

# Let's Decode This Algorithm

Start

Accept a Number

Divide the Number by 2

Is the remainder = 0

Yes : Display Number is Even

Go to End

No : Display Number is Odd

End

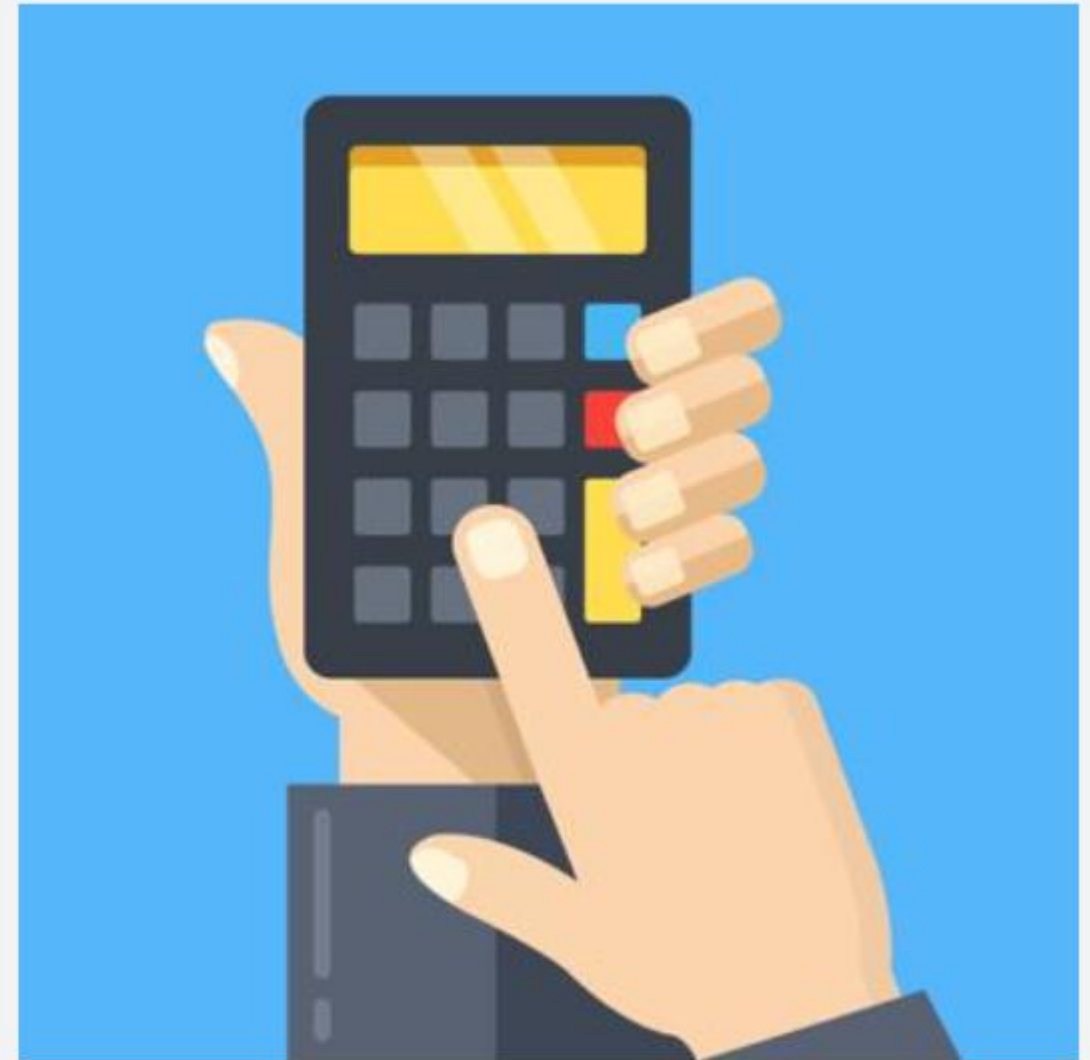


## Think and Tell

- How can we convert the **Algorithm** to a Java program?
- Are the operators sufficient to convert this algorithm to a Java Program?
- What do we need to learn about decision making?

