CHAPTER 5

DECISION MAKING AND BRANCHING

Dr.A.PRAVEEN, M.Sc.,M.Phil.,Ph.D.

C language supports the following control or decision making statements.

1. if statement
2. switch statement
3. conditional operator statement
4. goto statement.

Decision making with if statement

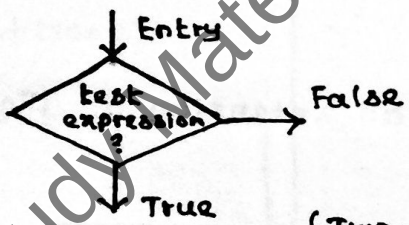
The if statement is a powerful decision making statement and is used to control the flow of execution of statements.

It is a two-way decision statement.

The general form of the if statement is:

```
if(test expression)
```

The computer evaluate the expression first and then, depending on whether the value of the expression (relation or condition) is 'true' (non-zero) or 'false' (zero), it transfers the control to a particular statement.



(Two-way branching)

The different forms of the if statement are:

1. Simple if statement
2. if... else statement
3. Nested if... else statement
4. else if ladder.

SIMPLE IF STATEMENT

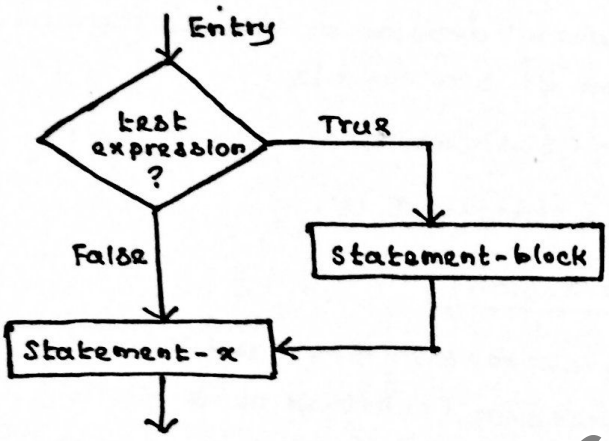
The general form of a simple if statement is

```
if(test expression)
  {
    statement-block;
  }
statement-x;
```

If the test expression is true, the statement-block will be executed; Otherwise the statement-block will be skipped and the execution will jump to the statement-x.

Note: ① The statement-block may be a single statement (or) a group of statements.

② When the condition is true both the statement-block and the statement-x are executed in sequence.



(Flowchart of simple if control)

The IF..ELSE Statement

The if...else statement is an extension of the simple if statement.

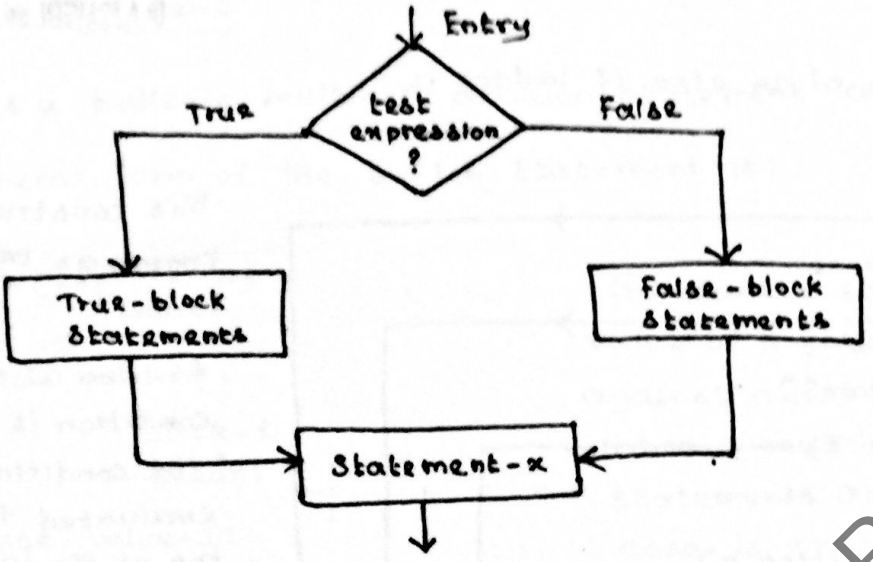
The general form is:

```
if (test expression)
  {
    True-block statement(s)
  }
else
  {
    False-block statement(s)
  }
Statement-x
```

If the test expression is true, then the true-block statement(s), immediately following the if statement are executed; Otherwise, the false-block statement(s) are executed.

