

# **FAST**

# National University of Computer and Emerging Sciences Peshawar

Lecture # 05

## Software Construction and Development (Java Programming)

**Instructor:** Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE



الذى علم بالقلم. علم الانسان ما لم يعلم.



# Decision Making Statements OR Decision Control Statements

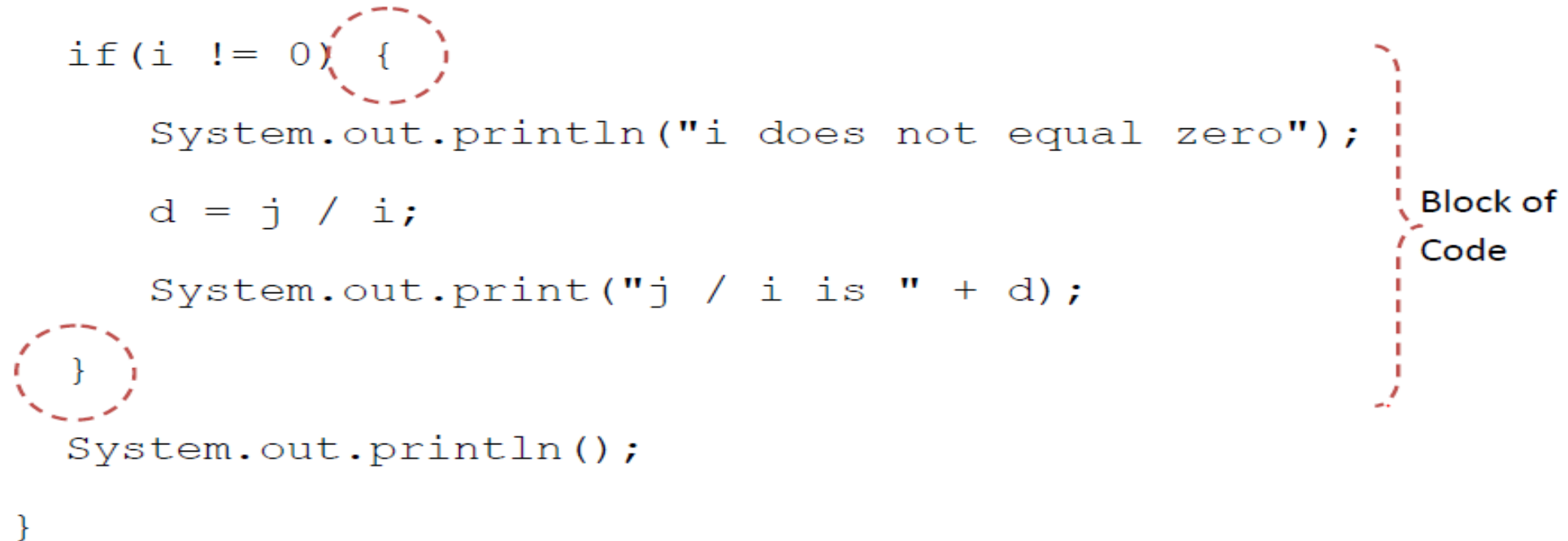
# Contents

1. If statement
2. If-else statement
3. Nested If-else statement
4. Multiple if statement
5. If else if ladder statement
6. Conditional operator.
7. Switch Statement

# Blocks of Code

Whenever we write an IF statement or a loop, if there is more than one statement of code which has to be executed, this has to be enclosed in braces, i.e. '{ ... }'

