

**Control Statements:** Normally the statements are executed sequentially as written in the program. But some times the operational flow of control of a program has to be altered depending upon a condition. Such statements are called control statements. They are classified into two. a). Decision making/selection statements b). Iteration statements.

**Selection Statements:** This statements allows to select an option from the alternatives based on a condition. They are **if** statement and **switch** statement. The if statement is used to select a set of statements for execution based on a condition. Syntax is `if (test expression) { statement block ; }`. If we want to execute some action when the condition becomes false, then we use **if.....else** statements. **Nested If** : If statement inside another if statement is called nested if . **The else if ladder:** It is used in the programs when multiple branching is required. Different conditions will be given and each condition will decide which statement is selected for execution. Ex.

**Switch – case statement:** This is used for multi branching. The expression is evaluated to get an integer or character constant and it is matched against the constants specified in the case statement. When a match is found the statement block associated with that case is executed until the break statement is reached. If none of the case is matched, the default case will be executed. The default statement is optional, and if it is missing no action takes place when all matches fail. **If a case statement does not include a break statement, then the control continues right on the next case statement until either a break is encountered or end of the switch is reached.**

**Note:** All switch statements can be replaced by else if ladder, but all else if ladder cannot be substituted by switch. Conditions involve only equality checking or it should be converted into equality expressions. Relational operators should not be used.

To display the day of a week using switch statement.

```
#include<iostream>
using namespace std;
int main()
{ int day;
  cout<<"Enter a number between 1 and 7"<<endl;
  cin>>day;
  switch(day)
  { case 1: cout<<"sunday ";
    break;
    case 2: cout<<"Monday";
    break;
    case 3: cout<<"Tuesday ";
    break;
    .....
    case 7: cout<<"saturday ";
    break;
    default: cout<<"invalid input";
  }
  return 0; }
```

Check whether the character is upper case, lower case, digits or other characters.

```
#include<iostream>
using namespace std;
int main()
{ char ch;
  cout<<"Enter a character "<<endl;
  cin>>ch;
  if (ch>= 'A' && ch<= 'Z')
    cout<<"Capital letter "<<endl;
  else if (ch>= 'a' && ch<= 'z')
    cout<<"Small letter "<<endl;
  else if (ch>=0 && ch<=9)
    cout<<"Digit"<<endl;
  else
    cout<<"Other character"<<endl;
  return 0; }
```

**Conditional operator ( ? : )** It is a ternary operator. It requires three operands to operate upon. It is an alternative to if ... else statement. The first operand will be a logical expression (condition) and the remaining two are values or expressions. If the condition is true, the first value will be selected other wise the second. Ex. Largest among two numbers. `max=(a>b) ? a : b ;` Largest among three numbers `max=(a>b) ? ((a>c) ? a : c) : ((b>c) ? b : c);`

**Looping statements (Iteration statements) :** They allow a set of instructions to be performed until a certain condition is fulfilled. A **looping statements contain initialisation expression, test expression, update expression, and loop body**. The loop body is used when an action is to be repeated for execution till the condition is true. The variable used in the test expression ( loop control variable) gets its initial value through the initialisation expression. Update expression changes the value of the loop control variable. They are divided into two. **Entry controlled loop** ( it first check the condition and if it is true, the loop body is executed and the execution will be continued as long as the condition will be true. Ex. **For loop, while loop** ) and **exit controlled loop** (This loop always execute at least once , even when the test expression evaluates to false initially. Here the loop body is executed first and then condition is checked. If it is true loop body starts its second execution and continues till the condition becomes false. Ex **do-while loop** ).

To find the sum of first n natural numbers. ( Using for loop )

```
#include<iostream>
using namespace std;
int main()
{ int n, sum=0;
  cout<<"Enter the limit ";
  cin>> n;
  for(int i=1 ; i<=n ; i++)
    sum=sum+i;
  cout<<"Sum of the first "<<n<<" natural numbers = "<<sum;
  return 0; }
```

To find the sum of first n natural numbers.

```
#include<iostream>
using namespace std;
int main()
{ int n, sum=0;
  cout<<"Enter the limit ";
  cin>> n;
  i=1;
  do
  { sum=sum+i;
    i=i+1;
  }while(i<=n);
  cout<<"Sum of the numbers is = "<<sum;
  return 0; }
```

To find the sum of first n natural numbers. (while loop)

```
#include<iostream>
using namespace std;
int main()
{   int n, sum=0;
    cout<<"Enter the limit ";
    cin>> n;
    i=1;
    while(i<=n)
    {   sum=sum+i;
        i=i+1;
    }
    cout<<"Sum of the first "<<n<<" numbers = "<<sum;
    return 0; }
```

