only be taken once, and must be completed within 30 minutes of starting. You will need a password to access the quiz, which will be announced during the lab.

The first of these quizzes is now available on Blackboard and should be taken NOW, before attempting the exercises below.

JAVA PROGRAMMING – EXERCISE 5

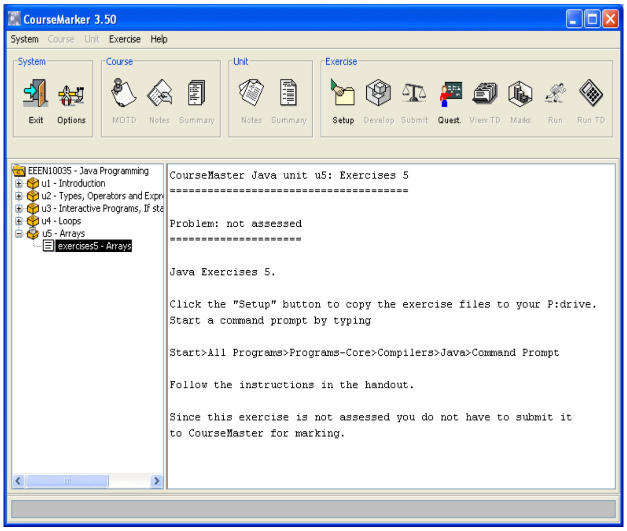
In this lab, we'll be dealing with arrays.

COURSE MARKER

1) Start Course Marker by selecting
Start>All Programs> CourseMarker >CourseMarker

After a few moments you will see the login window –
NOTE: User Name is your usual username (mch…), and the password is your student ID number:

2) Once you're in, access the relevant assignment by selecting:
EEEN10035 – Java Programming>u5 – Arrays:



```
CourseMaster Java unit u5: Exercises 5
======================================


Problem: not assessed
=====================


Java Exercises 5.

Click the "Setup" button to copy the exercise files to your P:drive.
Start a command prompt by typing

Start>All Programs>Programs-Core>Compilers>Java>Command Prompt

Follow the instructions in the handout.

Since this exercise is not assessed you do not have to submit it
to CourseMaster for marking.
```

3) Click the Setup button on top, and then click OK.
Clicking the Setup button copies the files needed for this exercise to the folder P:>mchXXYY> EEEN10035u5exercises5 (where mchXXYY is replaced with your username).

The files copied are:
ArrayTest1.java, ArrayTest2.java, ArrayTest3.java, ArrayTest4.java and ArrayTest5.java.
These are the java files you'll be working on this time.

4) That’s all we need from Course Marker, so now you can close it.

5) Everything we do from here on out will involve *notepad* and *command prompt*.

6) Open command prompt via,
Start>All Programs>Accessories>Command Prompt

In the resulting window type
P: *<press enter>*
and then type
cd mchXXYY\EEEN10035u5exercises5 *<press enter>*
(We’re just navigating to the folder where course marker copied all the necessary java files.)

7) Now, type the command **dir** and press enter.
You should see the following files in the folder: ArrayTest1.java, ArrayTest2.java, ArrayTest3.java, ArrayTest4.java and ArrayTest5.java.
Before we move on ahead, we need to learn about arrays.

ARRAYS

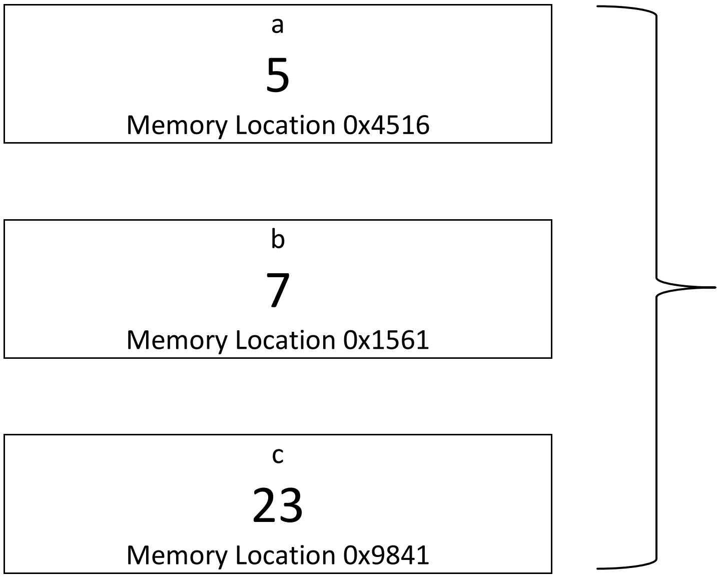
Simply put, an array is a set of variables.

