



Name : Rizqi Dewanto Y  
Number : 2341760176  
Class : 1-G  
Material : Jobsheet 9

## 2.1 Experiment 1: Fill in Array Element

### Experiment Time: 20 minutes

1. Open a text editor, create a new Java class with the name **arrayNumbersXX**.  
(XX=student ID number)
2. Write the basic structure of the Java programming language which contains the **main()** function
3. Create an array of integer type named **num** with a capacity of 4 elements
4. Fill each element of the array with numbers 5, 12, 7, 20
5. Display all contents of the elements to the screen
6. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
D: > polinema > Daspro.semester1 > Jobsheet > Jobsheet9 > arrayNumbers20.java > arrayNumbers20 > main(String[])

1  /**
2   * arrayNumbers20
3   */
4  public class arrayNumbers20 {
5
6      Run | Debug
7      public static void main(String[] args) {
8          int [] num = new int [4] ;
9          num[0] = 5;
10         num[1] = 12;
11         num[2] = 7;
12         num[3] = 20;
13
14         System.out.println(num [0]);
15         System.out.println(num [1]);
16         System.out.println(num [2]);
17         System.out.println(num [3]);
18     }
19 }
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\polinema\Daspro.semester1\Jobsheet> cd Jobsheet9
PS D:\polinema\Daspro.semester1\Jobsheet\Jobsheet9> javac arrayNumbers20.java
PS D:\polinema\Daspro.semester1\Jobsheet\Jobsheet9> java arrayNumbers20.java
5
12
7
20
```

7. Commit and push the changes to GitHub

## Questions!

1. If the contents of each element of the array **num** are changed with numbers 5.0, 12867, 7.5, 2000000. What happens? How can it be like that?
2. Modify the program code by initializing the array elements at the same time when declaring the array.

```
D: > polinema > Daspro.semester1 > Jobsheet > Jobsheet9 > J arrayNumbersModify20.java > arrayNumbersModify20 > main(String[])

1  /**
2   * arrayNumbersModify20
3   */
4  public class arrayNumbersModify20 {
5
6      Run | Debug
7      public static void main(String[] args) {
8          double num[] = {5.0, 7.5, 12867, 7.5, 2000000 };
9          for ( double p = 0; p <= 4; p++) {
10             System.out.println(num[(int)p]);
11         }
12         // double [] num = new double[4] ;
13         // num[0] = 5.0;
14         // num[1] = 12867;
15         // num[2] = 7.5;
16         // num[3] = 2000000;
17
18         // System.out.println(num [0]);
19         // System.out.println(num [1]);
20         // System.out.println(num [2]);
21         // System.out.println(num [3]);
22     }
23 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL

```
ress=localhost:56005' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\...\vscode\ws_30dc8\jdt_ws\jdt.ls-java-project\bin' 'arrayNumbersModify20'
5.0
7.5
12867.0
7.5
2000000.0
```

3. Change the statement in step 6 to be like this  
What is the result? How can it be like that?  
I added a typecast (int) to convert the loop variable p to an integer before using it as an array index. This is necessary because array indices must be integers in Java.
4. If the condition in the for-loop statement is changed to i <= 4, what is the output of the program? Why is the result like that?  
This is due to the loop's iteration over each number between 1 and 4, which prints the current number as well as the maximum, which stays at 4 for the duration of the loop. The loop's

iterations are guaranteed to contain the number 4 thanks to the  $i \leq 4$  condition.

5. Commit and push the changes to GitHub.

## 2.2 Experiment 2: Requesting User Input to Fill in an Array Element

**Experiment Time: 40 minutes**

1. Open a text editor, create a Java file then save it with the name **arrayValueXX**. (XX=student ID number)
2. Write the basic structure of the Java programming language which contains the **main()** function
3. Add the Scanner library
4. Create an array of integer type with the name **finalScore**, with a capacity of 10 elements
5. Using a loop, create an input to fill in the **finalScore** array element
6. Using a loop, display all the contents of the elements from the **finalScore** array

```
3  /**
4   * arrayValue20
5   */
6  public class arrayValue20 {
7
8      Run | Debug
9      public static void main(String[] args) {
10         Scanner arrayValue20 = new Scanner(System.in);
11
12         int[] finalScore = new int [10] ;
13         for (int i = 0; i < 10 ; i++) {
14             System.out.print("Enter the final score" + i + " : ");
15             finalScore[i] = arrayValue20.nextInt();
16         }
17     }
18 }
```