## Think and Tell

- What type of program can you write to create a calculator that can add, subtract and multiply two numbers?
- Can only operators help us create a calculator or do we need to learn something more to achieve this task?



# Think and Tell

- How can you calculate grades of the given marks?
- Are there any conditions to calculate it?
- Is there a condition that returns sometimes true and sometimes false to achieve the above said grade calculation?



# Implement Conditional Constructs





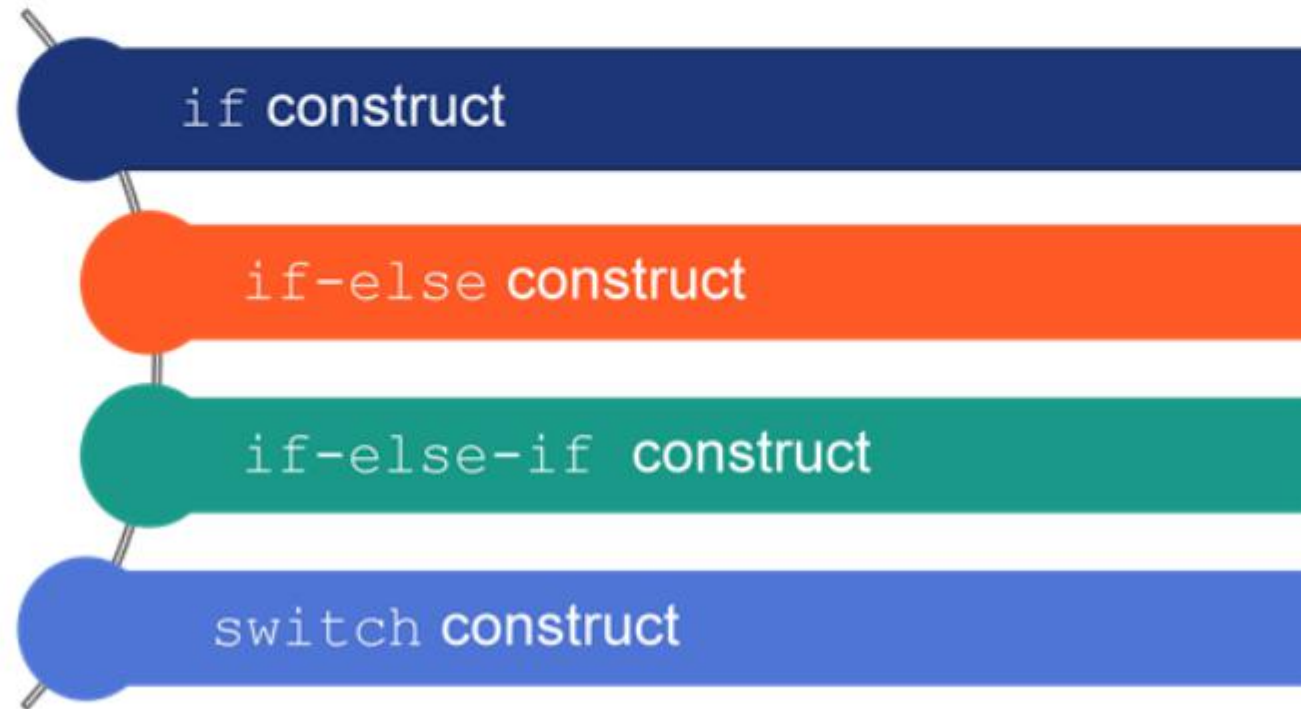
# Learning Objectives

- Describe Conditional Constructs
- Create *if* and *if-else* construct
- Employ ternary operator for decision-making
- Use Logical Operators
- Implement *if-else-if* and *nested* construct
- Apply *switch-case* construct
- Scanner Class



# Conditional Constructs

- A conditional **Construct** consists **of a condition and a task**. When the condition is true, the application performs the task. The condition part of a conditional statement is also called an expression.
- Look at the following Conditional Constructs:





# **If-construct**



# If-construct

- The **Java if statements** are the simplest decision-making statements.
- They are used to specify a block of code to be executed, if a specified condition is true.

```
if(condition)
{
    // Statements to execute if
    // condition is true
}
```

- Here the condition is an expression which results in either true or false.

