JOBSHEET 5

SELECTION part 1

1. Learning Outcome

After finishing this lesson, students must be able to :

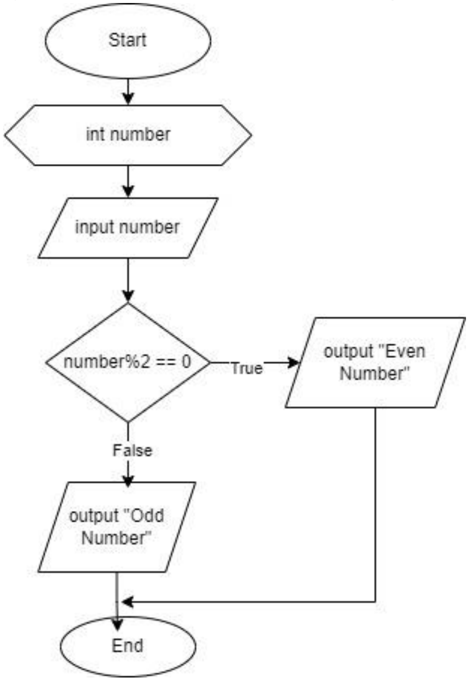
1. Explain the basic concept of selection,
2. Implement selection algorithm to solve a specific logical problem and visualize it by creating flowchart
3. Write the source code based on the flowchart to implement selection algorithm

2. Labs Activity

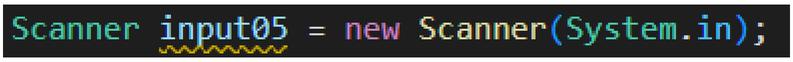
2.1 Experiment 1

Time: 40 minutes

1. The following flowchart visualizes the algorithm of odd/even number selection. We will create a program as the implementation of the following flowchart.

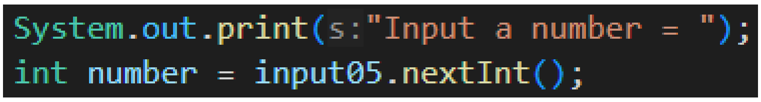


2. Open the text editor and create a new Java file named **Selection1StudentIDNumber.java.**
3. Create the basic structure of Java program, including class and main method.
4. Add a Scanner library since we will get the user input in this program.
5. Declare a Scanner variable or object named **inputStudentIDNumber**



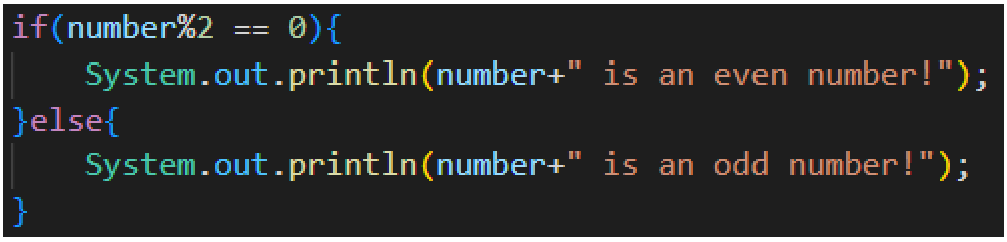
```
Scanner input05 = new Scanner(System.in);
```

6. Add the following statement inside the method main, to get the user input:



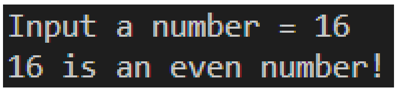
```
System.out.print(s:"Input a number = ");
int number = input05.nextInt();
```

7. Create the selection statement by using if-else to select the odd or even number of the user input.



```
if(number%2 == 0){
    System.out.println(number+" is an even number!");
}else{
    System.out.println(number+" is an odd number!");
}
```

8. Compile and run the program, and you mast get the following output:



```
Input a number = 16
16 is an even number!
```

9. Commit and push the changes into your repository.

Question!

1. Modify the above selection statement (if-else) by using Ternary Operator! We know that Ternary Operator could be used as a selection statement as well.
2. Compile, run and observe the result!
3. **Commit and push the changes into your repository!**
4. Finally, please explain why the output of the program before and after the changes has a similar output.

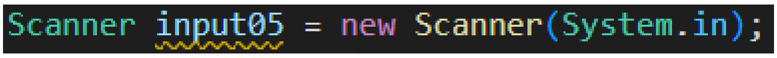
2.2 Experiment 2

Time: 40 minutes

Case Study:

