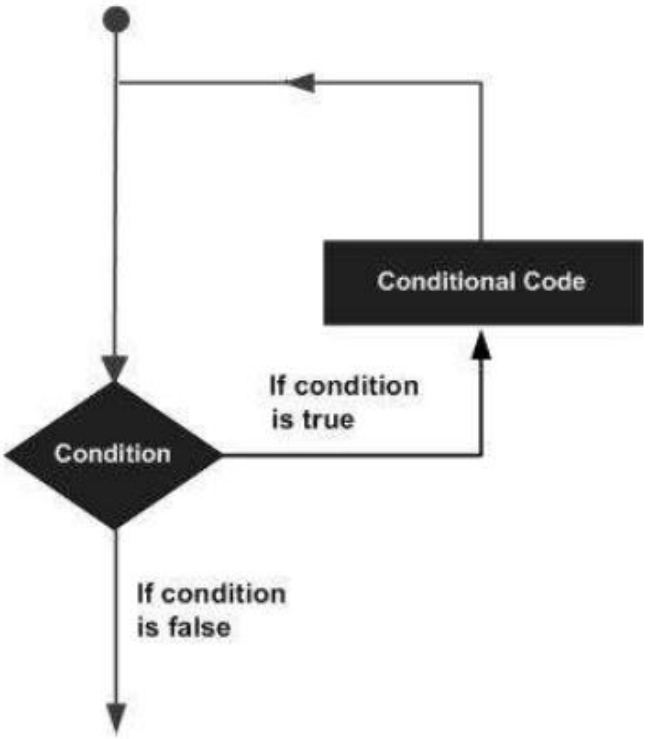
C++ PROGRAMMING LANGUAGE

Repeated Loops

There may be a situation, when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

Loops cause a section of your program to be repeated a certain number of times. The repetition continues while a condition is true. When the condition becomes false, the loop ends and control passes to the statements following the loop.



There are three kinds of loops in C++: the while loop, the do … while loop and for loop

C++ PROGRAMMING LANGUAGE

While Loop:

A **while** loop statement repeatedly executes a target statement as long as a given condition is true.

Syntax

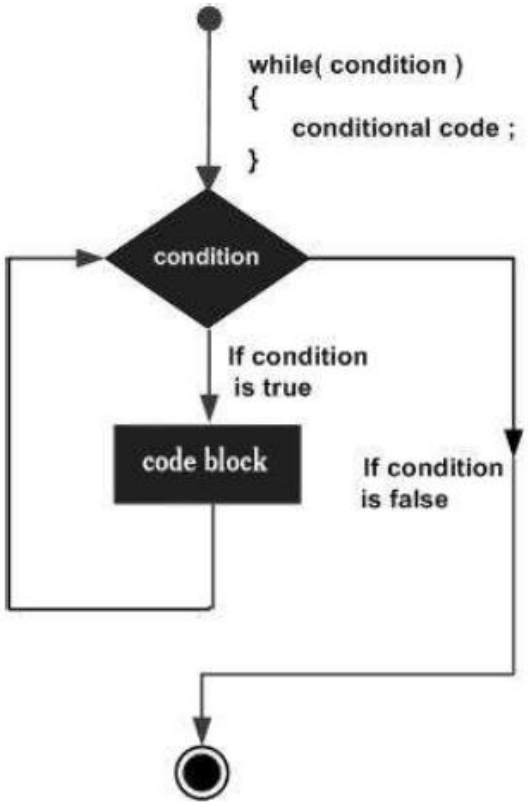
The syntax of while loop in C++ is:

```
while(condition)
{
      statement(s);
}
```

Here, statement(s) may be a single statement or a block of statements. The condition may be any expression, and true is any non-zero value. The loop iterates while the condition is true.
When the condition becomes false, program control passes to the line immediately following the loop.

Note: There is not semicolon (;) after while loop.

Flow Diagram



C++ PROGRAMMING LANGUAGE

Example: Write a program to print the numbers from 1 to 100.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int i=1;   // start of counter

    while(i<=100)    // condition
    {
        cout<<i<<endl;
        i++;          // increment or decrement operator
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Example: Follow the following program and write the output.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    char ch;
    cout<<"\nPress any key (q=exit): ";
    cin>>ch;
    while(ch!='q')
    {
        cout<<"\nPress any key (q=exit): ";
        cin>>ch;
    }
    cout << "GO!" << endl;
     system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

The Infinite Loop

A loop becomes infinite loop if a condition never becomes false.

Example:

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

int main()
{
    char ch;
    cout<<"\nPress any key (q=exit): ";
    cin>>ch;
    while(1)
    {
        cout<<"\nPress any key (q=exit): ";
        cin>>ch;
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

- **The program above will continue implementation indefinitely. To address this, we will use break statement as following:**

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

int main()
{
    char ch;
    cout<<"\nPress any key (q=exit): ";
    cin>>ch;
    while(1)
    {
        if(ch=='q')
            break;
        cout<<"\nPress any key (q=exit): ";
        cin>>ch;
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

- **Instead of repeating Lines below twice in the program**

```
cout<<"\nPress any key (q=exit): ";
cin>>ch;
```

We can improve the program by put the break statement at the end of while loop as following:

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

int main()
{
    char ch;
    while(1)
    {
        cout<<"\nPress any key (q=exit): ";
        cin>>ch;
        if(ch=='q')
            break;
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Do … While Loop

Unlike **while** loop, which test the loop condition at the top of the loop, the **do...while** loop checks its condition at the bottom of the loop.