# If-construct: Evaluation to True

```
public static void main(String[] args) {  
    int number = 111;  
    //Conditional Construct - This expression is true  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
    }  
    System.out.println("This will also get printed");  
}
```

```
3 or more digit number  
This will also get printed
```

- The *if*-construct executes statements based on the condition of expression.
- The expression is evaluated to be either **true** or **false**.
- If the expression is **true**, then the statements inside the block are executed and the statements outside the block gets printed.

# If-construct: Evaluation to False

```
public static void main(String[] args) {  
    int number = 99;  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
    }  
    System.out.println("This will only get printed");  
}
```

This will only get printed

- If the expression evaluates to **false**, then **only the** statements outside the block will get printed as shown in the demo.

# If-construct

```
public static void main(String[] args) {  
    int number = 50;  
    if (number % 2 == 0) {  
        System.out.println("EVEN number");  
    }  
    System.out.println("ODD number");  
    System.out.println("Hello");  
}
```

```
EVEN number  
ODD number  
Hello
```

- The expression is evaluated first.
- `number % 2 == 0` is true since the number is 50 and even.
- So, the expected output should be an "EVEN number" and "Hello".
- But the output is an "EVEN number, ODD number" and "Hello".
- Can we avoid this situation by adding another block?



# **If-else Construct**

# If-else Construct

While working with if-else constructs either the if or the else block will get executed.

- 'If' will be executed when the condition is true.
- 'else' will get executed when the condition is false.

```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```



## If-else Constructs

- The expression is evaluated first.
- `number%2 == 0` is true since the number is 50 and even.
- The statement in the if block is executed and 'EVEN Number' is displayed on the screen.
- The statement ODD number does not get printed here since the else block is not being executed.

```
public static void main(String[] args) {  
    int number = 50;  
    if (number % 2 == 0) {  
        System.out.println("EVEN number");  
    } else {  
        System.out.println("ODD number");  
    }  
    System.out.println("Hello");  
}
```

```
EVEN number  
Hello
```

# Quick Check

What will be the output for the following code:

- a) Syntax Error
- b) Hi
- c) Hello

```
public static void main(String[] args) {  
    if(5<4) {  
        System.out.println("Hi");  
    }  
    else {  
        System.out.println("Hello");  
    }  
}
```



# Quick Check: Solution

What will be the output for the following code:

- a) Syntax Error
- b) Hi
- c) **Hello**

```
public static void main(String[] args) {  
    if(5<4) {  
        System.out.println("Hi");  
    }  
    else {  
        System.out.println("Hello");  
    }  
}
```



# Quick Check

What will be the output for the following code:

- A) 3 or more-digit number  
This was executed
- B) Number is not a 3 or more-digit number
- C) This was executed

```
public static void main(String[] args) {  
    int number = 99;  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
        System.out.println("This was executed");  
    } else {  
        System.out.println("Number is not a 3 or more digit number ");  
    }  
}
```





# Quick Check: Solution

What will be the output for the following code:

A) 3 or more-digit number

This was executed

**B) Number is not a 3 or more-digit number**

C) This was executed

