

## Loops

One of the very powerful control structures is Repetition Statements in C++. Repetition statements allow to repeat a block of code until a certain condition is **true**. Repetition statements are commonly referred as loops and they can be implemented in the following ways

- i. while
- ii. for
- iii. do while

In this lesson, repetition statements i & ii are discussed. (Students can learn iii by themselves)

### while Loop

while loop is implemented using the following style.

```
while (rel exp)
{
    // statement 1
    // statement 2
    // statement n
    // one statement must
    // cause rel exp to
    //become false
}
```

Statements written inside the body of the while loop will continue to repeat until the rel exp becomes false. There must be a statement inside the body that will eventually make the rel exp false. Otherwise the loop will run infinite times. This is explained in the following codes.

les\_05\_code\_01.cpp

```
1. #include<iostream>
2. using namespace std;

3. int main (void)
4. {
5.     int count = 1;
6.     while (count <= 10)
7.     {
8.         cout<<"Hello world!\n";
9.         ++count;
10.    }
11.    return 0;
12. }
```

Output

```
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!
```

les\_05\_code\_02.cpp

```
// infinite while loop  
// objective of this code is to introduce the idea of infinite loop  
  
1. #include<iostream>  
2. using namespace std;  
  
3. int main()  
4.     {  
5.         int count = 0;  
6.         while(count < 10)  
7.             {  
8.                 cout<<"Hello World!\n";  
9.             }  
9.         return 0;  
10.    }
```

Generally, a while loop contains the following components

- i. Loop control variable: A variable mostly inside the relational expression.
- ii. Relational Expression
- iii. Body (Multiple statements)
- iv. A statement that makes the relational expression false

Based on the way, the statement which makes the relational expression false, while loop can be categorized into two types.

- i. Count Controlled
- ii. Sentinel Controlled

If the relational expression becomes false after a certain count is achieved, the while loop is count controlled. les\_05\_code\_01.cpp is the example of count controlled while loop. Let's see another example of count controlled while loop implementation.

les\_05\_code\_03.cpp

```
// count controlled while loop
// interest calculation

1. #include<iostream>
2. using namespace std;

3. int main()
4. {
5.     double balance, rate;
6.     int years, count;
7.     cout<<"What is the starting balance : ";
8.     cin>>balance;
9.     cout<<"What is the annual interest rate : ";
10.    cin>>rate;
11.    cout<<"How many years to calculate interest : ";
12.    cin>>years;
13.    count = 1;
14.    while(count <= years)
15.    {
16.        balance *= rate;
17.        ++count;
18.    }
19.    cout<<"After "<<years<<" years, balance will be "
20.    <<balance<<endl;
21.    return 0;
22. }
```

Output

```
What is the starting balance : 10000
What is the annual interest rate : 1.15
How many years to calculate interest : 10
After 10 years, balance will be 40455.6
```

les\_05\_code\_04.cpp

```
// count controlled while loop
// interest calculation modified

1. #include<iostream>
2. using namespace std;

3. int main()
```

```
4.  {
5.  double balance, rate;
6.  int years, count;
7.  cout<<"What is the starting balance : ";
8.  cin>>balance;
9.  cout<<"What is the annual interest rate : ";
10. cin>>rate;
11. cout<<"How many years to calculate interest : ";
12. cin>>years;
13. count = 1;
14. while(count <= years)
15.     {
16.         balance *= rate;
17.         cout<<count<<" : "<<balance<<endl;
18.         ++count;
19.     }
20. cout<<"After "<<years<<" years, balance will be "
21. <<balance<<endl;
22. return 0;
23. }
```

Output

```
What is the starting balance : 10000
What is the annual interest rate : 1.15
How many years to calculate interest : 10
1 : 11500
2 : 13225
3 : 15208.7
4 : 17490.1
5 : 20113.6
6 : 23130.6
7 : 26600.2
8 : 30590.2
9 : 35178.8
10 : 40455.6
After 10 years, balance will be 40455.6
```

### Sentinel Controlled

If the relational expression is controlled on the basis of a certain input, the while loop is termed as sentinel control. See the example

// les\_05\_code\_05.cpp

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int grade, gradeTotal, numGrades;
6.     numGrades = gradeTotal = 0;
7.     double average;
8.     cout<<"Enter grade or -1 to quit : ";
9.     cin>>grade;
10.
11.    while(grade != -1)
12.    {
13.        gradeTotal += grade;
14.        ++numGrades;
15.        cout<<"Enter grade or -1 to quit : ";
16.        cin>>grade;
17.    }
18.
19.    average = (float)gradeTotal/numGrades;
20.    cout<<"Average grade is : "<<average;
21.    return 0;
22. }
```

Output

