Lab 05: Conditional Statements in C Language

*Familiarization with else- if statement, To understand in depth the concept of using decisions in C instructions, To understand the concept of performing actions using decisions, To manipulate data using conditions.*

# The ELSE IF statement

Another use of else is when there are multiple conditional statements that may all evaluate to true, yet you want only one if statement's body to execute. You can use an "else if" statement following an if statement and its body; that way, if the first statement is true, the "else if" will be ignored, but if the if statement is false, it will then check the condition for the else if statement. If the if statement was true the else statement will not be checked. It is possible to use numerous else if statements to ensure that only one block of code is executed.

Syntax for each C decision control statements are given in below table with description.

|  |  |  |
| --- | --- | --- |
| **Decision control statements** | **Syntax** | **Description** |
| **if** | if (condition)  { Statements; } | In these type of statements, if condition is true, then respective block of code is executed. |
| **if…else** | if (condition)  { Statement1; Statement2;} else  { Statement3; Statement4; } | In these types of statements, group of statements are executed when condition is true. If condition is false, then else part statements are executed. |
| **nested if** | if  (condition1){ Statement1; } else if (condition2)  { Statement2; } else Statement 3; | If condition 1 is false, then condition 2 is checked and statements are executed if it is true. If condition 2 also gets failure, then else part is executed. |

Table 1 - Syntax and description of Decision control statements

**5.1.2 Examples to compare the above mentioned Decision Control Statements:**

#### Example 5.1 for if statement in C:

In “if” control statement, respective block of code is executed when condition is true. int main()

{

int m=40,n=40; if (m == n)

{

printf("m and n are equal");

}

}

#### Output:

**m and n are equal**

***Example 5.2 for if else statement in C:***

In C if else control statement, group of statements are executed when condition is true. If condition is false, then else part statements are executed.

#include <stdio.h> int main()

{

int m=40,n=20; if (m == n) {

printf("m and n are equal");

}

else {

printf("m and n are not equal");

}

}

#### Output:

**m and n are not equal**

***Example 5.3 for nested if statement in C:***

* In “nested if” control statement, if condition 1 is false, then condition 2 is checked and statements are executed if it is true.
* If condition 2 also gets failure, then else part is executed. #include <stdio.h>

int main()

{

Student Name: Roll No: int m=40,n=20;

if (m>n) {

printf("m is greater than n");

}

else if(m<n) {

printf("m is less than n");

}

else {

printf("m is equal to n");

}

}

#### Output:

**m is greater than n**

Section:

**Example 5.4:**

void main (void)

{

int age;

printf("Please enter your age"); scanf ("%d", &age);

if(age<100)

{

}

printf("You are pretty young!\n");

else if (age==100)

{

}

else

{

}

getch();

}

printf("You are old\n");

printf("You are really old\n");

Student Name: Roll No:

# Conditions using boolean operators

Section:

Boolean operators allow you to create more complex conditional statements. For example, if you wish to check if a variable is both greater than five and less than ten, you could use the boolean AND to ensure both var > 5 and var < 10 are true. In the following discussion of boolean operators, Boolean operators will be capitalized in order to distinguish them from normal English. The actual C operators of equivalent function are described later - the C symbols are not: OR, AND, NOT, although they are of equivalent function.

When using if statements, you will often wish to check multiple different conditions. You must understand the Boolean operators OR, NOT, and AND. The boolean operators function in a similar way to the comparison operators: each returns 0 if evaluates to FALSE or 1 if it evaluates to TRUE.

## NOT

The NOT operator accepts one input. If that input is TRUE, it returns FALSE, and if that input is FALSE, it returns TRUE. For example, NOT (1) evaluates to 0, and NOT (0) evaluates to 1. NOT (any number but zero) evaluates to 0. In C NOT is written as !. NOT is evaluated prior to both AND and OR.

## AND

This is another important command. AND returns TRUE if both inputs are TRUE (if 'this' AND 'that' are true). (1) AND (0) would evaluate to zero because one of the inputs is false (both must be TRUE for it to evaluate to TRUE). (1) AND (1) evaluates to 1. (any number but 0) AND (0) evaluates to 0. The AND operator is written && in C. Do not be confused by thinking it checks equality between numbers: it does not. Keep in mind that the AND operator is evaluated before the OR operator.

## OR

Very useful is the OR statement! If either (or both) of the two values it checks are TRUE then it returns TRUE. For example, (1) OR (0) evaluates to 1. (0) OR (0) evaluates to 0. The OR is written as

|| in C. Those are the pipe characters. On your keyboard, they may look like a stretched colon. On my computer the pipe shares its key with \. Keep in mind that OR will be evaluated after AND.

It is possible to combine several boolean operators in a single statement; often you will find doing so to be of great value when creating complex expressions for if statements. What is !(1 && 0)? Of course, it would be TRUE. It is true is because 1 && 0 evaluates to 0 and !0 evaluates to TRUE (ie, 1).

Student Name: Roll No: Section:

**Student Exercises and Tasks:**

### Task 1:

A company insures its drivers in the following cases

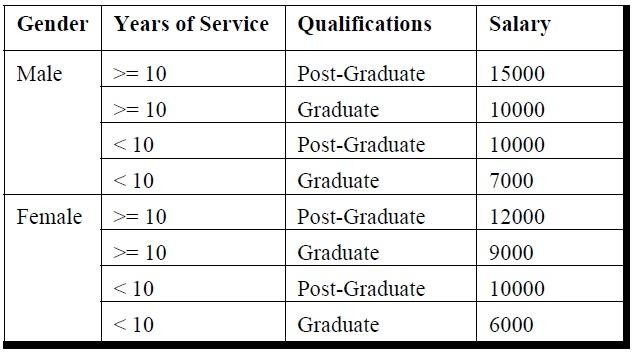
―If driver is married

―If the driver is unmarried, male & above 30 years of age

―If the driver is unmarried, female & above 25 years of age

In all other cases the driver is not insured. If the marital status, gender and age of the driver are entered through the keyboard, write a program to determine whether the driver is to be insured or not.

**Task 2:**

Write a program to calculate the salary as per the following table:

For Graduate enter ‘G’ and for Post-Graduate accept ‘P’.

**Task 3:**

Any year is input through the keyboard. Write a program to determine whether the year is a leap year or not.

**Task 4:**

If the ages of three students are input through the keyboard, write a program to determine the youngest of the three.

**Task 5:**

Write a program to check whether a triangle is valid or not, when the three angles of the triangle are entered through the keyboard. A triangle is valid if the sum of all the three angles is equal to 180 degrees.

Student Name: Roll No:

### Task 6:

A certain grade of steel is graded according to the following conditions:

* + - 1. Hardness must be greater than 50
      2. Carbon content must be less than 0.7
      3. Tensile strength must be greater than 5600

The grades are as follows:

Grade is 10 if all three conditions are met. Grade is 9 if conditions (i) and (ii) are met. Grade is 8 if conditions (ii) and (iii) are met. Grade is 7 if conditions (i) and (iii) are met. Grade is 6 if only one condition is met.

Grade is 5 if none of the conditions are met.

Section:

Write a program, which will require the user to give values of hardness, carbon content and tensile strength of the steel under consideration and output the grade of the steel.