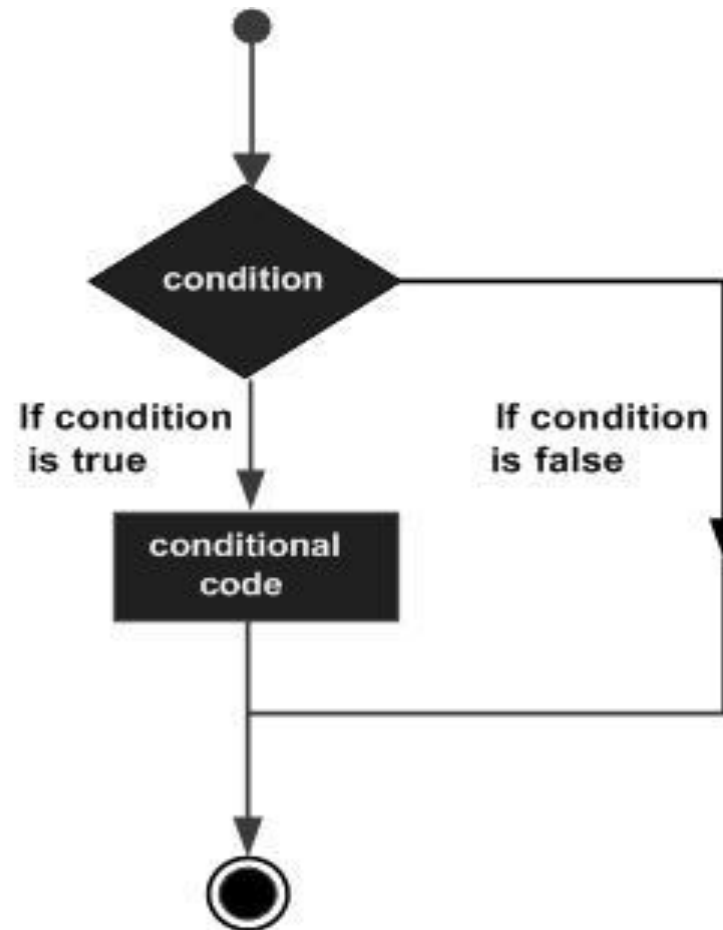
```
if (i != 0) {  
    System.out.println("i does not equal zero");  
    d = j / i;  
    System.out.print("j / i is " + d);  
}  
System.out.println();  
}
```



# Decision Making Statements OR Decision Control Statements

- ❖ Decision making structures have one or more conditions to be evaluated or tested by the program, along with a statement or statements that are to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.
- ❖ Used to change the flow.
- ❖ In these statements (conditions) order or sequence of the statements are changed.

# Decision Making Statements OR Decision Control Statements



# 1) If Statement

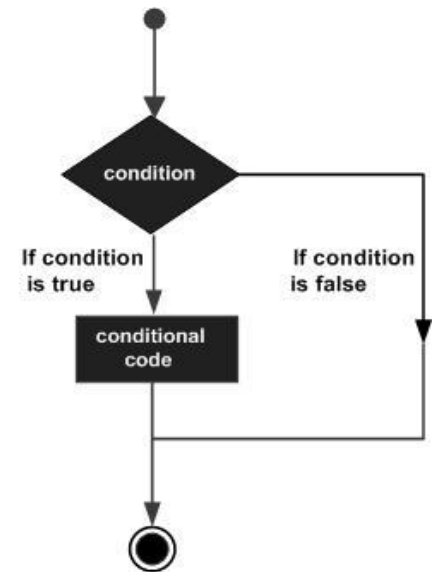
**if statement** will execute or skip or ignore a block of code depending one condition.

**Syntax:**

```
if(conditional expression)
{
```

```
// Statements will execute if the Boolean expression is true
```

```
}
```



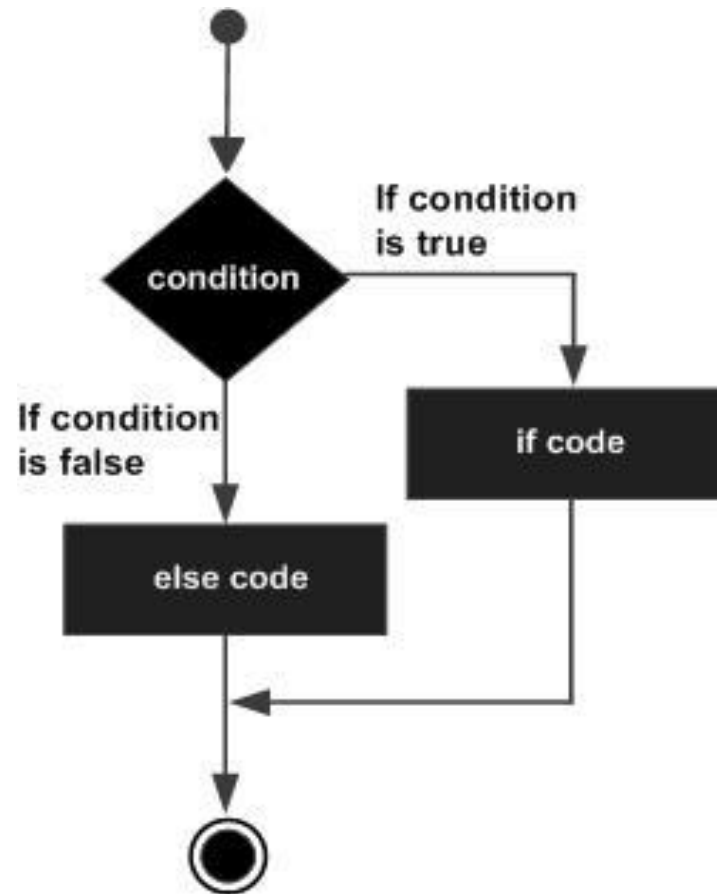
# 1) If Statement...