Observe:

Declaring & initializing 3 variables (No arrays) :

```
int a=5,b=6,c=9;
```

//This declares three integer variables called a,b and c, & initializes them as 5, 7 and 23.



When these variables are stored in memory, they’re assigned a specific location. Notice that the memory locations of a,b, and c are quite random, and are not connected in any way.

Declaring & initializing an array, containing 3 elements:

```
int a [ ]={4,6,19};
```

// This declares an array ‘a’ (of integer variables), which can hold 3 numbers, initialized as 4,6 and 19!

a[0]	a[1]	a[2]
3	6	19
Memory Location 0x1862	Memory Location 0x1863	Memory Location 0x1864

- ***When an array is stored in memory, the variables are all connected, and they are in order (Look at the memory locations).***
- ***Every array starts from 0, not 1. If you have an array of size 50, the elements inside, range from 0 to 49. In this case, the elements inside are a[0], a[1] & a[2].***
- ***There is an alternative way of declaring arrays in Java where the [] brackets come before the array name: int [] a = {4,6,19}, but we’ll stick with our way for now.***

Just declaring an array, without initializing it:

```
int a [ ]= new int [50];
```

// This declares an empty array containing 50 cells!

You can initialize some or all elements of the array by typing out,
a[0]=5;
a[1]=88;
a[2]=67;
…….and so on.

Special Notes:

- All arrays have a built-in public field (or ‘attribute’) **length**, that holds the number of elements stored in the array, and it is accessed with the ‘dot’ operator, for example , if ‘x’ was an array of size 5, and ‘y’ was an ordinary int variable,

  y = x.length; *// y = 5, since the array ‘x’ comprises of 5 elements*

- A **for-loop** is commonly used to process the contents of the array.
  For example, if you want to find out the total sum of all the elements in an array,

```
public class Anything {
      public static void main(String args[])  {

                  int x [] ={5,10,19,3,2,4,6,2,22};
                  int sum = 0;
                  for (int i = 0; i < x.length; i++)  {
                        sum = sum + x[i];
                  }
                  System.out.println("The total = " + sum);
      }
  }
```

Task 1: *(you can use ConTEXT instead of Notepad in this and the following tasks)*
Open up ArrayTest1.java using notepad.
The following data items (all of type double) are to be stored in an array, their sum and average taken and printed out.
3.2, 2.4, 6.6, 7.3, 4.2, 9.9, 12.0, -3.2, -4.1, 5.4

Task 2:
Convert the above so that a ***while-loop*** is used instead.

Task 3:
Complete the class ArrayTest2. The following data is first stored in an integer array, from which the sum of the square roots is calculated and printed out:
12, 34, 17, 2, 96, 44, 72, 63

Task 4:
Complete the class ArrayTest3. The class stores the following data items in an integer array and stores the squares of the items in a *new* array:
415, 243, 755, 452, 676, -210, -45, 6, 82, -73

Modify the class to print out the maximum and minimum values of the squares stored in the new array.

Task 5:
Complete the class ArrayTest4, that stores the square roots of numbers 1-100 in an array. Each value should be printed out to three decimal places of accuracy*, such that the output appears as a list in descending order , i.e.
Square root of 100 = 10.000
Square root of 99 = 9.950
…

…
Square root of 2 = 1.414
Square root of 1 = 1.000

**use System.out.printf*

<u>TWO-DIMENSIONAL ARRAYS</u>

An array can be considered to be a row of elements.
A two-dimensional array has two rows.

The declaration of such an array contains *two* brackets before the array name, e.g.
int x [] [];

The array is initialized with a statement such as
x = new int [3] [3];
(i.e. it has 3 rows and 3 columns. Remember it's marked as 0,1,2.)

As with one-dimensional arrays, if the elements are known in advance, the array can be initialised at the time it is declared. A two-dimensional array can be initialised with a statement such as
int x [ ] [ ] = {{2, 6, 8}, {3, 3, 5}, {4,7,3}};

Individual elements can be addressed by row and column number, for example:

```
        int y = x [1][2];          // y = 5
```

Task 6:

Complete the class ArrayTest5, that stores the following data in an array **val**:
int val [ ][ ] = {{8,16,9,52}, {3,15,27,6}, {14,25,2,10}};
and adds up the square roots of all the elements and prints the total.

Hint: use nested for-loops.

When you have finished, save your work and logout.

Task 9:
There are some additional , "advanced", java exercises, see "Week 5" under Further Exercises on Blackboard. See how many you can complete.

END