JOBSHEET 7

Loop 1

Name
Sherly Lutfi Azkiah Sulistyawati

NIM
2341720241

Class
1I

Department
Information Technology

Study Program
D4 Informatics Engineering

Labs Activity

Question! (Experiment 1)

1. There are 3 main components in FOR loop. Based on experiment 1 above, identify and explain these 3 components!

2. Explain how the following code works!

```
for(int i=1;i<=50;i++){
    if(i%multiple == 0){
        sum = sum + i;
        counter++;
        //System.out.print(i+"-");
    }
}
```

3. Modify the existing code by adding a new variable to calculate the average of all the specified multiples! Push and commit the program code to GitHub.

4. Create a new Java program file named WhileMultiplesStudentIDNumber.java. Create the equivalent code by using while loop. Push and commit the code to GitHub.

Answer!

1. The 3 main components in FOR loop based on experiment 1 are initialization(int i = 1), condition(i<=50), and update(i++).

2. The loop repeats this process for each value of i from 1 to 50, checking if each number is a multiple of the specified multiple. It accumulates the sum of those multiples in the sum variable and keeps track of how many multiples were found using the counter variable.

3.
```
public class ForMultiples24 {
    Run | Debug
    public static void main(String[] args) {
        Scanner input24 = new Scanner(System.in);
        int multiple, sum = 0, counter = 0;
        double average = 0;

        System.out.print(s:"Input the multiple = ");
        multiple = input24.nextInt();

        for(int i=1;i<=50;i++){
            if(i%multiple == 0){
                sum = sum + i;
                counter++;
                average = sum/counter;
               // System.out.print(i+"-");
            }
        }

        System.out.printf(format:"There are %d numbers that are multiple of %d in range 1 to 50.\n", counter, multiple);
        System.out.printf(format:"The sum from all multiples of %d in range 1 s.d. 50 is %d.\n", multiple, sum);
        average = sum/counter;
        System.out.println("Average = " + average);
```

4.
```java
public static void main(String[] args) {
    Scanner input24 = new Scanner(System.in);
    int multiple, sum = 0, counter = 0;
    double average = 0;

    System.out.print(s:"Input the multiple = ");
    multiple = input24.nextInt();

    int i=1;
    while(i<50){
        if(i%multiple == 0){
            sum = sum + i;
            counter++;
            average = sum/counter;
           // System.out.print(i+"-");
        }
        i++;
    }

    System.out.printf(format:"There are %d numbers that are multiple of %d in range 1 to 50.\n", counter, multiple);
    System.out.printf(format:"The sum from all multiples of %d in range 1 s.d. 50 is %d.\n", multiple, sum);
    average = sum/counter;
    System.out.println("Average = " + average);
```

Question! (Experiment 2)

1. Show the part of the program code used as a condition to stop the WHILE loop! How many times is the loop executed?
2. In this code,

```java
if(position.equalsIgnoreCase(anotherString:"director")){
    continue;
```

   What actually happens if the 'position' variable contains the value 'DIRECTOR'? What is the use of CONTINUE within the loop structure?

3. Why is the 'i++' iteration component placed in the middle, not at the end of the while block? Move 'i++' to the end of the while block, then run the program again by entering 'DIRECTOR' as the position for the first employee. What happens? Explain!
4. Modify the program code to handle invalid positions like the following example:

```
Employee number = 3
Position of employee 1 (director, manager, staff) = director
Employee 1 overtime hours = 5
Position of employee 2 (director, manager, staff) = manager
Employee 2 overtime hours = 10
Position of employee 3 (director, manager, staff) = programmer
Employee 3 overtime hours = 4
Invalid position!
Position of employee 3 (director, manager, staff) = staff
Employee 3 overtime hours = 4
Total of Overtime Pay = 1300000.0
```

5. Commit and push the changes to GitHub.

Answer!

