PROGRAMMING FUNDAMENTALS

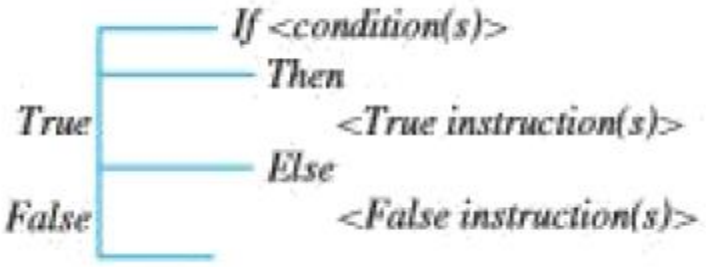
LAB 04

CONDITIONAL STATEMENTS IN C

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

IF-ELSE STRUCTURE

The decision logic structure uses the If/Then/Else instruction. It tells the computer that if a condition is true, then execute a set of instructions, or Else execute another set of instructions. The Else part is optional, as there is not always a set of instructions if the conditions are false.



The True instructions are processed when the resultant of the condition is True, and the False instructions are processed when the resultant of the condition is False. A condition can be one of four things:

- A logical expression, that is, an expression that uses logical operators (AND, OR, and NOT);
- An expression using relational operators (greater than, less than, greater than or equal to, less than or equal to, equal to, and not equal to);
- A variable of the logical data type (True or False);
- A combination of logical, relational, and mathematical operators.

Some examples of conditional expressions are as follows:

- A < B (A and B are the same data type—either numeric, character, or string)
- X + 5 (X and Z are numeric data)
- E < 5 or F > 10 (E and F are numeric data)
- DataOk (DataOk is a logical datum)

IF-ELSE-IF STRUCTURE

```
if (expression1)
        statement 1;
else if (expression 2)
        statement 2;
        ...
else
        statement n;
```

Multiple conditions can be written by making several else-if clauses. Once a condition is true, control will never go to other else-if conditions. An else clause can be added after else if statements.

EXAMPLE (IF-ELSE STRUCTURE)

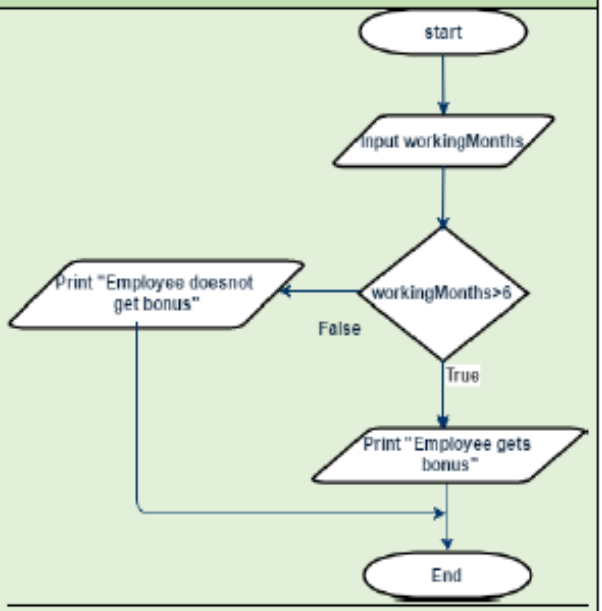
PROBLEM

A company gives a bonus at the end of each fiscal year. For an employee to get a bonus, the employee has been working at the company for more than six months.

ALGORITHM

1) Start
2) Input working months
3) IF workingMonths > 6
   then
      Print "Employee gets bonus"
   Else
      Print "Employee doesn't get bonus"
   END IF
4) End

FLOWCHART



C-IMPLEMENTATION

```
#include<stdio.h>

int main()
{
        int workingMonths;
        printf("Enter number of working months: ");
        scanf("%d", &workingMonths);

        if(workingMonths > 6)
                {
                        printf("\nEmployee gets bonus.");
                }
                else
                {
                        printf("\nEmployee doesnot get bonus.");
                }
}
```

OUTPUT

```
Enter number of working months: 5

Employee doesnot get bonus.
--------------------------------
```

EXAMPLE (IF-ELSE-IF STRUCTURE)

