

Statement Purpose:

Purpose of this Lab is to familiarize the students with the use of elementary data structure-Arrays. Another aim is to teach the students how to implement one dimensional and two dimensional arrays. The students are given small tasks related to 1-D and 2-D arrays which they complete during the lab session under the supervision of the lab instructor. This helps them understand the concepts well which they learn in their lectures.

Activity Outcomes:

The students will learn how to write methods related to some actions performed on 1-D and 2-D arrays. They will understand how to write simple methods to

- calculate average of all elements of 1-D array and display how many elements are below and how many are above this average
- count and display how many elements in 1-D array are even and how many are odd
- find and swap the largest and smallest elements in 1-D array
- find and display sum of each row and each column of 2-D array
- display elements of row which has the largest sum of elements
- display row(s) which contain(s) repeated elements

Theory Review (N/A):

Since an array is the basic data structure, the students are already familiar with in their previous courses, nothing can be added here as a review of this data structure. We shall practice solving some exercises related to 1-D and 2-D arrays.

Practice Activity with Lab Instructor: (10 Minutes)

Problem Statement 1:

We want to create a program that will randomly select four cards from a deck of 52 cards. Card numbers **0** to **12**, **13** to **25**, **26** to **38**, and **39** to **51** represent 13 Spades, 13 Hearts, 13 Diamonds, and 13 Clubs, respectively. **cardNumber / 13** determines the suit (i.e. name) of the card and **cardNumber % 13** determines the rank (i.e. number) of the card. After shuffling the array **deck**, pick the first five cards from **deck**. The program should display the cards from these four card numbers.



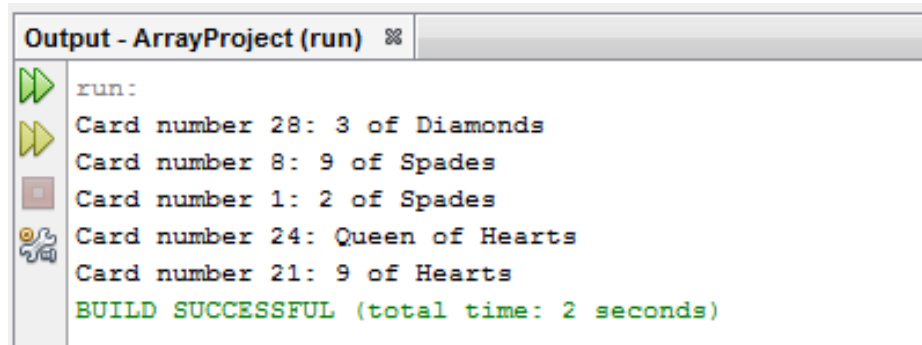
Solution:

1. Create a project with the name “ArrayProject”
2. Inside this project, create a main class with the name “DeckOfCards” and add the following code in this class

```
Start Page  DeckOfCards.java
Source History
import java.util.Scanner;
2 public class DeckOfCards {
3     public static void main(String[] args) {
4         //Create an array deck
5         int[] deck = new int[52];
6         //Create an array of Suits of cards
7         String[] suits = {"Spades", "Hearts", "Diamonds", "Clubs"};
8         //Create an array of rank of cards
9         String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};
10
11        // Initialize the cards
12        for (int i = 0; i < deck.length; i++)
13            deck[i] = i;
14
15        // Shuffle the cards
16        for (int i = 0; i < deck.length; i++) {
17            // Generate an index randomly
18            int index = (int) (Math.random() * deck.length);
19
20            //Swap card at index 'i' with card at index 'index'
21            int temp = deck[i];
22            deck[i] = deck[index];
23            deck[index] = temp;
24        }
25
26        // Display the first five cards
27        for (int i = 0; i < 5; i++) {
28            String suit = suits[deck[i] / 13];
29            String rank = ranks[deck[i] % 13];
30            System.out.println("Card number " + deck[i] + ": " + rank + " of " + suit);
31        }
32    }
33 }
```

3. When you run this program, it will display the following output on the screen:





```
Output - ArrayProject (run) %
run:
Card number 28: 3 of Diamonds
Card number 8: 9 of Spades
Card number 1: 2 of Spades
Card number 24: Queen of Hearts
Card number 21: 9 of Hearts
BUILD SUCCESSFUL (total time: 2 seconds)
```

Problem Statement 2:

We want to create a program that will store random integers between 0 and 99 in a two dimensional array. The program should find and display sum of all elements stored in the array.

