Lab Instructors: Romasha Khurshid Nida Munawar

Course Code: CL118

Credit Hour: 1

Program Section: SE-19

At the end of the lab, you will be able to understand the use of:

* Decision Control Statement:
* If, If-Else, Nested If -Else, Else-If, Switch case Statements

**C– Decision Control statement**

In any programming language, there is a need to perform different tasks based on the condition. For example, consider an online website, when you enter wrong id or password it displays error page and when you enter correct credentials then it displays welcome page. So there must be a logic in place that checks the condition (id and password) and if the condition returns true it performs a task (displaying welcome page) else it performs a different task (displaying error page).

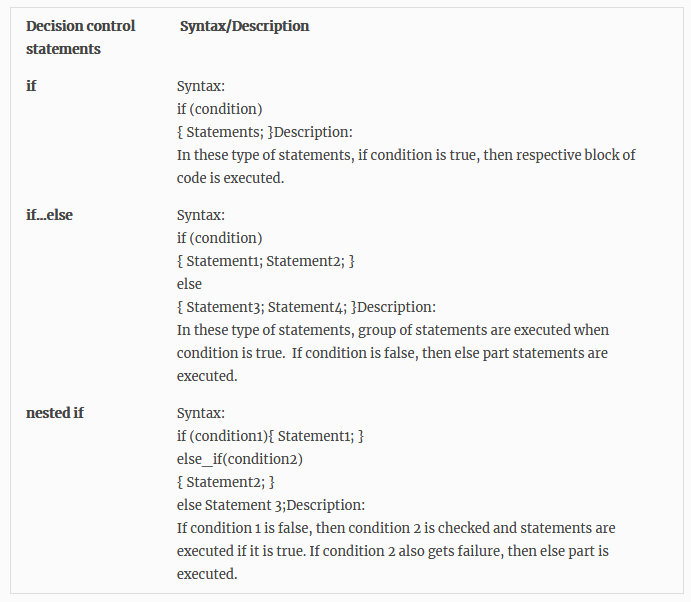
Using decision control statements, we can control the flow of program in such a way so that it executes certain statements based on the outcome of a condition (i.e. true or false).

In C Programming language we have following decision control statements.

* if statements
* if-else statements
* nested if-else statements
* else if Ladder
* switch-case statements

**“If”, “else” and “nested if” decision control statements in C:**

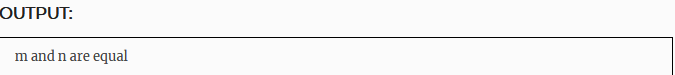
Syntax for each C decision control statements are given in below table with description.



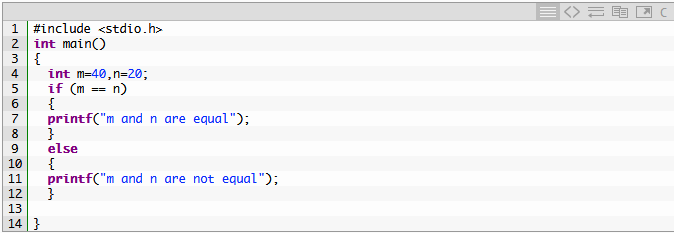
**Example program for “if” statement in C:**

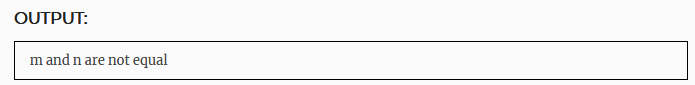
The statements inside if body executes only when the condition defined by if statement is true. If the condition is false, then compiler skips the statement enclosed in if’s body. We can have any number of if statements in a C program.





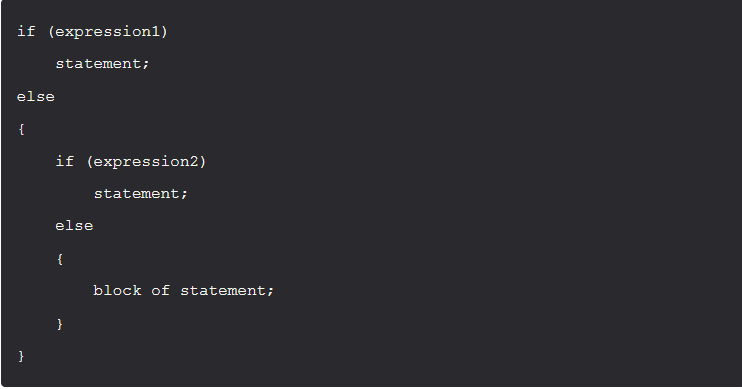
**Example program for “if-else” statement in C:**

