

Control Statements

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering, Pune
www.harishgadade.com

Control Statements

```
graph TD; A[Control Statements] --> B[Decision Making Statements]; A --> C[Looping / Iterative Statements]; A --> D[Jumping Statements]; B --> B1[- If - Statements]; B --> B2[- Switch-Case Statements]; B --> B3[- Conditional Statements]; C --> C1[- While Loop]; C --> C2[- Do-While Loop]; C --> C3[- For Loop]; D --> D1[- Break Statements]; D --> D2[- Continue Statements]; D --> D3[- Goto Statements];
```

Decision Making Statements

- If - Statements
- Switch-Case Statements
- Conditional Statements

Looping / Iterative Statements

- While Loop
- Do-While Loop
- For Loop

Jumping Statements

- Break Statements
- Continue Statements
- Goto Statements

Decision Statements

- If - Statement
- If-else Statement
- Switch-Case Statements
- Conditional Statements

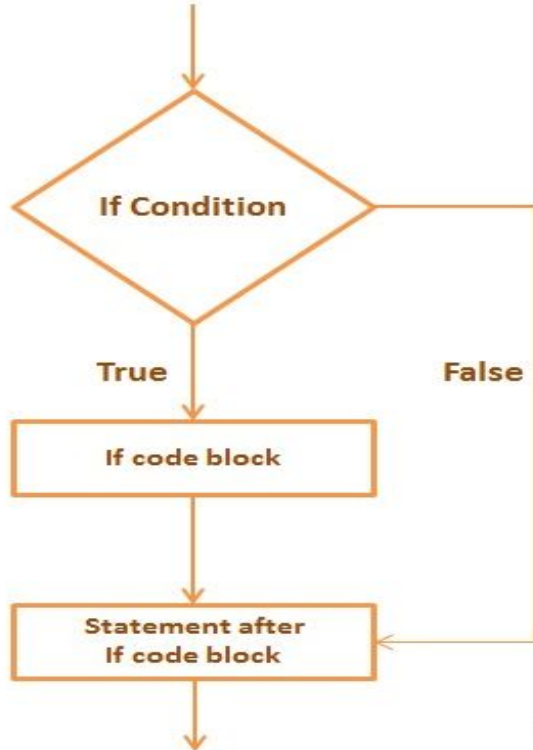
If- Statements

- It is used to control the flow of execution of the statements and also to test logically whether the condition is true or false.
- **Syntax**

```
if(condition)
{
    Block of Statements;
}
```

If- Statements

If Statement Flow Diagram



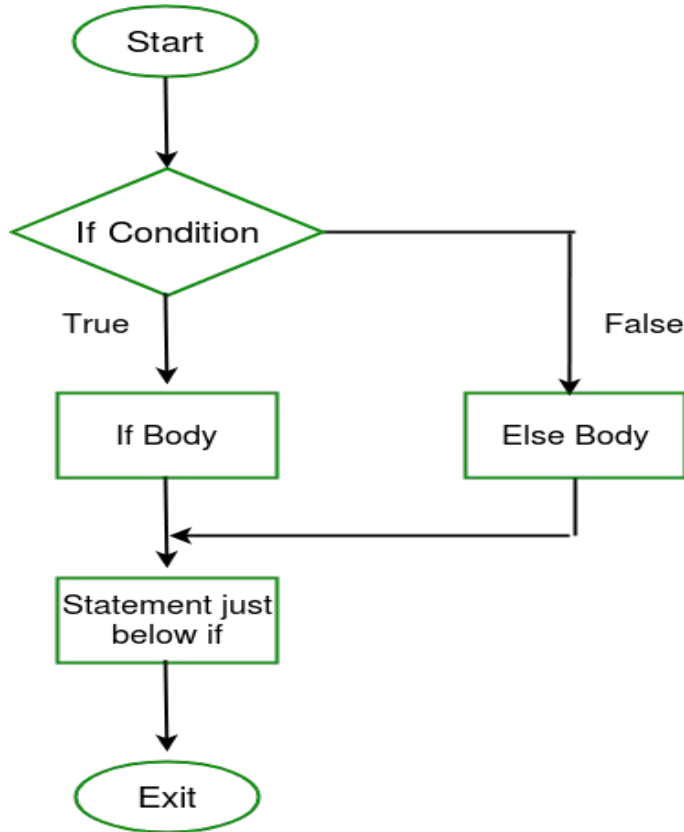
If condition is TRUE then block of statements following “if” will be executed and if Condition is false, it skip block of statements.

If-else Statements

- If-else is an extension of “if” Statement
- **Syntax**

```
if(condition)
{
    Block of If-Statements;
}
else
{
    Block of Else-Statements;
}
```

If- else Statements



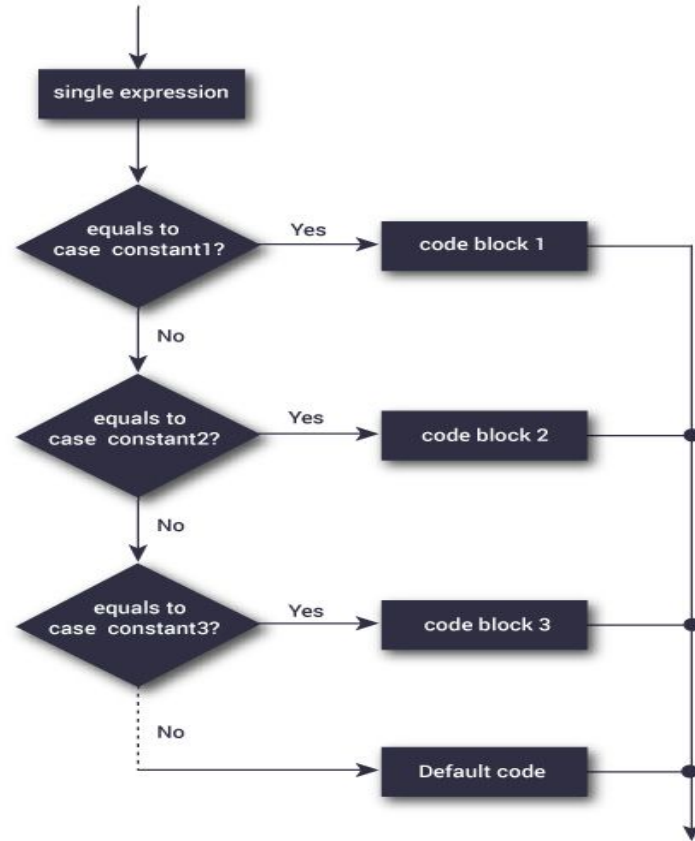
If condition is TRUE then block of if-statements will be executed, if condition is FALSE, block of else statements will be executed.

Switch-Case Statements

- The switch statement allows us to execute one code block among many alternatives.
- You can do the same thing with the if...else..if ladder. However, the syntax of the switch statement is much easier to read and write.

Switch-Case Statements

```
switch (expression)
{
    case constant1:
        // statements
        break;
    case constant2:
        // statements
        break;
    .
    .
    .
    default:
        // default statements
}
```



Conditional Operator

```
void main()  
{  
    int a=10,b=20;  
    a>b?printf("a>b"):printf("a<b");  
}
```

Conditional Operator

```
void main()
{
    int a=10,b=20;
    a>b?printf("a>b"):printf("a<b");
}
```

```
void main()
{
    int a=10,b=20,big;
    big=a>b ? a : b;
    printf("Big is %d", big);
}
```