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

**Compile**

```
PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ACER> CD\
PS C:\> cd D:
PS D:\> cd polinema
PS D:\polinema> cd daspro.semester1
PS D:\polinema\daspro.semester1> cd Jobsheet
PS D:\polinema\daspro.semester1\Jobsheet> cd Jobsheet9
PS D:\polinema\daspro.semester1\Jobsheet\Jobsheet9> javac arrayValue20.java
PS D:\polinema\daspro.semester1\Jobsheet\Jobsheet9> java arrayValue20
Enter the final score0 : █
```

## Run the program

```
PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ACER> CD\
PS C:\> cd D:
PS D:\> cd polinema
PS D:\polinema> cd daspro.semester1
PS D:\polinema\daspro.semester1> cd Jobsheet
PS D:\polinema\daspro.semester1\Jobsheet> cd Jobsheet9
PS D:\polinema\daspro.semester1\Jobsheet\Jobsheet9> javac arrayValue20.java
PS D:\polinema\daspro.semester1\Jobsheet\Jobsheet9> java arrayValue20
Enter the final score0 : █
```

8. Commit and push the changes to GitHub.

## Questions!

1. Change the statement in step 5 to be like this

Run the program. Have there been any changes? How can it be like that?

2. Apa yang dimaksud dengan kondisi `i < finalScore.length`?

The length of the array itself, the length of the array is the length or many elements of an existing array

3. Change the statement in step 6 to be like this, so that the program only displays the grades of students who passed, students who have a score > 70

Run the program and describe the flow of the program!

```

public static void main(String[] args) {
    Scanner arrayValue20 = new Scanner(System.in);

    int[] finalScore = new int [10] ;

    for (int i = 0; i < finalScore.length; i++) {

        System.out.print("Enter the final score" + i + " : ");
        finalScore[i] = arrayValue20.nextInt();

        if (finalScore[i] > 70) {
            System.out.println("Student " + i + "Pass");
        }else {
            System.out.println("Student are= " + i + "Failed");
        }
    }
}

```

4. Modify the program so that it displays all students, and mark which one passed, and which did not!

```

Enter the final score4 : 70
Student are= 4Failed
Enter the final score5 : 80
Student 5Pass
Enter the final score6 : 80
Student 6Pass
Enter the final score7 : 80
Student 7Pass
Enter the final score8 : 80
Student 8Pass
Enter the final score9 : 80
Student 9Pass
PS C:\Users\ACER> `

```

5. Commit and push the changes to GitHub.

## 2.4 Experiment 4: Searching

### Experiment Time: 45 minutes

1. Open a text editor, create a Java file then save it with the name **linearSearchXX**. (XX=student ID number)

2. Add the following code

```
6      public static void main(String[] args) {
7          int[] arrayInt = { 34, 18, 26, 48, 72, 20, 56, 63 };
8          int key = 20;
9          int result = 0;
10         for (int i = 0; i < arrayInt.length; i++) {
11             if (arrayInt[i] == key) {
12                 result = i;
13                 break;
14             }
15         }
16         System.out.println("The key in the array is located at index position " + result);
17     }
18 }
```

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

```
The key in the array is located at index position 5
```

4. Commit and push the changes to GitHub

## Questions!

1. Explain the meaning of the **break;** statement on line 13 of the program code in Experiment 4.
2. Modify the program code in experiment 4 so that the program can receive input in the form of the number of array elements, the contents of the array, and the key you want to search for. Then, print to the screen the index of the element positions of the searched key. Example of program results:

```
Enter the number of array elements: 8
Enter the array element 0: 12
Enter the array element 1: 18
Enter the array element 2: -6
Enter the array element 3: 10
Enter the array element 4: 6
Enter the array element 5: 15
Enter the array element 6: 11
Enter the array element 7: 9
Enter the key you want to search for: 10
The key in the array is located at index position 3
```

3. Modify the program in experiment 4 so that the program will give the message "key not found" if the key is not in the array. Example of program results:

```
Enter the number of array elements: 6
Enter the array element 0: 19
Enter the array element 1: 23
Enter the array element 2: 29
Enter the array element 3: 31
Enter the array element 4: 37
Enter the array element 5: 43
Enter the key you want to search for: 11
Key not found
```

## 1. Assignment

1. Create a program to produce the highest value, lowest value, and average from an array containing integer type numbers.

Terms:

- Input: Number of elements, value of each element
- Output: Highest value, lowest value, average value

2. Implement the flowchart that was created in the assignment for Week 9 of the Programming Fundamentals course related to the group project into Java program code.

Commit and push the results of your program code to your project's GitHub repository.

Note: assignments may only apply material from Week 1 to Week 9.