iMSCIT

SEMESTER:III

1601301

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

UNIT:II

Unit 2: Decision Control Statements and Looping Techniques

- **Decision Controls**
  - If
  - If-else
  - Nested If-else
  - If-else Ladder
  - Switch
- **Looping Techniques**
  - While loop
  - do-while loop
  - for loop
  - nested loop
- **Other Statements (Break, continue, GOTO, exit)**

Decision Making

- **Decision making statements in a programming language help the programmer to transfer the control from one part to other parts of the program.**
- This involves a decision making condition to see whether a particular condition is satisfied or note.
- On the basis of real time applications it is essential:

  – To alter the flow of a program

  – To test the logical conditions

  – To control the flow of execution as per the selection.

Decision Making

- Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements 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.
- C programming language **assumes any non-zero and non-null values as true**, and if it is either **zero or null, then it is assumed as false value.**

IF Statement

- **The if statement is used to check some given condition and perform some operations depending upon the correctness of that condition.**
- It is mostly used in the scenario where we need to perform the different operations for the different conditions. The syntax of the if statement is given below.
- **An if statement consists of a Boolean expression followed by one or more statements.**

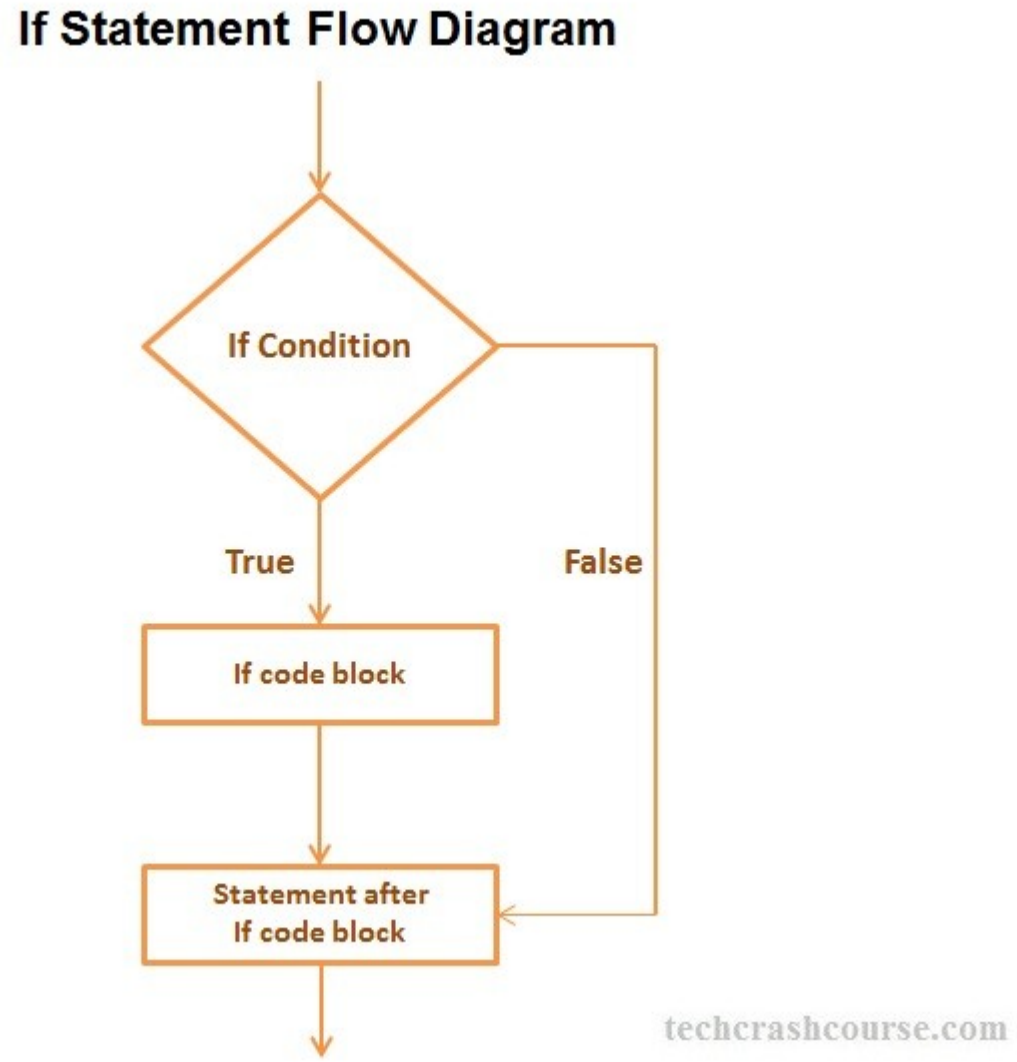
IF Statement

Syntax:

```
if(Boolean_Expression)
 {
      /* statement(s) will execute if the Boolean expression is true */
 }
```

- If the Boolean expression evaluates to **true**, then the **block of code inside the 'if' statement** will be executed.
- If the Boolean expression evaluates to **false**, then **the first set of code after the end of the 'if'** statement (after the closing curly brace) will be executed.

