A

PRESENTATION ON CONDITIONAL STATEMENT IN "C" LANGUAGE

Shaina Arora
Assistant Prof. (CS) Dept. AIET

C0NTENT

1. INTRODUCTION
2. IF CONDITION
3. IF ELSE CONDITION
4. NESTED IF ELSE CONDITION
5. LADDER ELSE IF CONDITION
6. SWITCH CASE CONDITION
7. BREAK STATEMENT
8. GOTO LABEL STATEMENT
9. CONTINUE STATEMENT

INTRODUCTION

Conditional Statements Are Used To Execute A Set Of Statements On Some Conditions. It Provides A Unit Of Block In Which We Can Either One Statement Or More Than One Statments. If The Given Condition Is True Then The Set Of Statements Are Executed Otherwise Body Is Skipped.

IF CONDITION

It is conditional statement, which is used to execute a set of statement on some conditions. The condition must be of Boolean type expression.

An expression, which returns only two value either TRUE or FALSE, is known as Boolean type expression.

Syntax: - if
(condition)
{
…………………….
…………………….
}

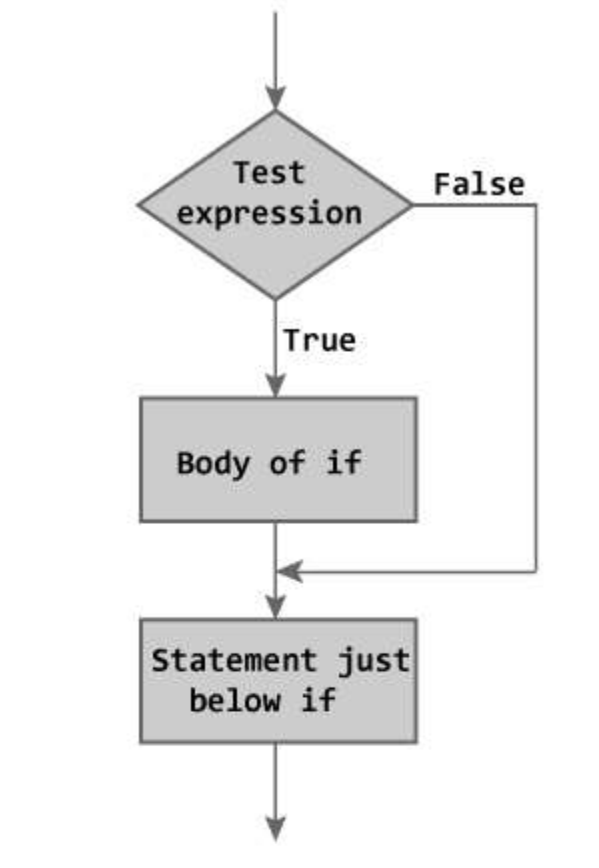


Figure: Flowchart of if Statement

In Another Words, It Can Also Be Said As Single Blocked Conditional Statements In Which Only One Part Is Mentioned. That Is Known As TRUE Part.

Example #1: C if statement
// Program to display a number if user enters negative number
// If user enters positive number, that number won't be displayed

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    // Test expression is true if number is less than 0
    if (number < 0)
    {
        printf("You entered %d.\n", number);
    }

    printf("The if statement is easy.");

