UNIT-3: Decision Making statements :
3.1 if statements :
3.1.1 simple if statements
3.1.2 if…else statements
3.1.3 if…else if….else statements
3.1.4 Nested if statements.
3.2 Switch..case statements
3.2.1 Use of break and default
3.2.2 Difference between switch and if statements.

UNIT-3: Decision Making statements :
3.1 if statements :

3.1.1 simple if statements

- syntax:

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

- Test expression is true then statement block is execute otherwise statement block is skipped and execution will jump to statement-x.
- E.g.

```
//check given number is positive number.
#include <conio.h>
#include <stdio.h>

void main()
{
        int n;
        clrscr();
        printf("\n enter no:");
        scanf("%d",&n);
        if(n>0)
            printf("\n %d is positive number",n);
        getch();
}
```

Output:
Enter no: 5
5 is positive number

3.1.2 if…else statements

- syntax:

```
if (test expression)
{
```

```
   True-Statement-block;
}
Else
{
   False-Statement-block;
}
Statement-x;
```

- Test expression is true then true statement block is execute otherwise false statement block.
- E.g.

```
//check given number is even or odd number.
#include <conio.h>
#include <stdio.h>

void main()
{
        int n;
        clrscr();
        printf("\n enter no:");
        scanf("%d",&n);
        if(n%2==0)
                printf("\n %d is even number",n);
        else
                printf("\n %d is odd number",n);
        getch();
}
```

Output:
Enter no:5
5 is odd number

3.1.3 if…else if….else statements (**THE ELSE IF LADDER)**

- syntax:

```
if (test1)
   Statement-1;
Else if (test2)
   Statement-2;
Else if (test3)
   Statement-3;
.
Else if (testn)
   Statement-n;
Else
    Default statement;
Statement-x;
```

- This construct is known as else if ladder.
- The tests are evaluated from top to down.

- As soon as a true test is found the statement associated with it is execute. And control transfer to statement-x.
- When all the tests become false, then the final else containing the default statement will be executed.

- E.g// disply marksheet

```
#include <conio.h>
#include <stdio.h>

void main()
{
        int rn,m1,m2,m3,m4,m5,t,p;
        char name[20];
        clrscr();
        printf("\n Enter name,rn,m1,m2,m3,m4,m5");
        scanf("%s %d %d %d %d %d %d",name,&rn,&m1,&m2,&m3,&m4,&m5);
        t=m1+m2+m3+m4+m5;
        p=t/5;
        printf("\n Name is:%s",name);
        printf("\n Roll no:%d",rn);
        printf("\n Marks1:%d",m1);
        printf("\n Marks2:%d",m2);
        printf("\n Marks3:%d",m3);
        printf("\n Marks4:%d",m4);
        printf("\n Marks5:%d",m5);
        printf("\n Total is:%d",t);
        printf("\n Percentage is:%d",p);
        if (p>70)
                printf("\n Distinction");
        else if(p>60)
                printf("\n First class");
        else if(p>50)
                printf("\n Second class");
        else if(p>40)
                printf("\n Pass class");
        else
                printf("\n Fail");
        getch();
}
```

Output:

```
Enter name,rn,m1,m2,m3,m4,m5 ishvari 1 80 80 80 80 80
Name is: ishvari
Roll no:1
Marks1:80
Marks2:80
Marks3:80
Marks4:80
Marks5:80
Total is: 400
  Percentage is:80
```

Distinction

3.1.4 Nested if statements.

```
if (test expression1)
{
        if(test expression2)
          {
                Statement-1;
           }
       Else
          {
                Statement-2;
           }
 }
 Else
{
          if(test expression3)
         {
                Statement-3;
          }
       Else
          {
                Statement-4;
           }
 }

Statement-x;
```

- If within if is called as nested if.
- Test expression1 is true then it checks test expression2. If it is true then statement-1 is executing otherwise statement -2 is execute.
- If test expression1 is false then it checks test expression3. If it is true then statement-3 is executing otherwise statement -4 is execute. Then control transfer to statement-x.
- // find larger number from three number.

```
#include <conio.h>
#include <stdio.h>

void main()
{
       int a,b,c;
       clrscr();
       printf("\n enter a,b,c:");
       scanf("%d %d %d",&a,&b,&c);
       if(a>b)
                if(a>c)
                        printf("\n %d is larger number",a);
                else
                        printf("\n %d is larger number",c);
       else
```

```
                if(b>c)
                        printf("\n %d is larger number",b);
                else
                        printf("\n %d is larger number",c);

        getch();
    }
```

Output:
Enter a,b,c 12 50 5
50 is larger number

3.2 Switch..case statements

3.2.1 Use of break and default

syntax

```
Switch (expression)
{
Case value1:
    Block1;
    Break;
Case value2:
    Block2;
    Break;
.
.
.
Default:
    Default-block;
    Break;
}
Statement-x;
```

- The expression is an integer expression or character .value1, value2….. are constant or constant expression and are known as case labels. Each of these values should be unique within a switch statement. Block1 ,block2 ….are statement lists and may contain zero or more statements. There is no need to put braces around these blocks. Case labels end with a colon (:) .
- When the switch is executed, the values of the expression are successfully compared against the values value1, value2…. If a case is found whose value matches with the value of the expression, then the block of statements that follow the case are executed.
- The break statement at the end of each block indicate the end of a particular case and causes an exit from the switch statement then transferring control to statement-x
- The default is 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 if all matches fail and control goes to the statement-x.
- //switch case example with integer type choice

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int d;
  clrscr();
  printf("\n enter day in number:");
  scanf("%d",&d);
  switch(d)
  {
    case 1:
            printf("\n monday");
            break;
    case 2:
            printf("\n tuesday");
            break;
    case 3:
            printf("\n wednesday");
            break;
    case 4:
            printf("\n thusday");
            break;
    case 5:
            printf("\n friday");
            break;
    case 6:
            printf("\n saturday");
            break;
    case 7:
            printf("\n sunday");
            break;
    default:
          printf("\n invalid day number");
          break;
    }

  getch();
  }
```

Output:

Enter day in number 1

Monday

```
//switch example with character type choice
#include <stdio.h>
#include <conio.h>
void main()
{
  char c;
  clrscr();
  printf("\n enter alphabet:");
  scanf("%c",&c);
  switch(c)
  {
        case 'A':
              printf("\n alphabet is A");
              break;
        case 'B':
              printf("\n alphabet is B");
              break;
        case 'C':
              printf("\n alphabet is C");
              break;
        case 'D':
              printf("\n alphabet is D");
              break;
        default:
             printf("\n invalid input");
             break;
        }

  getch();
  }
```

Output:
Enter alphabet: A
Alphabet is A

3.2.2 Difference between switch and if statements.

IF	SWITCH
1. It is knowns as decision statement.	It is use as multi conditional control statement.
2. It checks conditional expression.	It check constant expression.
3. The expression is Boolean type.	The expression is int or char expression
4. It works for equal or not equal or less or greater value.	It works for equal value only.

5. Syntax
If(condition)
{
Statements;
}
	Syntax
	Switch(expression)
	{
	Case value1:

	Block1; Break; Case value 2: Block2; Break ; Default : Default block; }