

Selection

- *Nested Conditional Statements* -

Programming Fundamentals Team Teaching

Learning Outcomes

- Students are able to understand nested conditional statement concept
- Students are able to understand the structure of nested conditional statement
- Students are able to solve complex logical problem using nested conditional statement
- Students are able to utilize nested conditional statement in Java

Nested Conditional Statement

What if, if you need more complex solution to solve complex logic condition

Nested IF

- **Nested IF** is a specific form of selection to perform a **multilevel** condition selection
- Nested IF means an **IF statement within another IF statement**. It allows us to test multiple criteria and increases the number of possible outcomes
- **Inside an IF** statement (or IF-ELSE or IF-ELSE IF-ELSE), it is **possible to have another IF** or IF-ELSE or IF-ELSE IF-ELSE statement

The Structure of Nested IF

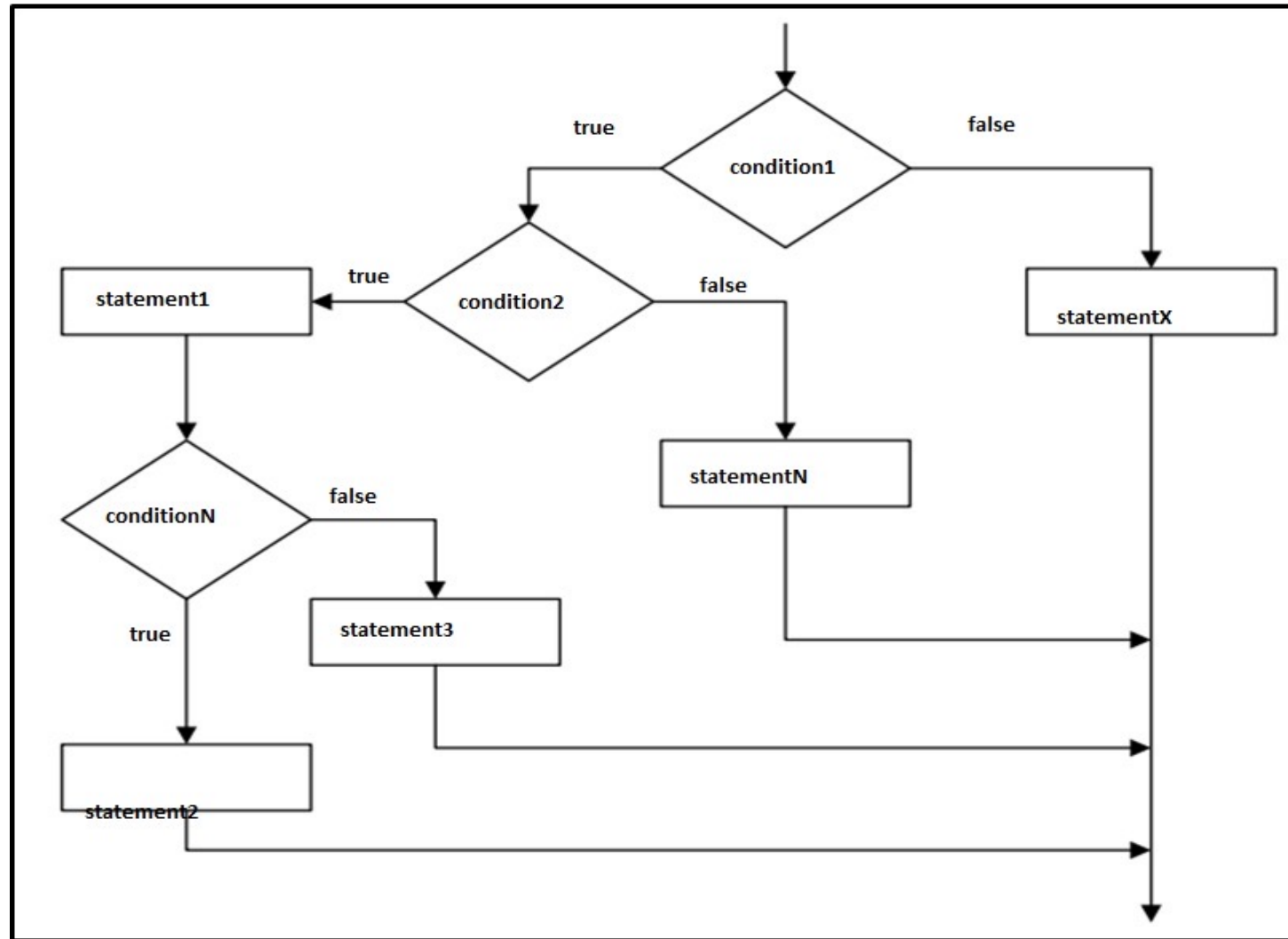
```
if (condition1){  
    if (condition2){  
        statement1;  
        ...  
        ...  
        if (conditionN){  
            statement2;  
        } else {  
            statement3;  
        }  
    } else {  
        statementN;  
    }  
} else {  
    statementX;  
}
```

