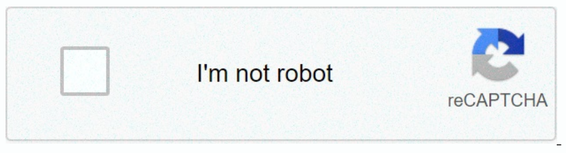


Continue

Difference between while and do while loop in c pdf

Difference between while and do-while loop in c pdf. Difference between while and while loop. What is the main difference between while loop and do while loop. What is difference between do and while loop. Can we use && in while loop.

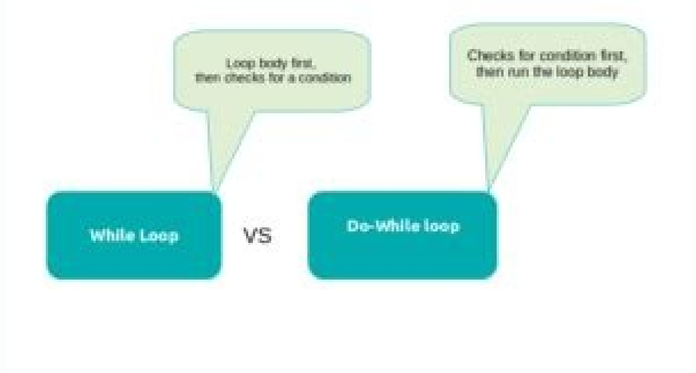
ReadDiscussCoursesPracticeImprove Article Save Article Like Article Loops in C language are the control flow statements that are used to repeat some part of the code till the given condition is satisfied. The do-while loop is one of the three loop statements in C, the others being while loop and for loop. It is mainly used to traverse arrays, vectors, and other data structures.What is do…while Loop in C?The do…while in C is a loop statement used to repeat some part of the code till the given condition is fulfilled. It is a form of an exit-controlled or post-tested loop where the test condition is checked after executing the body of the loop. Due to this, the statements in the do…while loop will always be executed at least once no matter what the condition is.do { // body of do-while loop } while (condition);The following example demonstrates the use of do…while loop in C programming language.#include int main(){   int i = 0;   do {   printf("Geeks");   i++;   } while (i < 3);   return 0;}How does the do…while Loop works?Syntax Structure of do while loopThe working of the do…while loop is explained below:When the program control first comes to the do…while loop, the body of the loop is executed first and then the test condition/expression is checked, unlike other loops where the test condition is checked first. Due to this property, the do…while loop is also called exit controlled or post-tested loop.When the test condition is evaluated as true, the program control goes to the start of the loop and the body is executed once more.The above process repeats till the test condition is true.When the test condition is evaluated as false, the program controls move on to the next statements after the do…while loop.As with the while loop in C, initialization and updation is not a part of the do…while loop syntax. We have to do that explicitly before and in the loop respectively.The flowchart below shows the visual representation of the flow of the do…while loop in C.Flowchart of do…while Loop in CAs with other loops, we can also nest one do…while loop into another loop. It is demonstrated using the following C program.Example of Nested do…while Loop in C:#include int main(){   int i = 0, j;   int count = 0;   do {   j = 0;   do {   printf("%d ", count++);   j++;   } while (j < 3);   printf("");   i++;   } while (i < 3);   return 0;}To know more about nested loops in C, refer to this article – Nested Loops in C with ExamplesExamples of do…while Loop in CExample 1. C Program to demonstrate the behavior of do…while loop if the condition is false from the start.#include #include int main(){   bool condition = false;   do {   printf("This is loop body.");   } while (condition);   return 0;}As we can see, even when the condition is false at the start, the loop body is executed once.

