Scientific and Technical Computing: SMR3821

Introduction to Scientific computing and Linux - Introduction to C Programming: Decisions and Iterations

Elliot S. Menkah, Ph.D

National Institute for Mathematical Sciences, Ghana.
Kwame Nkrumah University of Science and Technology, Ghana.
ICTP, Trieste.

November 16, 2022

Introduction to Scientific Computing - C programming - Decisions & Iterations

This materials is part of materials used at the National institute for Mathematical Sciences, Ghana, and have been contributed to over time by the follow authors and instructors:

1. Elliot Menkah, Ph.D
2. Peter Amoako-Yirenkyi, Ph.D
3. Ernest Ofori-Yirenkyi
4. Kwesi Smith
5. Shirley A. Akasreku

Iterations

National Institute for Mathematical Sciences Ghana

Outline

where are we?

National Institute for Mathematical Sciences Ghana

and Operators

$++$ is the increment operator.
Increment operator increases the value of a variable by 1.

$--$ is the decrement operator.
Decrement operator decreases the value of a variable by 1.

$x++$ is equivalent to x = x + 1;
$y--$ is equivalent to y = y - 1;

National Institute for Mathematical Sciences Ghana

E.g.: and Operators

```c
#include <stdio.h>

int main()
{
        int i=10;
        i++;
        printf("\n%d\n",i);

        float k=23.7;
        k++;
        printf("\n%f\n",k);

        char m='C';
        m--;
        printf("\n%c\n",m);

}
```

National Institute for Mathematical Sciences Ghana

In Pre-Increment: variable increment by 1 before assignement

Pre-Increment

```
1 y=++x;
```

Equivalent lines of code

```
1 x=x+1;
2 y=x;
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5         int x=5;
6         int y=10;
7
8         printf("\nBefore y=++x:   x=%d and y=%d\n",x,y);
9         y=++x;
10        printf("\nAfter  y=++x:   x=%d and y=%d\n",x,y);
11 }
```

National Institute for Mathematical Sciences Ghana

In Pre-Decrement: variable decrement by 1 before assignment.

Pre-Decrement

```
1 y=--x;
```

Equivalent lines of code

```
1 x=x-1;
2 y=x;
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5         int x=5;
6         int y=10;
7
8         printf("\nBefore y=--x:  x=%d and y=%d\n",x,y);
9         y=--x;
10        printf("\nAfter  y=--x:  x=%d and y=%d\n",x,y);
11 }
```

National Institute for Mathematical Sciences Ghana

In Post-increment: variable assignemnt before increment by 1.

Post-Increment

```
1 y=x++;
```

Equivalent lines of code

```
1 y=x;
2 x=x+1;
```

```
 1 #include<stdio.h>
 2
 3 int main()
 4 {
 5         int x=5;
 6         int y=10;
 7
 8         printf("\nBefore y=x++:  x=%d and y=%d\n",x,y);
 9         y=x++;
10         printf("\nAfter  y=x++:  x=%d and y=%d\n",x,y);
11 }
```

National Institute for Mathematical Sciences Ghana

In Post-Decrement: variable assignment before decement by 1.

Post-Decrement

```
1 y=x--;
```

Equivalent lines of code

```
1 y=x;
2 x=x-1;
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5         int x=5;
6         int y=10;
7
8         printf("\nBefore y=x--:  x=%d and y=%d\n",x,y);
9         y=--x;
10        printf("\nAfter  y=x--:  x=%d and y=%d\n",x,y);
11 }
```

where are we?