    getch();
}
```

IF ELSE CONDITION

It Is Known As Double Blocked Conditional Statements .It Means, It Has TRUE Parts As Well As FALSE Part.

If The Given Condition Is True Then The True Part Is Executed Otherwise False Part Is Executed.

SYNTAX: -

IF (CONDITION)

{

....

....

}

ELSE

{

....

....

}

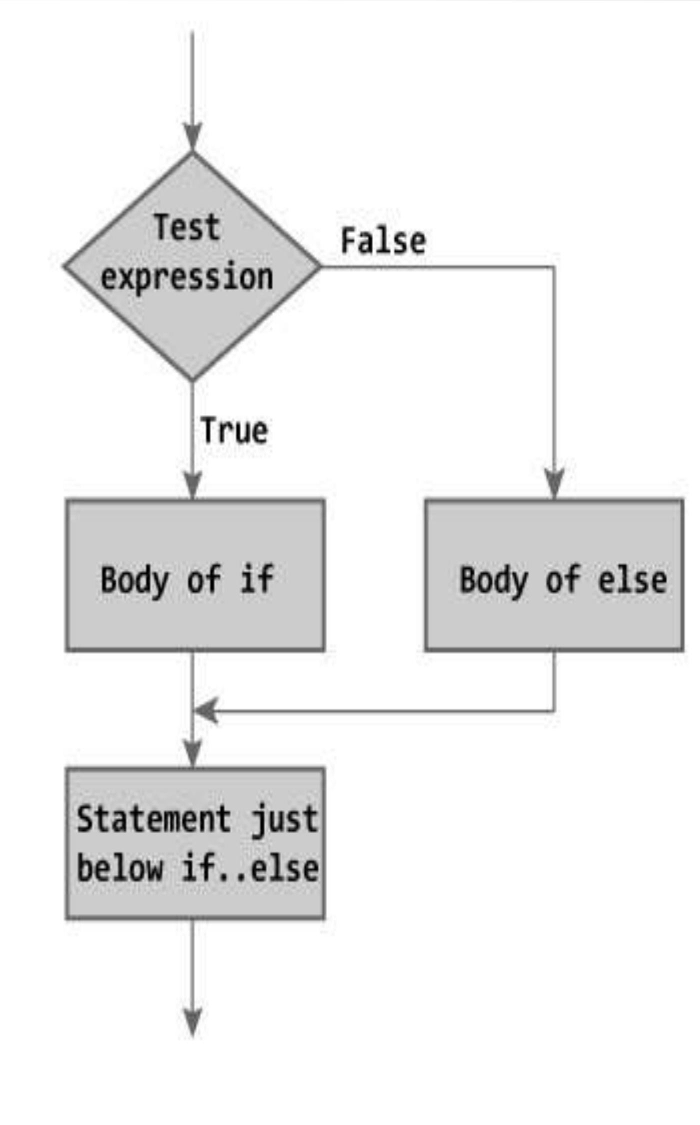


Figure: Flowchart of if...else Statement

EXAMPLE #2: C IF...ELSE STATEMENT
// PROGRAM TO CHECK WHETHER AN INTEGER ENTERED BY THE USER IS ODD OR EVEN

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int number;
    printf("Enter an integer: ");
    scanf("%d",&number);

    // True if remainder is 0
    if( number%2 == 0 )
        printf("%d is an even integer.",number);
    else
        printf("%d is an odd integer.",number);
    getch();
}
```

NESTED IF ELSE

Using Of One If Statement Within Another If Statement Is Known As Nested Of If else Statement.

syntax-

```
if (……….)
{
………….
………….
if (……….)
{
…………..
…………..
}
else
{
…………
…………
}
```

Q: -   WAP to input two numbers and print the greatest numbers.

```
#Include <stdio.h>
#Include<conio.h>
Void main()
{
Int a,b;
Clrscr();
 printf(“Enter Two
   Numbers”);

   scanf(“%d%d”,&a,&
   b);
If(a==b)
{
Printf(“a is equal to
```

```
else
If(a>b)
{
Printf(“a is greater”);
}
else
{
Printf(“b is greater”);
}
getch();
}
```

LADDER ELSE IF CONDITION

When We Want To Specify Condition With Else Part Then We Have To Use Ladder Of If Statement. In This Case If The Condition Specified With First If -Statement Is False, Then Control Goes To Second ―Else If‖ Statement. If It Is True Then Part (2) Is Executed Otherwise Control Goes To Third ―Else-if‖ Statement. If It Is True, Then Part (3) Is Executed Otherwise Part (4) Is Executed.

If The Condition Specified With First If Statement Is True, Then Part (1) Is Executed.

Syntax: -

```
if (……….)
{
…………..
…………..      1
}
else if (…….)
{
………….
………….       2
}
else if (………)
{
…………..
…………..          3
}
else
{
…………..
…………..           4
```

Q: - WAP to input three number and print the greatest number.

```
Include <stdio.h>
Include<conio.h>
Void main()
{
Int a,b,c;
Clrscr();
printf ("\n enter first number: -");
scanf("%d",&a);
printf("\n enter secend number : -");
scanf("%d",&b);
printf("\n enter third number : -");
scanf("%d",&c);
if(a>b)
{
if(a>c)
{
printf("\n a is greatest: -");
}
else
{
printf("\n c is greatest: -");
}
}
else
if(b>c)
{
printf("\n b is greatest: -");
}
else
{
printf("\n c is greatest: -");
}
getch();
}
```