IF Statement



IF-else Statement

- **The if-else statement is used to perform two operations for a single condition.**
- The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., **one is for the correctness of that condition**, and the **other is for the incorrectness of the condition.**
- Here, we must notice that if and else block cannot be executed simultaneously.
- **An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.**

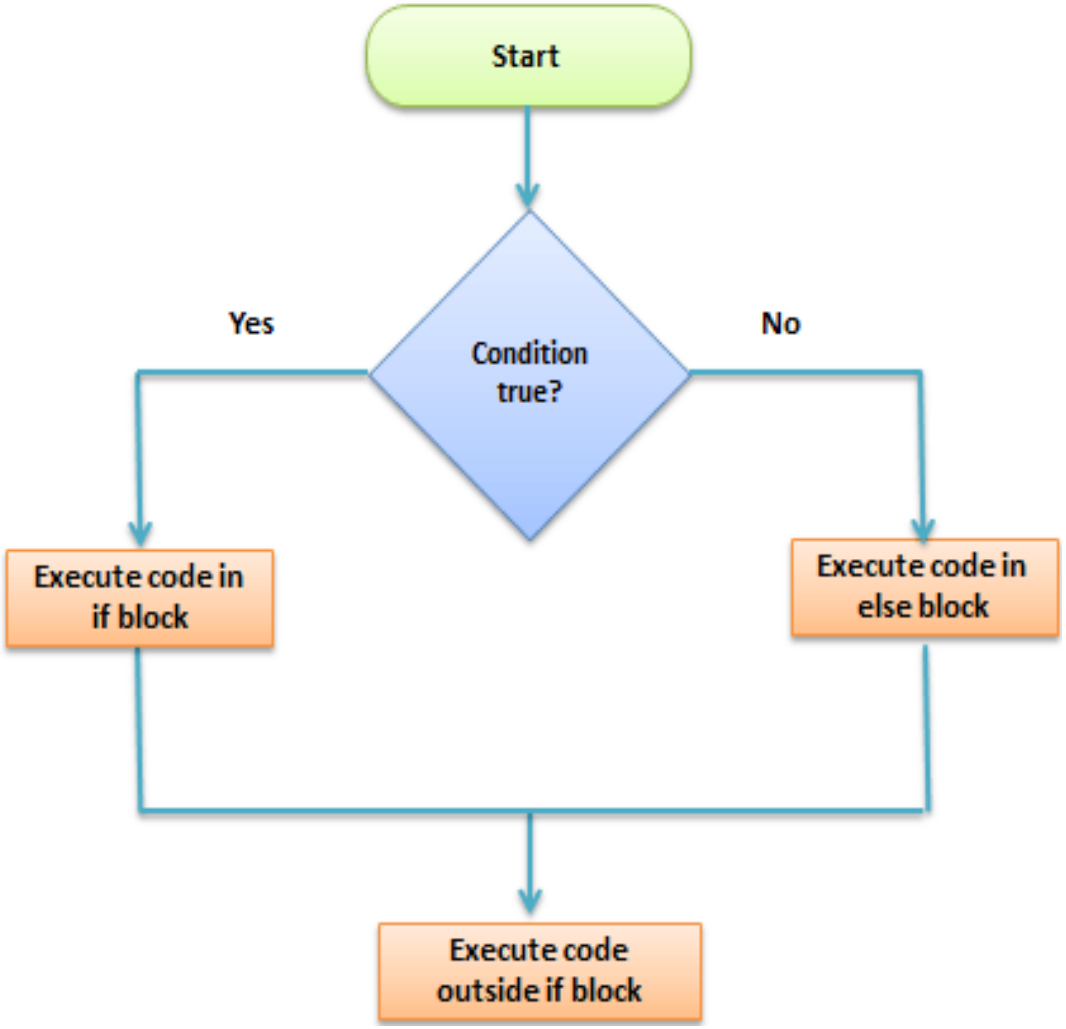
IF-else Statement

Syntax:

```
if(expression)
{
//code to be executed if condition is true
}
Else
{
//code to be executed if condition is false
}
```

