

**MIT AITI Mobile Application
Development in Java
Nigeria, Summer 2012
Lab 04: Arrays**



Complete these problems and show your solutions to the instructors. Be prepared to explain your code.

Create a new class called `GradeBook`. Add a main method to `GradeBook`. You should put the code required to complete all the subsequent parts of this lab in the `main()` method. Separate each question's output with a separator like "-----" on a new line;

1. A class of students received scores of 81.4, 75.3, 90, 96.9, and 85.6. Declare and initialize an array named `scores` to store these values.
2. Write a loop to display the contents of `scores`. Make sure that your printout is readable with spaces or commas between each grade. For example:

```
The scores are: 81.4 75.3 90 96.9 85.6
```

3. Compute the average and maximum scores and display it to the user. If the average grade is 85.84, and the maximum is 96.8 the output should be:

```
Average score: 85.84 Maximum score: 96.8
```

4. A grade of 90 or above is an A, 80 or above is a B, 70 or above is a C, 60 or above is a D, and anything lower is an F. Write a loop to print out each student's letter grade. The output should be of the form:

```
Letter grades: B C A A B
```

- Write code to output a histogram plot of the grade distribution using an 'X' to represent each person receiving that grade. For example, if there were three A's, four B's, two C's, no D's and one F, your code output should be:

```
A: XXX
B: XXXX
C: XX
D:
F: X
```

- Using a loop, increment all the grades by an extra credit amount of 5% and print the scores out as you did in problem 3. Be sure that the grades themselves are modified and you are not just displaying the modified values.
- Write code to sort and display the contents of `scores` in increasing order. Your code should work regardless of the size of `scores` or its contents. Test your code on arrays of other sizes to make sure that it works. The output should be of the form:

```
The sorted order is: 72.6 82.4 86.1 90.0 96.8
```

You can use an algorithm called *Selection Sort*. Selection sort loops over the array and calculates the maximum element from the elements it has not sorted yet. It has a nested structure with a `for` loop nested in another `for` loop:

```
for (int i = array.length; i >= 1; i--) {
    //find the maximum value in the array [0..i]
    ...
    //swap array[i] with array[currentMaxIndex]
    ...
}
```

Make sure that you understand how the above code works. Ask us if you have any questions.

- Make up a few other arrays of scores, and store all of these arrays in a multi-dimensional array. Each of these will represent the grades for one particular person. Then print out the 2D array of scores:

Student 1: 90.1 100 50.5 40.6 80.9

Student 2: 100 89.6 90.5 52.6 99.0

Student 3: 98.6 99 22.2 105 78