Control structure

➢ IF Statement

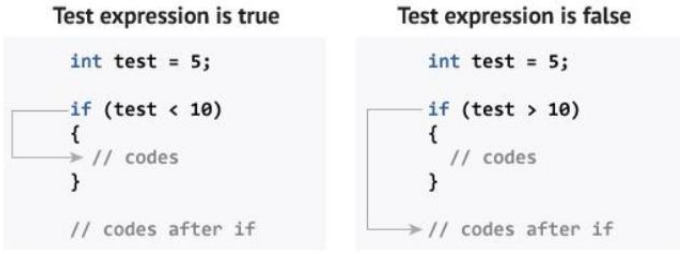
Lab	
Contents Covered:	**Control Structures** If statement If -else statement If else if statement Switch statement

The if statement is the simplest of the decision statements.
To specify the conditions under which a statement or group of statements should be executed.

```
if (test Expression)
{
// statements
}
```

The if statement evaluates the test expression inside parenthesis. If test expression is evaluated to true, statements inside the body of if is executed. If test expression is evaluated to false, statements inside the body of if is skipped.

How if statement works?



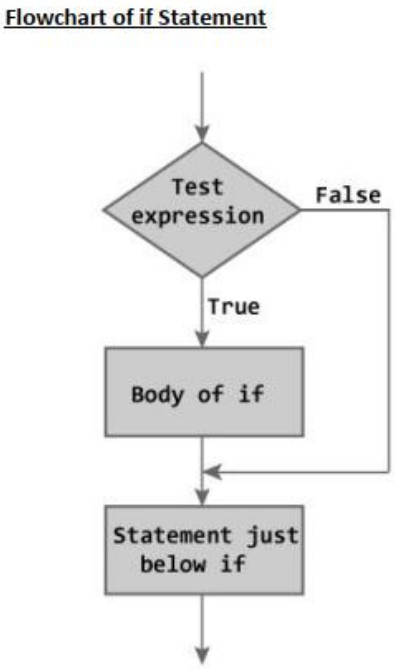


Figure: Flowchart of if Statement

Practice problem 1:

```
// ifdemo.cpp
// demonstrates IF statement
#include <iostream>
using namespace std;
int main()
{
int x;
cout << "Enter a number: ";
cin >> x;
if( x > 100 )
cout << "That number is greater than 100\n";
return 0;
}
```

Output

```
Enter a number: 150
That number is greater than
100
```

Example: program's output when the number entered by the user is greater than 100:

```
Enter a number: 2000
That number is greater than 100
```

If the number entered is not greater than 100, the program will terminate without printing the second line.

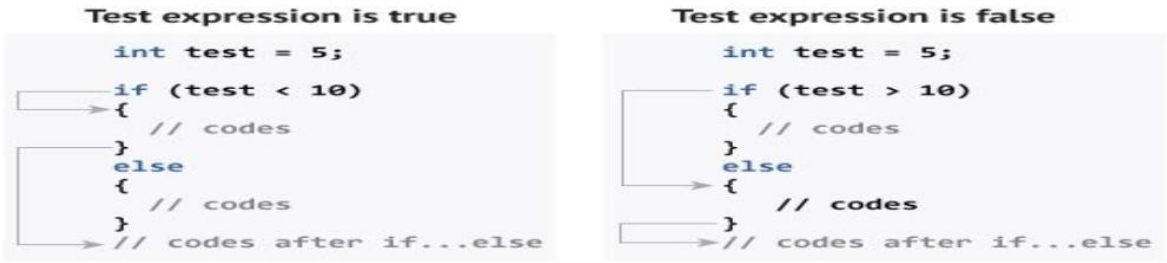
if...else

Syntax:

```
if(condition)
        statement;
else
        statement;
```

The if else executes the codes inside the body of if statement if the test expression is true and skips the codes inside the body of else. If the test expression is false, it executes the codes inside the body of else statement and skips the codes inside the body of if.

How if...else statement works?



Flowchart of if...else

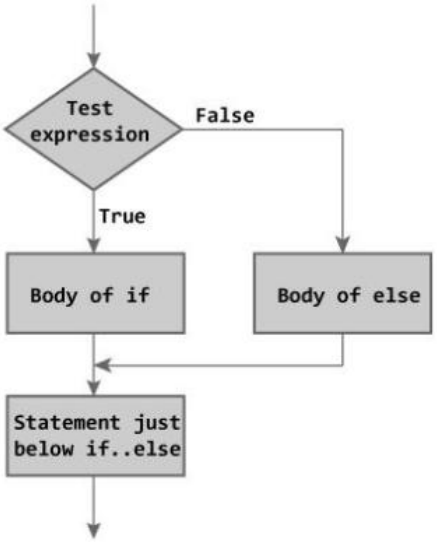


Figure: Flowchart of if...else Statement

PRACTICE PROBLEM 2:

```
#include <iostream>
using namespace std;
int main()
{
int x;
cout << "\n Enter a number: ";
cin >> x;
if( x > 100 )
cout << "That number is greater than 100\n";
else
cout << "That number is not greater than 100\n";
return 0;
}
```