To find the sum of digits of a number.

```
#include<iostream>
using namespace std;
int main()
{   int n, sum=0, dig, temp;
    cout<<"Enter a number ";
    cin>> n;
    temp=n;
    while(n>0)
    {   dig=n%10;
        sum=sum+dig;
        n= n/10;
    }
    cout<<"Sum of the digits of "<<n<<" is = "<<sum;
    return 0; }
```

### Nested Loop:

Placing a loop inside a body of another loop is called nesting of a loop. When we nest two loops, the outer loop counts the number of completed repetitions of the inner loop. The loop control variable for two loops should be different. Ex.



```
for(i=1; i<=2; i++)
{
    for(j=1; j<=3; j++)
    {
        cout<<"\n'<<i<<" "<<j;
    } }
```

The out put is of the form  
(outer loop)      (inner loop)

1	1
1	2
1	3
2	1
2	2
2	3

**Jump statements:** We can change this sequential manner of the program execution by using jump statements. They perform unconditional branching in a program and they are **return**, **goto**, **break** and **continue**. C++ provides a standard library function **exit()** which is used to terminate the program. a) **goto statement** ; By using goto, we can transfer the program control to anywhere in the program without any condition. The syntax is

```
goto label;

label : .....;
       .....;
       .....;
```



Ex. for goto statement

```
#include<iostream>
using namespace std;
int main()
{   float a, b;
    cout<<"Enter two numbers : ";
    cin>>a>>b;
    if(b==0)
        goto end;
    cout<<"The quotient is "<<a/b;
    return 0;
end: cout<<" Division by zero not defined ";
    return 0; }
```

b) **break statement** : A break statement skips the rest of the loop and jumps over to the statement following the loop. It is used with switch and loops. A break statement causes an exit only from the innermost loop. Program control goes out of the loop even though the test expression is true

Ex. for break statement

```
#include<iostream>
using namespace std;
int main()
{
    int i,j;
    for(i=1; i<=6; ++i)
    {
        cout<<"\n";
        for(j=1; j<=i; j++)
        {   cout<<" * ";
            if(j==3)
                break;
        }
    }
    return 0; }
```

### Output

```
*
* *
* * *
* * *
* * *
* * *
```

**Continue Statement:** This is used for skipping over a part of the code within the loop body and forces next iteration of the loop. It is used only with loops .

Ex. continue statement

```
#include<iostream>
using namespace std;
int main()
{   for(int i=1; i<=10; i++)
    {
        if(i==3 || i==7)
            continue;
        cout<<i<<"\t";
    }
    return 0;
}

out put is 1 2 4 5 6 8 9 10
```

Here 3 and 7 are not in the list. When the value i becomes 3 or 7 continue statement is executed as a result the out put statement is skipped and control goes to next iteration.

## Questions :

- Write the four components of the loop ?
- In C++ ----- is used as a statement terminator ? ( semi column )
- Differentiate between 'a' and "a" in C++? ( a character constant which require 1 byte of memory, a string constant needs 2 bytes of memory).
- What happens if **break** statement is not used in **switch** statement ?
- What is meant by type conversion? Differentiate between **implicit** and **explicit type conversion**? Explain with examples ?
- Differentiate between **entry controlled loop** and **exit controlled loop**? Give examples .
- What is the usage of **default** in **switch** statement ?
- What is the difference between the expression **a=5** and **a==5** .
- is the key word used with **goto** statement ( label )
- operator in C++ is a ternary operator ( ? : or conditional operator )
- statement is used for unconditional jump in a program. ( goto )
- statement is used to exit a loop even if the condition is true ( break )
- A break statement causes an exit ..... ( from the inner most loop or switch )
- Predict the output : `int a = 5; int b = a++ + ++a; cout<<b<<"t<<"a; (5+7=12, use then change & change then use. 12 7)`
- predict the output of the following code code segment. `Int a, b, n=1000; cin>>a; b=n+a>1800?400:200; cout<<b; ( a) if the input value is 500. (b) if the input is 1000 (200,400)`
- predict the output of the following code segment. `for(int i=1; i<=10; i++); cout<<i; ( 11 , there is a semi column after the loop )`
- Rewrite the following code using while loop and do while loop. `for(int i=2, sum=0; i<=20; i=i+2 ) { sum+= i ; }`
- Write a program to produce the following design
- What is wrong with the following while statement  
if the value of k=2  

```
while(k>=0)
sum+=k;
```

( There is no updation statement. The value of k is always 2 and it is an infinite loop )
- Write the declaration statement for using the identifier PI in the program code in which the value is always 3.14
- Meaningful and processed form of data is known as ..... (information )
- Predict the out put of the following program code if we give ( a) 0 as input (b) 2 as input ( c ) 7 as input justify the answer `int a; cin>>a; switch(a) { case 0: cout<<"Zero "; case 1: cout<<"one"; case 2: cout<<"Two"; break; default: cout<<"invalid number "; }`

