JOBSHEET 9
Array 2

1. Objective

- Students are able to understand two-dimensional array creation
- Students are able to access array elements in Java programs along with enrichment of case studies in a matrix operation

2. Laboratory

2.1 Experiment 1: Declare, Initialize, and Display 2-Dimensional Array

1. Create a new project
2. Create a new class, name it **Arr1**
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Create an array of integer type named **number** with a row capacity of 2 elements and a column of 3 elements

```
int[][] number = new int[2][3];
```

5. Fill in each element of the value array as follows:

```
number[0][0] = 12;
number[0][1] = 14;
number[0][2] = 34;
number[1][0] = 20;
number[1][1] = 24;
number[1][2] = 67;
```

6. Display all contents of the elements to the screen

```
System.out.println(number[0][0] + " " + number[0][1] + " " + number[0][2]);
System.out.println(number[1][0] + " " + number[1][1] + " " + number[1][2]);
```

7. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
12 14 34
20 24 67
```

Questions!

1. Should the array elements be filled sequentially? Explain!
2. In step 5, modify the code so that the filled elements are only array elements in odd row positions! Can this be done? Prove it!

2.2Experiment 2: Display 2 Dimensional Array Elements Using Loop

1. Create a new class, name it **Arr2**
2. Write the basic structure of the Java programming language which contains the **main()** function
3. Create an array of integer type named **number** with a row capacity of 2 elements and a column of 3 elements

```
int[][] number = new int[2][3];
```

8. Fill in each element of the value array as follows:

```
number[0][0] = 12;
number[0][1] = 14;
number[0][2] = 34;
number[1][0] = 20;
number[1][1] = 24;
number[1][2] = 67;
```

4. Using a loop, display all the contents of the elements from the **number** array

```
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
        System.out.print(number[i][j] + " ");
    }
    System.out.println("");
}
```

5. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
12 14 34
20 24 67
```

Questions!

1. How many columns was the array in Experiment 2? Change the number of columns to 4 so that the array declaration and instantiation looks like the following code

```
int[][] number = new int[2][4];
```

   Then, fill in the array elements with any value, corresponding to the addition of these columns. Run the program again, what happened?

2. In step 5, change the program code as follows

```
for (int i = 0; i < number.length; i++) {
    for (int j = 0; j < number[0].length; j++) {
        System.out.print(number[i][j] + " ");
    }
    System.out.println("");
}
```

   Run the program after the change, what happened?

3. Regarding displaying all array elements, change the program code to display array elements as follows

```
for (int array[] : number) {
    for (int r : array) {
        System.out.print(r + " ");
    }
    System.out.println("");
}
```

Run the results of these changes, what happened?

2.3 Experiment 3: Filling in 2 Dimensional Array Elements via Keyboard

1. Create a new class, name it **Arr3**
2. Write the basic structure of the Java programming language which contains the **main()** function
3. Add the Scanner library
4. Make a **Scanner** declaration with the name **input**
5. Create an array of integer type named **number** with a row capacity of 2 elements and a column of 3 elements

```
int[][] number = new int[2][3];
```

6. Using a loop, create an input to fill in the **number** array element

```
for (int i = 0; i < number.length; i++) {
    for (int j = 0; j < number[0].length; j++) {
        System.out.print("Enter a number [" + i + "][" + j + "]: ");
        number[i][j] = input.nextInt();
    }
    System.out.println("----------------");
}
```

7. Using a loop, display all the contents of the elements from the **number** array

```
for (int i = 0; i < number.length; i++) {
    for (int j = 0; j < number[0].length; j++) {
        System.out.print(number[i][j] + " ");
    }
    System.out.println("");
}
```

8. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
Enter a number [0][0]: 7
Enter a number [0][1]: 3
Enter a number [0][2]: 9
---------------
Enter a number [1][0]: 11
Enter a number [1][1]: 4
Enter a number [1][2]: 2
---------------
7 3 9
11 4 2
```

Questions!

1. In step 6 can position j be replaced with position i? Explain!
2. Add program code to determine the number of rows and columns of array elements dynamically (rows and columns are determined when the program runs through the keyboard)!
3. Modify the program code to display array elements using **foreach**!

3. Assignment

1. Create a program that has two arrays as follows:
   - The first array is a one-dimensional array **char CODE [10]**, containing the license plate codes
   - The second array is a two-dimensional array containing the city names which are paired with the license plate codes

   The illustration of the array display is as follows:

A	B	A	N	T	E	N				
B	J	A	K	A	R	T	A			
D	B	A	N	D	U	N	G			
E	C	I	R	E	B	O	N			

F
G
H
L
N
T

B	O	G	O	R					
P	E	K	A	L	O	N	G	A	N
S	E	M	A	R	A	N	G		
S	U	R	A	B	A	Y	A		
M	A	L	A	N	G				
T	E	G	A	L					

2. Create a program containing a two-dimensional array having the row and column sizes obtained from keyboard input. Then, make input to fill the array elements. Next, make a menu choice that consists of:
   a. **(MIN Value)**. Display the value of the smallest array element to the screen
   b. **(MIN Value & Amount)**. Display to the screen the smallest value and how many the smallest value is, and also display the row and column location of the minimum value.
   c. **(Array conditions)**. Display the word "FOUND" on the screen if there is a value of 50 between the two-dimensional array elements, otherwise print the word "NOT FOUND".