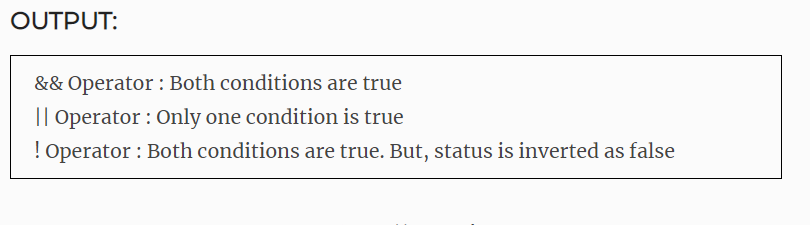
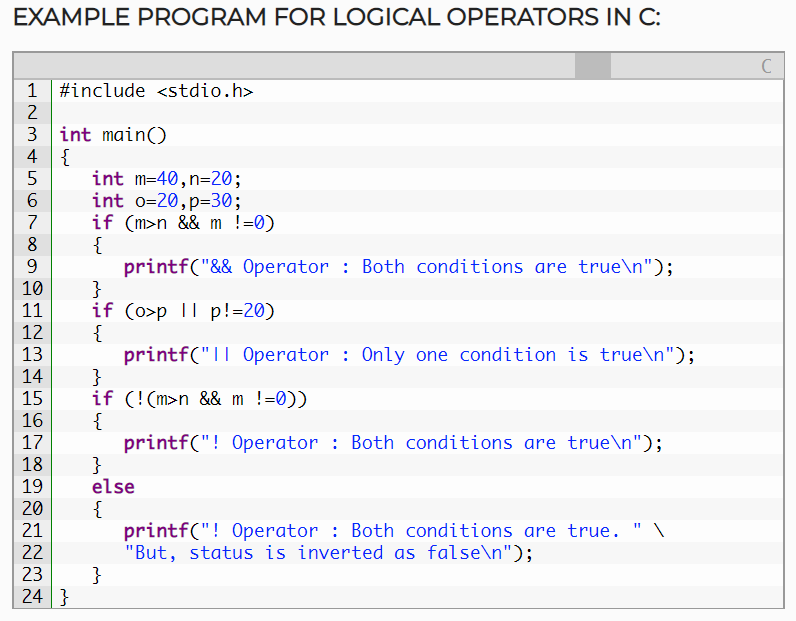
In this decision control statement, we have two block of statements. If condition results true, then if block gets executed else statements inside else block executes. else cannot exist without if statement. In this tutorial, I have covered else-if statements as well.

