

Computer science
class: 10th
ch# 05

By: Sir Akhtar Ali

Difference between for loop() & while loop.

Ans: For loop:

The **for** statement is used to executed a set of statements repeatedly for a fixed number of times in a program.

While loop:

The while statement is used to implement repetition structure in a program when the number of iterations is not known in advance and the repetition continues until some condition remains true.

Difference between while and do while loop

Ans: Difference between While() and Do-While() Loops:

	While() loop	Do-WHILE () Loop
1.	While() loop is pre-tested loop.	Do-while loop is post-tested loop.
2.	The syntax or general form of while() loop is: <pre>while(condition) { Statements; //body of loop }</pre>	The syntax or general form of do-while() loop is: <pre>Do{ Statements; // body of loop. }while(condition);</pre>
3.	In 'while' loop the controlling condition appears at the start of the loop.	In do-loop the controlling condition appears at the end of the loop.
4.	The iterations do not occur if, the condition at the first iteration appears false	The iteration occurs at least once even if the condition is false at the first iteration.

What is looping structure? Explain for loop with example.

Ans: A loop Structure:

A loop is a structure that enables the programmer to execute the same sequence of statements repeatedly until a particular condition is met.

For loop/ For Statement:

The for is a looping statement which is used to execute a set of statements repeatedly for a fixed number of times. It is also known as counter loop. It has the **general form:**

```
For (initialization; test condition; increment/decrement)
```

```
{
```

```
Body of the loop
```

```
}
```

When for statement is executed, a variable (also known as loop variable) is assigned an initial value in the initialization part of the loop, such as $k=1$ or $count=0$. The value of the loop variable is checked with the given test condition. The test condition is a relational expression, such as $k<10$. If the condition is true, the control enters the body of the loop otherwise it will exit the loop.

...

incremented or decremented using an assignment statement such as $k=k+1$. The new value of loop variable is again checked with the test condition. If the condition is satisfied then the body of the loop is again executed. This process goes on till the last condition becomes false.

Body of the loop may have one or more statements. If it contains only a single statement then braces are not needed.

Note: all the three expressions such as initialized, test condition and increment are optional. You can omit any or all in for statement.

Example: For example the following all for statements are valid.

For (; ;)

For(int i=1 ; ;)

For(; k<10 ; k++)

For(; ; k++)

For(; x<12 ;)

classNotes

While & do while loop with examples:

Ans: While loops/The While Statement:

A repetition structure when the number of iterations is known in advance and the repetition continues until test condition remains true.

The **while** statement has the **general form**:

While (test condition)

{

Body of the loop

}

When a while statement is executed, the computer first evaluates the

...

test condition. If it is true, body of the while loop is executed. After the execution of the body of the loop, the test condition is again evaluated and if it is true, the body of the loop is executed once again. The process continues until the test condition becomes false. When it becomes false, the control is transferred to the first statement following the end of body of loop.