Solution:

4. Create a project with the name “ArrayProject2”
5. Inside this project, create a main class with the name “SumElements” and add the following code in this class



```
Start Page 88 SumElements.java 88
Source History
1 import java.util.Scanner;
2 public class SumElements {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         System.out.print("Enter number of rows: ");
6         int m = sc.nextInt();
7         System.out.println();
8         System.out.print("Enter number of columns: ");
9         int n = sc.nextInt();
10        System.out.println();
11        //Create a two dimensional array
12        int[][] matrix = new int[m][n];
13        //Initialize array with random values from 0 to 99
14        System.out.println("Entering random numbers in matrix of " + matrix.length + " rows and " + matrix[0].length + " columns: ");
15        for (int row = 0; row < matrix.length; row++) {
16            for (int column = 0; column < matrix[0].length; column++) {
17                matrix[row][column] = (int)(Math.random() * 100);
18            }
19        }
20        System.out.println();
21        //Print the elements of the array
22        System.out.println("The elements of two dimensional array with rank "+m+"X"+n+" are as follows:");
23        for (int row = 0; row < matrix.length; row++) {
24            for (int column = 0; column < matrix[0].length; column++) {
25                System.out.print("Matrix("+row+", "+column+")= "+matrix[row][column]+"\\t\\t");
26            }
27            System.out.println();
28        }
29        //Summing all elements
30        int total = 0;
31        for (int row = 0; row < matrix.length; row++) {
32            for (int column = 0; column < matrix[row].length; column++) {
33                total += matrix[row][column];
34            }
35        }
36        System.out.println();
37        System.out.println("The sum of all elements of the above array is "+total);
38    }
39 }
```

6. When you run this program, it will display the following output on the screen:



Output - ArrayProject2 (run) %

```
run:
Enter number of rows: 5
Enter number of columns: 4
Entering random numbers in matrix of 5 rows and 4 columns:

The elements of two dimensional array with rank 5X4 are as follows:
Matrix(0,0)= 32      Matrix(0,1)= 72      Matrix(0,2)= 48      Matrix(0,3)= 82
Matrix(1,0)= 41      Matrix(1,1)= 62      Matrix(1,2)= 21      Matrix(1,3)= 82
Matrix(2,0)= 93      Matrix(2,1)= 48      Matrix(2,2)= 97      Matrix(2,3)= 53
Matrix(3,0)= 50      Matrix(3,1)= 54      Matrix(3,2)= 39      Matrix(3,3)= 6
Matrix(4,0)= 69      Matrix(4,1)= 64      Matrix(4,2)= 1       Matrix(4,3)= 54

The sum of all elements of the above array is 1068
BUILD SUCCESSFUL (total time: 3 seconds)
```



LAB EXERCISES: (70 Minutes)

1. Write code in the method BelowAboveAvg() in "ArrayExercises" class which first finds the average of all random numbers entered in an array and then finds and displays the number of elements less than and greater than the average.

Sample Run:

```
Output - Lab1ExercisesSolutions (run) %
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average |
2. Find how many elements are even and how many are odd  |
3. Swap largest and smallest elements in the array       |
4. Summing elements in each row and each column of 2-D array |
5. Find row of 2-D array which has the largest sum       |
6. Display row(s) of 2-D array containing repeated elements |
7. Quit                                                  |
-----

> Enter your choice: 1
How many elements of the array?
15
Elements of the array are as follows:
87, 14, 33, 31, 67, 56, 81, 85, 77, 20, 77, 62, 12, 1, 24,
The average of all elements of the array is 48

The number of elements above average is: 8
The elements above average are as follows:
87, 67, 56, 81, 85, 77, 77, 62,
The number of elements below average is: 7
The elements below average are as follows:
14, 33, 31, 20, 12, 1, 24,
```



2. Write code in the method CountEvenOdd() in "ArrayExercises" class which finds and displays the number of even and odd elements present in the array.

Sample Run:

```
Output - Lab1ExercisesSolutions (run) %
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average |
2. Find how many elements are even and how many are odd   |
3. Swap largest and smallest elements in the array        |
4. Summing elements in each row and each column of 2-D array |
5. Find row of 2-D array which has the largest sum       |
6. Display row(s) of 2-D array containing repeated elements |
7. Quit                                                  |
-----

> Enter your choice: 2
How many elements of the array?
15
Elements of the array are as follows:
98, 49, 7, 35, 84, 2, 66, 67, 25, 79, 4, 47, 48, 11, 62,

The number of even elements in the array is: 7

The number of odd elements in the array is: 8

Even elements of the array are as follows:
98, 84, 2, 66, 4, 48, 62,
Odd elements of the array are as follows:
49, 7, 35, 67, 25, 79, 47, 11,
```



3. Write code in the method SwapLargestSmallest() in "ArrayExercises" class which swaps largest and smallest elements in the array.