```
public static void main(String[] args) {  
    int number = 99;  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
        System.out.println("This was executed");  
    } else {  
        System.out.println("Number is not a 3 or more digit number ");  
    }  
}
```



## Positive or Negative Number

Write a program that will check if the given number is greater than zero.

If the number is greater than zero, print the "Number is a positive number".

If the number is not greater than zero, then print the "Number is a negative number".

DEMO



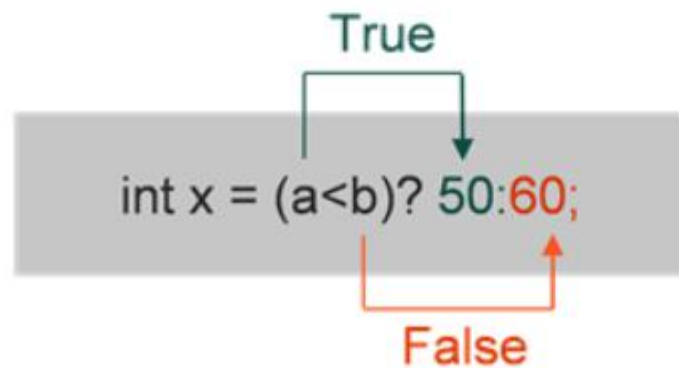




# **Ternary Operator**

# Ternary Operator

- The **ternary operator** is a type of Java conditional operator. The **ternary operator (?:)** consists of three operands (**A value involved in an operation** is called an operand).
- It is a one-liner replacement for the if-else statement.
- **The syntax of ternary operator is:**
  - `boolean_expression ? expression 1 : expression 2`
- Here, `boolean_expression` , `expression1` and `expression 2` are the three operands.



# Ternary Operator

```
public static void main(String[] args) {  
    int num1 = 23;  
    int num2 = 24;  
    int result = 0;  
    //Ternary Operator  
    result = (num1 > num2) ? num1 : num2;  
    System.out.println("Greatest Number: " + result);  
}
```

Greatest Number: 24

- The purpose of a ternary operator is to make your **code more concise and readable**.
- Here (num1 > num2) is an **expression**.
- If the expression is **true**, then **num1 will get printed**.
- If the expression is **false**, then **num2** will get printed.
- As the **expression** is false, the num2 (24) is printed as a result.


# Ternary Operator vs. if-else

Ternary Operator is a programming statement.

```
public static void main(String[] args) {  
    int num1 = 23;  
    int num2 = 24;  
    int result = 0;  
    //Ternary Operator  
    result = (num1 > num2) ? num1 : num2;  
    System.out.println("Greatest Number: " + result);  
}
```

"If-else" is a programming block.

```
public static void main(String[] args) {  
    int num1 = 23;  
    int num2 = 24;  
    if(num1>num2){  
        System.out.println("Greatest Number is " + num1);  
    }  
    else  
    {  
        System.out.println("Greatest Number is " + num2);  
    }  
}
```



Check whether the given number is a  
***three-digit number, a two-  
digit number, or neither.***  
How will you write a program for this?



# **If-else-if construct**



# The if-else-if construct

```
public static void main(String[] args) {  
    int number = 111;  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
    } else if (number >= 10) {  
        System.out.println("2 digit number");  
    } else {  
        System.out.println("Number is neither 2 nor 3 digit number");  
    }  
}
```

3 or more digit number

- In this scenario, we have two conditions.
- First:  
The expression **number>=100** if true then execute the statement inside the first if block.
- Second:  
The if **number>=10** if true then execute the statement inside the second if block.
- If none of the conditions are true execute the statement inside the else block.
- As the first expression is true i.e. **(111>=100)**, the output is 3 or more-digit number.

# Quick Check

Predict the output for the following code:

```
public static void main(String[] args) {  
    int marks = 80;  
  
    if (marks > 70)  
        System.out.println("Distinction");  
    if (marks > 35)  
        System.out.println("Pass");  
    else  
        System.out.println("Fail");  
    System.out.println("Better luck next time");  
}
```

1. Distinction
2. Distinction  
Pass  
Better luck next time
3. Fail  
Better luck next time
4. Pass



# Quick Check: Solution

Predict the output for the following code:

```
public static void main(String[] args) {  
    int marks = 80;  
  
    if (marks > 70)  
        System.out.println("Distinction");  
    if (marks > 35)  
        System.out.println("Pass");  
    else  
        System.out.println("Fail");  
    System.out.println("Better luck next time");  
}
```

1. Distinction

2. **Distinction**  
**Pass**  
**Better luck next time**

3. Fail  
Better luck next time

4. Pass





## Digit Number

Write a program that assigns a grade based on the value of a test score.

Grade A for scoring 90 or above.

Grade B for scoring 80 or above.

Grade C for scoring 70 or above.

Take the input of the score from the user using the Scanner.

DEMO




Check whether the given number is a **three-digit number**. If it has three digits, then find out if the number fall between 100 and 150. If the number is not of three digits, then only print its **two-digit number**.

How will you write a program for this?

# Logical Operator

- Logical Operators work on two or more expressions to produce one final output.
- There are three logical operators :
  - **Logical AND (&&)** - Used to compare two expressions and return true if both the expressions are true.
    - (5> 10&& 23<56) - The first expression is false, and the second expression is true as the 'logical AND' operator is used, so the result will be false.
  - **Logical OR (||)** - Used to compare two expressions and return false if both the expressions are false, or else return true if any one of the expressions is true.
    - (5> 10&& 23<56) - The first expression is false, and the second expression is true as the logical OR operator is used so the result will be true.
  - **Logical NOT(!)** - This is a unary operator and returns true when the condition under consideration is not satisfied or is a false condition.
    - !(5<10) - (5<10) - This condition is true but since the NOT(!) expression is used before the condition; the result will be false.





# **Nested if-else**

# Logical Operator AND (&&)

```
public static void main(String[] args) {  
    //Initializing variables  
    int num1 = 10, num2 = 20, num3 = 20, result = 0;  
    //if ((10<20) && (20 == 20))  
    // if((true) && (true)) ---> condition becomes true  
    if ((num1 < num2) && (num2 == num3)) {  
        result = num1 + num2 + num3;  
        System.out.println("The sum is: " + result);  
    }  
    else{  
        System.out.println("False conditions");  
    }  
}
```

The sum is: 50

- Condition1: num1 < num2 is true  
Condition2: num2 == num3 is true
- If(condition1 & condition2) is **true**, then add all numbers and store them in a variable result.
- If the “if condition” is false, then print the statement or else block it.

# Nested if-else

```
public static void main(String[] args) {  
    int number = 111;  
    //check if condition is true then it will go to next if block  
    //if condition is false it will go to else block  
    if (number >= 100) {  
        System.out.println("3 or more digit number");  
        if(number>100 || number<150)  
        {  
            System.out.println("Number is between 100 and 150");  
        }  
    } else if (number >= 10) {  
        System.out.println("2 digit number");  
    } else {  
        System.out.println("Number is neither 2 nor 3 digit number");  
    }  
}
```

3 or more digit number

Number is between 100 and 150

- When an 'if statement' is found within another 'if statement', it is known as a nested 'if statement'.
- As the first expression evaluates to be true, the 'if block' gets executed. Since there is one more if condition inside the block, the nested 'if condition' also gets executed.
- Now the inner block has two expressions having OR as the logical operator and both conditions are true, so the result is true, and the statement is printed.

# Quick Check

What will be the output for the following code:

```
public static void main(String[] args) {  
    // initializing variables  
    int num1 = 10, num2 = 1, num3 = 10, num4 = 30;  
    // using logical OR to verify  
    if (num1 > num2 || num3 == num4){  
        System.out.println("One or both"  
            + " the conditions are true");  
    }  
    else{  
        System.out.println("Both the"  
            + " conditions are false");  
    }  
}
```

1. Syntax Error.
2. One or both conditions are true.
3. Both conditions are true.
4. One or both conditions are false.
5. Both conditions are false.





# Quick Check: Solution

What will be output for the following code:

```
public static void main(String[] args) {  
    // initializing variables  
    int num1 = 10, num2 = 1, num3 = 10, num4 = 30;  
    // using logical OR to verify  
    if (num1 > num2 || num3 == num4){  
        System.out.println("One or both"  
            + " the conditions are true");  
    }  
    else{  
        System.out.println("Both the"  
            + " conditions are false");  
    }  
}
```

1. Syntax Error.
2. **One or both conditions are true.**
3. Both conditions are true.
4. One or both conditions are false.
5. Both conditions are false.





## Eligible to Donate Blood

Write a program creating two variables: one with age and the other with weight.

If the age is 18 years or more, then check the weight.

If the weight is between 50 to 70 then print "You are eligible to donate blood."

If any of the criteria are false, then display the message saying, "You are not eligible to donate blood."

DEMO



# How Do You Take Input From a User?