- If the Boolean expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

IF-else Statement



Nested if-else Statement

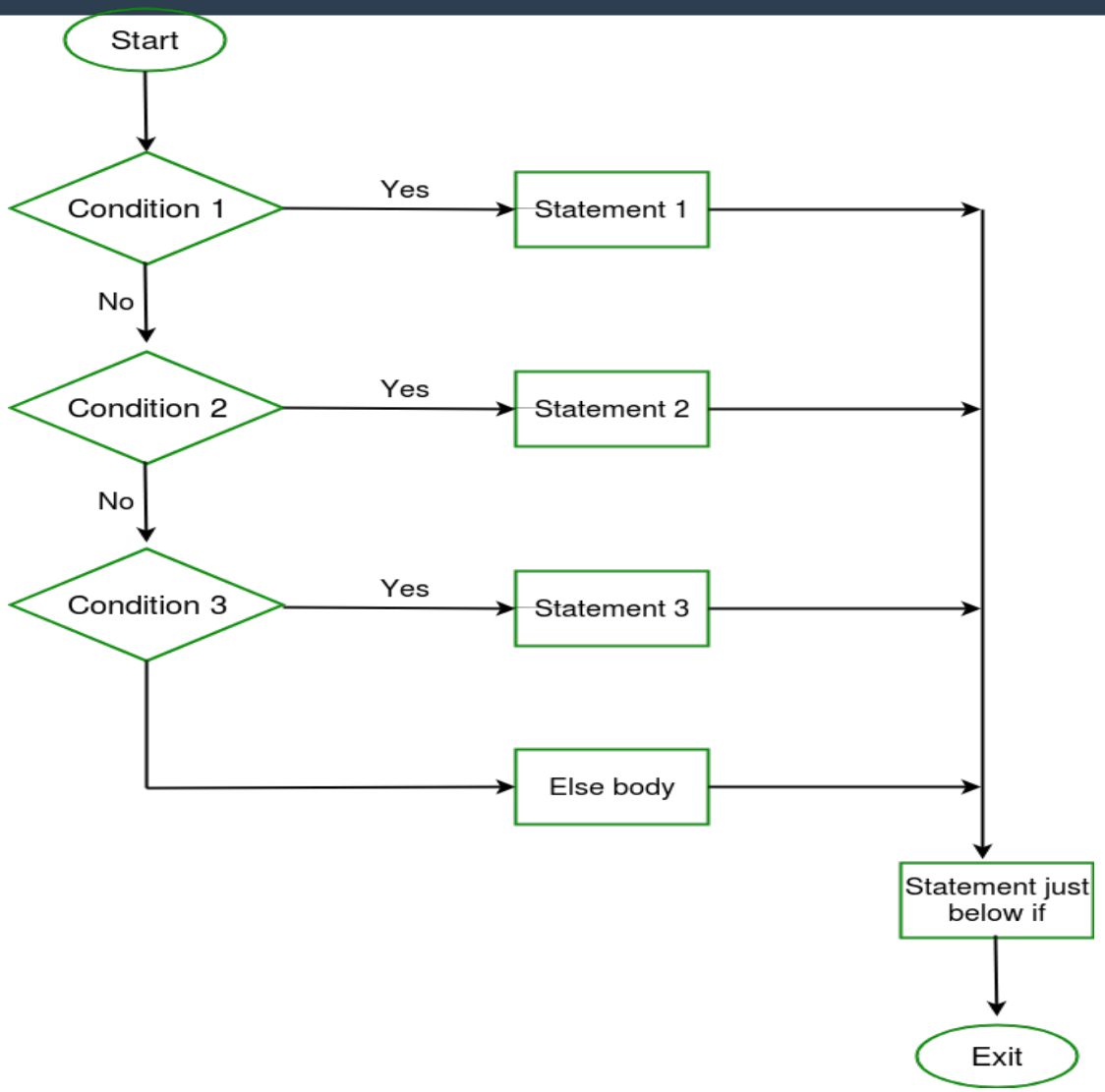
- Nested if-else can be chained with one another.
- If the condition is true control passes to the block following first if. In this case, we may have one more if statement whose condition is again checked. This process continues till there is no if statement in the last if block.
- If the condition is false control passes to else block. In that case, we may have one more if statement whose condition is again checked. This process continues till there is no if statement in the last else block.

