C++ LESSON MODULE: WHILE LOOP

INTRODUCTION

In C++, the while loop is an entry-controlled loop that repeatedly executes a block of code as long as a specified condition remains true. Unlike the for loop, which is useful when the number of iterations is known, the while loop is typically used when the number of repetitions is unknown and depends on a condition evaluated during runtime.

SYNTAX AND STRUCTURE

```
while (condition) {
    // body of loop
}
```

Explanation:

- Condition: A Boolean expression that controls the loop.
- Body: Contains the statements to be executed repeatedly while the condition is true.
- Entry-controlled: The condition is evaluated before the body is executed.
- If the condition is false initially, the body is skipped altogether.

EXAMPLE 1: PRINT NUMBERS FROM 1 TO 5

```
#include <iostream>
using namespace std;
int main() {
    int i = 1; while (i <= 5) {
        cout << i << " ";
        i++;
    }
    return 0;
}
```

OUTPUT:

1 2 3 4 5

Explanation: The loop initializes i = 1. The condition i <= 5 is checked before each iteration. Each loop iteration prints i, then increments it.

VISUALIZATION OF WHILE LOOP EXECUTION

Visualization steps of the while loop:

1. Condition Check: Evaluate if the condition is true.
2. Execute Body: If true, execute all statements inside the loop.
3. Update Variables: Modify loop-control variables as needed.
4. Repeat: Return to step 1 until the condition becomes false.

EXAMPLE 2: CALCULATING THE SUM OF NATURAL NUMBERS

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

int main() {
    int n = 5;
    int sum = 0;

    while (n > 0) {
        sum += n;
        n--;
    }

    cout << "Sum = " << sum;
    return 0;
}
```

OUTPUT: Sum = 15

Explanation: The loop keeps adding n to sum and decrements n until the condition n > 0 becomes false.

EXAMPLE 3: NESTED WHILE LOOPS

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

int main() {
    int i = 0;
    while (i < 4) {
        int j = 0;
        while (j < 4) {
            cout << "* ";
            j++;
        }
        cout << endl;
        i++;
    }
    return 0;
}
```

OUTPUT:

```
* * * *
* * * *
* * * *
* * * *
```

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

EXAMPLE 4: INFINITE WHILE LOOP

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

int main() {
    while (true) {
        cout << "Infinite loop example" <<
endl;
    }
    return 0;
}
```

Warning: Always ensure the condition eventually becomes false or use control statements like break to exit manually.

REFERENCES:

- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/CPP/CPP-WHILE-LOOP/
- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/CPP/NESTED-LOOPS-IN-C-WITH-EXAMPLES-2/
- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/CPP/INFINITE-LOOP-IN-CPP/
- GEEKSFORGEEKS: HTTPS://WWW.GEEKSFORGEEKS.ORG/DSA/DIFFERENCE-BETWEEN-FOR-AND-WHILE-LOOP-IN-C-C-JAVA/
- PYTHON TUTOR VISUALIZER: HTTPS://PYTHONTUTOR.COM/VISUALIZE.HTML

THANK YOU