This is because in the do…while loop, the condition is checked after going through the body so when the control is at the start,It goes through the loop body.Executes all the statements in the body.Checks the condition which turns out to be false.Then exits the loop.Example 2. C Program to print Multiplication Table of N using do…while loopThe following example demonstrates the use of do…while loop for printing the multiplication table of N.#include int main(){   int N = 5, i = 1;   do {   printf("%d\tx\t%d\t=\t%d", N, i, N * i);   } while (i++ < 10);   return 0;}Output5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50The following table lists the important differences between the while and do…while Loop in C.while Loopdo…while LoopThe test condition is checked before the loop body is executed.The test condition is checked after executing the body.When the condition is false, the body is not executed not even once.The body of the do…while loop is executed at least once even when the condition is false.It is a type of pre-tested or entry-controlled loop.It is a type of post-tested or exit-controlled loop.Semicolon is not required.Semicolon is required at the end.To know more about these differences, please refer to this article – Difference between while and do-while loop in C, C++, JavaConclusionIn conclusion, the use of the only exit-controlled loop in C, the do…while loop is also to iterate a particular part of code but the way it works makes it different from entry-controlled loops such as the while loop and the for loop. It is useful in cases where we need to execute the statement inside the loop body at least once such as in traversing circular linked lists.FAQs on C do…while Loops1. How many types of loops are there in C?Ans: There are 3 types of loops in C language:for Loopwhile Loopdo…while Loop2. What are the entry-controlled or pre-tested loops?Ans: The entry-controlled loops or pre-tested loops are those loops in which the loop condition is checked before executing the body of the loop.3. What are the exit-controlled or post-tested loops?Ans: The exit-controlled loops or post-tested loops are those loops in which the program control comes to the body of the loop before checking the loop condition.4. Which loop is guaranteed to execute at least once?Ans: The do…while loop is guaranteed to execute the statements in the body of the loop at least once as it is a type of exit-controlled loop.5. Can we skip braces in C do…while loop syntax if there is only one statement in the body?Ans: No, we cannot skip braces in C do…while syntax even if there is only a single statement unlike while and for loop.6. How to create an infinite loop in C using the do…while loop?Ans: We can create an infinite loop in C by specifying a condition that will always be true as the loop condition. The below program demonstrates how to do that:#include int main(){   do {   printf("gfg ");   } while (1);   return 0;}Outputgfg gfg gfg gfg gfg gfg gfg .... (infinite)Related Articles:C – LoopsC for Loopwhile Loop in CLast Updated : 24 Feb, 2023Like Article Save Article While loop checks the condition first and then executes the statement(s), whereas do while loop will execute the statement(s) at least once, then the condition is checked. While loop is entry controlled loop, whereas do while is exit controlled loop.

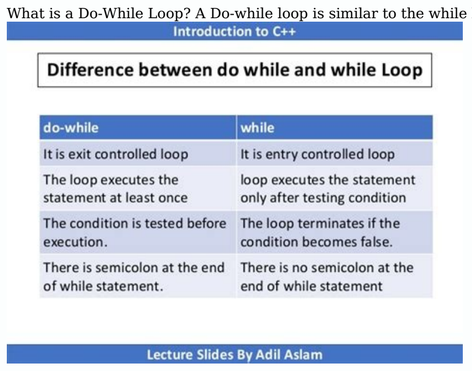
In the while loop, we do not need to add a semicolon at the end of a while condition, but we need to add a semicolon at the end of the while condition in the do-while loop. While loop statement(s) is executed zero times if the condition is false, whereas the do-while statement is executed at least once. While loop allows initialization of counter variable before starting the body of a loop, whereas do while loop allows initialization of counter variable before and after starting the body of a loop. What are loops? A Loop executes the sequence of statements many times until the stated condition becomes false. A loop consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false.

The purpose of the loop is to repeat the same code a number of times. What is While Loop? A While loop is the most straightforward looping structure. It is an entry-controlled loop. In a while loop, a condition is evaluated before processing a body of the loop. If a condition is true, then and only then the body of a loop is executed. After the body of a loop is executed, the control again goes back to the beginning, and the condition is checked. If it is true, the same process is executed until the condition becomes false. Once the condition becomes false, the control goes out of the loop.

In a while loop, if the condition is not true, then the body of a loop will not be executed, not even once.



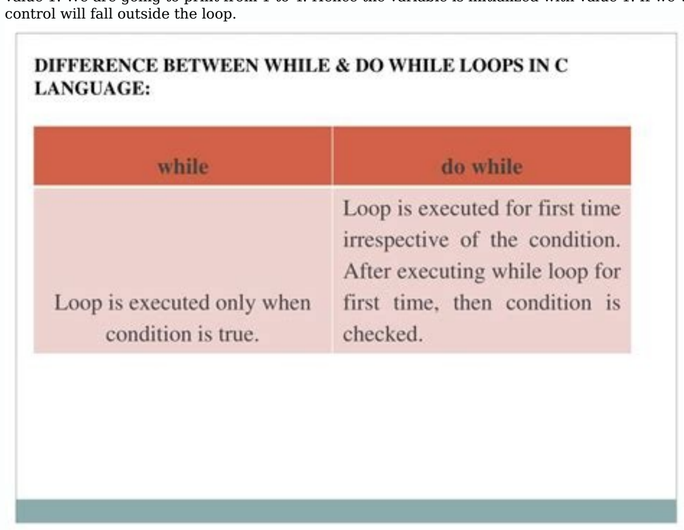
What is a Do-While Loop? A Do-while loop is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop. In the do-while loop, the body of a loop is always executed at least once.