Example:

```
public class Test {  
    public static void main(String args[]) {  
        int x = 10;  
        if( x < 20 )  
        {  
            System.out.print("This is if statement");  
        }  
    }  
}
```



## 2) if...else statement



## 2) if...else statement...

- ❖ Used for making two way decision.
- ❖ It will execute if block if condition is true and will execute another block (else block) if condition is false.
- ❖ It will take one action if the condition is true and take another action if condition is false.

```
if(condition) {  
    // Executes this block when the Boolean expression is true  
}  
else {  
    // Executes this block when the Boolean expression is false  
}
```

## 2) if...else statement...

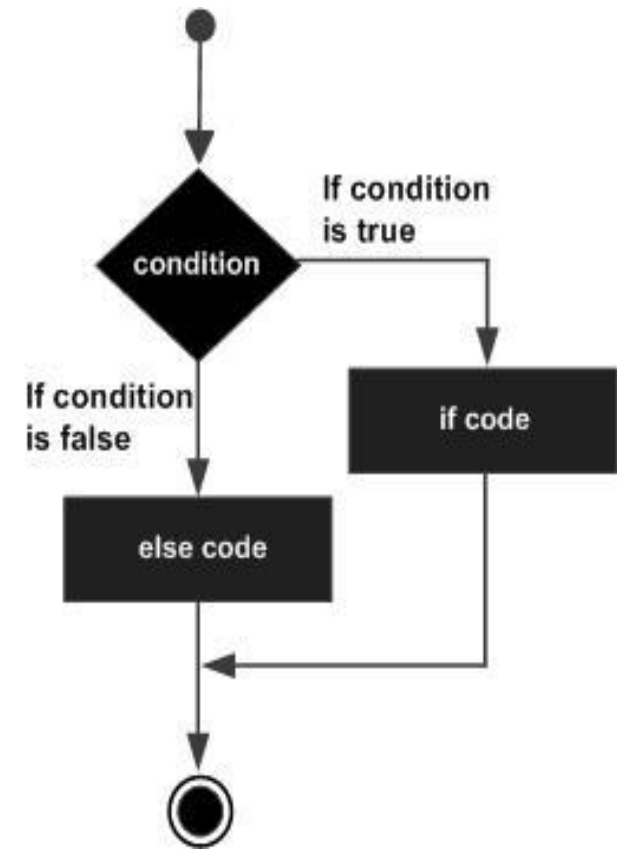
```
int x = 30;
```

```
if( x < 20 ) {
```

```
    System.out.print("This is if statement");  
}
```

```
else
```

```
{  
    System.out.print("This is else statement");  
}
```



## 2) if...else statement Tasks

1. Find positive and negative numbers using if else statement.
2. Find even and odd numbers using if else statement.
3. Find leap year using if else statement.
4. Write a java program which will get two numbers from user and find large number between them using if else statement.

**Leap year Hints:** common year has 365 days (feb 28 days)

Leap year has 366 days (feb 29 days)

`year%4==0`   leap year

### 3) Nested if statement

You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).

```
public class Test {  
    public static void main(String args[]) {  
        int x = 30;  
        int y = 10;  
        if( x == 30 ) {  
            if( y == 10 ) {  
                System.out.print("X = 30 and Y = 10");  
            }  
        }  
    }  
}
```

## 4) Multiple if statement

- This statement is used when we have to check multiple conditions.
- Used for multiple way decision making.

