



JOB SHEET 6

LOOPING 1

1. Objective

- Students are able to explain the writing format of looping part 1
- Students are able to implement the flowchart of looping part 1 using Java programming language

2. Laboratory

2.1 Experiment 1: Calculate factorial values using iteration

a) Loop using for

1. Open a text editor. Create a new file, name it **factorialFor.java**
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 multiple **int** type variables with names **number**, **factorial**, and **i**
6. Write down the syntax for entering the value from keyboard

```
System.out.print("Enter a number: ");  
number = input.nextInt();
```

7. Create a for loop structure to calculate the factorial

```
factorial = 1;  
for (i = 1; i <= number; i++) {  
    factorial = factorial * i;  
}
```

8. Display factorial calculation results

```
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!

**b) Loop using while**

1. Open a text editor. Create a new file, name it **factorialWhile.java**
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 multiple **int** type variables with names **number**, **factorial**, and **i**
6. Write down the syntax for entering the value from keyboard

```
System.out.print("Enter a number: ");  
number = input.nextInt();
```

7. Create a while loop structure to calculate the factorial

```
factorial = 1;  
i = 1;  
while (i <= number) {  
    factorial = factorial * i;  
    i++;  
}
```

8. Display factorial calculation results

```
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!

c) Loop using do-while

1. Open a text editor. Create a new file, name it **factorialDoWhile.java**
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 multiple **int** type variables with names **number**, **factorial**, and **i**
6. Write down the syntax for entering the value from keyboard



```
System.out.print("Enter a number: ");  
number = input.nextInt();
```

7. Create a do-while loop structure to calculate the factorial

```
factorial = 1;  
i = 1;  
do {  
    factorial = factorial * i;  
    i++;  
} while (i <= number);
```

8. Display factorial calculation results

```
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!
10. Match the results of the running programs that you have created according to the following display

```
Enter a number: 6  
The factorial of 6 is 720
```

2.2 Experiment 2: Exit loop using break

1. Open a text editor. Create a new file, name it **loopBreak.java**
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 multiple **int** type variables with names **number** and **b**
6. Add the following code to enter the value from keyboard in 'for' loop structure.
In 'for' loop there is also a condition to stop the process using the **break** statement



```
for (b = 0; true;) {  
    System.out.print("Enter a number: ");  
    number = input.nextInt();  
    b += number;  
    if (b > 50) {  
        break;  
    }  
}  
System.out.printf("The numbers stop at the sum of the numbers %d\n", b);
```

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

```
Enter a number: 15  
Enter a number: 9  
Enter a number: 12  
Enter a number: 24  
The numbers stop at the sum of the numbers 60
```

2.3 Experiment 3: Exit loop using continue

1. Open a text editor. Create a new file, name it **loopContinue.java**
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 multiple **int** type variables with names **number**, **b**, **i**, and **count**. Then also create two **double** type variables with names **avg** and **total**
6. Add the following code to enter the value from keyboard in 'for' loop structure. In 'for' loop there is also a condition to stop the process using the **continue** statement



```
b = 0;
count = 0;
for (i = 0; i < 5; i++) {
    System.out.print("Enter a number: ");
    number = input.nextInt();
    if (number >= 50) {
        continue;
    }
    b += number;
    count++;
}
total = (double) b;
System.out.printf("The total number is less than 50: %.2f\n", total);
avg = (double) b / count;
System.out.printf("Average number less than 50: %.2f\n", avg);
```

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

```
Enter a number: 25
Enter a number: 35
Enter a number: 45
Enter a number: 55
Enter a number: 30
The total number is less than 50: 135.00
Average number less than 50: 33.75
```

3. Questions!

1. Explain the difference between Experiment 2 and Experiment 3!
2. Suppose you are asked to create a Java program that asks for input of an



integer **n**. Then, the program displays the character '*' on the screen **n** times.

Which of the two pieces of the program below is better and safer? Why?

<pre>/* for example: user input n has been stored in integer variable n */ int i = 0; while (i < n) { System.out.print("*"); i++; }</pre>	<pre>/* for example: user input n has been stored in integer variable n */ int i = 0; while (i != n) { System.out.print("*"); i++; }</pre>
---	---

3. What is the output of the following three code snippets?

<pre>int r = 1; int i = 1; int a = 2; int n = 4; while (i <= n) { r = r * a; i++; } System.out.print(r);</pre>	<pre>int n = 5; boolean stop = false; int i = 1; while (!stop) { if (i >= n) { stop = true; } else { if (i % 2 == 1) { System.out.print("#"); } else { System.out.print("*"); } i++; } }</pre>	<pre>int n = 5; long result = 1; for (int i = 1; i <= n; i++){ hasil = hasil * i; } System.out.println(n+"!="+ result);</pre>
--	--	---

4. Assignment

1. Create a program to display numbers from 1 to the user input numbers sequentially and skip the multiples of 5 as shown below!



```
Enter a number: 12
1
2
3
4
6
7
8
9
11
12

Enter a number: 9
1
2
3
4
6
7
8
9
```

2. Create a program using the Java programming language that requests input of an integer N ($N > 0$) from user. The program then displays the sum of the **first N positive even numbers** (even numbers ≥ 0).

Example:

- If the user enters $N = 10$, the program will count the number of positive numbers in the range 1-10 then display the sum of the positive numbers between 1-10, namely:

$$0 + 2 + 4 + 6 + 10 = 30.$$

After that the program will display the average of the positive numbers that were added earlier.

- Example of program output

```
Enter a number range: 10
The number of even numbers from 1 to 10 is 5
Even number 1 is 2
Even number 2 is 4
Even number 3 is 6
Even number 4 is 8
Even number 5 is 10
The sum of the even numbers from 1 to 10 is 30
The average of the even numbers from 1 to 10 is 6.00
```



You can design your own for the program display