Decision Control

In sequential Instructions the various statements get executed in the same order they appear in the program. In many programming situations, we want one set of instructions to be executed in one situation and a different set in another situation. These situations are dealt with using decision controlled instructions.

1. Its not necessary that the statement after ? Or : be only arithmetic.
2. The Conditional Operators can be nested.
3. The limitation of conditional operator is that after the ? Or : only one statement can occur.

Multiple Initialisations in the FOR loop

1. for(I=1,j=2; j<=10; j++)
2. For (I=1,j=2; d

Switch Case

# Syntax

switch(integer expression) {

Case constant 1:

Do this;

Break;

Case constant 2:

Do this;

Break;

Case constant 3:

Do this;

Break;

Default:

Do this;

Break;

* The control instruction that allows us to make a decision from the number of choices is called a switch OR a switch case default.
* The cases in a switch can be put in any order i.e it does not have to follow 1,2,3 order
* Even of there are multiple statements to be executed in case there is no need to enclose them within a pair of braces/parenthesis.
* Every statement in a switch must belong to some case or the other.
* If we have no default case, and no case is satisfied then the control exits the switch and continues with the next instruction.
* At times, we may want to execute a common set of statements
* The disadvantage of SWITCH over IF is that. One cannot have a case in a switch which is as follows: case i <= 20
* The value of an expression can be checked provided it is a constant.
* In principle, a switch may occur within another switch, however in practice this is barely done.
* Switch statement is mainly used in Menu-Driven programs.

The “GOTO” Statement

The GOTO Statement causes an unconditional jump to another statement in the same function.

* The destination of the jump is specified using a label following the GOTO Keyword.