Output:

Enter a number: 150
That number is greater than 100
Enter a number: 50
That number is not greater than 100

➢ C++ Program to Check Whether a character is Vowel or Consonant.

```
#include <iostream>
using namespace std;
int main() {
   char c;
   int isLowercaseVowel, isUppercaseVowel;

   cout << "Enter an alphabet: ";
   cin >> c;

   // evaluates to 1 (true) if c is a lowercase vowel
   isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

   // evaluates to 1 (true) if c is an uppercase vowel
   isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

   // show error message if c is not an alphabet
   if (!isalpha(c))
     cout<<"Error! Non-alphabetic character.";
   else if (isLowercaseVowel || isUppercaseVowel)
      cout << c << " is a vowel.";
   else
      cout << c << " is a consonant.";

   return 0;
}
```

Output:

```
Enter an alphabet: a
a is a vowel.
```

➢ C++ Program to Check Whether a character is Vowel or Consonant and handle multiple inputs

```
#include <iostream>
#include <cctype> // For isalpha function
using namespace std;

int main() {
   char c;
   int isLowercaseVowel, isUppercaseVowel;
   char choice;

   do {
      cout << "Enter an alphabet: ";
      cin >> c;

      // evaluates to 1 (true) if c is a lowercase vowel
      isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

      // evaluates to 1 (true) if c is an uppercase vowel
      isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
```

```
    // show error message if c is not an alphabet
    if (!isalpha(c)) {
        cout << "Error! Non-alphabetic character." << endl;
    } else if (isLowercaseVowel || isUppercaseVowel) {
        cout << c << " is a vowel." << endl;
    } else {
        cout << c << " is a consonant." << endl;
    }

    // Ask the user if they want to continue
    cout << "Do you want to enter another character? (y/n): ";
    cin >> choice;
  } while (choice == 'y' || choice == 'Y');

  return 0;
}
```

Output:

```
Enter an alphabet: a
a is a vowel.
Do you want to enter another
character? (y/n): y
Enter an alphabet: B
B is a consonant.
Do you want to enter another
character? (y/n): n
```

Check Whether Number is Even or Odd using if else

```
#include <iostream>
using namespace std;
int main()
{
   int n;
   cout << "Enter an integer: ";
   cin >> n;
if ( n % 2 == 0)
      cout << n << " is even.";
   else
      cout << n << " is odd.";

   return 0;
}
```

Output

```
Enter an integer: 2
2 is even
```

Nested If:

In C++ A nested if statement is an if condition placed inside another if condition, allowing you to test multiple conditions in a hierarchical manner.

Syntax : C++ Nested If

```
if(boolean_expression1)
{
// Executes when the boolean expression 1 is true
if(boolean_expression2)
{
// Executes when the boolean expression 2 is true
}
}
```

- **Example: write a program that take input from user and compare if it is less than 15 or 12.**

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   cout << "Enter a value: ";
   cin >> i;

   if (i < 15)
   {
      cout << "The number is less than 15." << endl;

      if (i < 12)
      {
         cout << "The number is also less than 12." << endl;
      }
      else
      {
         cout << "The number is 12 or greater but less than 15." << endl;
      }
   }
   else
   {
      cout << "The number is 15 or greater." << endl;
   }

   return 0;
}
```

Output

Enter a value: 13
The number is less than 15.
The number is 12 or greater but less than 15.

Lab 3 TASKS:

TASK 1: Find the largest of three numbers input by the user?
TASK 2: Find the grades of students? IF the entered marks are greater than 75, it should be the first class, if the entered marks are greater than 65 it should be in second class, below 55 it should be low class or fourth class?
TASK 3: Write a C++ program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer, calculate Average and grade according to given conditions:
If Average >= 90% : Grade A
If Average >= 80% : Grade B
If Average >= 70% : Grade C
If Average >= 60% : Grade D
If Average >= 40% : Grade E
If Average < 40% : Grade F

TASK 4: Write C++ program to check whether a character is Upper case or Lowercase?
TASK 5: Write a Program to check alphabet, digit or special character.

➢ Selection Statements

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.

Syntax:

Lab	
Contents Covered:	**Conditional statements and execution flow for conditional statements** • Switch statement • Nested Switch

The syntax for a switch statement in C++ is as follows −