Nested IF – How it works

- **Condition1** will be **firstly evaluated**. If it is **false**, then the compiler will go to **outermost else** block
- If **condition1** is **true**, then it will continue to evaluate **condition2**.
- If **condition2** is **true**, then **statement1** will be executed. If **condition2** is **false**, then it will go to else block to execute **statementN**.
- After executing **statement1**, it will continue to evaluate **conditionN**. If **conditionN** is **true**, then **statement2** will be executed, otherwise **statement3** will run.

```
if (condition1){  
    if (condition2){  
        statement1;  
        ...  
        ...  
        if (conditionN){  
            statement2;  
        } else {  
            statement3;  
        }  
    } else {  
        statementN;  
    }  
} else {  
    statementX;  
}
```

Nested IF – The Flowchart



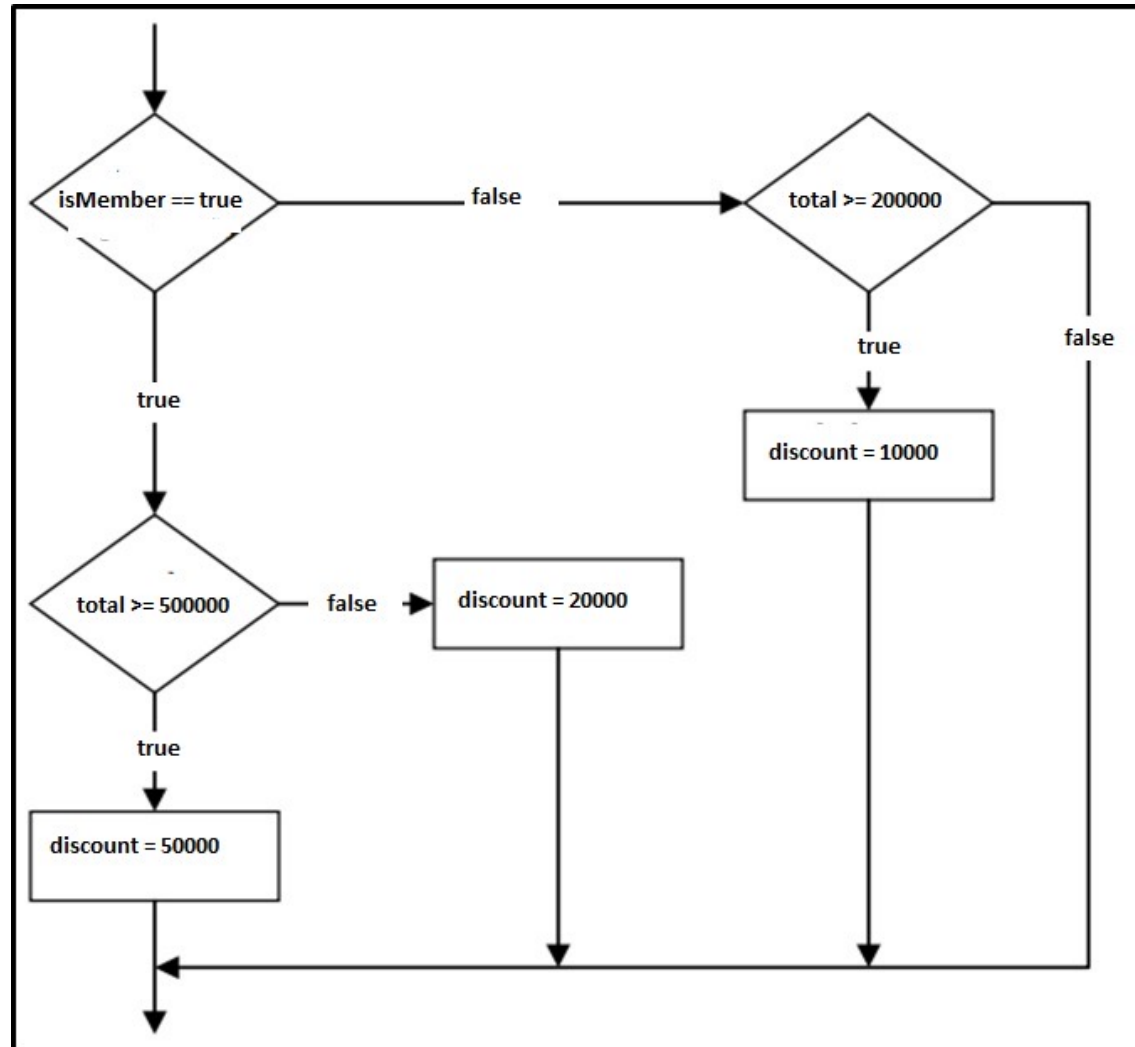
Nested IF – An Example

In a Point of Sales application, there is a rule as follows:

If the customer is a member?

- TRUE:
 - **Is the total amount bigger than Rp 500.000?**
 - ❖ TRUE: The customer will get discount Rp 50.000
 - ❖ FALSE: The customer will get discount Rp 25.000
- FALSE:
 - **Is the total amount bigger than Rp 200.000?**
 - ❖ TRUE: The customer will get discount Rp 10.000
 - ❖ FALSE: The customer will not get any discount

Nested IF – An Example: The Flowchart