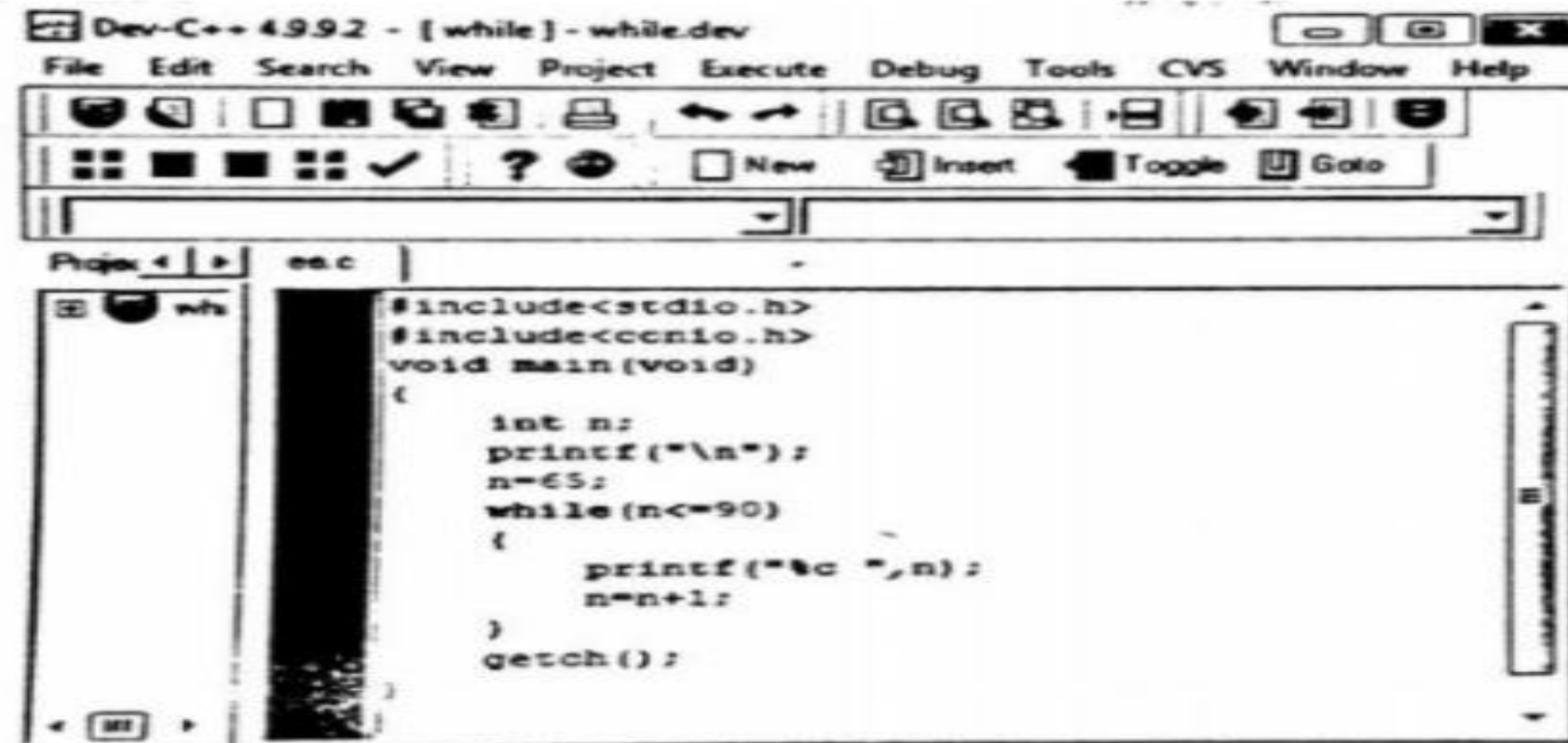
The body of the loop can be a single statement or it can be multiple statements.

Examples:

Program: The program in Fig prints all the upper case letters on a single line using while loop. The ASCII code for upper case letters are in the range 65 to 90.

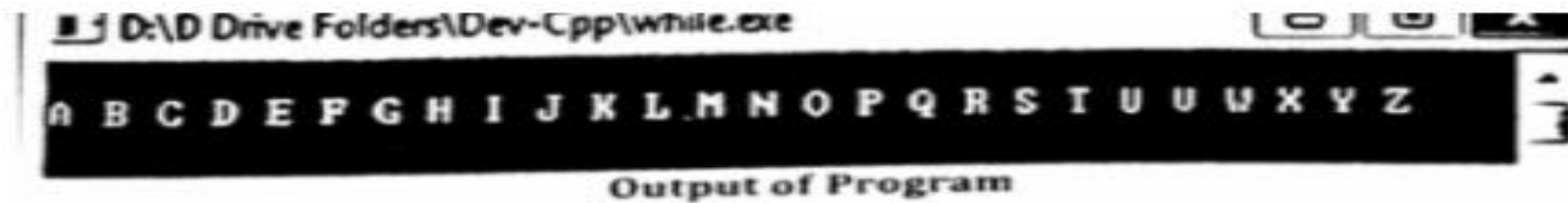
...