```
Enter grade or -1 to quit : 91
Enter grade or -1 to quit : 67
Enter grade or -1 to quit : 55
Enter grade or -1 to quit : 43
Enter grade or -1 to quit : 78
Enter grade or -1 to quit : 85
Enter grade or -1 to quit : -1
Average grade is : 69.8333
```

Write a program that take two numbers from user and multiply them without using \* (multiplication) operator.

Solution is in les\_05\_code\_06.cpp

## for Loop

One another way to implement the repetition statements in C++ is to use for loop. For loop is always count controlled. The following code will help you learn the syntax and idea of for loop. (Note it is possible to use while instead of for and for instead of while in many cases, so as a student you should learn both and practice enough to recognize when to use what)

les\_05\_code\_07.cpp

```
// for loop
// syntax

1.  #include<iostream>
2.  using namespace std;

3.  int main()
4.  {
5.      for(int i = 1; i<11; i++)
6.      {
7.          cout<<"Hello World\n";
8.      }
9.      return 0;
10. }
```

Output

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

les\_05\_code\_08.cpp

```
// for loop with manual increment

1. #include<iostream>
2. using namespace std;

3. int main()
```

```
4.    {
5.    for(int i = 1; i<11; i+=2)
6.    {
7.        cout<<i<<" Hello World\n";
8.    }
9.    return 0;
10.   }
```

Output

```
1 Hello World
3 Hello World
5 Hello World
7 Hello World
9 Hello World
```

les\_05\_code\_09.cpp

```
// interest calculation with for loop
```

```
1. #include<iostream>
2. using namespace std;

3. int main()
4.    {
5.        double balance, rate;
6.        int years;
7.        cout<<"What is the starting balance : ";
8.        cin>>balance;
9.        cout<<"What is the annual interest rate : ";
10.       cin>>rate;
11.       cout<<"How many years to calculate interest : ";
12.       cin>>years;
13.       for(int i = 1; i<=years; ++i)
14.       {
15.           balance *= rate;
16.       }
17.       cout<<"After "<<years<<" years, balance will be "
18.       <<balance<<endl;
19.       return 0;
20.   }
```

Output

What is the starting balance : 10000  
What is the annual interest rate : 1.15  
How many years to calculate interest : 10  
After 10 years, balance will be 40455.6

## Nested for Loop

les\_05\_code\_10.cpp

```
// nested for loop

1. #include<iostream>
2. using namespace std;

3. int main()
4. {
5.     char star = '*';
6.     for (int i = 0; i < 5; ++i)
7.     {
8.         for(int j = 0; j < 5; ++j)
9.         {
10.            cout<<star;
11.        }
12.        cout<<endl;
13.    }
14.    return 0;
15. }
```

Output

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

les\_05\_code\_11.cpp

```
// nested for loop

1. #include<iostream>
2. using namespace std;
```



```
3. int main()
4. {
5.     char star = '*';
6.     for (int i = 0; i < 5; ++i)
7.     {
8.         for(int j = 0; j <= i; ++j)
9.         {
10.            cout<<star;
11.        }
12.        cout<<endl;
13.    }

14.    return 0;
15. }
```

Output

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

les\_05\_code\_12 (UIY Understand it yourself)

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     char star = '*';
6.     for (int i = 0; i < 5; ++i)
7.     {
8.         for(int j = 0; j <= i; ++j)
9.         {
10.            cout<<star;
11.        }
12.        cout<<endl;
13.    }

14.    for (int i = 5; i > 0; --i)
15.    {
16.        for(int j = i; j > 0; --j)
17.        {
18.            cout<<star;
19.        }
```

```

20.         cout<<endl;
21.     }
22.     return 0;
23. }

```

Output

```

*
**
***
****
*****
*****
*****
****
***
**
*

```

les\_05\_code\_13.cpp

```

1. #include<iostream>
2. using namespace std;

3. int main()
4. {
5.     for(int row = 1; row <= 5; ++row)
6.     {
7.         for(int col = 1; col <= 5; ++col)
8.         {
9.             cout<<row<<" * "<<col<<" = "
10.            <<row * col<<"\t";
11.         }
12.         cout<<endl;
13.     }
14. }

```

Output

1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25