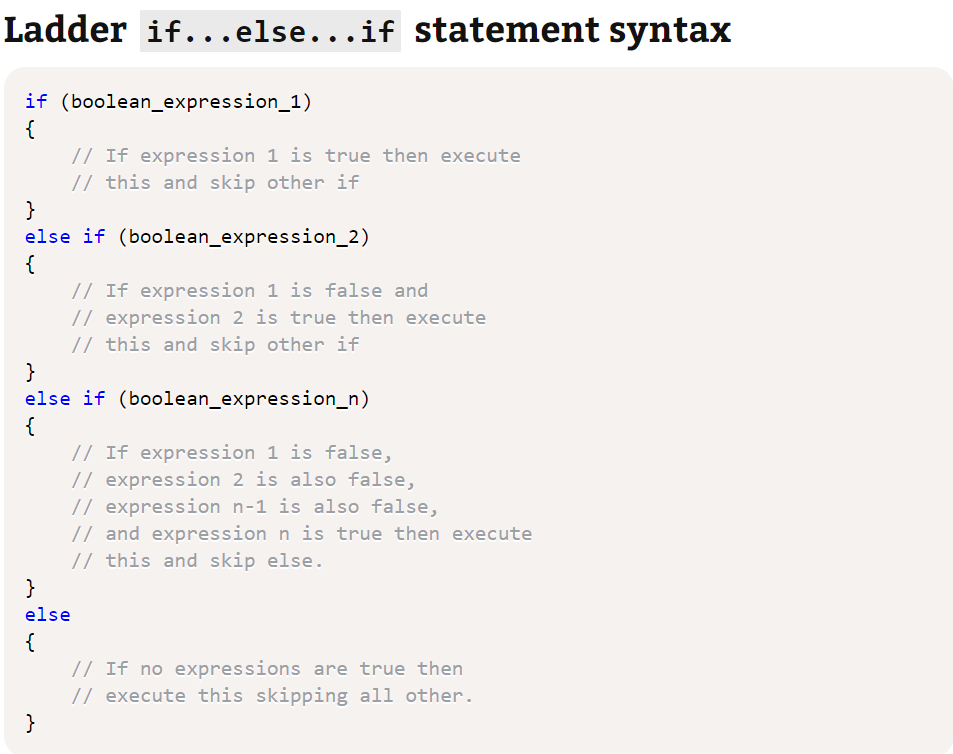


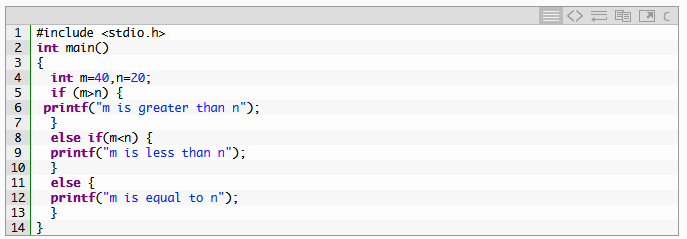
# Example program for nested if-else statement in C:

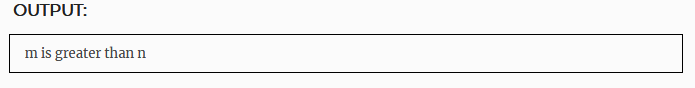
It is possible to nest if-else statement within one another. An if-else statement can occur within another if-else statement. The inner if-else is said to be nested in the outer if-else. Nesting can go up to any level.