### Syntax:

```
    if(condition)
{   statements(s);   }
    if(condition)
{   statements(s);   }
    if(condition)
{   statements(s);   }
```

## 4) Multiple if statement Tasks

1. Find positive, negative and neutral numbers using multiple if statement.
2. Take value of temperature from user and find status of weather accordingly.
3. Take value of percentage from user and find grades based on percentage value.

## 5) if else if ladder statement

- This statement is used to check multiple conditions.
- Used to execute one condition from multiple statements.

- **Syntax:**

```
    if(condition)
{   statements(s);   }
    else if(condition)
{   statements(s);   }
    else if(condition)
{   statements(s);   }
    else {   statements(s);   }
```



## 5) if else if ladder statement Tasks

1. Find positive, negative and neutral numbers using multiple if statement.
2. Take value of temperature from user and find status of weather accordingly.
3. Take value of percentage from user and find grades based on percentage value.
4. Make a calculator using if else if ladder statement which perform the addition, subtraction, multiplication, division and remainder operations. Take value and operators from user on runtime.

# if else if ladder statement VS multiple if

In multiple if statement all conditions are checked due to which compiler is bored and it is difficult task for compiler.

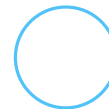
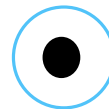
That's why we are using if else if ladder statement instead of multiple if.

## Note:

1) check boxes is the example of multiple if



2) Radio buttons are the example of if else if ladder statement.



## 6) Conditional Operator (? :)

- It is ternary operator and work on three operands.
- It works like if else statement.
- **Syntax:**

(condition) ? statement 1 : statement 2;

Statement must be single and it is the limitation of conditional operator.

### **Example:**

```
int n1=4;
```

```
int n2=5;
```

```
int result= (n1>n2) ? n1 : n2;
```

```
print(result);
```

## 6) Conditional Operator (? :) Tasks

1. Write a java program which will get two numbers from user and find large number between them using conditional operator.
2. Find positive and negative numbers using conditional operator.
3. Find even and odd numbers using conditional operator.

# Logical Operators

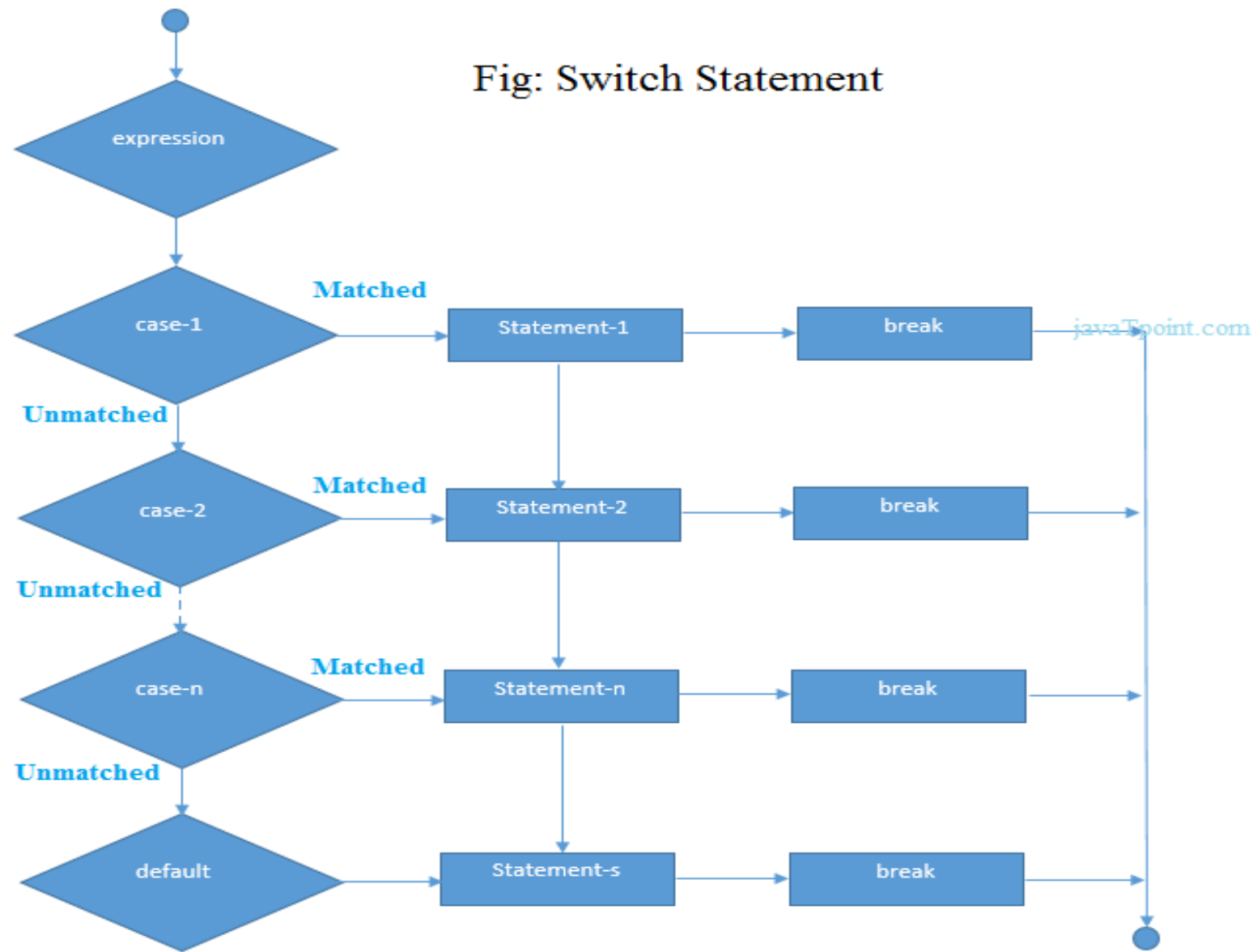
Used for compound condition or expression

