

3. Nested if

- When an if statement inside another if block is called nested if.

Syntax

```
if (test - exp1)
{
    if (test - exp2)
    statement 1;
    else
    statement 2;
}
else
{
    body of else;
}
```

- Write a C++ program to check greater of 3 numbers.

Ans)

```
#include <iostream.h>
void main()
{
    int a, b, c;
    cout << "enter input";
    cin >> a >> b >> c;
    if (a > b)
    {
        if (a > c)
        {
            cout << "a is greater"
        }
        else
        {
            cout << "c is greater";
        }
    }
    else
    {
        if (b > c)
```

```
{
    cout << "b is greater";
}
else
{
    cout << "c is greater";
}
```

4. if else if Ladder

- * It is used when multiple branching is required.
- * Different conditions will be given and for each condition will decide which statement is to be selected for execution.

Syntax

```
if (test expression 1)
```

```
{
```

```
statement block 1 ;
```

```
}
```

```
else if (test expression 2)
```

```
{
```

```
statement block 2 ;
```

```
}
```

```
else if (test expression 3)
```

```
{
```

```
statement block 3 ;
```

```
}
```

```
else
```

```
{
```

```
body of else;
```

```
}
```

- * When the test expression 1 returns true the statement block 1 will get executed skipping all the else if condition and the statement block associated with it.
- * When the test expression 1 returns false control of the program moves to the test expression 2, when this condition returns true statement block 2 will be executed and when the condition returns false it will move to the test expression 3.

* When all of the test expression 1, test expression 2 and test expression 3 returns a false value body of else will be executed.

• Write a C++ program to print student grade according to the following table.

Scores	Grade
80 or more	A
From 60 to 79	B
From 40 to 59	C
From 30 to 39	D
Below 30	E

Ans

```
#include <iostream.h>
void main ( )
{
    int s;
    cout << "enter the score";
    cin >> s;
    if (s >= 80)
    {
        cout << "a grade";
    }
    else if (s >= 60)
    {
        cout << "b grade";
    }
    else if (s >= 40)
    {
        cout << "c grade";
    }
}
```

```
else if (s >= 40)
{
    cout << "c grade";
}
else if (s >= 30)
{
    cout << "d grade";
}
else
{
    cout << "e grade";
}
}
```

- Write a C++ program to check whether the given year is a leap year or not.

Ans) at print

```
#include <iostream.h>
void main ( )
{
    int y;
    cout << "enter the year";
    cin >> y;
    if (y % 100 == 0)
    {
        if (y % 400 == 0)
        {
            cout << "leap year";
        }
        else
        {
            cout << "not a leap year";
        }
    }
}
```

```
else if (y % 4 == 0)
{
    cout << "leap year";
}
else
{
    cout << "not leap year";
}
}
```

- Write a C++ program to display the name of the day for a given day number.

Ans)

```
#include <iostream.h>
void main ( )
{
    int d;
    cout << "enter the day number";
    cin >> d;
    if (d == 1)
    {
        cout << "sunday";
    }
    else if (d == 2)
```

```
{
    cout << "monday";
}
else if (d == 3)
{
    cout << "tuesday";
}
else if (d == 4)
{
    cout << "wednesday";
}
}
```

```

else if (d == 5)
{
cout << "thursday";
}
else if (d == 6)
{
cout << "friday";
}
else if (d == 7)
{
cout << "saturday";
}
else
{
cout << "invalid";
}
}

```

5. ★ Switch statement

- * Switch is a multiple branching statement.
- * Programs written using else if ladder can also be rewritten by using switch statement.

Syntax

```

switch (expression)
{
case constant - 1 :
statement block 1 ;
break ;
case constant - 2 :
statement block 2 ;
break ;
case constant - 3 :
statement block 3 ;
break ;

```

```

case constant - 4 :
statement block 4 ;
break ;
-----
-----

```

```

case n :
statement block - n ;
break ;
default :
default statement block ;
}

```

- * switch, case, default and break are keywords.
- * Any number can be included with in the switch.
- * Expression may be a variable or any type of expression.
- * If it's a variable the value stored with the variable when the value and the case number matches the particular block of statement with in the case will be executed skipping all other case.
- * case 1, case 2, case 3 are the way in which the cases are numbered while coding the program.
- * Skipping all other cases that are non matching is done by the keyword break.
- * Break keyword transfers the program control out of the switch.
- * When no matching case is not found the statement under the default will get executed same as the else part execution in else if ladder.