SWITCH CASE CONDITION

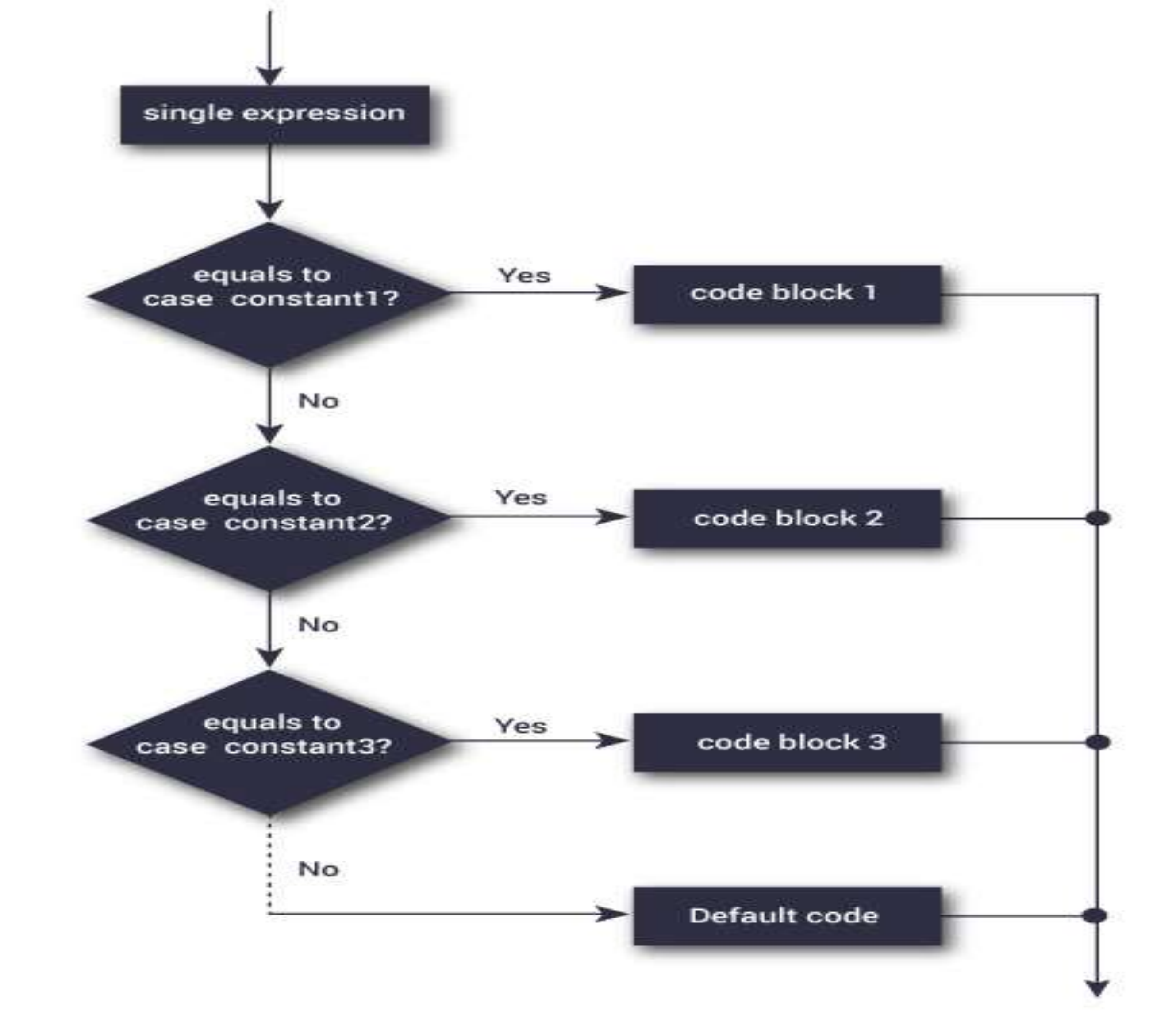
It Is Multiple Conditioned Checking Statements, Which Is Generally Used For Menu- Driven Program Where We Have To Select One Option Out Of Several Options At A Time.

The number of ―case within switch – statement is same as the number of options present in menu. Each ―case is used to do only one work at a time.

Default section: -

It is optional section, which is generally used for error handling. It means by using this section we can displays error messages. In case of wrong choice entry or out of range choice. It is automatically executed when any user enters wrong choice.

SWITCH CASE FLOW CHART



Q: - WAP to accept day no. (1 to 7) and print the day name

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("enter day number\n****************");
scanf("%d",&a);
switch(a)
{
case 1:
printf("monday");
break;
case 2:
printf("tuesday");
break;
case 3:
printf("wednesday");
break;
case 4:
printf("thirsday");
break;
case 5:
printf("friday");
break;
case 6:
printf("saturday");
break;
case 7:
printf("sunday");
break;
default:
printf("day not vailed\n**************");
}
getch();
}
```

GOTO LABELCONDITION

This statement is used to send the control from one position to any desired position within a program.

Since program is executed from top to bottom statement by statement. So if we want to execute certain part of program directly, then we can use this statement to do the same work.

Label: -

May Be Any User Defined Name Or Identifiers. It Is Just Like A Variable But Not A Variable. It Should Not Be Declared As Variables By Any Data Type.

Label Must Be Indicated At That Place Where We Want To Send The Control And It Must Be Followed By Colon (:) Sign.

When It Is Directly Used In A Program Then It Just Works As Infinite Loop. So It Must Be Used On Some Condition.

Syn: -

```
goto label;
```

Ex: -

```
void main ()
{
ram:
-----
------
-------
---------
goto ram;
----------
----------
---------
}
```

THANK YOU

ANY QUERY ???