Reading:

1. Deitel & Deitel, Chapter 7

Objectives:

1. To learn how to declare and initiate Arrays
2. To be able to search arrays
3. To be able to sort arrays using Bubble, binary, or linear sort
4. To be able to perform operations on a standard java array

Concepts:

1. Java Arrays
2. Overview of Lab 2

Outline:

1. Introduction
2. Arrays
   a. Declaring and creating Arrays
      - Examples using arrays
   b. Sorting Arrays
      - Bubble
   c. Searching Arrays
      - Linear, and binary
3. Multidimensional Arrays

Reference:

1. Arrays

- Data Structures
- Related data items of same type
- Remain same size once created
  - Fixed-length entries
- Group of variables
  - Same type
- Index
  - Also called subscript
  - Position number in square brackets
  - Positive integer or integer expression

- Example
  
  ```java
  int a = 5;
  int b = 6;
  c[ a + b ] += 2;
  // Adds 2 to c[ 11 ]
  
  int c[] = new int[12];
  ```

- Declaring and Creating arrays
- Arrays are objects that occupy memory
- Created dynamically with keyword new

  ```java
  String b[] = new String[100];
  ```

- Sorting data
  - Attracted intense research in computer-science field
  - Bubble sort
    - Smaller values “bubble” their way to top of array
    - Larger values “sink” to bottom of array
    - Use nested loops to make several passes through array
    - Each pass compares successive pairs of elements
    - Pairs are left alone if increasing order (or equal)
    - Pairs are swapped if decreasing order

- Searching
  - Finding elements in large amounts of data
  - Determine whether array contains value matching key value
  - Linear search
    - Compare each array element with search key
    - If search key found, return element index
    - If search key not found, return –1 (invalid index)
- Works best for small or unsorted arrays
- Inefficient for larger arrays

  o Binary Search
  - Efficient for large, sorted arrays
  - Eliminates half of the elements in search through each pass
  - Compare middle array element to search key
  - If element equals key, return array index
  - If element is less than key, repeat search on first half of array
  - If element is greater then key, repeat search on second half of array
  - Continue search until element equals search key (success)
  - Search contains one element not equal to key (failure)

2. Multidimensional arrays
   - Example
     Tables with rows and columns
     Two-dimensional array
     Declaring two-dimensional array b[2][2]
     \[
     \begin{bmatrix}
     1 & 2 \\
     3 & 4 \\
     \end{bmatrix}
     \]
     - 1 and 2 initialize b[0][0] and b[0][1]
     - 3 and 4 initialize b[1][0] and b[1][1]
     \[
     \begin{bmatrix}
     1 & 2 \\
     3 & 4 & 5 \\
     \end{bmatrix}
     \]
     - row 0 contains elements 1 and 2
     - row 1 contains elements 3, 4 and 5

   - Creating multidimensional arrays
     o Can be allocated dynamically
     - 3-by-4 array
       \[
       \begin{bmatrix}
       \end{bmatrix}
       \]
       - Rows can have different number of columns

       \[
       \begin{bmatrix}
       \end{bmatrix}
       \]
       - Allocate rows
         b[ 0 ] = new int[ 5 ]; // allocate row 0
         b[ 1 ] = new int[ 3 ]; // allocate row 1
Objectives:

1. Demonstrate knowledge of arrays
2. To perform a set of operations on an array

Assignment:
Write a Java program that does the following:
1. Create an integer array with the following values {1,5,2,-2,20,3}.
2. Perform the following operations on the array. After each operation, output all the values and indices of the array.
   a. Set the 5th value of the array equal to 200.
   b. Swap the 2nd and the 6th value of the array.
   c. Subtract 8 from the 7th value of the array.
   d. Find the index of the value that is equal to -2 by searching the array using the binary or linear sort.
   e. Sort the values of the array in increasing order using a bubble sort

Your code should be clear and well commented. You are graded on neatness as well as correctness.
Objectives:

3. Demonstrate knowledge of arrays
4. To perform a set of operations on an array

Assignment:
Write a Java program that does the following:
3. Create an integer array with the following values {1,5,2,-2,20,3}.
4. Perform the following operations on the array. After each operation, output all the values and indices of the array.
   a. Set the 5th value of the array equal to 200.
   b. Swap the 2nd and the 6th value of the array.
   c. Subtract 8 from the 7th value of the array.
   d. Find the index of the value that is equal to -2 by searching the array using the binary or linear sort.
   e. Sort the values of the array in increasing order using a bubble sort

Your code should be clear and well commented. You are graded on neatness as well as correctness.

```java
public class lab4solution {
    double[] array = {1,5,2,-2,20,3};
    public static void main (String[] args){
        //a
        array[4] = 200;
        System.out.println("Array values and indices");
        for(int i=0; i< array.length; i++){
            System.out.println(i+" : "+array[i]);
        }
        //b
        int temp = array[1];
        array[1] = array[5];
        array[5] = temp;
        System.out.println("Array values and indices");
        for(int i=0; i< array.length; i++){
            System.out.println(i+" : "+array[i]);
        }
        //c
        array[7] = array[7]-8;
        System.out.println("Array values and indices");
        for(int i=0; i< array.length; i++){
            System.out.println(i+" : "+array[i]);
        }
        //d
        int index = 0;
        for(int i=0; i<array.length; i++){
```
if(array[i] == -2) index = i;
}

System.out.println("Array values and indices");
for(int i=0; i< array.length; i++){
    System.out.println(i+" : "+array[i]);
}

//e
int out, in;
for(out=array.length-1; out>1; out--)
    for(in=0; in<out; in++)
        if( array[in] > array[in+1] ){
            int temp = array[in];
            array[in] = array[in+1];
            array[in+1] = temp;
        }
}

System.out.println("Array values and indices");
for(int i=0; i< array.length; i++)
    System.out.println(i+" : "+array[i]);
}