**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE110** | **Credits: 1.5** |
| **Course Name: Programming Language I** | **Semester: Fall’18** |

**Lab 10  
Arrays II**

1. **Topic Overview:**

This week we will not be starting any new topics. Instead, we will continue working on the array problems from last week, and a few additional problems. Student will learn sorting, uses and manipulation of arrays in details

1. **Lesson Fit:**

There is pre-requisite to this lab. However, it is a practical lesson for the theory covered in the theory class

1. **Learning Outcome:**

After this lecture, the students will be able to:

* 1. Visualize how array works
  2. Detect when we need arrays
  3. Sort an array

1. **Anticipated Challenges and Possible Solutions**
   1. Students will start the array index from 1, as it is natural to start counting from 1. Moreover, they will have a lot of syntax error in declaring arrays, manipulating them.
   2. Task 4: students will face difficulty understanding sorting algorithm. Nested loop in array will give them hard times visualizing what is actually happening.

**Solutions:**

* + 1. The instructor should write on whiteboard and show the students how to declare and initialize arrays of different data types. He/she should also teach them the default values of different data types while creating an array.
    2. Instructor will explain the algorithm first and show the students how selection sort (easiest one to visualize) works. He/she should simulate the sorting algorithm for at least 3 index so that the students have a better visual understanding how sorting is done.

1. **Acceptance and Evaluation**

Students will show their progress as they complete each problem. They will be marked according to their class performance. Their maybe students who might not be able to finish all tasks they will submit them later and give a viva to get their performance mark.

1. **Activity Detail**
   1. **Hour: 1  
      Discussion:**Explain syntax of declaring arrays of different data types, how indexing works, how to print or access array elements.  **Problem Task:**
      1. Task 1 to 3
   2. **Hour: 2**

**Discussion:**

Explaination of sorting algorithm (Selection sort, if possible bubble sort too). In both ascending and descending order.

**Problem Task:**

* 1. **Hour: 3**

Students are now capable of solving different problems by themselves. They will solve the remaining problems and if they face specific problem, they will seek help from instructors and instructors will help them solving their issues but they will not explain how to solve the problem.

**Discussion:**

Explanation of printf() method and its necessity.

**Problem Task:**

* + 1. Task 6 to 10

1. **Home tasks**a. Unfinished task

**Lab 10 Activity List**

**Task 1**

Write a program which reads 5 numbers into an array and prints the largest number.

If the user enters 7, 13, 2, 10, 6 then your program should print 13.

**Task 2**

Write a program which reads 5 numbers into an array and prints the largest number and its location in the array.

If the user enters 7, 13, 2, 10, 6 then your program should print “largest number 13 was found at location 1”.

**Task 3**

Write a program which reads 5 numbers into an array and prints the smallest and largest number and their location in the array.

If the user enters 7, 13, -5, 10, 6 then your program should print

“Smallest number -5 was found at location 2”.

“Largest number 13 was found at location 1”.

**Task 4**

Write a program which reads 5 numbers into an array, sorts/arranges the numbers from low to high and prints all numbers in the array.

If the user enters 7, 13, 2, 10, 6 then your program should print 2, 6, 7, 10, and 13.

**Task 5**

Write a program which reads 5 numbers into an array, sorts/arranges the numbers from high to low and prints all numbers in the array.

If the user enters 7, 13, 2, 10, 6 then your program should print 13, 10, 7, 6, 2.

**Task 6**

Write a program which asks the user how many numbers to take. Then takes that many numbers and prints the median value. Read http://www.mathsisfun.com/median.html

If the user gives 10, 50, 40, 20, 30. Then the median is 30 (because 30 falls in middle 10, 20, **30**, 40, 50)

If the user gives 30, 10, 40, 20. Then the median is 25 because, (20+30)/2=25(average of two middle values from 10, **20**, **30**, 40)

**Task 7**

Write a java program that reads 10 numbers from the user. Write the program in such a way so that

If the user enters 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, then the output should be 10, 30, 50, 70, 90, 20, 40, 60, 80, 100.

If the user enters 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, then the output should be 1, 3, 5, 7, 9, 2, 4, 6, 8, 10.

If the user enters 2, 5, 6, 9, 12, 13, 14, 15, 16, 17 then the output should be 2, 6, 12, 14, 16, 5, 9, 13, 15, 17.

**Task 8**

Create a String array (size 10) consisting of the words “zero”, “one”, “two”……, “nine”. Then take a number (between 0 and 9) from the user and print that number in words from the array. If the user enters 5, you should print a[5] and output should be “five”.

**Task 9**

Write a java program that reads 15 numbers from the user, all the numbers within the range 0-9. Then print the number of times each number has been entered by the user. You can try this problem in at least 3 ways:

Task 9a) Using nested loop to search within array for each number between 0-9

Task 9b) using ten variables as counter/tally: zerCount, oneCount …., nineCount to count during input

Task 9c) Modify task 9b and use a 2nd array as the counter instead of ten variables

**Task 10**

Read from the following link and try to use printf () for all variable types you know

http://web.cerritos.edu/jwilson/SitePages/java\_language\_resources/Java\_printf\_method\_quick\_reference.pdf

Change the following System.out.println() to a System.out.printf() method.

**Current output:**5.984807753012208

**Desired output after the change**: 5.9848

double z;

z = 5+ Math.sin(80\*Math.PI/180);

System.out.println(z);