```
public class EmployeeDetails {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Enter Age :");  
        int age = scanner.nextInt();  
        System.out.println("Enter Salary :");  
        float salary = scanner.nextFloat();  
        System.out.println("Enter Name :");  
        String name = scanner.next();  
        scanner.close();  
    }  
}
```

- The values for the variables can be accepted from the user.
- This can be done using the Scanner class provided by Java.
- This is a simple text scanner that can parse primitive types and strings.

# Steps to Read Input

- Initialize the Scanner

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

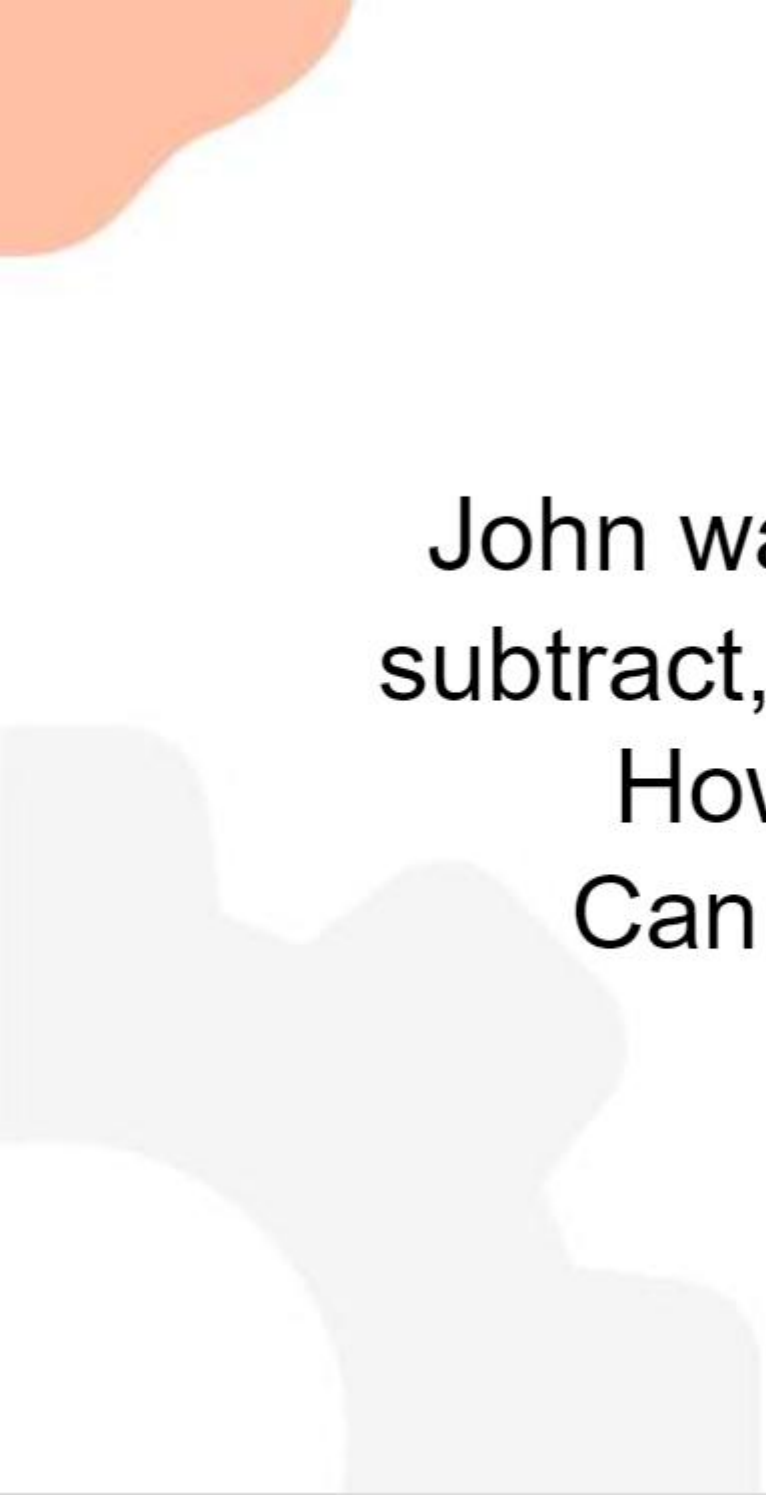
- Utilize the Scanner functionality to read values – To read `int` values from the user, the `nextInt()` method of the `Scanner` is used