Nested if-else Statement

Syntax

```
if ( condition1 )
{
    if ( condition2 )
    {
        ....
        True block of statements 1;
    }
    ....
}
else
{
     False block of condition1;
}
```

Nested if-else Statement



If else-if ladder Statement

- The if-else-if ladder statement is an extension to the if-else statement.
- It is used in the scenario where there are multiple cases to be performed for different conditions
- In if-else-if ladder statement, if a condition is true then the statements defined in the if block will be executed, otherwise if some other condition is true then the statements defined in the else-if block will be executed, at the last if none of the condition is true then the statements defined in the else block will be executed.
- There are multiple else-if blocks possible

If else-if ladder Statement

```
if(condition1)
{
//code to be executed if condition1 is true
}else if(condition2)
{
//code to be executed if condition2 is true
}
else if(condition3){
//code to be executed if condition3 is true
}
.....
else
{
//code to be executed if all the conditions are false
}
```

If else-if ladder Statement

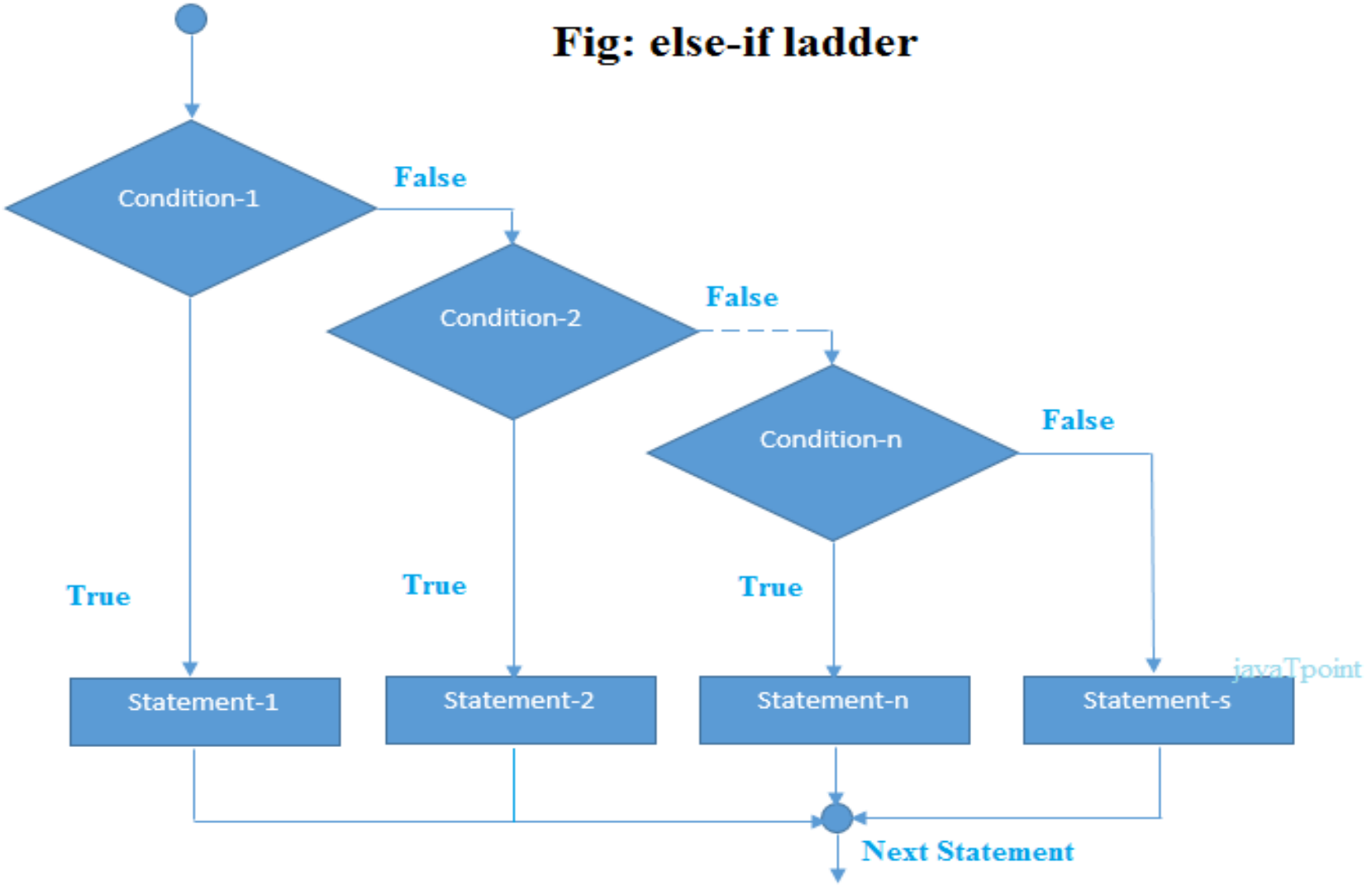