```
switch(expression) {
  case constant-expression  :
    statement(s);
    break; //optional
  case constant-expression  :
    statement(s);
    break; //optional
  // you can have any number of case statements.
  default : //Optional
    statement(s);
}
```

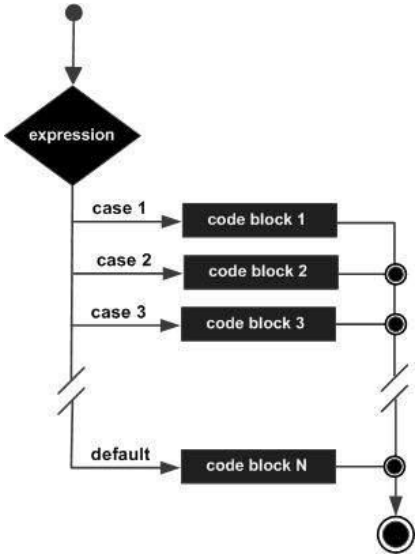
Rules of switch statement:

The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.

When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

- **Flow Diagram**



Switch case :

switch...case is a branching statement used to perform action based on available choices, instead of making decisions based on conditions. Using switch...case you can write more clean and optimal code than if...else statement. switch...case only works with integer, character and enumeration constants.

C++ switch statement

Example 1:Write a C++ program that uses a switch statement to evaluate a grade (A, B, C, D, or F). The program should print a message based on the grade entered, such as "Excellent!" for an A, "Well done" for B or C, "You passed" for a D, and "Better try again" for an F. If the input is not one of these grades, it should print "Invalid grade". Finally, display the grade entered by the user

Code:

```
#include <iostream>
using namespace std;
int main () {
  // local variable declaration:
  char grade = 'D';
  switch(grade) {
    case 'A' :
      cout << "Excellent!" << endl;
break;
    case 'B' :
    case 'C' :
      cout << "Well done" << endl;
      break;
```

```
    case 'D' :
      cout << "You passed" << endl;
      break;
    case 'F' :
      cout << "Better try again" << endl;
      break;
    default :
      cout << "Invalid grade" << endl;
  }

  cout << "Your grade is " << grade << endl;

  return 0;
}
```

Output:

You passed
Your grade is D

Example 2:Write a C++ program that asks the user to enter their grade (A, B, C, D, or F) and prints a message based on the grade. Ensure the program handles invalid inputs and displays the entered grade at the end.

Code:

```
#include <iostream>
using namespace std;

int main() {
   char grade;

   // Prompt the user to enter their grade
   cout << "Enter your grade (A, B, C, D, F): ";
   cin >> grade;

   // Switch statement to evaluate the grade
   switch(grade) {
      case 'A':
         cout << "Excellent!" << endl;
         break;
      case 'B':
      case 'C':
         cout << "Well done" << endl;
         break;
      case 'D':
         cout << "You passed" << endl;
```

```
            break;
        case 'F':
            cout << "Better try again" << endl;
            break;
        default:
            cout << "Invalid grade" << endl;
    }

    // Display the entered grade
    cout << "Your grade is " << grade << endl;

    return 0;
}
```

OUTPUT:

Enter your grade (A, B, C, D, F): **B**
Well done
Your grade is B

Lab 4 Tasks:

Task 1: Program to build a simple calculator using switch Statement
Task 2: Program to check vowel and consonant using switch statement

Task 3: Write a program to use weekday numbers to calculate weekday name using switch statement.

Task 4: Write a program to check positive, negative or zero using switch statement.

- **List of switch case programming exercises**

1-Write a C++ program print total number of days in a month using switch case.

2-Write a C program to check whether a number is even or odd using switch case.

Nested Switch

It is possible to have a switch as part of the statement sequence of an outer switch. Even if the case constants of the inner and outer switch contain common values, no conflicts will arise.

C++ specifies that at least 256 levels of nesting be allowed for switch statements.

Syntax

The syntax for a **nested switch** statement is as follows −

```
switch(ch1) {
case 'A':
cout << "This A is part of outer switch";
switch(ch2) {
case 'A':
cout << "This A is part of inner switch";
break;
case 'B': // ...
}
break;
case 'B': // ...
}
```

Example

```
#include <iostream>
using namespace std;

int main () {
  // local variable declaration:
  int a = 100;
  int b = 200;

  switch(a) {
    case 100:
      cout << "This is part of outer switch" << endl;
      switch(b) {
```

```
         case 200:
            cout << "This is part of inner switch" << endl;
      }
   }
   cout << "Exact value of a is : " << a << endl;
   cout << "Exact value of b is : " << b << endl;

   return 0;
}
```

This would produce the following result −

```
This is part of outer switch
This is part of inner switch
Exact value of a is : 100
Exact value of b is : 200
```