After the body is executed, then it checks the condition. If the condition is true, then it will again execute the body of a loop. Otherwise, the control is transferred out of the loop. Syntax of While loop in C Here is a syntax of While loop in C programming: while (condition) { statements; } In the while loop, we have to write a condition that needs to be evaluated. The statement inside curly braces indicates the code to be executed. Syntax Do While Loop in C Here is a syntax of Do while loop in C programming: do { statements } while (expression); In the do-while loop, we need to first write the statement inside curly braces, which indicates the code to be executed.
After this, we need to mention Java, C, or C++ program expressions that need to be evaluated.

```
// PROGRAM TO PRINT EVEN NUMBERS TILL 20
#include<iostream>
using namespace std;

int main()
{
    int number=2;

    do
    {
        cout<<"\n"<<number;
        number+=2;

    }
    while(number<=20);

    return 0;
}

```

How While Loop Works? While loop works as follows: Flow Chart Explanation: Step 1) Start of while loop Step 2) The test expression or condition is evaluated Step 3) Next, if the test expression is true, the program executes the body of do-while loop Step 4) If the test expression is false, the program outside while loop is executed How Do-While Loop Works? The Do-while loop works as follows: Flow Chart Explanation: Step 1) Start the do-while loop Step 2) The body of do-while loop is executed Step 3) The test expression or condition is evaluated Step 4) If the test expression is true, the compiler executes the body of do-while loop Step 5) Next, if the test expression is false, the compiler executes the statements after the loop body Step 6) Statements that come after the loop body are executed While vs Do-While Loop: Difference Between Them Here is an important difference between While and Do While Loop: While Do While It checks the condition first and then executes statement(s) This loop will execute the statement(s) at least once, then the condition is checked. While loop allows initialization of counter variables before starting the body of a loop. Do while loop allows initialization of counter variables before and after starting the body of a loop. It is an entry controlled loop. It is an exit controlled loop. We do not need to add a semicolon at the end of a while condition. We need to add a semicolon at the end of the while condition. In case of a single statement, we do need to add brackets. Brackets are always needed. In this loop, the condition is mentioned at the starting of the loop. The loop condition is specified after the block is executed. Statement(s) can be executed zero times if the condition is false.
Statement is executed at least once. Generally while loop is written as: while (condition) { Statements; // loop body } Generally do while loop is written as: do{ Statements; //loop body } while (condition); While Loop Example in C Following program illustrates while loop in C programming with an example: #include #include int main() { int num=1; //initializing the variable with value 1 while(num<=4) //while loop with condition { printf("%d",num); num++; //incrementing operation } return 0; } Output: 1 2 3 4 The above program illustrates the use of a while loop. In the above code, we have printed a series of numbers from 1 to 4 using a while loop. We have initialized a variable called num with value 1. We are going to print from 1 to 4. Hence the variable is initialized with value 1. If we want to print from 0, then assign the value 0 during initialization. Next, in a while loop, we have provided a condition (num<=4), which means the loop will execute the body until the value of num becomes 4. After that, the loop will be terminated, and control will fall outside the loop.



In the body of a loop, we have a print function to print our number and an increment operator to increment the value per execution of a loop.
An initial value of num is 1, after the execution, it will become 2, and during the next execution, it will become 3. This process will continue until the value becomes 4, and then it will print the series on the console and terminate the loop. Do While Loop Example in C The following program is a Do-while loop example to print a table of number 2 in C: #include #include int main() { int num=1; //initializing the variable with value 1 do //do-while loop { printf("%d",2*num); num++; //incrementing operation } while(num<=4); return 0; } Output: 2 4 6 8 In the above example, we have printed a multiplication table of 2 using a do-while loop. First, we have initialized a variable ‘num’ with the value 1. Then we have written a do-while loop. In a loop, we have a print function that will print the series by multiplying the value of num with 2. After each increment, the value of num will increase by 1, and it will be printed on the screen. Initially, the value of num is 1.
In a body of a loop, the print function will be executed in this way: 2*num where num=1, then 2*1=2. Hence the value 2 will be printed. This will go on until the value of num becomes 10. Next, the loop will be terminated, and a statement which is immediately after the loop will be executed. In this case, it will return 0. Which One Should We Choose? When checking a condition, if the first iteration is compulsory, we need to use the while loop. It can also be used if the number of iterations is unknown or uncertain. Do while loop mainly requires in the case where we have to execute the loop minimum one time. The do-while loop is typically needed in a menu-driven programming language where the final condition is based upon the end-user.