IT1010 – Note for Lecture 04

What are the basic control structures in C language?

The basic control structures in C are the ways to control the flow of execution of a program. There are three basic control structures:

1. Sequence
2. Selection
3. Repetition

Sequence control structure

This is the default mode of execution, where the statements in a program are executed one after the other.

For example, the following code will print "Hello, world!" followed by "This is a sequential statement":

```
printf("Hello, world!\n");
printf("This is a sequential statement.\n");
```

Selection control structure

The selection control structure in C is used to make decisions, such as whether to execute a certain block of code or not. The most common selection statements in C are the if, else and switch statements.

The if statement is used to make a simple decision. The syntax for the if statement is:

```
if (condition)
{
	// block of code to be executed if condition is true
}
```

```
int x = 10;

if (x > 5)
{
	printf("x is greater than 5.\n");
}
```

The else statement is used to execute a block of code if the condition in the if statement is not true. The syntax for the else statement is:

```
if (condition)
{
        // block of code to be executed if condition is true
}
else
{
        // block of code to be executed if condition is false
}
```

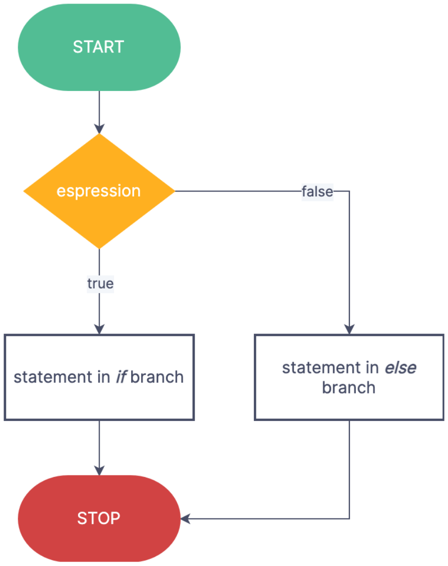

Figure 1 : Flow chart of if – else control structure

For example, the following code will print "x is less than or equal to 5" if the variable x is not greater than 5 and will print "x is greater than 5" if the variable x is greater than 5:

```
int x = 10;

if (x > 5)
{
  printf("x is greater than 5\n");
}
else
{
  printf("x is less than or equal to 5\n");
}
```

What operators can be used with if condition?

The if control structure in C can be used to test a wide variety of conditions. Some of the most common conditions that can be tested with the if statement include:

- Equality: **x == y** tests whether the values of x and y are equal.
- Inequality: **x! = y** tests whether the values of x and y are not equal.
- Less than: **x < y** tests whether the value of x is less than the value of y.
- Less than or equal to: **x <= y** tests whether the value of x is less than or equal to the value of y.
- Greater than: **x > y** tests whether the value of x is greater than the value of y.
- Greater than or equal to: **x >= y** tests whether the value of x is greater than or equal to the value of y.
- Logical operators: The && and || operators can be used to combine multiple conditions.

  For example, **x > 5 && x < 10** tests whether the value of x is greater than 5 and less than 10.
- Relational operators: The <, >, <=, and >= operators can be used to compare the values of two expressions. For example, x < y tests whether the value of x is less than the value of y.

The next form of conditional control structure can be used in C language is the Switch-Case statement block.

The switch case control structure in C is a way of executing different code blocks based on the value of a variable. The syntax for a switch statement is as follows:

```
switch (expression)
{
  case value1:
              // code block for value1
              break;
  case value2:
              // code block for value2
              break;
  ..........................
  default:
              // code block for all other values
}
```

The expression in the switch statement is evaluated once, and the value of the expression is compared to the values of each case label. If the value of the expression matches a case label, the code block associated with that case label is executed.

The break statement is used to exit the switch statement. If the value of the expression does not match any of the case labels, the code block associated with the default label is executed.

For example, the following code uses a switch statement to print a different message depending on the value of the variable day:

```
int day = 3;

switch (day)
{
  case 1:
    printf("Monday");
    break;

  case 2:
    printf("Tuesday");
    break;

  case 3:
    printf("Wednesday");
    break;

  default:
    printf("Unknown day");
}
```

If the value of day is 1, the printf("Monday") statement will be executed. If the value of day is 2, the printf("Tuesday") statement will be executed, and so on. If the value of day is not 1, 2, or 3, the printf("Unknown day") statement will be executed.

The switch statement is a powerful tool for controlling the flow of execution in a C program. It can be used to simplify code and make it more readable.

The difference between if and switch control structures

The if and switch control structures in C are both used to make decisions about the flow of execution in a program. However, there are some key differences between the two.

The if statement is used to make a decision **based on a Boolean expression**. The Boolean expression can be any expression that evaluates to a Boolean value, such as true or false. If the Boolean expression evaluates to true, the code block associated with the if statement is executed.

If the Boolean expression evaluates to false, the code block associated with the if statement is skipped.

The switch statement is used to make a decision **based on the value of a variable**. The variable can be of any type that can be compared to other values, such as int, char, or string. **The switch statement compares the value of the variable to a series of case labels. If the value of the variable matches a case label, the code block associated with that case label is executed**. If the value of the variable does not match any of the case labels, the code block associated with the default label is executed.

Here is a table that summarizes the key differences between if and switch control structures:

Feature	**if statement**	**switch statement**
Purpose	Makes a decision based on a Boolean expression	Makes a decision based on the value of a variable
Expression	Boolean expression	Variable
case labels	None	One or more case labels
default label	Optional	Optional