Program: The program in Fig prints all the upper case letters on a single line using while loop. The ASCII code for upper case letters are in the range 65 to 90.



Program to print upper – case letters

Output:



```
D:\D Drive Folders\Dev-Cpp\while.exe
A B C D E F G H I J K L M N O P Q R S T U U X Y Z
Output of Program
```

Do-while loops/The Do While Statement:

The do while statement is used to implement loop structure when it is required to execute the loop at least once. The general form of the **do while** loop is given below.

Do

{

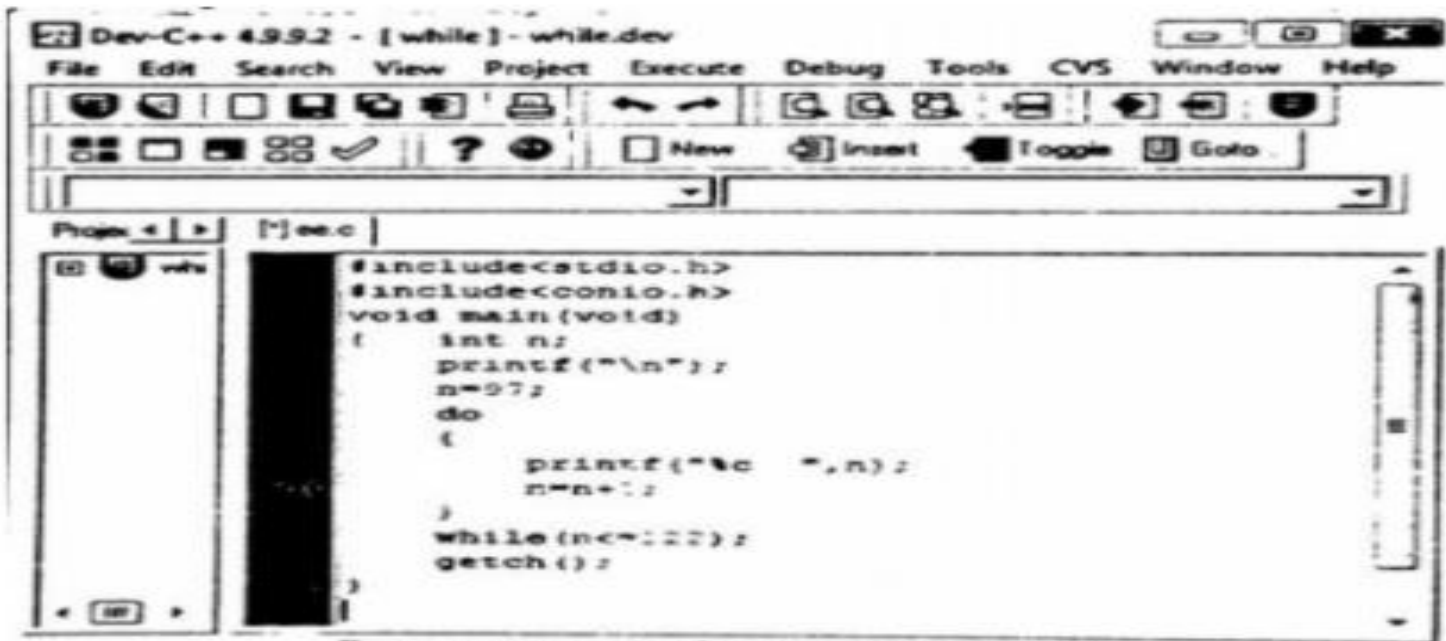
Body of the loop

}

While (test condition)

The statement **while (test condition)** is placed at the end of the loop so that the body of the loop is executed at least once whether the condition is true or false. There is a semicolon after the test condition because it is at the end of loop. If the body of loop contains a single statement then braces are not required.

...



