do-while loop in C++ with example

as the loop name is – **do-while** meaning "**First do it, then check"**.

C++ do-while loop

a do-while loop is different from the other loops (for loop, while loop), initially, the variable is initialized, then the **body of loop** is executed, then the condition is checked.

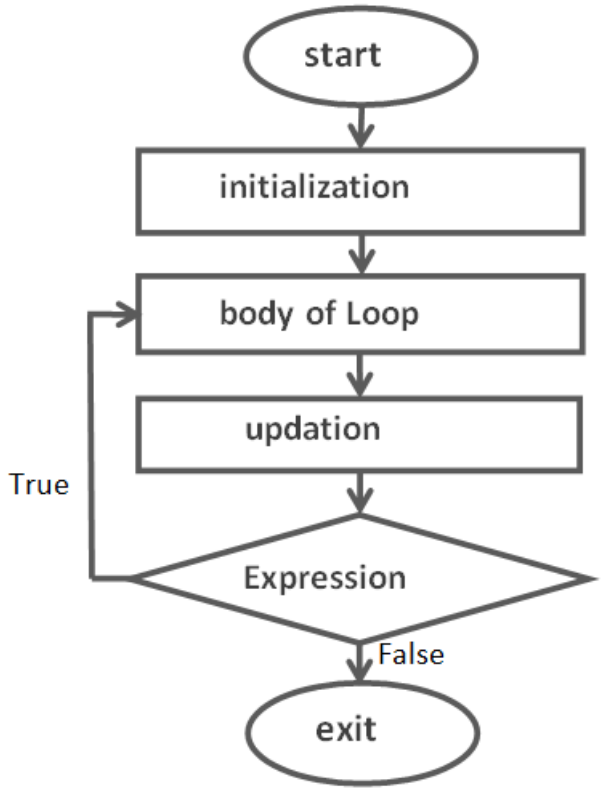
a do-while loop is always executed whether the condition is true or false whereas the other two The first condition is checked in the loop and then the **body of loop** is executed.

The syntax is given below.

```
initialization;

 do {

     body of loop;

     updation;

  }

    while (expression);
```

flow-diagram of a do-while loop will as follows,

do-while loop flow diagram in C++



Example of do-while loop in C++

In the above program, the condition is given that **loop-body** only execute when user-given value less than 5.

```
#include<iostream.h>

using namespace std;



int main()

 {
```

```
   int num;



   cout<<"Enter number less than 5:";

   cin>>num;



   int x=num; //5

   do{

       cout<<x<<".statement executed\n"; // body of loop

       x++;

    }

      while(x<5);

  return 0;

 }
```

OUTPUT

Here we will execute the above program 2 times, in the **1st execution** user given value is **num = 1**; then, the output will be as

```
Enter number: 1

1.statement executed

2.statement executed

3.statement executed

4.statement executed
```

because condition becomes True (1<5) so it will execute 4th times as we can see above, Program will execute as follows,

body of loop	value increased by 1 , x++;	Expression/condition
1.statement executed	2	2 < 5 (True)
2.statement executed	3	3 < 5 (True)
3.statement executed	4	4 < 5 (True)
4.statement executed	5	5 < 5 (False)

but now in the **2nd Execution**, this time user given value is **num = 6;** then,

```
6.statement executed
```

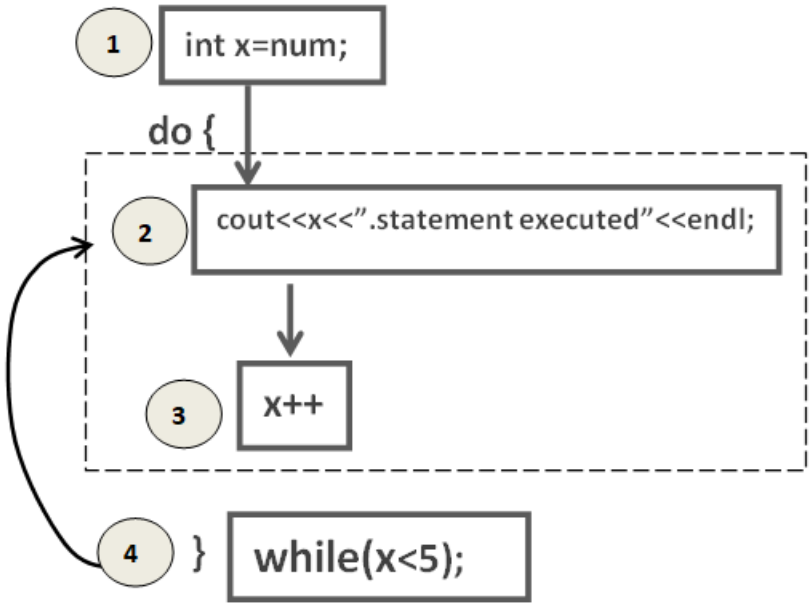
this times condition was becoming False (6<5) but **body of loop** still executed, this will execute follows,

