Note 4:
Conditional Branching Statements

1 Conditional Branching Statements

There are 4 conditional Statements in C:

1. If statement
2. If-else statement
3. if-else-if statement
4. switch statement

1.1 If Statement

The if statement is the simplest form of decision control statements that is frequently used in decision making. The if block may include one statement or n statements enclosed with curly brackets. First the test expression is evaluated. If the test expression is true, the statement of if block (statement 1 to n) are executed otherwise these statements will be skipped and the execution will jump to statement x.

General Syntax:

```
if (test expression) {
    statement 1;
    ............
    statement n;
}
statement x;
```

The statement in an if block is any valid c language statement and the text expression is any valid C language expression that may include logical operators.
Note that there is no semi colon after the test expression.

Ex:

```
#include<stdio.h>
int main() {
    int x = 10;
    if (x>0) {
        x++;
    }
    printf("x = %d", x);
    return 0;
}
```

this is will output x = 11.

In the above code, we take a variable x and initialize it to 10. In the text expression we check if the value of the variable x is greater than 0. If it is, we increment the value of x by one, if it isn't we do nothing to change the value.

1.2 If - Else Statement

If we want to execute a different set of expressions if our test expression isn't true rather than just doing nothing, we use the if - else statement.

General Syntax :

```
if (test expresssion) {
    statement 1;
    ....
    statement n;
} else {
    statement x;
}
statement y;
```

Ex:

```
#include<stdio.h>
int main() {
    int x;
    printf("Enter the value of x: ");
    scanf("%d", &x);

    if (x % 2 == 0) {
        printf("%d is an even number", x);
    } else {
        printf("%d is an odd number", x);
    }
    return 0;
}
```

In the above code, we ask for an input from the user, we then check the reminder of that number after dividing with 2. We know that any even number divided by 2 will give a reminder of 0, and so if the reminder is 0, we print that the number entered is 0 else we print that the number we printed is odd.

1.3 If - Else - If Statement

What if we want to check a number of text expressions instead of just 2? Well in that case we use an if-else-if statement. This basically allows us to check as many cases as we want.

General Syntax:

```
if ( text expression ) {
    statement 1;
} else if (text expression 2) {
    statement 2;
}
...............................................
} else {
    statement block x;
}
statement y;
```

Ex:

```
#include<stdio.h>
int main() {
    int x, y;
    printf("Enter the values of x and y");
    scanf("%d %d", &x, &y);

    if (x == y) {
        printf("The two numbers are equal");
    } else if (x > y) {
        printf("%d is greater than %d", x, y);
    } else {
        printf("%d is greater than %d", y, x);
    }
    return 0;
}
```

Here are some rules regarding if, if-else and if-else-if statements.

1. The expression must be enclosed in parentheses.
2. No semicolon is placed after the if/if-else/if-else-if statement. Semi colons are only placed at the end of statements in the statement block.
3. A statement block begins and ends with a curly brace. No semicolon is placed after the opening/closing braces.

1.4 Switch Case

A switch case statement is a multi-way decision statement that is a simplified version of an if-else block that evaluates only one variable.

Switch statements are mostly used in two situations:

1. When there is only one variable to evaluate in the expression.
2. When many conditions are being tested.

General Syntax:

```
switch (variable)
{
case value1:
    Statement Block 1;
    break;
case value2:
    Statement block 2;
    break;
....................................................

case valuen:
    Statement block n;
    break;
default:
    Statement block D;
    break;
}
Statement X;
```

Default is also a case that is executed when the value of the variable does not match with any of the values of the case statement, i.e., the default case is executed when no match is found between the values of switch and case statements and thus there are no statements to be executed.

Ex:

```
#include<stdio.h>
int main() {
    char grade = 'C';
    switch(grade)
    {
    case 'O':
        printf("Outstanding");
        break;
    case 'A':
        printf("Excellent");
        break;
    case 'B':
        printf("Good");
        break;
    case 'C':
        printf("Fair");
        break;
    default:
        printf("Invalid grade");
        break;
    }
}
```

In the next note, we will go through the remaining topics which are the looping statements.