CL1002 – Programming Fundamentals Lab

Lab # 09

While & do while

Instructor: Engr. Muhammad Usman

Email: usman.rafiq@nu.edu.pk

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar

C break statement

The break is a keyword in C which is used to bring the program control out of the loop. The break statement is used inside loops or switch statement.

Example 1

```
#include<stdio.h>
int main ()
{
   int i;
   for(i = 0; i<10; i++)
   {
       printf("%d ",i);
       if(i == 5)
       break;
   }
   printf("came outside of loop i = %d",i);

}
```

Output

```
0 1 2 3 4 5 came outside of loop i = 5
```

C continue statement

The continue statement in C language is used to bring the program control to the beginning of the loop. The continue statement skips some lines of code inside the loop and continues with the next iteration. It is mainly used for a condition so that we can skip some code for a particular condition.

Example 2

```
#include<stdio.h>
int main(){
int i=1;
//starting a loop from 1 to 10
for(i=1;i<=10;i++){
 if(i==5){
```

```
 //if value of i is equal to 5, it will continue the loop
 continue;
}
printf("%d \n",i);
}//end of for loop
return 0;
}
```

Output

```
1

2

3

4

6

7

8

9

10
```

while loop

While loop is also known as a pre-tested loop.In general, a while loop allows a part of the code to be executed multiple times depending upon a given boolean condition. The while loop is mostly used in the case where the number of iterations is not known in advance.

The syntax of the while loop is:

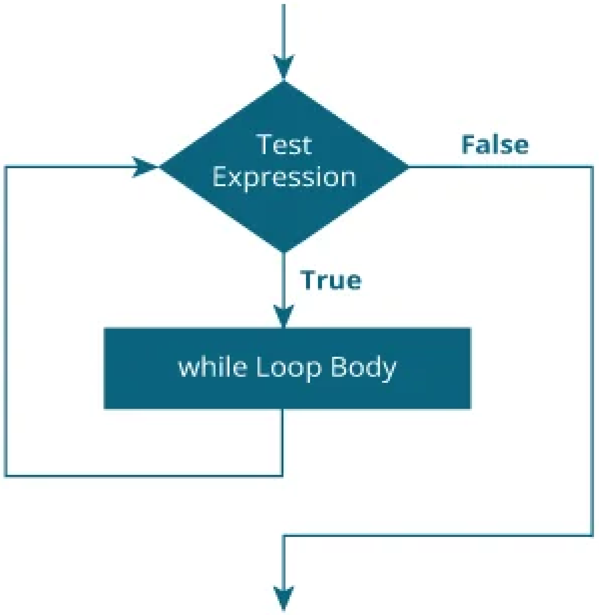
```
while (testExpression) {

 // the body of the loop

}
```

How while loop works?

- The while loop evaluates the testExpression inside the parentheses ().
- If testExpression is true, statements inside the body of while loop are executed. Then, testExpression is evaluated again.

- The process goes on until testExpression is evaluated to false.
- If testExpression is false, the loop terminates (ends).

Flowchart of while loop



Example 3

```
// Print numbers from 1 to 5
#include <stdio.h>
int main() {
 int i = 1;

 while (i <= 5) {
   printf("%d\n", i);
   ++i;
 }
 return 0;
}
```

Output

```
1 2 3 4 5
```

Here, we have initialized i to 1.

1. When i = 1, the test expression i <= 5 is true. Hence, the body of the while loop is executed. This prints 1 on the screen and the value of i is increased to 2.
2. Now, i = 2, the test expression i <= 5 is again true. The body of the while loop is executed again. This prints 2 on the screen and the value of i is increased to 3.
3. This process goes on until i becomes 6. Then, the test expression i <= 5 will be false and the loop terminates.

do...while loop

The do while loop is a post tested loop. Using the do-while loop, we can repeat the execution of several parts of the statements. The do-while loop is mainly used in the case where we need to execute the loop at least once. The do-while loop is mostly used in menu-driven programs where the termination condition depends upon the end user.

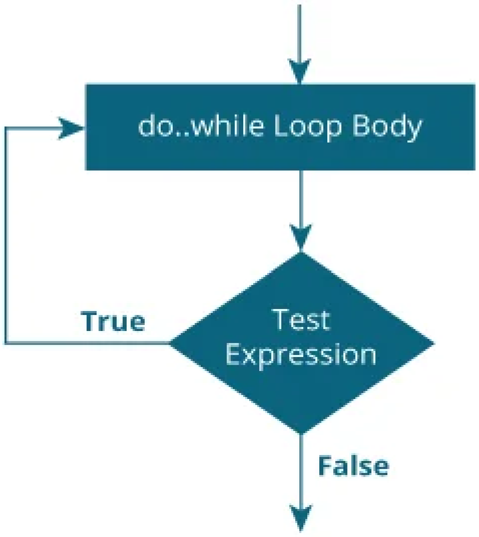
The syntax of the do...while loop is:

```
do {
 // the body of the loop
}
while (testExpression);
```

How do...while loop works?

- The body of *do...while* loop is executed once. Only then, the testExpression is evaluated.
- If testExpression is true, the body of the loop is executed again and testExpression is evaluated once more.
- This process goes on until testExpression becomes false.
- If testExpression is false, the loop ends.

Flowchart of do...while Loop



Example 4

```
#include <stdio.h>
int main()
{
   int i = 0;
do {
 printf("%d ",i);
 i++;
}
while (i < 5);

   return 0;
}
```

Output

```
0 1 2 3 4
```

Example 5

```c
#include <stdio.h>
int main()
{
  char exit;
  do
{
   printf("In Loop\n");
   printf("Exit Loop (y/n)");
   scanf("%c",&exit);
}
while(exit!='y');

return 0;
}
```

Output

```
In Loop
Exit Loop (y/n) n
In Loop
Exit Loop (y/n)In Loop
Exit Loop (y/n) y
```

References:

https://www.programiz.com/c-programming/c-do-while-loops

https://www.javatpoint.com/c-break