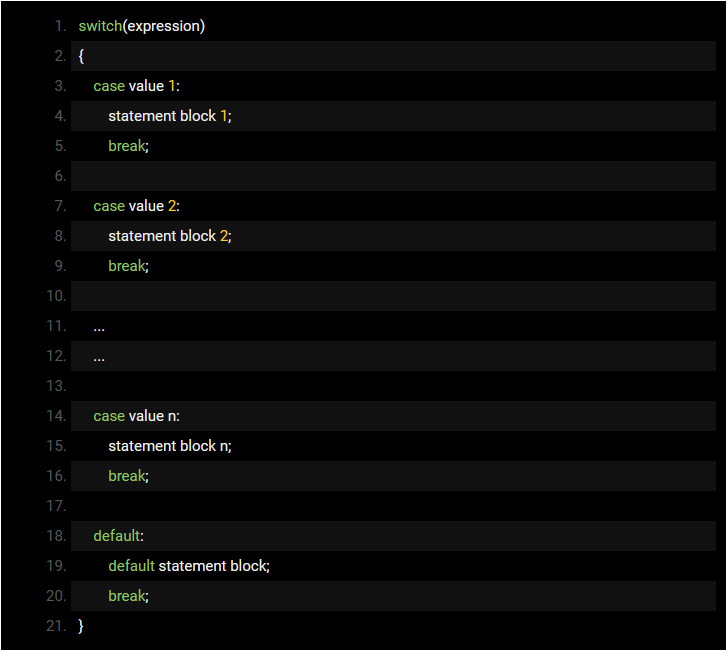






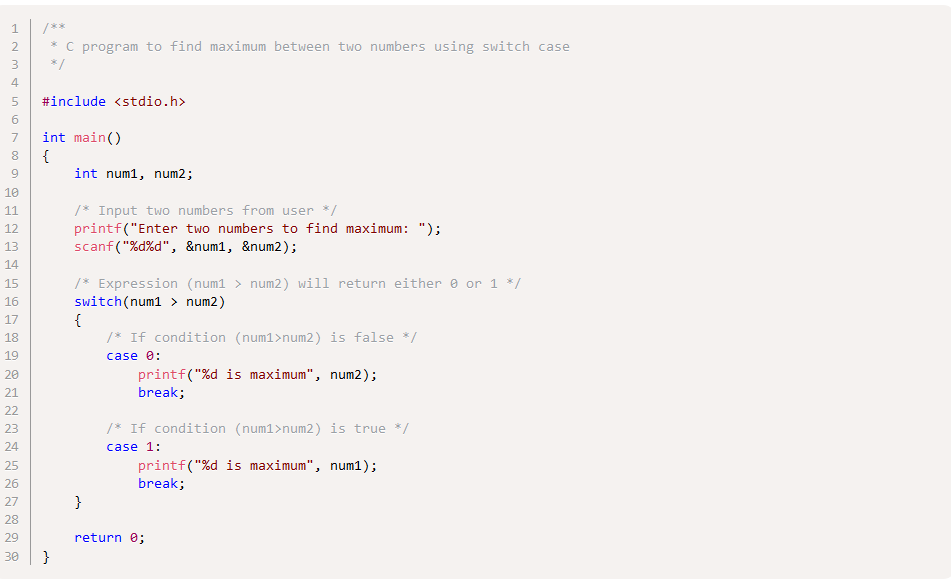
**'switch' Statements**

The alternative of nested 'if' statement is called 'switch' statement, which has different statement blocks for different cases. Basically 'switch' statement tries to match evaluated value of an expression against different cases. Syntax of 'switch' statement is as follows.



The expression 'expression' is evaluated first and the value is checked with different case values (value 1, value 2, etc.) If one of the case values matches the result of expression, then respective statement block is executed. If none is matched, then the statement block of 'default' case is executed. You may notice another keyword in the switch statement which is **break.** If **break** keyword is omitted, all the statements after that will be executed even if the corresponding case values do not match the value of expression. This is called a 'fall through' statement. For example, if 'break' is omitted and say, case 'value 1' is matched, then after execution of 'statement block 1' the code will execute statement block 2, 3 and so on until a 'break' statement is reached.

# Example program for switch-case statement in C:



**Rules for Writing 'switch' Statements in C Language**

* The values of case label must be unique, else complier will throw error
* The values of case should be an integral constant (integer or character). If any variable is specified as part of a case, compiler will throw error.
* The 'default' case is optional and it can be placed anywhere in the switch block.
* 'switch' statements can be nested; that means you can have another 'switch' statement as part of a statement block associated with a case.
* The statement block of a case is optional and it can be shared by more than one cases

**Jump Statements:**

# C – goto statement with example:

* The goto statement is rarely used because it makes program confusing, less readable and complex. Also, when this is used, the control of the program won’t be easy to trace, hence it makes testing and debugging difficult.

## C – goto statement:

When a goto statement is encountered in a C program, the control jumps directly to the label mentioned in the goto statement.



**Disadvantages of using goto statement:**

* The use of goto statement is highly discouraged as it makes the program logic very complex.
* use of goto makes the task of analyzing and verifying the correctness of programs (particularly those involving loops) very difficult.

**Lab Tasks**

1. Suppose there is a person named, Amanda who lives in Berlin and on the coming weekend she will have to attend her best friend wedding ceremony held in Paris on coming Saturday at 10 am, now on Friday she is got free from her job related work around 5 pm. She has only few hours to get prepared and leave for Paris and she kept only $200 for travelling expenses, now make a C program using switch control statement that helps Amanda to take best decision among the different choices of traveling.

Note: she has to reach Paris two hours prior to the event.

* Train takes about 8h28min and cost about $234
* Car takes 12h8m and cost about $70
* Air Plane takes about 1h50mins and cost $247
* Bus takes about 14h15mins and cost $41.

1. A patient went to the hospital as he has been suffering from fever for few days now make a program that helps physician to prescribe patient either for follow-up or to get admitted in hospital for proper care.
2. Suppose XYZ company has suffered a serious downfall, so company executives decided to downsize the company according to performance of the employees for that they consider monthly evaluation report for every employee and those who got their evaluation percentage less than 45 will get fire immediately and if percentage is less than equal to 50 generate a warning for them. Now write a C program that exactly defines the above scenario.
3. Write a C program to check whether a number is even or odd.
4. Write a C program to check whether a year is leap year or not.
5. Write a C program to input any alphabet and check whether it is vowel or consonant.
6. Write a C program to input any character and check whether it is alphabet or not.
7. Write a C program to print day of week name using switch case.
8. Write a C program to find maximum between three numbers using switch case.
9. Write a C program to check whether a number is positive, negative or zero using switch case.
10. Write a C program to create Simple Calculator using switch case.