National Institute for Mathematical Sciences Ghana

Compound Statement

+= e.g.: x+=10; $\Longleftrightarrow$ x=x+10;

-= e.g.: x-=10; $\Longleftrightarrow$ x=x-10;

= e.g.: x=10; $\Longleftrightarrow$ x=x*10;

/= e.g.: x/=10; $\Longleftrightarrow$ x=x/10;

%= e.g.: x%=10; $\Longleftrightarrow$ x=x%10;

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

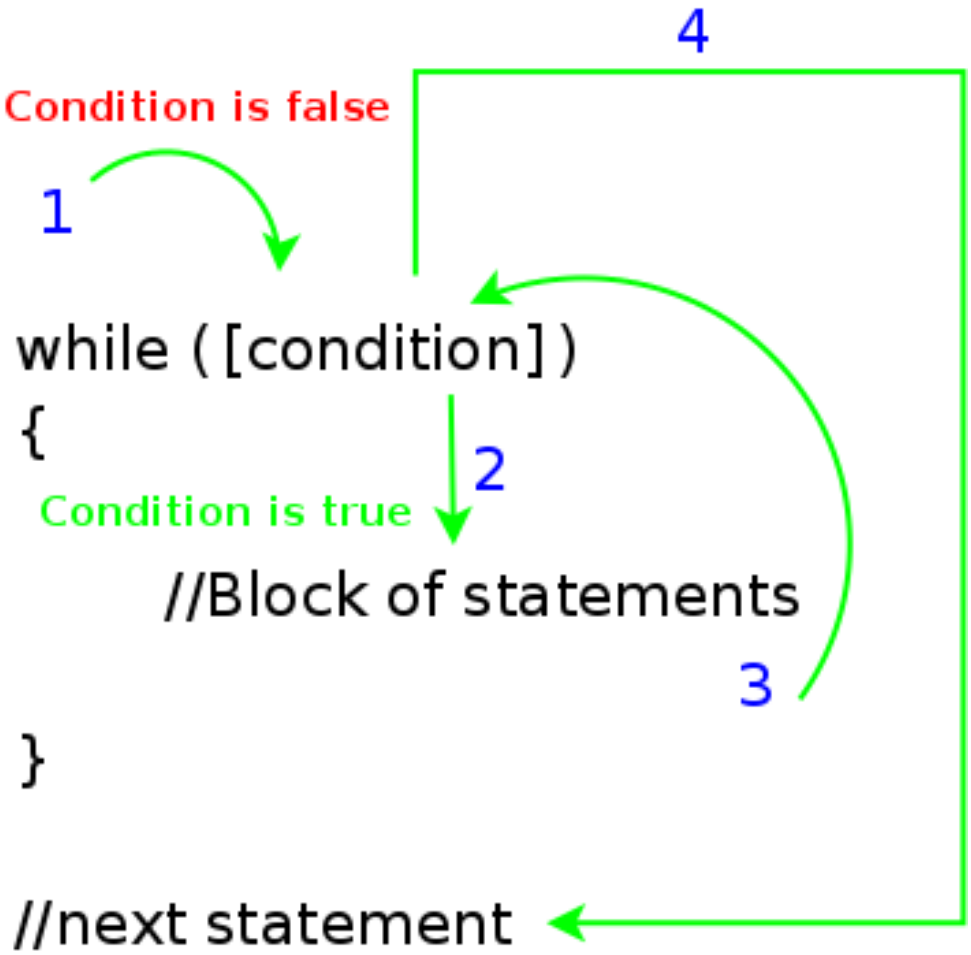
Syntax:

```
1 while (condition)
2 {
3       [statement_1;
4        statement_2;
5        statement_k;]
6 }
```

1. Condition is tested first before body of loop is executed.
2. Iteration continues as long as the conditon remains true.
3. At least one of the statements in the body must eventually cause the loop condition to turn false or force a break out from the body.

National Institute for Mathematical Sciences Ghana

Logical flow of control in While Loop



National Institute for Mathematical Sciences Ghana

E.g.:

```
1 #include<stdio.h>
2 int main()
3 {
4         int i=0;
5         while(i<10)
6         {
7                 i++;
8                 printf("i = %d\n",i);
9         }
10 }
```

National Institute for Mathematical Sciences Ghana

Syntax:

```
1 Do
2 {
3         [statement_1;
4          statement_2;
5          statement_k;]
6 }while (condition);
```

1. Body of code is executed at least once before condition is tested.
2. Iteration continues as long as the conditon remains true.
3. At least one of the statements in the body must eventually cause the loop condition to turn false or force a break out from the body.
4. The Do-While loop structure must be terminated!