Note: In either case, either the true-block or the false-block will be executed, not both. In both the cases, the control is transferred subsequently to statement-x.

Dr.A.PRAVEEN.,M.Sc.,M.Phil.,Ph.D.

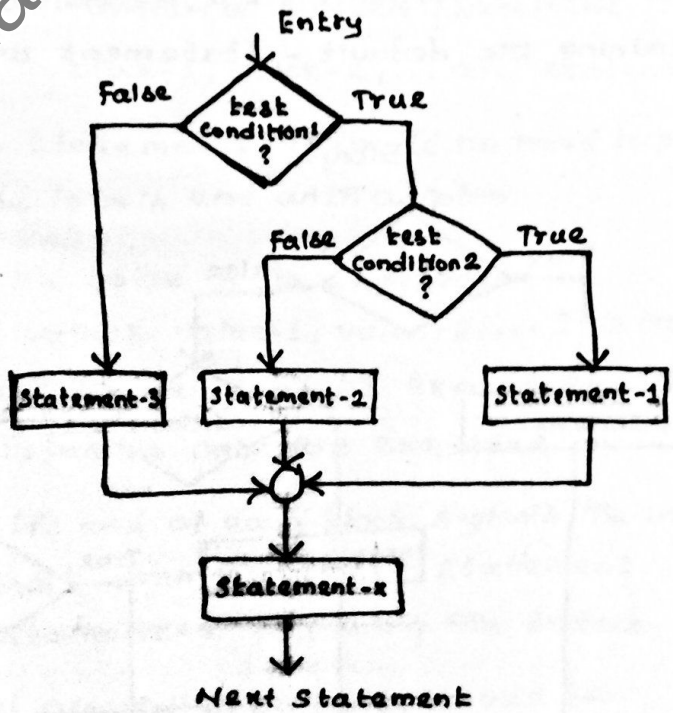


(flowchart of if... else control)

Nesting Of if...else Statements:

When a series of decisions are involved, we may have to use more than one if...else statement in nested form.

```
if (test condition 1)
  {
  if (test condition 2)
    {
      statement-1;
    }
  else
    {
      statement-2;
    }
  }
else
  {
    statement-3;
  }
statement-x;
```



(flowchart of nested if...else statements)

If the condition-1 is false, the statement-3 will be executed; otherwise it continues to perform the second test. If the condition-2 is true, the statement-1 will be evaluated; otherwise statement-2 will be evaluated, and then the control is transferred to the statement-x.

