C++ LESSON MODULE: DO-WHILE LOOP

INTRODUCTION

In C++, the do-while loop is an exit-controlled loop that executes a block of code at least once and continues as long as a given condition remains true. Unlike the while loop, which checks the condition before executing the body, the do-while loop checks the condition after the body has executed, guaranteeing that the loop body runs at least once.

SYNTAX AND STRUCTURE

```
do {
    // body of loop
    // update
expression
} while (condition);
```

Explanation:

- Body: The block of code that will always execute at least once.
- Update Expression: Modifies the loop variable to bring it closer to the termination condition.
- Condition: Checked after the body executes. If true, the loop repeats; if false, it exits.
- Semicolon: The semicolon after the condition is required.

EXAMPLE 1: PRINT NUMBERS FROM 1 TO 5

```
#include <iostream>
using namespace std;

int main() {
    int i = 0;
    do {
        cout << "Hi" << endl;
        i++;
    } while (i < 5);
    return 0;
}
```

OUTPUT:

Hi
Hi
Hi
Hi
Hi

Explanation: The loop prints "Hi" five times. The body executes first, then the condition is checked.

VISUALIZATION OF DO-WHILE LOOP EXECUTION

Visualization steps:

1. Execute Body: Run all statements inside the loop.
2. Update Variables: Modify loop-control variables.
3. Check Condition: Evaluate the condition.
4. Repeat: If true, return to step 1; if false, exit.

EXAMPLE 2: PRINT NUMBERS (EVEN IF CONDITION IS FALSE)

```
#include <iostream>
using namespace std;

int main() {
    int i = 1;
    do {
        cout << i << endl;
        i++;
    } while (i < 0);
    return 0;
}
```

OUTPUT: 1

Explanation: Even though the condition is false from the start, the body executes once.

EXAMPLE 3: USER INPUT VALIDATION

```
#include <iostream>
using namespace std;

int main() {
   int n;
   do {
      cout << "Enter a positive number: ";
      cin >> n;
   } while (n <= 0);
   cout << "Entered number: " << n << endl;
   return 0;
}
```

OUTPUT:

Enter a positive number: -1
Enter a positive number: -999
Enter a positive number: 2
Entered number: 2

Explanation: The loop ensures the user enters a valid number by repeating until a positive number is provided.

EXAMPLE 4: NESTED DO-WHILE LOOPS

```
#include <iostream>
using namespace std;

int main() {
    int i = 0;
    do {
        int j = 0;
        do {
            cout << "* ";
            j++;
        } while (j < 3);
        cout << endl;
        i++;
    } while (i < 3);
    return 0;
}
```

OUTPUT:

* * *

* * *

* * *

Explanation: The outer loop controls rows, and the inner loop prints columns. Each iteration of the outer loop triggers a full execution of the inner one.

REFERENCES:

- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/CPP/CPP-DO-WHILE-LOOP/
- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/CPP/NESTED-LOOPS-IN-C-WITH-EXAMPLES-2/
- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/DIFFERENCE-BETWEEN-WHILE-AND-DO-WHILE-LOOP-IN-C-C-JAVA/

THANK YOU