A **do...while** loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

Syntax

The syntax of a do...while loop in C++ is:

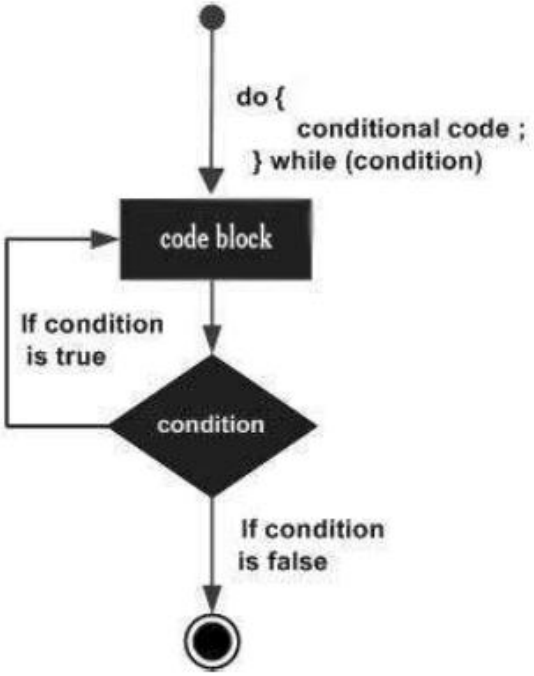
```
do
{
      statement(s);
} while( condition );
```

Note:

- **There is semicolon (;) after while loop.**
- **The conditional expression appears at the end of the loop, so the statement(s) in the loop execute once before the condition is tested.**

C++ PROGRAMMING LANGUAGE

Flow Diagram



Example: Write a program to print the numbers from 1 to 100.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int i=1;   // start of counter

    do
    {
        cout<<i<<endl;
        i++;          // increment or decrement operator
    } while(i<=100);   // condition
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

Example: Follow the following program and write the output.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    char ch;
    do
    {
        cout<<"\nPress any key (q=exit): ";
        cin>>ch;
    } while(ch!='q');
    cout << "GO!" << endl;
     system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

To note the different between two loops above, follow the following programs:

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n=1, p=1;
    while(n<5)
    {
        cout<<"n="<<n<<"
p="<<p<<endl;
        n++;
        p=n*n;
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Output:

```
n=1 p=1
n=2 p=4
n=3 p=9
n=4 p=16
GO!
Press any key to continue . . .
```

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n=1, p=1;
    do
    {
        cout<<"n="<<n<<"
p="<<p<<endl;
        n++;
        p=n*n;
    } while(n<5);
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Output:

```
n=1 p=1
n=2 p=4
n=3 p=9
n=4 p=16
GO!
Press any key to continue . . .
```

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n=1, p=1;
    while(n>5)
    {
        cout<<"n="<<n<<"
p="<<p<<endl;
        n++;
        p=n*n;
    }
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Output:

```
GO!
Press any key to continue . . .
```

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n=1, p=1;
    do
    {
        cout<<"n="<<n<<"
p="<<p<<endl;
        n++;
        p=n*n;
    }while(n>5);
    cout << "GO!" << endl;
    system("pause");
    return 0;
}
```

Output:

```
n=1 p=1
GO!
Press any key to continue . . .
```

C++ PROGRAMMING LANGUAGE

Examples:

1- Write a program to calculate the summation of n of values.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n, k=1;
    float sum=0,x;
    cout<<"\nHow many values: ";
    cin>>n;
    cout<<"\nEnter the values :";
    while(k<=n)
    {
        cin>>x;
        sum=sum+x;
        k++;
    }
    cout<<"\nThe summation of values= "<<sum<<endl;
    system("pause");
    return 0;
}
```

2- Write a program to find the average of a student's grades.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n, k=1;
    float sum=0,grade,average;
    cout<<"\nHow many subject: ";
    cin>>n;
    cout<<"\nEnter the grades: ";
    do
    {
       cin>>grade;
       sum=sum+grade;
       k++;
    }while(k<=n);
    average=sum/n;
    cout<<"\nThe Average of grades= "<<average<<endl;
    system("pause");
    return 0;
}
```

(H.W) 3- Rewrite a the previous program by using (while loop)

C++ PROGRAMMING LANGUAGE

4- Write a program to find the maximum value of n values.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int n, k=1;
    float val,maxval;
    cout<<"\nHow many values: ";
    cin>>n;
    cout<<"\nEnter the values: ";
    cin>>val;
    maxval=val;
    while(k<=n-1)
    {
        cin>>val;
        if(val>maxval)
            maxval=val;
        k++;
    }
    cout<<"\nThe maximum value is:"<<maxval<<endl;
    system("pause");
    return 0;
}
```

5- Write a program to find the summation of following series:

$$1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\ldots\ldots+\frac{1}{2^n}$$

References:
- Object-Oriented Programming in C++, Fourth Edition
- Tutorials Point https://www.tutorialspoint.com/cplusplus/