The ELSE IF ladder

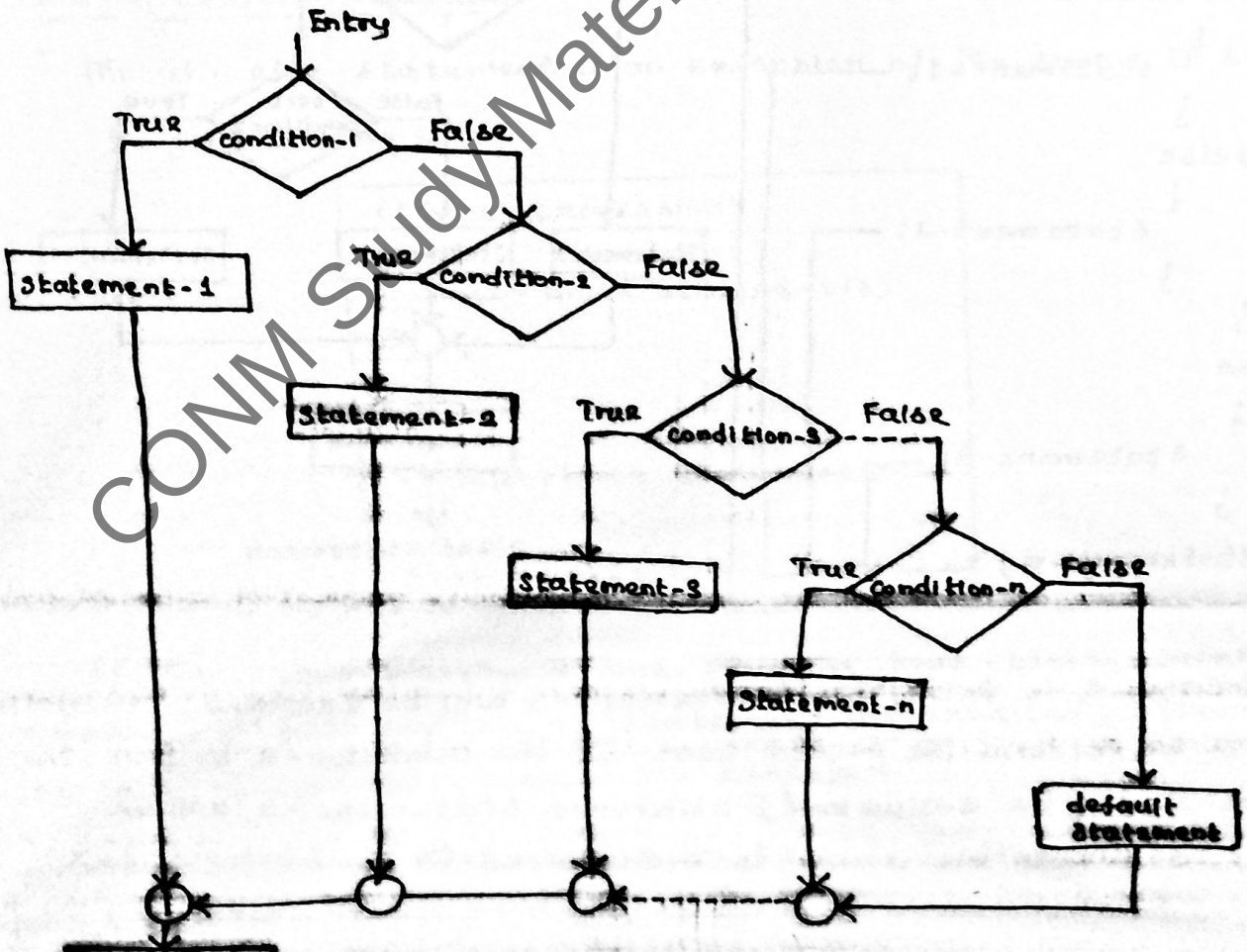
Dr. A. PRAVEEN M.Sc., M.Phil., Ph.D.

The general form of an else if ladder is :

```
if (condition 1)
   statement-1;
else if (condition 2)
      statement-2;
   else if (condition 3)
         statement-3;
      ::::::
      else if (condition n)
            statement-n;
         else
            default-statement;
Statement-x;
```

This construct is known as the else if ladder.

As soon as a true condition is found, (the conditions are evaluated from the top of the ladder downwards) the statement associated with it is executed and the control is transferred to the Statement-x. When all the n conditions become false, then the final else containing the default-statement will be executed.



(flow chart of the else if ladder)

The SWITCH Statement

Dr.A.PRAVEEN KUMAR

C has a built-in multiway decision statement known as a switch.

The general form of the switch statement is:

```
switch (expression)
{
   case value-1 :
            block-1
            break;
   case value-2:
            block-2
            break;
    :::::
   default:
            default-block
            break;
}
statement-x;
```

The switch statement tests the value of a given variable (or expression) against a list of case values and when a match is found, a block of statements associated with that case is executed.