```
System.out.println("Enter Age : ");  
int age = sc.nextInt();
```

- Close the Scanner

```
sc.close();
```

**Note:** Methods will be discussed in detail in the later sessions



John wants to create a calculator to add, subtract, multiply, and divide two numbers.

How will you write this program?

Can the if-else construct be used?





# **Switch-Case**



```
switch (expression){  
    case 1:  
        //statement of case 1  
        break;  
    case 2:  
        //statement of case 2  
        break;  
    case 3:  
        //statement of case 3  
        break;  
    .  
    .  
    .  
    case N:  
        //statement of case N  
        break;  
    default;  
        //default statement  
}
```

## Switch Case

- The switch statement is used **to test the equality of a variable against several values specified in the test cases.**
- The expression is evaluated once and compared with the values of each case.
- Here the expression can be a conditional expression and it can be any data type.

```
public static void main(String[] args) {  
    //Declaring a variable for switch expression  
    int number=100;  
    //Switch expression  
    switch(number){  
        //Case statements  
        //case value1  
        case 50: System.out.println("50");  
            break;  
        //case value2  
        case 100: System.out.println("100");  
            break;  
        //case value3  
        case 150: System.out.println("150");  
            break;  
        //Default case statement  
        default: System.out.println("Not in 50, 100 or 150");  
    } //end of switch  
}
```

**Here 100 will get printed**

## The Switch Construct

- If the expression matches with the case value1 (50), the statement of case value1 is executed.
- Similarly, the code of case value2 (100) is executed if the expression matches with value2.
- If there is no match, the code of the default case is executed.
- Each case statement can have an optional break statement. When the control reaches the break statement, it jumps the control after the switch expression. If a break statement is not found, it executes the next case. ( break is a statement that is used to break the current execution flow of the program)

**Note** – More about the break keyword will be explored in next Sprint

# Quick Check

Predict the output for the following code:

1. TWENTY
2. Syntax Error
3. FORTY
4. TWENTY  
THIRTY  
FORTY

```
public static void main(String[] args) {  
    int num = 20;  
    switch (num) {  
        case 10:  
            System.out.println("TEN");  
            break;  
        case 20:  
            System.out.println("TWENTY");  
        case 30:  
            System.out.println("THIRTY");  
        default:  
            System.out.println("Forty");  
    }  
}
```





# Quick Check: Solution

Predict the output for the following code:

1. TWENTY
2. Syntax Error
3. FORTY
4. **TWENTY**  
**THIRTY**  
**FORTY**

```
public static void main(String[] args) {  
    int num = 20;  
    switch (num) {  
        case 10:  
            System.out.println("TEN");  
            break;  
        case 20:  
            System.out.println("TWENTY");  
        case 30:  
            System.out.println("THIRTY");  
        default:  
            System.out.println("Forty");  
    }  
}
```



## Creating a Simple Calculator

Write a program to create a simple calculator that performs addition, subtraction, multiplication, and division based on the user input.

The program takes the value of both the numbers (entered by the user) and then the user is asked to enter the operation (+, -, \* , and /), based on the input program performing the selected operation on the entered numbers using Switch Case.

DEMO

