

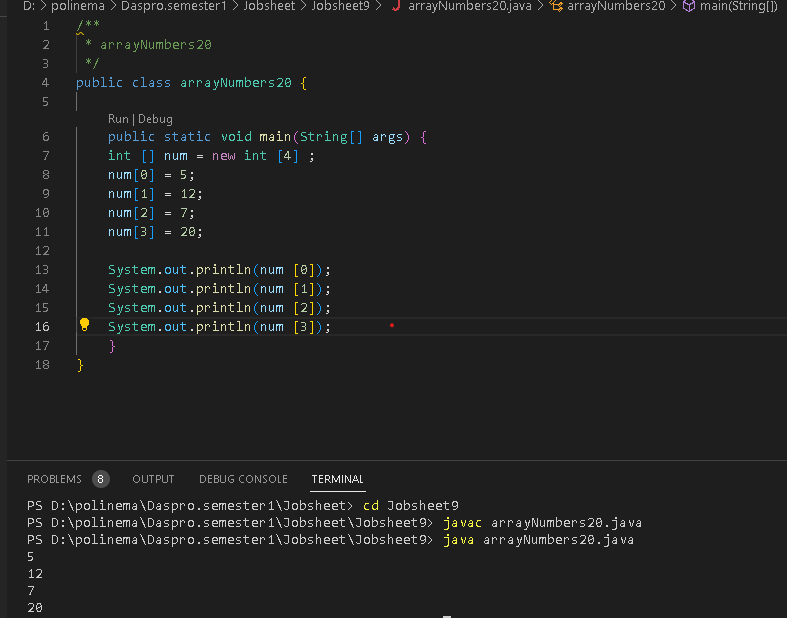
**Name : Rizqi Dewanto Y Number : 2341760176**

**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

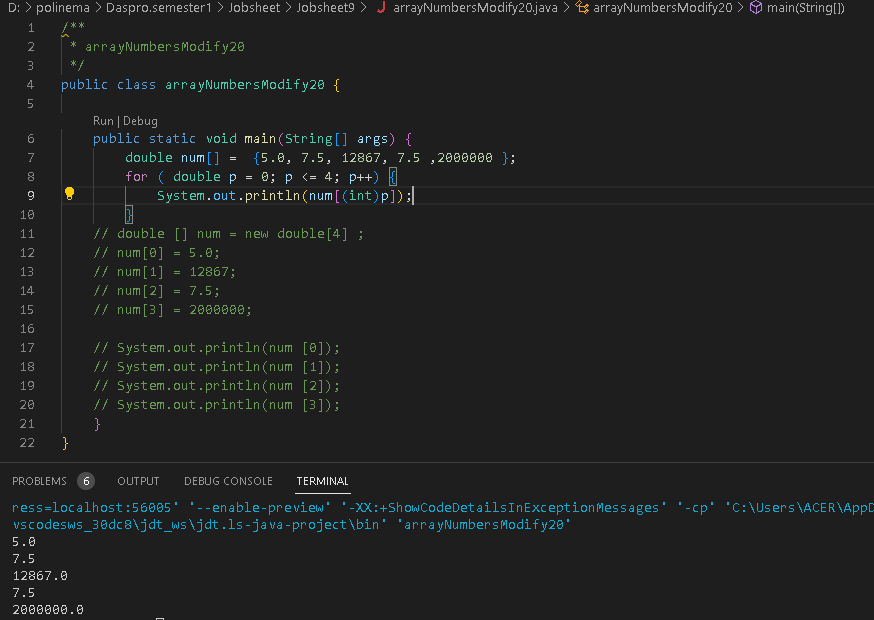


7. Commit and push the changes to GitHub

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.

  
3. Change the statement in step 6 to be like this  
What is the result? How can it be like that?

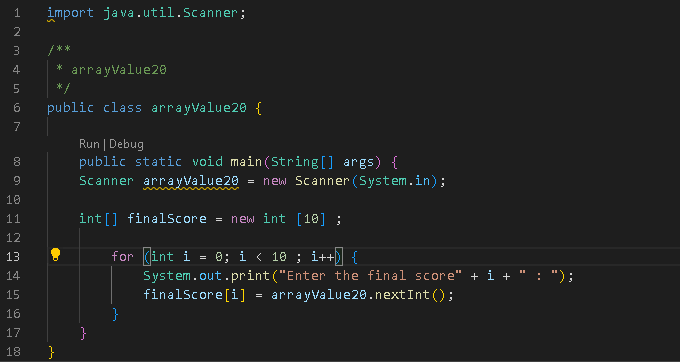
I added a typecast **(int)** to convert the loop variable **p** to an integer before using it as an array index. This is necessary because array indices must be integers in Java.

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?

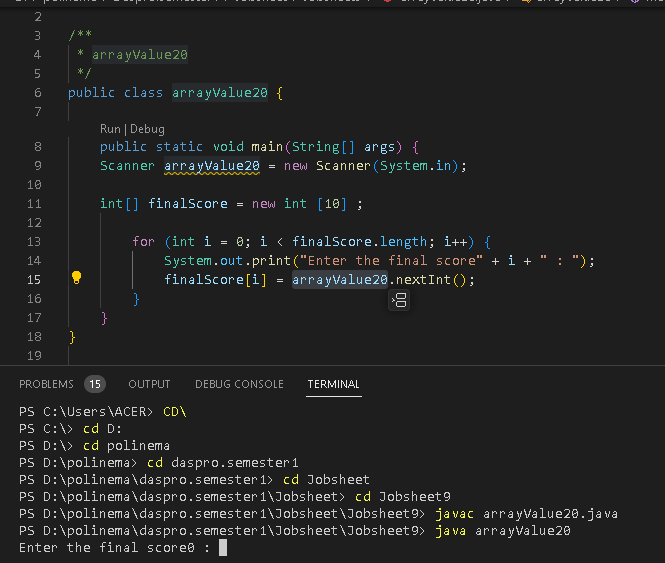
This is due to the loop's iteration over each number between 1 and 4, which prints the current number as well as the maximum, which stays at 4 for the duration of the loop. The loop's iterations are guaranteed to contain the number 4 thanks to the i <= 4 condition.

