

Chapter 5 - Control Flow –Loops

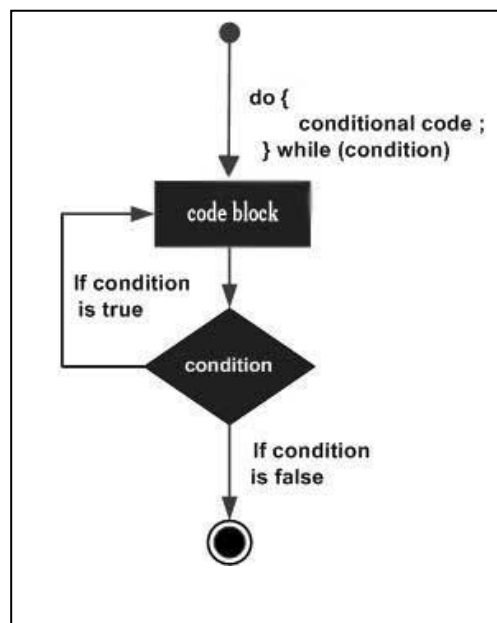
do . . while loop

- the do-while loop has the test expr at the end (post test):

```
do  
    statement  
while (expression);
```

- statement is executed
- expression is evaluated.
 - if true, repeat.
 - if/when false, exit the loop.
- statement always executes at least once.

Flow Diagram



do .. while example

Example:

```
int number = 1;
do
{
    cout << "Student Number " << number << endl;
    number++;
} while (number <= 3);

cout << "Done" << endl;
```

Output:

```
Student Number  1
Student Number  2
Student Number  3
Done
```

Watch out

What is output?

```
int number = 10;
do
{
    cout << "Student Number " << number << endl;
    number++;
} while (number <= 3);

cout << "Done" << endl;
```

- The body (statement) of the do-while is **ALWAYS** executed at least once, even if the test expression is false from the beginning.

do-while for asking user to repeat

```
double cel, fahr;
char repeat;
do {
    cout << "Enter the temp in Celsius: ";
    cin >> cel;
    fahr = 9.0/5.0*cel + 32;
    cout << "Fahrenheit: " << fahr << endl;
    cout << "Do you want to convert another temperature (Y/N)? ";
    cin >> repeat;
} while(repeat == 'Y' || repeat == 'y');
```

Output:

```
Enter the temp in Celsius: 0
Fahrenheit: 32
Do you want to convert another temperature (Y/N)? y
Enter the temp in Celsius: 100
Fahrenheit: 212
Do you want to convert another temperature (Y/N)? N
```

do-while with menus

```
char choice;

do {
    cout << "\nA: Make a reservation." << endl;
    cout << "B: View flight status." << endl;
    cout << "C: Check-in for a flight." << endl;
    cout << "D: Quit the program." << endl;
    cout << "\nEnter your choice: ";
    cin >> choice;
    switch (choice) {

        case 'A': // code to make a reservation
            break;
        case 'B': // code to view flight status
            break;
        case 'C': // code to process check-in
            break;
        case 'D': break ;
        default : cout << "\nInvalid Choice \n" ;

    }

} while(choice != 'D');
```

```
// rest of the code
```

Breaking out of a loop

- Sometimes we want to abort a loop before it has completed.
- The **break** statement can be used to terminate the loop from within.

```
cout << "guess a number between 1 and 10 " ;
int number;
while (true) {
    cin >> number;
    if (number == 8)
        break;
    cout << "again , guess a number between 1 and 10 " ;
}

cout << "\nOut of the loop - Done " << endl;
```

Don't do this. It makes your code hard to read and debug.

Sample run

```
guess a number between 1 and 10 2
again , guess a number between 1 and 10 3
again , guess a number between 1 and 10 10
again , guess a number between 1 and 10 5
again , guess a number between 1 and 10 8
```

Out of the loop - Done

Stopping an iteration

- Sometimes want to abort an iteration before it is done.
- The **continue** statement can be used to **terminate** the **current** iteration:

```
for (int i=1; i <= 10; i++) {  
    if (i == 4 || i == 7 )  
        continue;  
    cout << i << " ";  
}  
  
cout << "\nOut of the loop - Done " ;
```

Output:

1 2 3 5 6 8 9 10

Out of the loop - Done

Don't do this either. It makes your code hard to read and debug.

Rewrite the following code segment using **while** and **do .. while**:-

```

cout << "How many students?";
cin >> numStudents;
cout << "How many test scores? ";
cin >> numTests;
for (int student=1; student <= numStudents; student++) {
    float total = 0, score;
    cout << "Enter the " << numTests
    << " test scores for student " << student << endl;
    for (int test=1; test <= numTests; test++) {
        cin >> score;
        total = total + score;
    }
    float avgScore = total/numTests;
    cout << "Average for student" << student
    << " is: " << avgScore << endl;
}

```

Rewrite the following code segment using **for** and **do .. while**:-

```

int count = 10;
while (count > 0)
{
    cout << count << endl;
    count--;
}
cout << "Liftoff! Godspeed, John Glenn!" << endl;

```

Rewrite the following code segment using **for** and **while**:-

```

int counter;
cout << "How many hellos? ";
cin >> counter;
do
{
    cout << "Hello\n";
    counter--;
} while (counter >0 );
cout << "Counter is: " << counter << endl;

```

What is the exact output of the following program :

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

int main() {
    int v = 0;
    do
        cout << v++ << " ";
    while (v < 5);

    cout << endl<<endl<<endl;

    int count = 0, funny = 1, serious = 0, limit = 4;
    do {
        funny++;
        serious += 2;
    } while (count++ < limit);
    cout << funny << "\n"
        << serious << "\n";
    cout << count << endl;
    return 0;
}
```