Fig: else-if ladder

Switch Statement

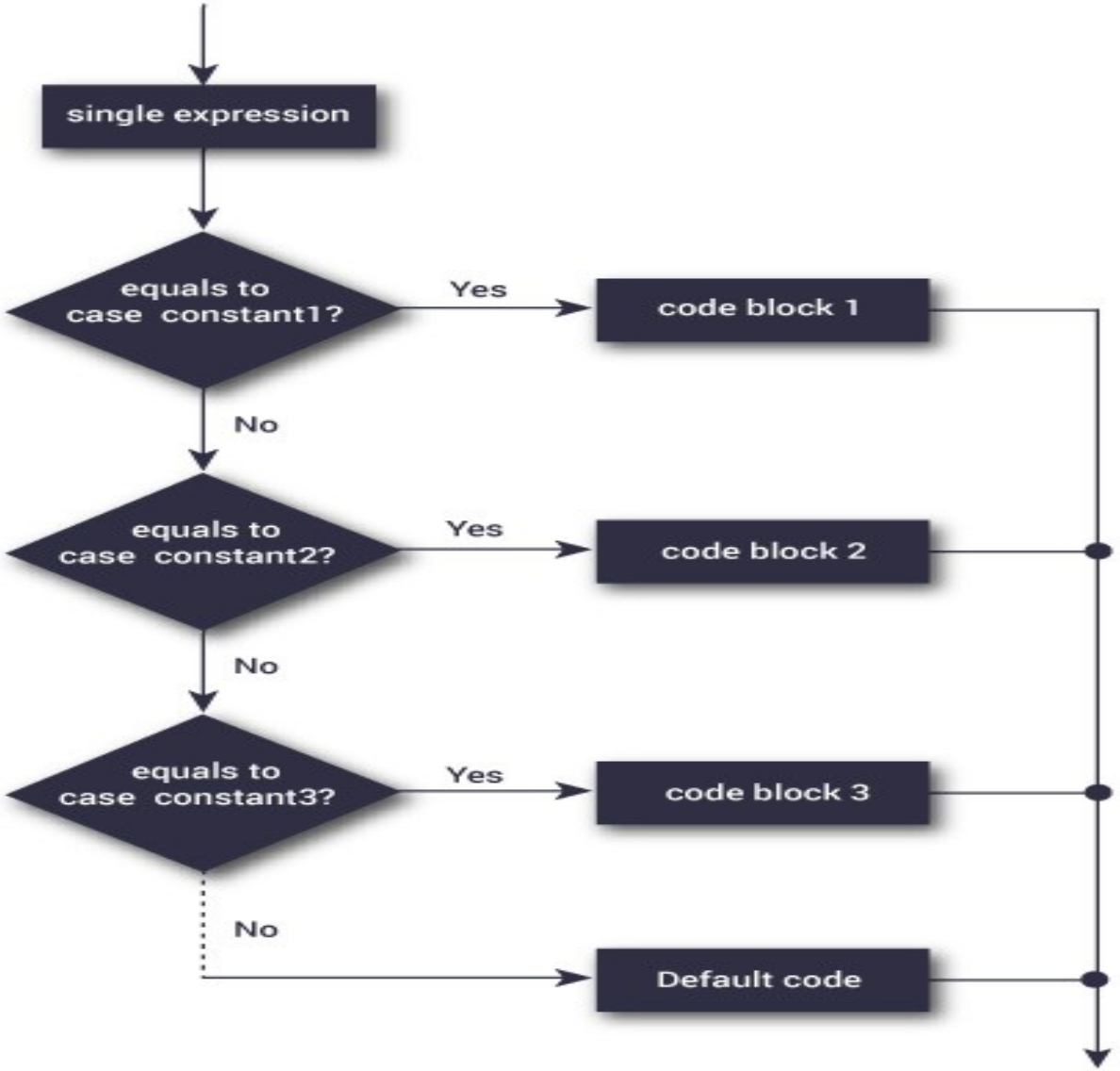
- The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possibles values of a single variable called switch variable.
- We can define various statements in the multiple cases for the different values of a single variable.

Switch Statement

Syntax:

```
switch(expression){
  case value1:
            //code to be executed;
            break;
   case value2:
            //code to be executed;
              break;
        ......
    Default:
              code to be executed if all cases are not matched;
}
```

Switch Statement



How does the switch statement work?

- The expression is evaluated once and compared with the values of each case label.
- If there is a match, the corresponding statements after the matching label are executed. For example, if the value of the expression is equal to value2, statements after case value2: are executed until break is encountered.
- If there is no match, the default statements are executed.
- If we do not use break, all statements after the matching label are executed.
- The default clause inside the switch statement is optional.

THANK YOU