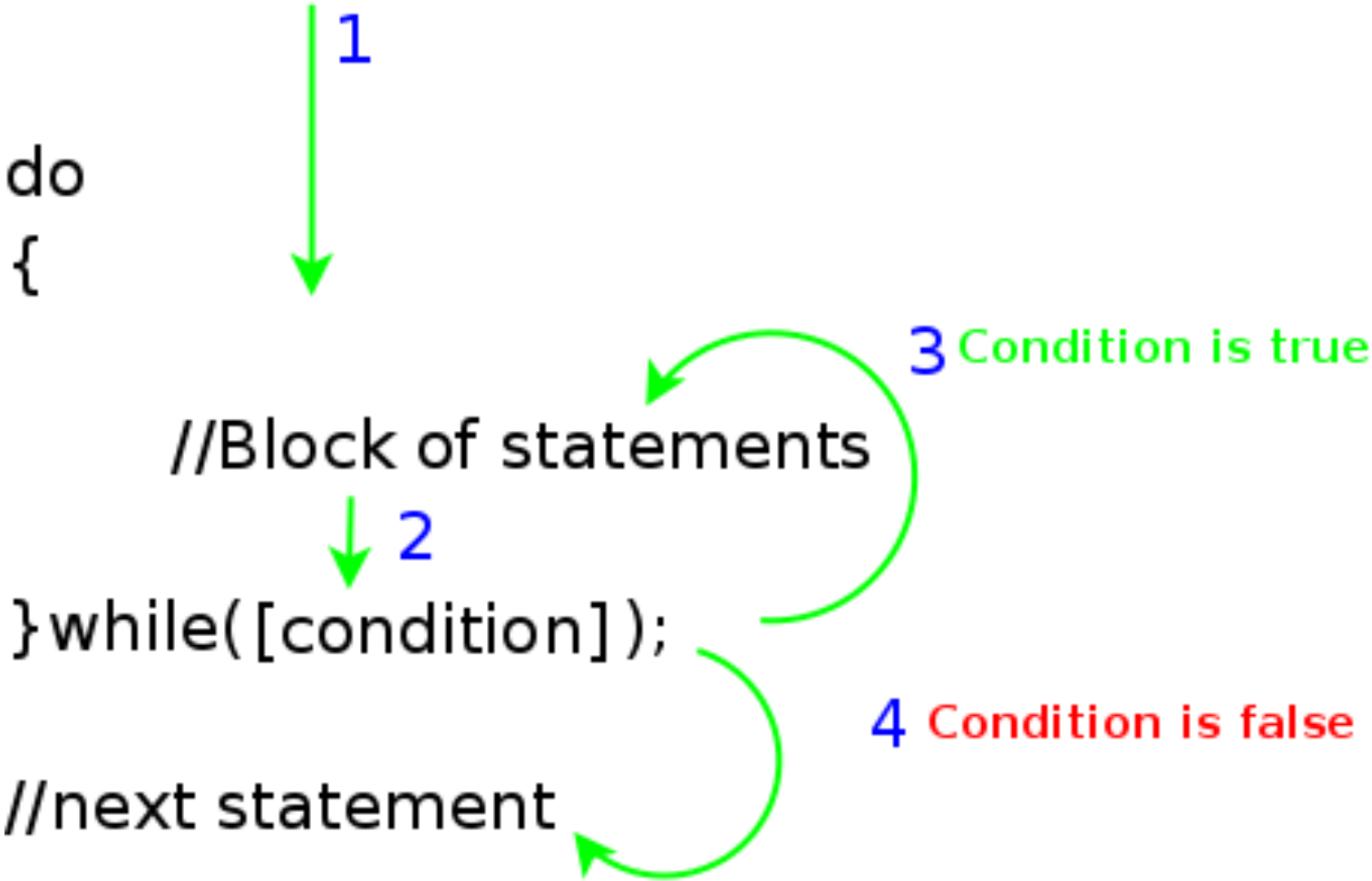
National Institute for Mathematical Sciences Ghana

E.g.:

```
1 #include<stdio.h>
2 int main()
3 {
4         int i=0;
5         do
6         {
7                 i++;
8                 printf("i = %d\n",i);
9         }while(i<10);
10 }
```

