DECISION AND CONTROL STATEMENTS

CONTROL FLOW

- C program executes one command after another , from top to bottom.
- Control statements enable
  - Alter the flow of program
  - Repeat commands
- The if statement enable us to skip commands or execute command depending on condition

if(condition)

‘C’ CONTROL STATEMENTS

STATEMENT TYPE	KEYWORD
Decision and Selection Control	***if-else***, ***switch case***, conditional operator
Looping	***while,*** ***do-while,*** ***For***
Branching	***break,*** ***continue, return,*** ***goto, label***

if-else STATEMENT

```
 if (condition)
{
    block of statements;
}
else
{
    block of statements;
}
```

NESTED *if-else*

```
if (condition1)
{
    if (condition2)
    block of statements;
    else
    block of statements;
}
else
{
    if (condition3)
    block of statements;
    else
    block of statements;
}
```

else if LADDER

```
 if (condition)    //Sequence of conditions need to be executed.
 {                          // If true, control comes out of ladder.
     statements;      // If false, control goes to next else if condtion
}
else if(condition)
{
     statements;
}
else if(condition)  {
     statements;
}
else                        //Default when nothing comes true
{
     statements;
```

USE OF LOGICAL OPERATOR

```
if (marks >= 75 )
    grade = ‘A’;
else if(marks >= 60 && marks <75)
    grade = ‘B’;

if(gender ==‘m’ || age>30)
printf(“Insured”);
else
printf(“Not Insured”);
```

- E1 && E2, if E1 is false E2 is not evaluated
- E1 || E2, if E1 is true E2 is not evaluated

CONDITIONAL OPERATOR

- Also known as ternary operator.
- Syntax:

expression1 **?** expression2 **:** expression3;

- Example: To find maximum of 2 numbers.

  max = (num1 > num2) ? num1 : num2;

- This is equivalent to

```
if(num1>num2)
        max = num1;
else
        max=num2;
```

switch STATEMENT

- Syntax

switch (expression)

{

case value1:

statements;

break;

case value2:

statements;

break;

…..

default:

statements;