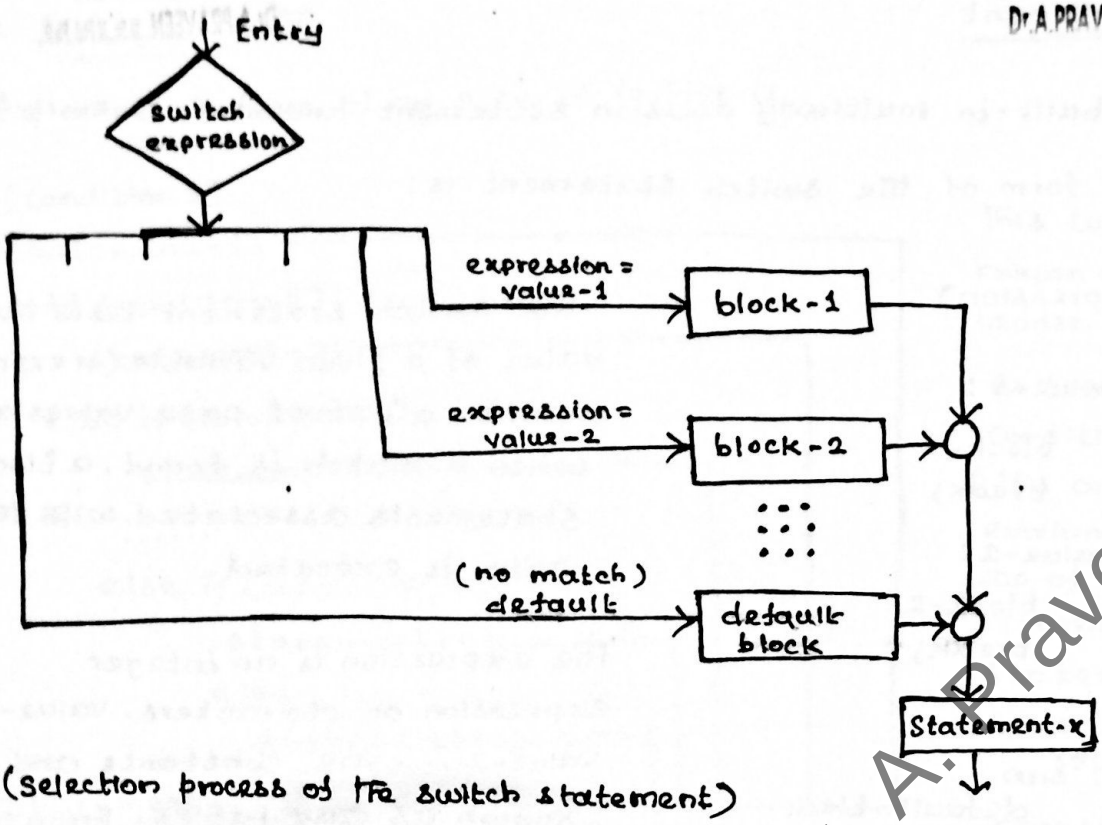
The expression is an integer expression or characters. value-1, value-2,... are constants and are known as case labels. Each of these values must be unique within a switch statement.

block-1, block-2,... are statement lists and may contain zero or more statements. There is no need to put braces around these blocks. The case labels end with a colon.

When the switch is executed, the value of the expression is successively compared against the values value-1, value-2,... If a case is found whose value matches with the value of the expression, then the block of statements that follows the case are executed.

The break statement at the end of each block signals the end of a particular case and causes an exit from the switch statement, transferring the control to the statement-x following the switch.

The default is an optional case. When present, it will be executed if the value of the expression does not match with any of the case values. If not present, no action takes place and the control goes to the statement-x, if all matches fail.



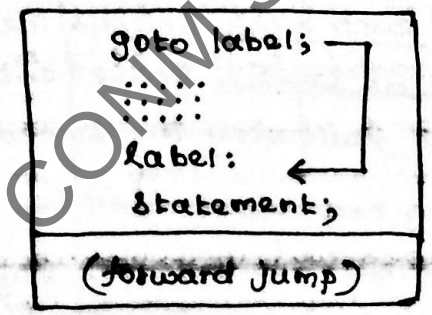
(Selection process of the switch statement)

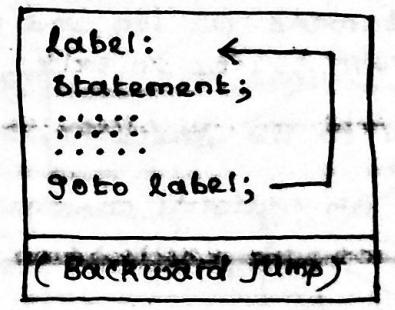
The GOTO statement

C supports the goto statement to branch unconditionally from one point to another in the program.

The goto requires a label in order to identify the place, where the control is to be transferred. A label is any valid variable name and must be followed by a colon. The label is placed immediately before the statement where the control is to be transferred.

The general forms of goto and label statements are shown below:





The label: can be anywhere in the program either before or after the goto label; statement.

If the label: is before the statement goto label; a loop will be formed and some statements will be executed repeatedly. Such a jump is known as a backward jump. On the other hand, if the label: is placed after the goto label; some statements will be skipped and the jump is known as a forward jump.

- A goto is often used at the end of a program to direct the control to go to the input statement to read further data. The goto statement is used