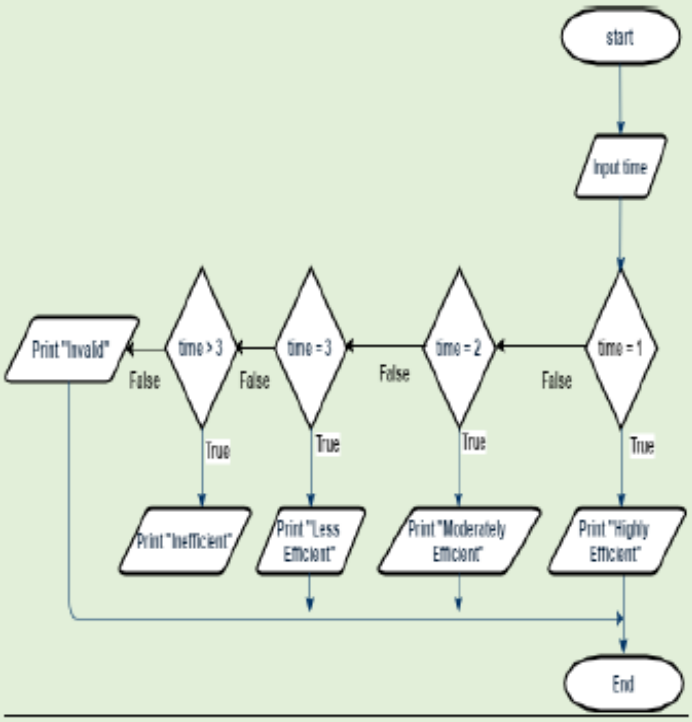
PROBLEM

In a company, worker efficiency is determined on the basis of the time required for a worker to complete a particular job. If the time taken by the worker is 1 hour, then the worker is said to be highly efficient. If the time required by the worker is 2 hours, then the worker is said to be moderately efficient. If the time taken is 3 hours, the worker is said to be less efficient and if the time taken by the worker is more than 3 hours, then the worker is said to be inefficient. Find the efficiency of the worker.

ALGORITHM

1) Start
2) Input time
3) IF time = 1
   then
   Print "Highly Efficient."
   Else
   If time = 2
   Then
   Print "Moderately Efficient."
   Else
   If time = 3
   Then
   Print "Less Efficient."
   Else
   If time > 3
   Then
   Print "Inefficient."
   else
   Print "Invalid Input"
   END IF
4) End

FLOWCHART



C-IMPLEMENTATION

```
#include<stdio.h>

int main()
{
        int time;
        printf("Enter the time(hours) taken by the worker to complete a particular job.");
        scanf("%d",&time);
```

```
        if(time == 1)
                {
                        printf("Worker is highly efficient.");
                }

                else if(time == 2)
                        {
                                printf("Worker is moderately efficient.");
                        }

                        else if(time == 3)
                        {
                                printf("Worker is less efficient.");
                        }

                        else if(time > 3)
                        {
                                printf("Inefficient.");
                        }

                        else
                        {
                                printf("Invalid Input.");
                        }
        return 0;
}
```

OUTPUT

```
Enter the time(hours) taken by the worker to complete a particular job.6
Worker is not efficient.
--------------------------------
Process exited after 9.083 seconds with return value 0
Press any key to continue . . .
```

SWITCH STATEMENT

A series of decisions in which a variable or expression is tested separately for each of the constant integral values it may assume, and different actions are taken. This is called multiple selection. C provides the switch multiple-selection statement to handle such decision making.

The switch statement consists of a series of case labels, an optional default case and statements to execute for each case.

The switch statement is especially useful when the selection is based on the value of a single variable or of a simple expression (called the controlling expression). The value of this expression may be of type int or char , but not of type double.

C- SYNTAX

```
switch ( controlling expression ) {

label set 1 :
        statements1 ;
        break;

label set 2 :
        statements 2 ;
        break;
        .
        .
        .
label set n :
        statements n ;
        break;

default:
        statements d ;
}
```

INTERPRETATION:

- The controlling expression, an expression with a value of type int or type char, is evaluated and compared to each of the case labels in the label sets until a match is found. A label set is made of one or more labels of the form case followed by a constant value and a colon.
- When a match between the value of the controlling expression and a case label value is found, the statements following the case label are executed until a break statement is encountered. Then the rest of the switch statement is skipped.
- The statements following a case label may be one or more C statements, so it is not need to make multiple statements into a single compound statement using braces.
- If no case label value matches the controlling expression, the entire switch statement body is skipped unless it contains a default label. If so, the statements following the default label are executed when no other case label value matches the controlling expression.