Nested IF – An Example: The Code

```
4  public class Kasir {  
    Run | Debug  
5  public static void main(String[] args) {  
6      int total, discount, purchase;  
7      boolean isMember;  
8      Scanner input = new Scanner(System.in);  
9  
10     System.out.print(s:"Is member? (true/false) = ");  
11     isMember = input.nextBoolean();  
12     System.out.print(s:"Total = ");  
13     total = input.nextInt();  
14  
15     if(isMember)  
16     |     if(total > 500000)  
17     |         discount = 50000;  
18     |     else  
19     |         discount = 25000;  
20     | else  
21     |     if(total > 200000)  
22     |         discount = 10000;  
23     |     else  
24     |         discount = 0;  
25  
26     purchase = total - discount;  
27     System.out.println("Total = "+total);  
28     System.out.println("Discount = "+discount);  
29     System.out.println(x:"-----");  
30     System.out.println("Purchase = "+purchase);  
31 }  
32 }
```



Logical Operators

Commonly used logical operator in (nested) conditional statement

Logical Expression #1

- There are 3 logical operators which are frequently used to construct condition statement:
 - && : AND
 - || : OR
 - ! : NOT
- **Logical operators** are generally used for combining two or more relational statements that represent two or more conditions.
- In logical operator, the condition statement will be evaluated from left to the right