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations and execution. The main editor window displays the following C code:

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    int n;
    printf("\n");
    n=97;
    do
    {
        printf("%c  ",n);
        n=n+1;
    }
    while(n<=122);
    getch();
}
```

Program to print lower - case letters

The output of this program is shown in Fig.



The screenshot shows a command prompt window with the following output:

```
D:\D Drive Folders\Dev-Cpp\while.exe
a b c d e f g h i j k l m n o p q r s t u v w x y z _
```

...

- In this program, the integer variable `n` is initialized to 97 which is the ASCII value of letter `a`, before entering the loop.
- The expression in the do while statement, `n<=122` acts as the test condition. The statements within the loop execute as long as this expression remains true.
- The statement `n=n+1` increment the value of `n` by one with each iteration. Note that the ASCII value of `z` is 122.

Purpose of break & continue statement:

Ans: The Break Statement:

C language provides the break statement to exit from a loop as soon as certain condition occurs. It is used in **for**, **while** and **do while** statements. Break statement is also used to exit the body of **switch** statement after executing the statements under a **case** and transfers control to the first statement following the end of the **switch** statement.

The Continue Statement:

Sometimes during the execution of a loop, when a certain condition occurs after executing a statement, it may be required to skip the remaining statements within the body of the loop and continue for next iteration until the test condition increment. C language provides the continue statement to achieve this task. The continue statement causes the loop to be continued with the next iteration after skipping the remaining statements within the body of the loop.

The general format of continue statement is:

Continue;

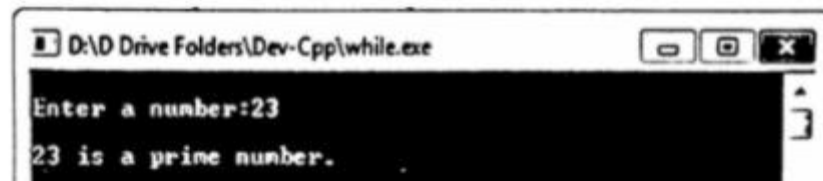
Program to check for prime number by using break statement:

...



```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    int num, k, a=0;
    printf("\nEnter a number:");
    scanf("%d", &num);
    for(k=2; k<=num/2; k++)
        if(num%k==0)
        {
            a=1;
            break;
        }
    if(a==0)
        printf("\n%d is a prime number.", num);
    else
        printf("\n%d is not a prime number.", num);
    getch();
}
```

The output of the program is shown in Fig.



```
D:\D Drive Folders\Dev-Cpp\while.exe
Enter a number:23
23 is a prime number.
```

- When the above program is executed, it reads a number and determines whether it is a prime number or not. The integer variable a is assigned the value 0 to assume that the number num is a prime number.
- The remainder operator (%) is used inside the for loop to check if the number is exactly divisible by any number in the range 2 to num/2.
- If this condition is true, the variable a is assigned the value 1, indicating that num is not a prime number.
- At this point, the break statement is used to exit for loop immediately. After the execution of the for loop, the value of a is checked, if it is 0 then the message telling the user that num is a prime number is printed otherwise the message, num is not a prime number is printed.

Nested loop:

In C language it is allowed to nest loops within another loop. Nested loops can be of any kind. For example, the programmer can nest a for loop inside a **while** loop or inside a **do while** loop. Loops can be mixed in any way required.

Example # 1:

Programs uses nested loop to print the products of numbers as given below.

1 × 1 = 1

1 × 2 = 2

1 × 3 = 3

1 × 4 = 4

2 × 1 = 2

2 × 2 = 4

2 × 3 = 6

2 × 4 = 8

3 × 1 = 3



Output of Program

The program prints the products of number



...

- The loop variable j is assigned its initial value 1 and the nested loop

```
For (k=1; k<=4; k++)
```

```
{
```

```
    Prod=j*k;
```

```
    Printf("\n%2d x %2d = %2d", j, k, prod);
```

```
}
```

is executed. This calculates and displays the first four products 1×1 , 1×2 , 1×3 and 1×4

- The value of j is then incremented by 1 and the inner loop is executed again. This calculates and displays the next four products, 2×1 , 2×2 , 2×3 and 2×4 .
- Finally, j is incremented to 3, giving the last four products, 3×1 , 3×2 , 3×3 and 3×4 .
- In this program braces must be used in the inner for loop because there are more than one statements to be executed.