5. Commit and push the changes to GitHub.

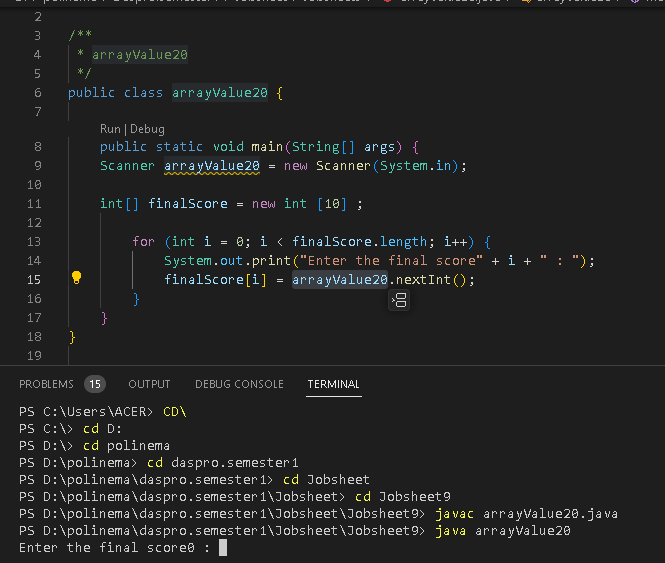
**2.2 Experiment 2: Requesting User Input to Fill in an Array Element  
Experiment Time: 40 minutes**1. Open a text editor, create a Java file then save it with the name **arrayValueXX**.  
(XX=student ID number)  
2. Write the basic structure of the Java programming language which contains the **main()**function  
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 element6. Using a loop, display all the contents of the elements from the **finalScore** array

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

**Compile**

****

**Run the program**

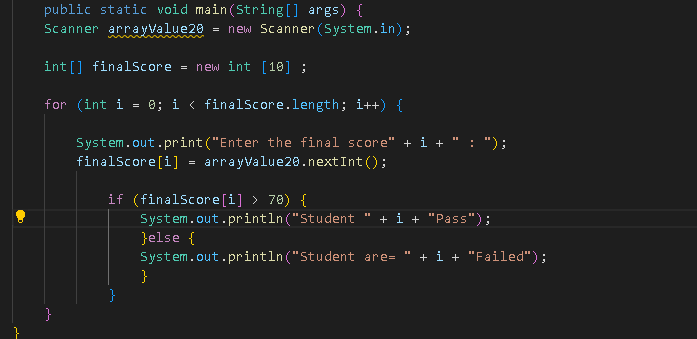


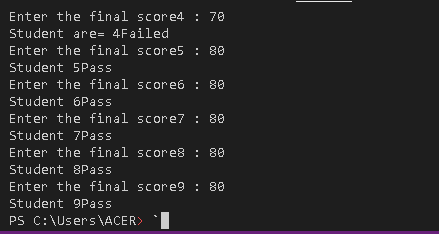
8. Commit and push the changes to GitHub.

**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!

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

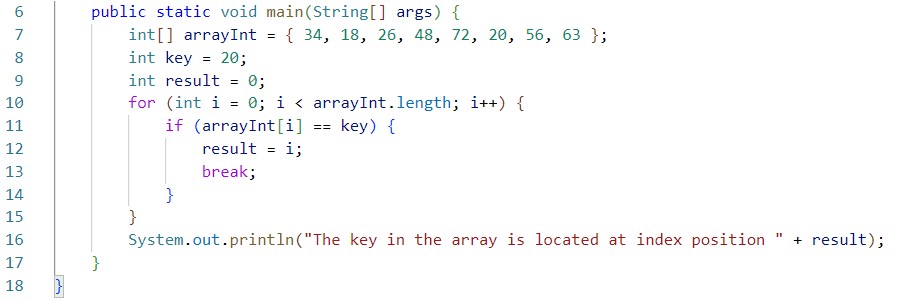


5. Commit and push the changes to GitHub.

**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



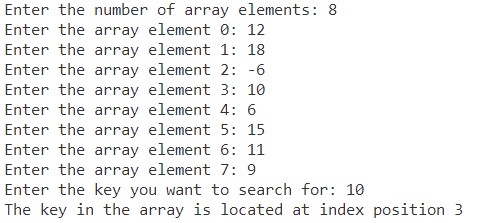
1. Compile and run the program. Match the results of the running programs that you have created according to the following display



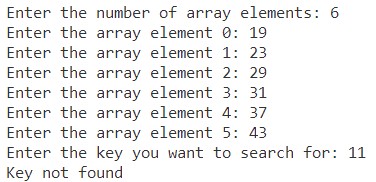
1. Commit and push the changes to GitHub

**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:



1. 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:



# 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

1. 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.