Sample Run:

```
Output - Lab1ExercisesSolutions (run) %  
run:  
*****  
***** Array Exercises *****  
*****  
1. Find how many elements are below and above the average |  
2. Find how many elements are even and how many are odd |  
3. Swap largest and smallest elements in the array |  
4. Summing elements in each row and each column of 2-D array |  
5. Find row of 2-D array which has the largest sum |  
6. Display row(s) of 2-D array containing repeated elements |  
7. Quit |  
-----  
  
> Enter your choice: 3  
How many elements of the array?  
15  
  
Elements of the array before swapping largest and smallest elements are as follows:  
12, 35, 10, 42, 29, 75, 43, 21, 12, 51, 14, 16, 47, 76, 98,  
Largest element of the array is 98  
  
Smallest element of the array is 10  
  
Elements of the array after swapping largest and smallest elements are as follows:  
12, 35, 98, 42, 29, 75, 43, 21, 12, 51, 14, 16, 47, 76, 10,
```



4. Write code in the method SumRowsColumns() in "ArrayExercises" class which finds and displays sum of each row and each column of the array.

Sample Run:

```
Output - Lab1ExercisesSolutions (run) %
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average |
2. Find how many elements are even and how many are odd   |
3. Swap largest and smallest elements in the array         |
4. Summing elements in each row and each column of 2-D array |
5. Find row of 2-D array which has the largest sum         |
6. Display row(s) of 2-D array containing repeated elements |
7. Quit                                                    |
-----

> Enter your choice: 4
Enter number of rows: 4

Enter number of columns: 5

Entering random numbers in matrix of 4 rows and 5 columns:
The elements of array with the sum of rows and columns are as follows:

      88      69      8      97      29      Sum: 291
      58      36      8       4      14      Sum: 120
      33      75      37      63      74      Sum: 282
      78      30      62       4      75      Sum: 249
Sum:    257     210     115     168     192
```



5. Write code in the method LargestSumRow() in "ArrayExercises" class which displays all row elements that has largest sum of its elements.

Sample Run:

```
Output - Lab1ExercisesSolutions (run) %
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average |
2. Find how many elements are even and how many are odd  |
3. Swap largest and smallest elements in the array        |
4. Summing elements in each row and each column of 2-D array |
5. Find row of 2-D array which has the largest sum        |
6. Display row(s) of 2-D array containing repeated elements |
7. Quit                                                    |
-----
> Enter your choice: 5
Enter number of rows: 4

Enter number of columns: 5

Entering random numbers in matrix of 4 rows and 5 columns:

The array elements are as follows:

60      22      71      88      28
79      75      70      12      85
18      21      19      85      96
28      7       27      4       78
Row 1 has the maximum sum of 321
Its elements are as follows:
79      75      70      12      85
```



6. Write code in the method RepeatedElementsRow() in "ArrayExercises" class which displays row(s) which contain(s) repeated elements.

Sample Run 1:

```
Output - Lab1ExercisesSolutions (run) %
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average      |
2. Find how many elements are even and how many are odd       |
3. Swap largest and smallest elements in the array            |
4. Summing elements in each row and each column of 2-D array  |
5. Find row of 2-D array which has the largest sum            |
6. Display row(s) of 2-D array containing repeated elements   |
7. Quit                                                        |
-----

> Enter your choice: 6
Enter number of rows: 6

Enter number of columns: 6

Entering random numbers in matrix of 6 rows and 6 columns:
The array elements are as follows:

28      80      70      19      39      97
73       5      30      63      53      24
74      58      22       8      14      60
27      24      22      33      47      75
1       42      46      79      41      76
66      56      45      76      83      78

No row contains duplicate elements
```



Sample Run 2:

```
run:
*****
***** Array Exercises *****
*****
1. Find how many elements are below and above the average |
2. Find how many elements are even and how many are odd  |
3. Swap largest and smallest elements in the array       |
4. Summing elements in each row and each column of 2-D   |
5. Find row of 2-D array which has the largest sum       |
6. Display row(s) of 2-D array containing repeated       |
7. Quit                                                  |
-----

> Enter your choice: 6
Enter number of rows: 6

Enter number of columns: 6

Entering random numbers in matrix of 6 rows and 6 columns:
The array elements are as follows:

92      50      1      1      20      25

12      32      1      68      73      1

19      0      36      0      77      55

97      40      42      51      58      16

25      96      22      19      4       55

4       96      43      75      37      69

Row(s) containing repeated elements
      92      50      1      1      20      25

      12      32      1      68      73      1

      19      0      36      0      77      55
```