body of loop	increment in initialization value x = 6	Expression/condition
6.statement executed	6	6 < 5 (False)

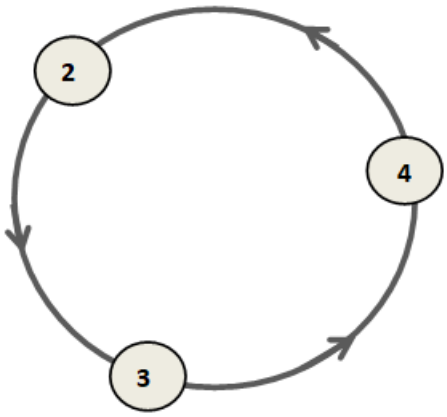
Why did this happen?

Because in the do-while loop first value, **initialized** then **body of loop** executed and in the **last condition** is checked.

a do-while loop will work as follows,



repeatition of the above program will as follows,



Difference between while and do-while loop in c++

In the first while-loop, the first condition is checked, then the body of loop is executed, then the initialize value of the loop is incremented, whereas, in the do-while loop, the first body of loop is executed, then the condition is checked Then there is an increment in the value in the Last.

in the while-loop

initialization – condition – body of loop – increment

in the do-while loop

initialization – body of loop – increment – condition

do-while loop with switch statement in C++

as we know in the do-while loop first execution will be done, then after condition will be checked. creating a menu wizard is a good example of the do-while loop.

In the below Program, First Do, meaning that first display the menu then check the condition if the condition becomes true then execute.

We have conditioned that if the value given by the user matches the serial no[1,2,3,0] of the options so only that option is executed.

```
#include<iostream>

#include<stdlib>  //exit()

using namespace std;

int main()

  {

    int select;

    int num1,num2,result;



 do {
```

```
cout<<"1.Sum \n";

cout<<"2.Subtraction\n";

cout<<"3.Multiply\n";

cout<<"0.Exit";



cout<<"\nEnter Your Choice: ";

cin>>select;



switch(select)

 {

     case 1:

              cout<<"\nEnter Two Number: ";

              cin>>num1>>num2;

              result = num1+num2;

              cout<<"\nTotal: "<<result;

              getch();

              break;

    case 2:

              cout<<"\nEnter Two Number: ";
```

```
                cin>>num1>>num2;

                result = num1-num2;

                cout<<"\nSubstraction: "<<result;

                getch();

                break;

       case 3:

                cout<<"\nEnter Two Number: ";

                cin>>num1>>num2;

                result = num1*num2;

                cout<<"\nMultiply: "<<result;

                getch();

                break;

       case 0:

                exit(0);



        default: cout<<"\ninvalid selection....";

                return 0;

     }

  }while(select!=0);

}
```

OUTPUT

```
1.Sum

2.Subtraction

3.Multiply

0.Exit



Enter Your Choice:
```

The program will execute until the condition becomes false (select==0), note that the `exit()` statement will terminate the program while **do-while** loop will display the menu until the user does not select the Exit option.

what will happen if the do-while loop is not used there, the answer is- Program will execute only a single option so there is no need to `exit()` statement because, after a single execution, the program will terminate automatically.

here is the output, of without using do-while loop

Note: always avoid to write the same statement multiple times int program, so here in the above program, will not be considered as good programming because of the following statement

```
cout<<"Enter number:";

cin>>num1>>num2;
```

which is repeating in every case statement, use function with switch statement in C++ to avoid this.

Related Exercise

- Printing only even number using do-while loop
- a simple menu demonstration of two number operation using switch and do – while loop