( a) zero one two - no break statement after each statement. (b) Two (c) invalid number )
- How many times the following loop will be executed ?  
`Int s=0, i=0; do { s+=i; i++; } while(i<=5);` ( 6 times i have values 0, 1,2,3,4,5 )
- Write C++ program for the following questions ? (a). To check whether the given number is prime or not. (b). To print the terms of the Fibonacci series. (c). To find the reverse of a given number . (d). To check whether a given number is palindrome or not. (e).
- Distinguish between **break** and **continue** statements ? Explain with example ?

break	continue
<ul style="list-style-type: none"> <li>Used with switch and loop</li> <li>Forces termination of the loop even though the condition is true.</li> <li>A break inside a loop will abort the loop and transfer control to the statement following the loop.</li> </ul>	<ul style="list-style-type: none"> <li>Used with loops</li> <li>It forces next iteration of the loop.</li> <li>A continue statement will just abandon the current iteration and let the loop start next iteration.</li> </ul>

### Program to display a triangular stars.

```
#include<iostream>
using namespace std;
int main( )
{
    short int i, j;
    for(i=1; i<=5; i++) // outer loop
    {
        for(j=1; j<=i; j++) //inner loop
            cout<<"*";
    }
    return 0;
}
06.07.2021
```

### Multiplication Table

```
#include<iostream>
using namespace std;
int main( )
{
    int i, n;
    cout<<" Enter a number ";
    cin>>n;
    cout<<" Multiplication table of " <<n<<endl;
    for(i=1; i<=10; i++)
        cout<<i<<" X " <<n<<" = " <<i*n<<endl;
    return 0; }
Downloaded from www.Hsslive.in @
```

**To find the average height of a group of students.**

```
#include<iostream>
using namespace std;
int main( )
{
    float ht, sum=0, avg_ht;
    short n=0;
    char ch;
    do
    {
        cout<<"Enter the height ";
        cin >>ht;
        n++;
        sum=sum+ht;
        cout<<"Any more student (Y/N) ? ";
        cin>>ch;
    } while(ch=='Y' || ch=='y');
    avg_ht=sum/n;
    cout<<" Average height is  = "<<avg_ht;
    return 0;    }
```

**// Prime or not**

```
#include<iostream>
using namespace std;
int main( )
{
    int i,n;
    cout<<" Enter a number ";
    cin>>n;
    for(i=2; i<=n/2; i++)
        if(n%i==0)
        {
            cout<<"\n not a prime "<<endl;
            return 0;
        }
    cout<<"\n"<<n<<" is a prime number ";
    return 0;
}
```

**Number is palindrome or not**

```
#include<iostream>
using namespace std;
int main( )
{
    int n, t, d, rev=0;
    cout<<" Enter a number ";
    cin>>n;
    t=n;
    while(n!=0)
    {
        d=n%10;
        rev=rev*10+d;
        n=n/10;    }
    cout<<"\n The reverse is "<<rev<<endl;
    if(rev==t)
        cout<<" The number is palindrome ";
    else
        cout<<" The number is not palindrome ";
    return 0;    }
```

**Check whether a number is Armstrong or not  
(like  $1^3 + 5^3 + 3^3 = 153$ , the number itself )**

```
#include<iostream>
using namespace std;
int main( )
{
    int n, t, d, sum=0;
    cout<<" Enter a number ";
    cin>>n;
    t=n;
    while(n!=0)
    {
        d=n%10;
        sum=sum+d*d*d;
        n=n/10;    }
    cout<<"\n The sum of cubes of its digits is "<<sum<<endl;
    if(sum==t)
        cout<<" The number is Armstrong ";
    else
        cout<<" The number is not Armstrong ";
    return 0;    }
```

**Factors of a number**

```
#include<iostream>
using namespace std;
int main( )
{
    int i, n;
    cout<<" Enter a number ";
    cin>>n;
    cout<<"The factors of "<<n<<" are "<<endl;
    for(i=1; i<=n; i++)
    {
        if(n%i==0)
            cout<<i<<"\t";
    }
    return 0;    }
```

**Fibonacci series**

```
#include<iostream>
using namespace std;
int main( )
{
    int n, i, f=1, s=1, t ;
    cout<<"Enter number of terms in the series ";
    cin>>n;
    cout<<f<<" "<<s<<" ";
    for(i=3; i<=n; i++)
    {
        t=f+s;
        cout<<t<<" ";
        f=s;
        s=t;
    }
    return 0;    }
```