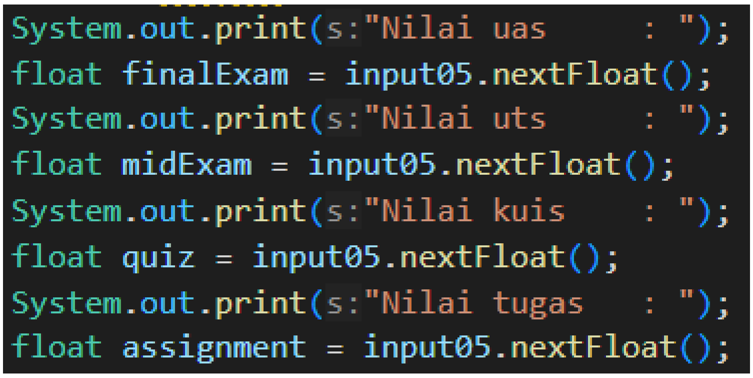
At the end of the semester, a lecturer calculates the final grade for students, which consists of midterm exam, final exam, quizzes, and assignments. The final grade is determined by 40% for the final exam, 30% for the midterm exam, 10% for quizzes, and 20% for assignments. If a student's final grade is below 65, they will be required to re-take the exam. Create a program to help to identify students who need to re-take the exam based on their final grades!

1. Create a new Java program file named **Selection2StudentIDNumber.java.**
2. Create a basic structure for Java program that consists of class and main method.
3. Add the Scanner.
4. Declare a Scanner variable named **inputStudentIdNumber**



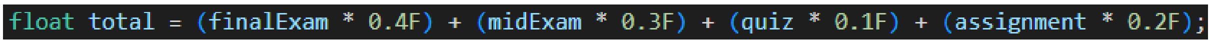
```
Scanner input05 = new Scanner(System.in);
```

5. Get the user input by using the following statements.



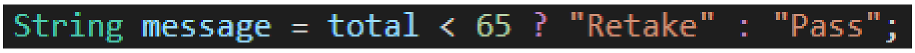
```
System.out.print(s:"Nilai uas     : ");
float finalExam = input05.nextFloat();
System.out.print(s:"Nilai uts     : ");
float midExam = input05.nextFloat();
System.out.print(s:"Nilai kuis    : ");
float quiz = input05.nextFloat();
System.out.print(s:"Nilai tugas   : ");
float assignment = input05.nextFloat();
```

6. Add the final grade calculation as follows:



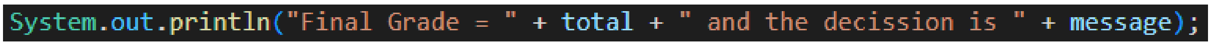
```
float total = (finalExam * 0.4F) + (midExam * 0.3F) + (quiz * 0.1F) + (assignment * 0.2F);
```

7. Add the following statement to select the condition, by using Ternary Operator:



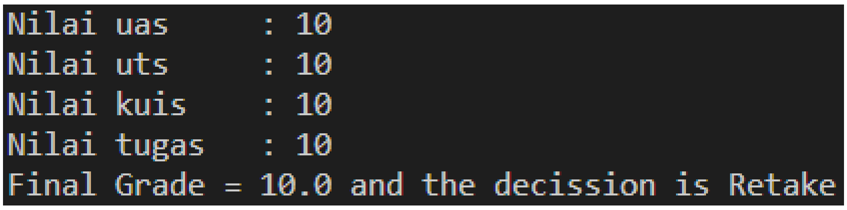
```
String message = total < 65 ? "Retake" : "Pass";
```

8. Output the final grade and the final decision either retake or not.



```
System.out.println("Final Grade = " + total + " and the decission is " + message);
```

9. Run the program and the output must be like this:



```
Nilai uas     : 10
Nilai uts     : 10
Nilai kuis    : 10
Nilai tugas   : 10
Final Grade = 10.0 and the decission is Retake
```

10. Commit and push the changes into your repository.

Question!

1. Modify the above program so that it now can convert from numerical grade into letter grade, based on the following rule!

Nilai Angka	Nilai Mutu		
	Nilai Huruf	Nilai Setara	Kualifikasi
80<N≤ 100	A	4	Sangat Baik
73 <N≤ 80	B+	3,5	Lebih dari Baik
65 <N≤ 73	B	3	Baik
60 <N≤ 65	C+	2,5	Lebih dari Cukup
50 <N≤ 60	C	2	Cukup
39 <N≤ 50	D	1	Kurang
N ≤ 39	E	0	Gagal

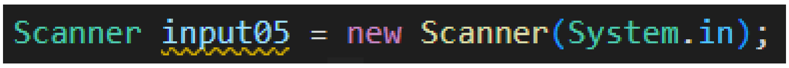
Commit and push the changes into your repository!