1. The loop is executed as long as the value of i is less than the value of numEmployee. The loop will continue until i becomes equal to or greater than numEmployee. So, the loop is executed as many times as there are employees specified by the user in numEmployee.
2. The purpose of using continue is to handle different types of employees differently. Continue allows the program to skip the standard overtime pay calculations and continue to the next employee's input without processing the 'director' employee's overtime pay calculation.
3. Moving i++ to the end of the while block would cause the loop to behave incorrectly when 'DIRECTOR' is entered as the first employee's position and cannot continue to selection.
4.

```
15      int i=0;
16      while(i<numEmployee){
17          System.out.print("Position of employee "+(i+1)+" (director, manager, staff) = ");
18          position = input24.next();
19          System.out.print("Employee "+(i+1)+" overtime hours = ");
20          overtimeHours = input24.nextInt();
21          i++;
22
23          if(position.equalsIgnoreCase(anotherString:"director")){
24              continue;
25          } else if(position.equalsIgnoreCase(anotherString:"manager")){
26              overtimePay=overtimeHours*100000;
27          } else if(position.equalsIgnoreCase(anotherString:"staff")){
28              overtimePay=overtimeHours*75000;
29          } else {
30              System.out.println(x:"Invalid position!");
31              i--;
32              continue;
33          }
34
35          totalOvertimePay += overtimePay;
36      }
37
38      System.out.println("Total of Overtime Pay = "+totalOvertimePay);
```

Question! (Experiment 3)

1. What is the use of the BREAK within the loop syntax?
2. Modify the program so that if the number of leave days requested is greater than the remaining entitlement, the program does not stop, allowing the user to enter the number of days according to the entitlement.
3. Commit and push the program code to GitHub.
4. When typing "t" as the confirmation input, what happens? Why?

5. Modify the program code so that when the user enters "t" as the confirmation input, the program will stop.

6. Commit and push the program code to GitHub.

Answer!

1. The break statement is used within the loop to exit the loop prematurely when a specific condition is met.

2.
```
do {
    System.out.print(s:"Do you want to take a leave? (y/n)? ");
    confirmation = input24.next();

    if (confirmation.equalsIgnoreCase(anotherString:"y")) {
        System.out.print(s:"Leave number: ");
        numLeave = input24.nextInt();

        if (numLeave <= leaveEntitlement) {
            leaveEntitlement -= numLeave;
            System.out.println("The remaining leave entitlement: " + leaveEntitlement);
        } else {
            System.out.println(x:"The remaining leave entitlement is not sufficient!");
            continue;
        }
    } else {
        break;
    }
} while (leaveEntitlement > 0);
}
```

4. The program will exit the loop and terminate.

5.
```
do {
    System.out.print(s:"Do you want to take a leave? (y/n)? ");
    confirmation = input24.next();

    if (confirmation.equalsIgnoreCase(anotherString:"y")) {
        System.out.print(s:"Leave number: ");
        numLeave = input24.nextInt();

        if (numLeave <= leaveEntitlement) {
            leaveEntitlement -= numLeave;
            System.out.println("The remaining leave entitlement: " + leaveEntitlement);
        } else {
            System.out.println(x:"The remaining leave entitlement is not sufficient!");
            continue;
        }
    } else {
        break;
    }
} while (leaveEntitlement > 0);
}
```

Assignment

Times: 130 minutes

- Implement the flowchart that was already created in the assignment for the Basic Programming course related to the project into your program.
- Push and commit the result of your program code to your project repository.
- Note: the assignment should only apply the material from sessions 1 to 7.

Answer!

```
while(true){
    System.out.print(s:"\nPayment Method(BNI/BRI/Bank Jatim/Mandiri/BCA): ");
    payMethod = sc.nextLine();

    switch(payMethod) {
    case "BNI" :
        System.out.print(s:"Input BNI account number: ");
        int BNI = sc.nextInt();
        System.out.println(x:"Salary is currently being processed for transfer to the account...");
        return;
    case "BRI":
        System.out.print(s:"Input BRI account number: ");
        int BRI = sc.nextInt();
        System.out.println(x:"Salary is currently being processed for transfer to the account...");
        return;
    case "Bank Jatim":
        System.out.print(s:"Input Bank Jatim account number: ");
        int bankJatim = sc.nextInt();
        System.out.println(x:"Salary is currently being processed for transfer to the account...");
        return;
    case "Mandiri":
        System.out.print(s:"Input Mandiri account number: ");
        int mandiri = sc.nextInt();
        System.out.println(x:"Salary is currently being processed for transfer to the account...");
        return;
    case "BCA":
        System.out.print(s:"Input BCA account number: ");
        int BCA = sc.nextInt();
        System.out.println(x:"Salary is currently being processed for transfer to the account...");
        return;
    }
}
```