National Institute for Mathematical Sciences Ghana

Logical flow of control in Do-While Loop



National Institute for Mathematical Sciences Ghana

Syntax:

```
1  for([initializer];[condition];[iterator])
2  {
3        [statement_1;
4         statement_2;
5         statement_k;]
6  }
```

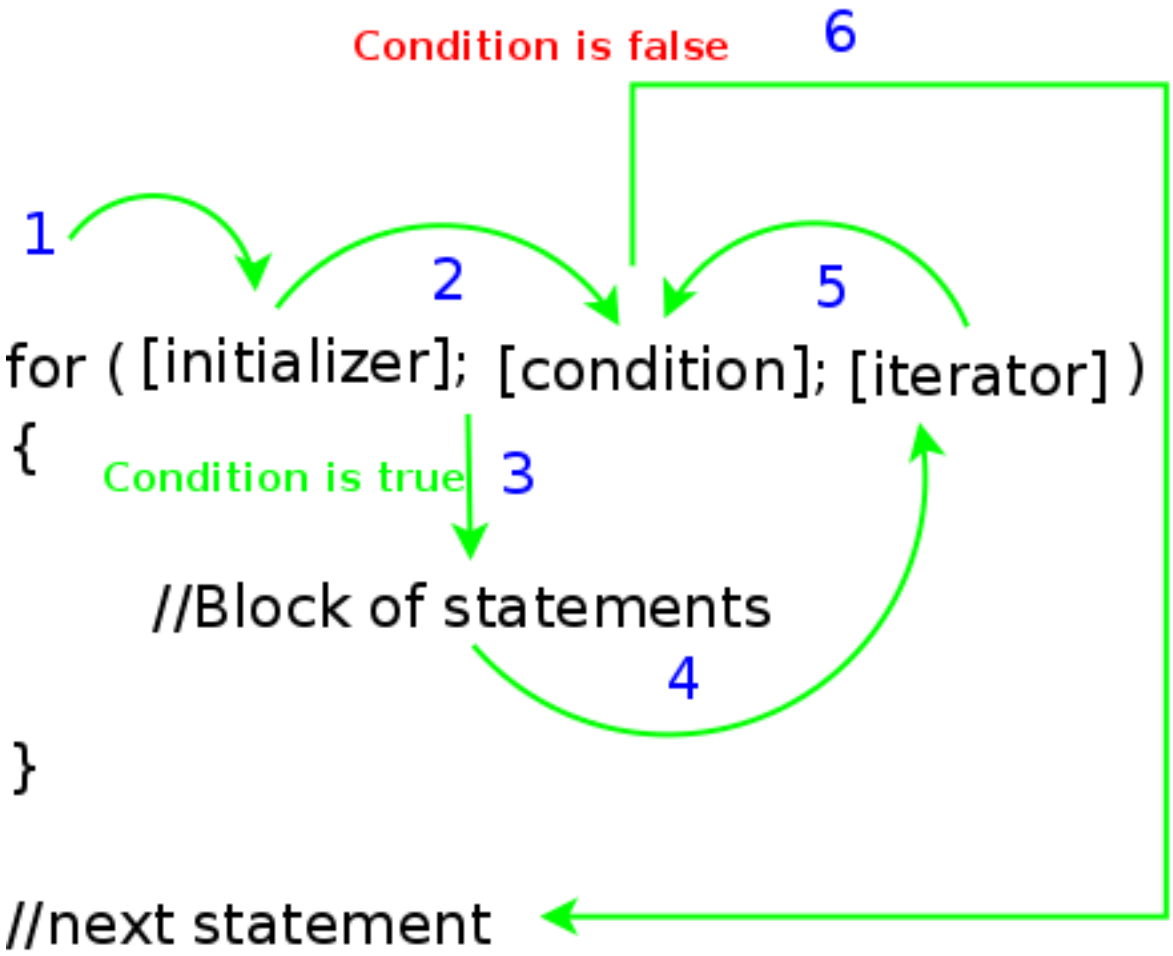
INITIALIZER is/are expression(s) evaluated prior to the execution of the loop. It is primarily used to initialize the loop *counter*.