2. After the above modification, how many conditions are there and what type of operator are used?

Experiment 3

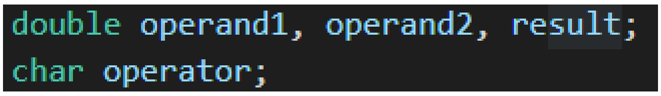
Time: 60 minutes

1. Create a new Java program file named **Selection3StudentIDNumber**.java.
2. Create a basic structure for Java program, including class definition and main method.
3. Declare a Scanner variable or object named **inputStudentIDNumber**



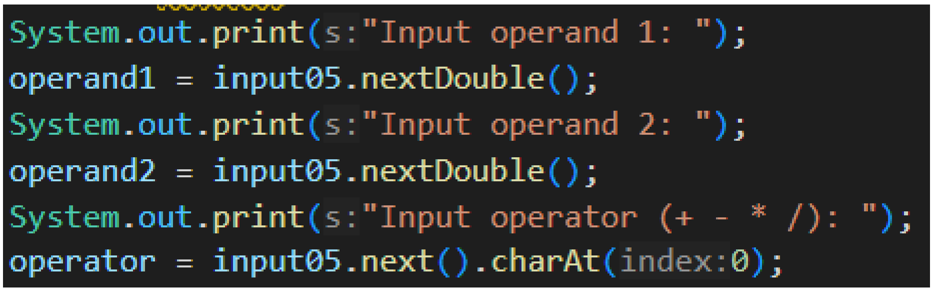
```
Scanner input05 = new Scanner(System.in);
```

4. Declare the following variables.



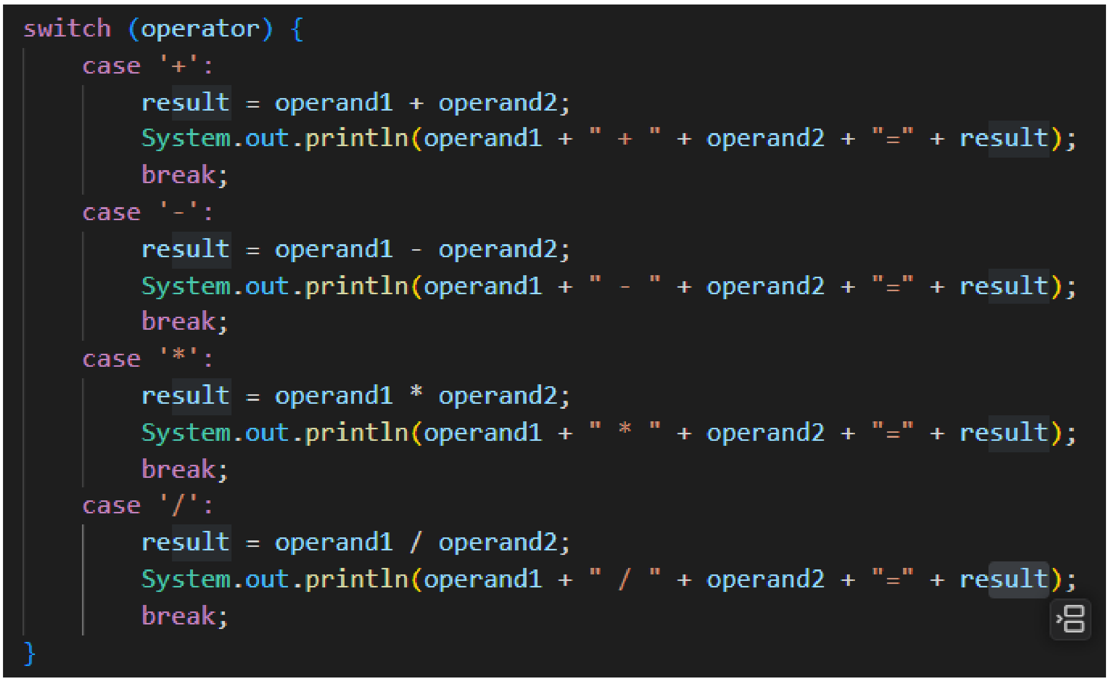
```
double operand1, operand2, result;
char operator;
```

5. Create the following statements to get the user input.



```
System.out.print(s:"Input operand 1: ");
operand1 = input05.nextDouble();
System.out.print(s:"Input operand 2: ");
operand2 = input05.nextDouble();
System.out.print(s:"Input operator (+ - * /): ");
operator = input05.next().charAt(index:0);
```

6. Add the following selection statements.



```
switch (operator) {
    case '+':
        result = operand1 + operand2;
        System.out.println(operand1 + " + " + operand2 + "=" + result);
        break;
    case '-':
        result = operand1 - operand2;
        System.out.println(operand1 + " - " + operand2 + "=" + result);
        break;
    case '*':
        result = operand1 * operand2;
        System.out.println(operand1 + " * " + operand2 + "=" + result);
        break;
    case '/':
        result = operand1 / operand2;
        System.out.println(operand1 + " / " + operand2 + "=" + result);
        break;
}
```

7. Explain how your program works!
8. Commit and push your program into your repository!

Question!

1. What is the use of ***break*** and ***default*** statement?
2. Modify the above program by deleting **break** statement in the first **case**. Run the program, observe the result, and explain what it is the effect if there is no **break** in **case** block!
3. **Commit and push the changes into your repository.**
4. Please explain the function of the following statement

```
operator = sc.next().charAt(0);
```

5. Assignment

Time: 160 minutes

Create a program based on the flowchart that was already created in Assignment 5 in the Slide of Selection part 1! Commit and push the code results to your project repository!

Note: The assignment may only apply the material covered from Week 1 to Week 5.