if-else vs switch

What is an if-else statement?

An if-else statement in C programming is a conditional statement that executes a different set of statements based on the condition that is true or false. The 'if' block will be executed only when the specified condition is true, and if the specified condition is false, then the else block will be executed.

Syntax of if-else statement is given below:

```
if(expression)
{
    // statements;
}
else
{
   // statements;
}
```

What is a switch statement?

A switch statement is a conditional statement used in C programming to check the value of a variable and compare it with all the cases. If the value is matched with any case, then its corresponding statements will be executed. Each case has some name or number known as the identifier. The value entered by the user will be compared with all the cases until the case is found. If the value entered by the user is not matched with any case, then the default statement will be executed.

Syntax of the switch statement is given below:

```
switch(expression)
{
  case constant 1:
  // statements;
  break;
  case constant 2:
  // statements;
  break;
  case constant n:
  // statements;
  break;
 default:
// statements;
}
```

Similarity b/w if-else and switch

Both the if-else and switch are the decision-making statements. Here, decision-making statements mean that the output of the expression will decide which statements are to be executed.

Differences b/w if-else and switch statement

The following are the differences between if-else and switch statement are:

- **Definition**

if-else

Based on the result of the expression in the 'if-else' statement, the block of statements will be executed. If the condition is true, then the 'if' block will be executed otherwise 'else' block will execute.

Switch statement

The switch statement contains multiple cases or choices. The user will decide the case, which is to execute.

- **Expression**

If-else

It can contain a single expression or multiple expressions for multiple choices. In this, an expression is evaluated based on the range of values or conditions. It checks both equality and logical expressions.

Switch

It contains only a single expression, and this expression is either a single integer object or a string object. It checks only equality expression.

- **Evaluation**

If-else

An if-else statement can evaluate almost all the types of data such as integer, floating-point, character, pointer, or Boolean.

Switch

A switch statement can evaluate either an integer or a character.

- **Sequence of Execution**

If-else

In the case of 'if-else' statement, either the 'if' block or the 'else' block will be executed based on the condition.

Switch

In the case of the 'switch' statement, one case after another will be executed until the **break** keyword is not found, or the default statement is executed.

- **Default Execution**

If-else

If the condition is not true within the 'if' statement, then by default, the else block statements will be executed.

Switch

If the expression specified within the **switch** statement is not matched with any of the cases, then the default statement, if defined, will be executed.

- **Values**

If-else

Values are based on the condition specified inside the 'if' statement. The value will decide either the 'if' or 'else' block is to be executed.

Switch

In this case, value is decided by the user. Based on the choice of the user, the case will be executed.

- **Use**

If-else

It evaluates a condition to be true or false.

Switch

A **switch** statement compares the value of the variable with multiple cases. If the value is matched with any of the cases, then the block of statements associated with this case will be executed.

- **Editing**

If-else

Editing in 'if-else' statement is not easy as if we remove the 'else' statement, then it will create the havoc.

Switch

Editing in **switch** statement is easier as compared to the 'if-else' statement. If we remove any of the cases from the switch, then it will not interrupt the execution of other cases. Therefore, we can say that the **switch** statement is easy to modify and maintain.

- **Speed**

If-else

If the choices are multiple, then the speed of the execution of 'if-else' statements is slow.

Switch

The case constants in the switch statement create a jump table at the compile time. This jump table chooses the path of the execution based on the value of the expression. If we have a multiple choice, then the execution of the switch statement will be much faster than the equivalent logic of 'if-else' statement.

Let's summarize the above differences in a tabular form.

	If-else	switch
Definition	Depending on the condition in	The user will decide which statement is

	the 'if' statement, 'if' and 'else' blocks are executed.	to be executed.
Expression	It contains either logical or equality expression.	It contains a single expression which can be either a character or integer variable.
Evaluation	It evaluates all types of data, such as integer, floating-point, character or Boolean.	It evaluates either an integer, or character.
Sequence of execution	First, the condition is checked. If the condition is true then 'if' block is executed otherwise 'else' block	It executes one case after another till the break keyword is not found, or the default statement is executed.
Default execution	If the condition is not true, then by default, else block will be executed.	If the value does not match with any case, then by default, default statement is executed.
Editing	Editing is not easy in the 'if-else' statement.	Cases in a switch statement are easy to maintain and modify. Therefore, we can say that the removal or editing of any case will not interrupt the execution of other cases.
Speed	If there are multiple choices implemented through 'if-else', then the speed of the execution will be slow.	If we have multiple choices then the switch statement is the best option as the speed of the execution will be much higher than 'if-else'.