CONDITION is the expression that is checked before the commencement of each new iteration. As long as the condition remains true, the body of the for-loops is executed.

ITERATOR is an expression that is evaluated after each iteration. The iterator is primarily used to increase / decrease the loop counter.

National Institute for Mathematical Sciences Ghana

Logical flow of control in For Loop



National Institute for Mathematical Sciences Ghana

E.g.:

At least one of the statements in the body or the iterator must eventually cause the loop condition to turn false or force a break out from the body

Example

```
1 #include<stdio.h>
2 int main()
3 {
4         int i;
5         for(i=1;i<11;i++)
6         {
7                 printf("i = %d\n",i);
8         }
9 }
```

National Institute for Mathematical Sciences Ghana

Syntax:

Using For Loop

```
1  for([initializer];[condition];[iterator])//outer loop
2  {
3    for([initializer];[condition];[iterator])//inner loop
4    {
5      [statement_1;
6       statement_2;
7       statement_k;]
8    }
9  }
```

1. Nested loops arise when one loop structure is embedded within the body of another loop structure.
2. The embedded loop is refered to as the the inner loop and the main loop is refered to as the outer loop.
3. For each iteration of the outer loop, the inner loop must start and complete all of its iterations.

National Institute for Mathematical Sciences Ghana

E.g.: Loop Using For Loop

```
1 #include<stdio.h>
2
3 int main()
4 {
5    int i,j;
6    for(i=0;i<11;i++)
7    {
8       printf("\n New outer loop iteration: %d",i);
9
10      for(j=0;j<5;j++)
11      {
12           printf("\n i=%d, j=%d",i,j);
13      }
14   }
15 }
```

where are we?

National Institute for Mathematical Sciences Ghana

Statement

Break statement is used within the switch and loop structures to exist the structure.

Upon exit, control is passed to the immediate statement following the structure.

If a break statement is placed outside a loop or switch structure, a compile time error is generated.

National Institute for Mathematical Sciences Ghana

E.g.: Statement

```
1 #include<stdio.h>
2 int main()
3 {
4    int i=10;
5    while(i<26)
6    {
7       if(i%9==0)
8       {
9         printf("1st multiple of 9 btn 10 & 25 is %d",i
10        break;
11      }
12      i++;
13   }
14   printf("\nDone!\n");
15 }
```

Statement

Continue statement is used within the loop structures to exit *only* the current iteration.

If a Continue statement is placed outside a loop structure, a compile time error is generated.

Example

```
1 #include<stdio.h>
2 int main()
3 {
4    int i;
5    for(i=1;i<21;i++)
6    {
7           if( i % 2 ==0)
8           {
9                  continue;
10          }
11          printf("\n %d",i);
12   }
13 }
```

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

and Statements

label is used to prefix a statement.
goto statements are used to transfer control to a labelled statement.

```
1 #include<stdio.h>
2 int main()
3 {
4         printf("\nPharoh! ");
5
6         goto myline;
7
8         printf("let my people "); //skipped by goto
9
10        myline:printf("go!\n");
11 }
```

Goto statements should be avoided as it leads to the so called spaghetti code making it difficult to follow the logical flow of a program.

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

1. Write a C program to generate the product table shown below

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
...	...	...	...	...	...
10	10	20	30	40	50

2. Write a C program that receives an integer as input and outputs the product of of its digits. E.g. $1234 = 24$, $705 = 0$

3. Write a C program that has the following menu options:

1. Area of a Triangle
2. Area of a Rectangle
3. Area of a Circle
4. Exit

The user selects the number corresponding to an option. Upon selecting a menu option, the program takes the appropriate input values from the user and computes the result. After displaying the result, the system returns the user to the menu option.
The program is terminated if menu option 4 is selected.

Useful formulae:
Area of a Triangle $= \frac{base * height}{2}$
Area of a Rectangle $= Length * Width$
Area of a Circle $= \pi r^2$ where π=3.142

Special thanks to the following for their initial work:

- *Kwesi A. Smith*
- *Shirley A. Akasreku*