Unit 3

Conditional Program Execution / Decision Making

3 Introduction

We have seen that a C program is a set of statements which are normally executed sequentially in the order in which they appear. However, in practice, we have a number of situations where we may have to change the order of execution of the statements based on certain conditions or repeat a group of statements until certain specified conditions are met. This involves a kind of decision making to see whether a particular condition has occurred or not and then direct the computer to execute certain statements accordingly.

C language possesses such decision-making capabilities by supporting the following statements:

- 1.1 IF statement
- **1.2 IF-ELSE statement**
- **1.3 Nesting of IF-ELSE statements**
- **1.4 ELSE-IF ladder**
- **1.5 SWITCH statement**
- **1.6 GOTO statement**

3.1 IF Statement:
Theif statement:

Atement and is used to control the flow of Theif statement is a powerful decision making execution of statements. It is lasic in various way decision statement and is used in conjunction with an expression. It takes the following form: if (test expression

{ statement block 1; } statement x;

The statement block 1 may be a single statement or a group of statements. If the test expression is true, the statement block 1 will be executed; otherwise the statement block 1 will be skipped and the control will jump to the statement-x. When the condition is true both the statement block 1 and the statement x are executed in sequence.

Program

```
// Program to classify a number as positive, negative or zero.
#include<stdio.h>
main()
{
    int num;
    printf("\n Enter any number : ");
    scanf("%d", &num);
    if(num==0)
        printf("\n The value is equal to zero");
    else if(num>0)
        printf("\n The number is positive");
    else
        printf("\n The number is negative");
    return 0;
}
```

3.5 SWITCH Statement:

We have seen that when one of the many alternatives is to be selected we can use and statement to control the selection. However the complexity of such a program increases dramatically when the number of alternatives increases. The program becomes difficult to read and follow. At times it may confuse even the person who designed it. Fortunately, C has a built in multi-way decision statement known as switch. The switch statement tests the value of a given variable (or expression) against the list of ease values and when a match is found a block of statement associated with that case in exercise.

```
case value-1:
block-1
break;
case value-2:
block-2
break;
```

```
default:
```

default-block

```
}
```

{

statement-x;

3.19 Function Call

- The function call statement invokes the function.
- When a function is invoked the compiler jumps to the called function to execute the statements that are a part of that function.
- Once the called function is executed, the program control passes back to the calling function.
- Function call statement has the following syntax.

function_name(variable1, variable2, ...);

Program

```
#include<stdio.h>
int sum(int a, int b); // FUNCTION DECLARATION
int main()
{
      int num1, num2, total = 0;
      printf("\n Enter the first number : ");
                         i", total);
// FUNCTION CALL
Notesale.co.uk
      scanf("%d", &num1);
      printf("\n Enter the second number : ");
      scanf("%d", &num2);
      total = sum(num1, num2);
      printf(``\n Total = %d", total);
      return 0;
                                NCTION HEADEROF 2
}
// FUNCTION DEFNITION
int sum (int a, int b)
}
```

3.20 Return statement

- The return statement is used to terminate the execution of a function and return control to the calling function. When the return statement is encountered, the program execution resumes in the calling function at the point immediately following the function call.
- By default, the return type of a function is int.
- For functions that has no return statement, the control automatically returns to the calling function after the last statement of the called function is executed.

3.21 Passing parameters to the function

- There are two ways in which arguments or parameters can be passed to the called function.
- Call by value in which values of the variables are passed by the calling function to the called function.
- Call by reference in which address of the variables are passed by the calling function to the called function.

- 11. What are iterative control statements? Differentiate between while loop and do-while loop.
- 12. Define recursion. Give its advantage. Which data structure is used to implement recursion? Write a program to calculate factorial of a number using recursion.
- 13. Write a program to check whether a number is perfect number or not. If the sum of factor is equal to number itself then it is a perfect number. E.g Factor of 6 are 1, 2, 3 whose sum 1+2+3=6.
- 14. Write a program to find the prime numbers between the given range.

2011-12

- 1. Write a program to generate following pattern.
 - Α

ΒA

- ABA
- BABA
- ABABA
- 2. Write a program o read five digit number if it is even then add the digits otherwise multiply them.
- 3. Write a program to generate the given series upto less than 200. $1 - 4 + 9 - 16 + 25 \dots$
- 4. Write a program to read age of 100 persons and count the number of persons in the age group 5. Write a program to check whether a number is even or odd aither else option. **Note** 27 of 27 **Preview page** 27 of 27