Switch Case Statement As A Substitute For Long If Statements:

EXAMPLE USING IF - ELSE	EXAMPLE USING SWITCH

#include <stdio.h>
 	
int main ()
{
char grade = 'E';
 	
if(grade == 'A')
printf("Superb!\n");
 	
else if(grade == 'B')
printf("Very good!\n");
 	
else if(grade == 'C')
printf("Good.\n");
 	
else if(grade == 'D')
printf("Passed.\n");
 	
else if(grade == 'F')
printf("Try again.\n");
 	
else
printf("Invalid grade.\n");
 	
printf("Your grade is %c.\n", grade);
 	
return 0;
}
	#include <stdio.h>
	
	int main ()
	{
	char grade = 'F';
	
	switch(grade) {
	case 'A' :
	printf("Superb!\n");
	break;
	case 'B' :
	printf("Very good!\n");
	break;
	case 'C' :
	printf("Good\n");
	break;
	case 'D' :
	printf("Passed\n");
	break;
	case 'F' :
	printf("Try again\n");
	break;
	default :
	printf("Invalid grade\n");
	}
	
	printf("Your grade is %c.\n", grade);
	
	return 0;
	}

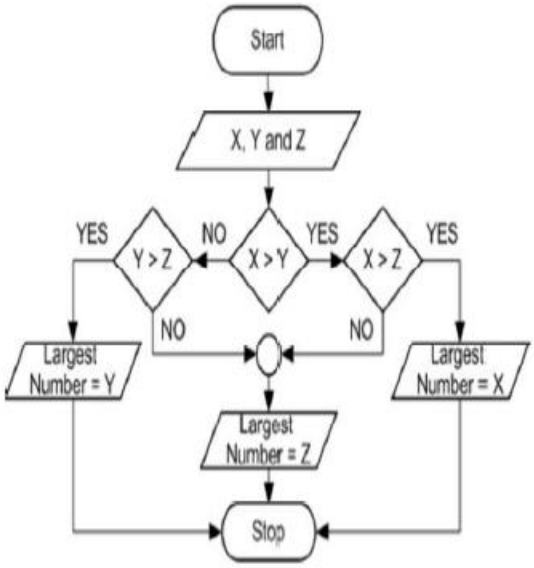
Nested If-else Statement

Placing the block of if else statement inside an existing if or else block statement is called nested If else statement. Each block of nested if else, logically perform same as simple if else statements. Whenever a user wants to check more than one condition at a time, the appropriate way is to use nested if statements. Following is the structure of nested if else statement.

```
IF (logical-expression) THEN
    statements
    IF (logical-expression) THEN
        statements
    ELSE
        statements
    END IF
    statements
ELSE
    statements
    IF (logical-expression) THEN
        statements
    END IF
    statements
END IF
```

Example Nested If-else Statement

Algorithm	Flowchart

Step 1: Input X,Y,Z
Step 2: if(X>Y) then
If(X>Z) then
Max= X [X>Y, X>Z]
Else
Max= Z [Z>X>Y]
Endif
Else
If(Y>Z) then
Max = Y [Y>X, Y>Z]
Else
Max = Z [Z>Y>X]
Endif
Endif
Step 3: Print "The largest number is ",Max
	

C-Implementation

```
#include<stdio.h>
main(){
	intx,y,z;
	printf("Enter value of X");
	scanf("%d",&x);
	printf("Enter value of Y");
	scanf("%d",&y);
	printf("Enter value of Z");
	scanf("%d",&z);
	if(x>y){
		if(x>z){
			printf("The largest value is of x = %d",x);
		}
		else{
			printf("The largest value is of z =%d", z);
		}
	}
	else{
		if(y>z){
			printf("The largest value is of y= %d",y);
		}
		else{
			printf("The largest value is of z= %d",z);
		}
	}
}
```

Output

```
Enter value of X67
Enter value of Y45
Enter value of Z89
The largest value is of z =89
```