- 1) AND (&&)
- 2) OR (||) pip sign

## 7) Switch statement

- Used when multiple choices are given and one is to be selected. It is like if else if ladder statement.
- Used to select one several actions based on the value of variable or expression.
- **Switch case statement** is used when we have number of options (or choices) and we may need to perform a different task for each choice.

Fig: Switch Statement



# 7) Switch statement...

**Syntax:**

```
switch(variable/expression)
```

```
{
```

```
case value 1:
```

```
Statement(s); // code to be executed
```

```
break;
```

```
case value 2:
```

```
Statement(s); // code to be executed
```

```
break;
```



## 7) Switch statement...

**Syntax:**

.

.

.

case value n:

Statement(s);    // code to be executed

break;

default:

Statement(s);    // code to be executed if all cases are not matched

}

# 7) Switch statement...

## Example

```
int num=2;  
switch(num+2) {  
case 1:  
    System.out.println("Case1: Value is: "+num);    break;  
case 2:  
    System.out.println("Case2: Value is: "+num);    break;  
case 3:  
    System.out.println("Case3: Value is: "+num);    break;  
default:  
    System.out.println("Default: Value is: "+num);  
}
```

## 7) Switch statement Tasks

1. Make a calculator using switch statement which perform the following addition, subtraction, multiplication, division and remainder value. Take value from user on runtime.
2. Write a java program using switch statement which get month name from user and display month number accordingly.

# Switch statement VS if else if

- ❖ If a program contains conditions or compound conditions then we use if else if
- ❖ If program contains single variable or expression then we use switch statement.

## Compound conditions

- 1) `if(a>b && a>c)`
- 2) `if(a>b || a>c)`

# Java Switch statement is fall through

It means it executes all statements after match if break statement is not used with switch cases.

## **Example:**

```
alphabet= input.nextLine();  
switch(alphabet)  
{  
case "a" :  
case "e" :  
case "i" :  
case "o" :  
case "u" :
```

# Java Switch statement is fall through...

```
case "A" :  
case "E" :  
case "I" :  
case "O" :  
case "U" :
```

```
System.out.println("You entered vowel");  
break;  
default:  
System.out.println("You entered consonant");  
  
} // switch body Closed
```

# Break Statement

The break statement is used to exit from the body of the switch structure or loop structure.

The break statement terminates the execution of the loop when it is used inside the body of the loop.

**Syntax:** break;

# boolean variable example

```
Scanner input= new Scanner(System.in);  
int number;  
System.out.println("Enter any number");  
number = input.nextInt();  
boolean even;  
even = (n%2==0);  
if(even)  
{      System.out.println("Even number")  }
```



# boolean variable example...

```
else
```

```
{
```

```
System.out.println("Odd Number");
```

```
}
```

# THANK YOU