Logical Expression #2

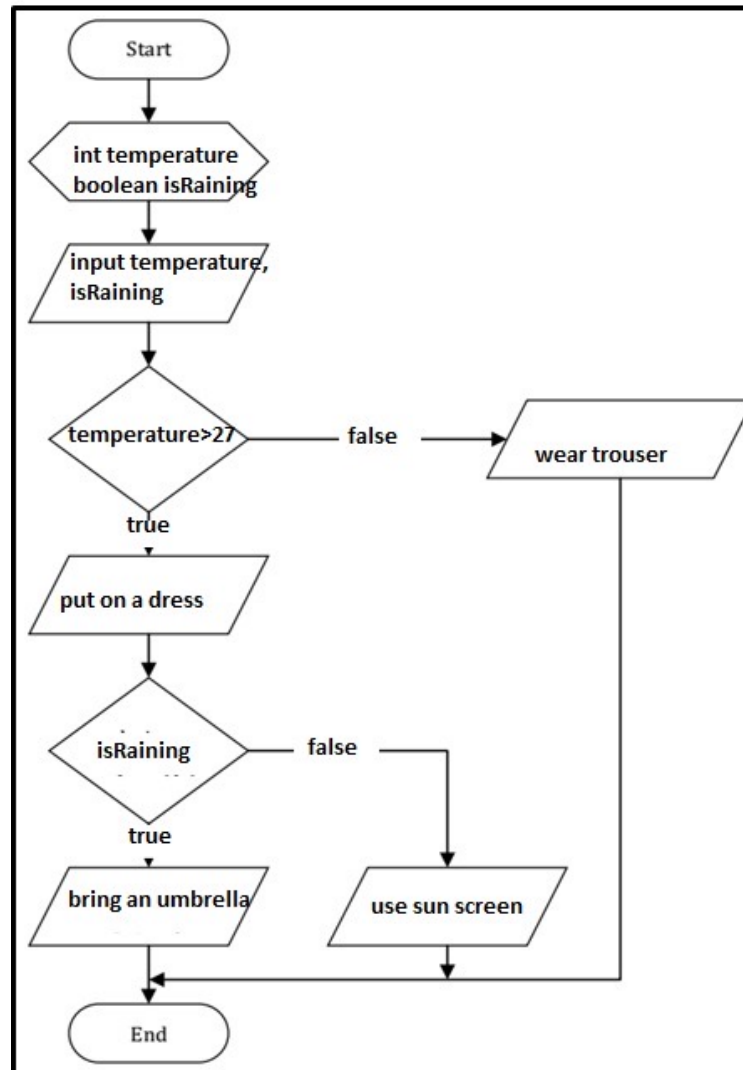
- When evaluating $(e1 \ || \ e2)$, if **e1** returns **TRUE**, then **e2 will not** be evaluated. Thus, the value of the entire expression $(e1 \ || \ e2)$ will be considered **true**
- However, if **e1** returns **FALSE**, then **e2 will** then be evaluated to determine the value of the entire expression
- Example:
 - `If(speed == 0 || machineOn == true)`
 - `System.out.println("Turn off the machine");`

Case Study

The Case

- A system was created to determine the clothing and equipment that users should carry according to weather conditions. If the temperature is more than 27C, the user is advised to put on a **dress**, then check whether it is currently raining, if it is raining then the user is advised to bring an **umbrella**, whereas if it is not raining then the user is advised to use **sunscreen**. However, if the temperature is less than or equal to 27C, then users are advised to wear **long trousers**
- Make a flowchart for the system!

The Case: The Flowchart





Practice

Practice #1

Design an algorithm using a flowchart to determine the maximum number from the 3 numbers input

input :

num1 = 28

num2 = 54

num3 = 15

output :

max number = 54

Practice #2

In every Wednesday, a bookshop gives discount to all customers depend on the book category, as the following rules:

- A 10% discount will be given to customers who buy **dictionary**, then an additional 2% discount will be given if the customers buy more than 2 books.
- A 7% discount will be given to customers who buy **novel**, then an additional 2% discount will be given if they buy more than 3 novels, but if there are only 3 or less than 3 novels they buy, then additional 1% discount will be given
- The customers who buy other categories, then they will get 5% discount if they buy more than 3 books

Make a flowchart (using logical operators) to determine the total amount that must be paid by customer. The input are the **book category** and **number** of books bought, while the output is the discount amount

Practice #3



- To determine and print the minimum number of currency notes of the denominations: \$1, \$5, \$10, \$20, \$50, \$100, \$500 and \$1000 required to pay any given amount.

Practice #4

Accept three integers representing the angles of a triangle in degrees to determine whether they form a valid set of angles of a triangle. If it is not a valid set, then generate a message and terminate the process. If it is a valid set, then the process determines whether it is equiangular (all three angles are the same). It also determines if the triangle is right angled (has one angle with 90 degrees), obtuse angled (one angle above 90), or acute angled (all three angles are below 90 degrees). Finally, it shows conclusion about the triangle.

Practice #5

An electricity board charges the following rates to domestic users to discourage large consumption of energy:

- for the first 100 units—\$.85 per unit
- for the next 200 units—\$1.45 per unit
- Beyond 300 units—\$1.85 per unit

All users are charged a minimum of \$ 500.00. If the total cost is more than \$ 2,500.00, then an additional surcharge of 3% of the total cost is added to the total cost to determine the final bill.

