**Control Statements**

**Sequence**

**Selection**

**Iteration**

=========================================

Conditional ControlStatements

1. if
2. if… else
3. else if Ladder
4. Nested if else

LoopingControlStatements

1. while
2. do… while
3. for

JumpControlStatements

1. switch
2. goto

Other IMP Terms

break

continue

return

exit

**1. if**

Definition:

Syntax :

if(<condition>)

{

//Statements

}

Example :

if(N1 == N2)

{

printf(“\n Both Numbers are Equal.”);

}

**2.if… else**

Definition:

Syntax :

if(<condition>)

{

//True Block Statements

}

else

{

//False Block Statements

}

Example :

if(N1 > N2)

{

printf(“\n Number1 is Maximum.”);

}

else

{

printf(“\n Number2 is Maximum.”);

}

**3.else if Ladder**

Definition:

Syntax :

if(<condition1>)

{

//True Block Statements C1

}

elseif(<condition2>)

{

//True Block Statements C2

}

elseif(<condition3>)

{

//True Block Statements C3

}

.

.

.

else

{

// All Conditions False

//Default Statements

}

Example :

if(N1 == N2)

{

printf(“\n Both Numbers areEqual.”);

}

elseif(N1 > N2)

{

printf(“\n Number1 is Maximum.”);

}

else

{

printf(“\n Number2 is Maximum.”);

}

4. Nested if… else

Definition :

Syntax:

if(<condition1>) // False

{

if(<condition2>)

{

// True Statements Block of Condition 1 & 2

}

else

{

// True Statements Block of Condition 1

}

}

else if(<condition3>) // False

{

if(<condition4>)

{

// True Statements Block of Condition 3 & 4

}

else

{

// True Statements Block of Condition 3

}

}

else

{

// False Block For Conditions 1 & 3

}

Switch

It is Only supports for char & int data type expression.

Syntax:

switch(<Expression>) // datatype char

{

case ‘<Value of Expr1>’:

Statements;

break;

case ‘<Value of Expr2>’:

Statements;

break;

case ‘<Value of Expr3>’:

Statements;

break;

.

.

.

.

case ‘<Value of ExprN>’:

Statements;

break;

default :

Default case Statements;

break;

}

switch(<Expression>) // datatypeint

{

case<Value of Expr1>:

Statements;

break;

case<Value of Expr2>:

Statements;

break;

case<Value of Expr3>:

Statements;

break;

.

.

.

.

case<Value of ExprN>:

Statements;

break;

default :

Default case Statements;

break;

}