Decision Making

If statement

C if Statement

The syntax of the if statement in C programming is:

```
if (test expression)
{
// statements to be executed if the test expression is true
}
```

Example

```
/*Program to display a number if it is negative*/
#include <stdio.h>
int main() {
int number;
printf("Enter an integer: ");
scanf("%d", &number);
if (number < 0) {
printf("You entered %d.\n", number);
}
printf("The if statement is easy.");
return 0;
}
```

C if...else Statement

The if statement may have an optional else block. The syntax of the if..else statement is:

```
if (test expression) {
// statements to be executed if the test expression is true
}
else {
// statements to be executed if the test expression is false
}
```

Example

```
/* Check whether an integer is odd or even*/
#include <stdio.h>
int main() {
int number;
printf("Enter an integer: ");
scanf("%d", &number);
if (number%2 == 0) {
printf("%d is an even integer.",number);
}
else {
printf("%d is an odd integer.",number);
}
return 0;
}
```

How if...else statement works?

If the test expression is evaluated to true,

statements inside the body of if are executed.

statements inside the body of else are skipped from execution.

If the test expression is evaluated to false,

statements inside the body of else are executed

statements inside the body of if are skipped from execution.

C if...else Ladder

The if...else statement executes two different codes depending upon whether the test expression is true or false. Sometimes, a choice has to be made from more than 2 possibilities.

The if...else ladder allows you to check between multiple test expressions and execute different statements.

Syntax

```
if (test expression1) {
statement(s)
}
else if(test expression2) {
 statement(s)
}
else if (test expression3) {
statement(s)
}
.
.
else {
 statement(s)
}
```

```
#include <stdio.h>
int main() {
int number1, number2;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
if(number1 == number2) {
printf("Result: %d = %d",number1,number2);
}
else if (number1 > number2) {
printf("Result: %d > %d", number1, number2);}
else {
printf("Result: %d < %d",number1, number2);
}return 0;
}
```

- Write a program to check whether the given integer is positive even, negative even, postitive odd or negative odd.
- Write a program to check whether a percentage score of an student is Distinction, first division, second division, third division or fail.

Nested If

Syntax

The syntax for a **nested if** statement is as follows :

```
if( boolean_expression 1)
 { /* Executes when the boolean expression 1 is
    true */
 if(boolean_expression 2)
{ /* Executes when the boolean expression 2 is
    true */ }
 }
```

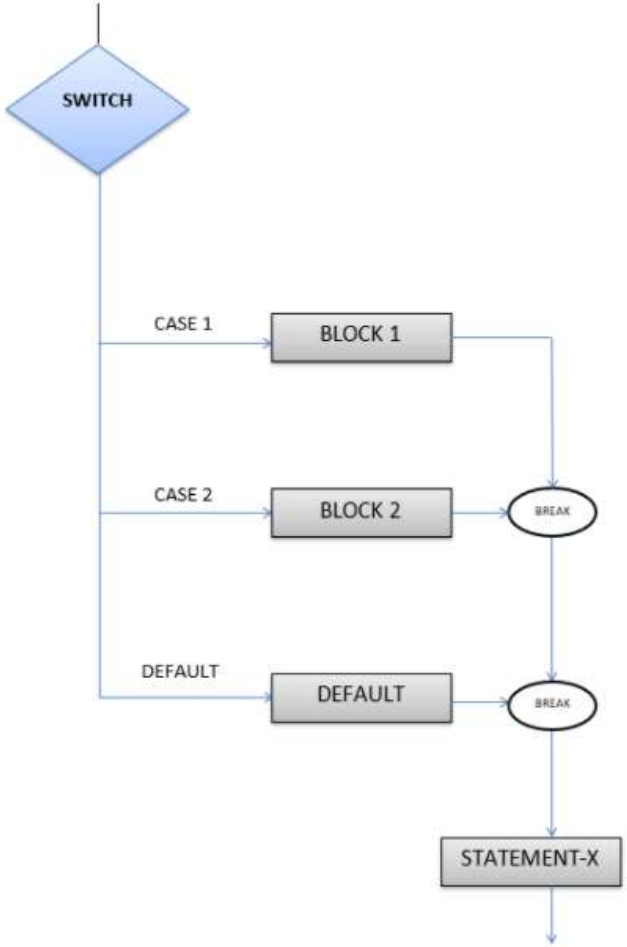
Nested if...else

It is possible to include an if...else statement inside the body of another if...else statement.

```
#include <stdio.h>
int main() {
int number1, number2;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
if (number1 >= number2) {
if (number1 == number2) {
printf("Result: %d = %d",number1,number2);
}
else {
printf("Result: %d > %d", number1, number2);
}
}
else {
printf("Result: %d < %d",number1, number2);
}
return 0;
}
```

Switch statement

- **Switch statement in C** tests the value of a variable and compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is executed.
- Each case in a block of a switch has a different name/number which is referred to as an identifier.


SWITCH
CASE 1
BLOCK 1
CASE 2
BLOCK 2
BREAK
DEFAULT
DEFAULT
BREAK
STATEMENT-X

Example

```
// Program to create a simple calculator
#include <stdio.h>

int main() {
   char operator;
   double n1, n2;

   printf("Enter an operator (+, -, *, /): ");
   scanf("%c", &operator);
   printf("Enter two operands: ");
   scanf("%lf %lf",&n1, &n2);

   switch(operator)
   {
      case '+':
         printf("%.1lf + %.1lf = %.1lf",n1, n2, n1+n2);
         break;

      case '-':
         printf("%.1lf - %.1lf = %.1lf",n1, n2, n1-n2);
         break;

      case '*':
         printf("%.1lf * %.1lf = %.1lf",n1, n2, n1*n2);
         break;

      case '/':
         printf("%.1lf / %.1lf = %.1lf",n1, n2, n1/n2);
         break;

      // operator doesn't match any case constant +, -, *,
/
      default:
         printf("Error! operator is not correct");
   }

   return 0;
}
```