Do-while Loop:

Loops come into use when we need to repeatedly execute a block of statements. Like while the do-while loop execution is also terminated on the basis of a test condition. The main difference between a do-while loop and a while loop is in the do-while loop the condition is tested at the end of the loop body, i.e do-while loop is exit controlled whereas the other two loops are entry-controlled loops.

Key characteristics:

Executes at least once: The loop body always executes one time, even if the condition is initially false. This is because the condition is checked at the end of the loop, not the beginning.

Syntax:

do {

// statements to be executed

} while (condition);

How it works:

Execution of the loop body: The statements within the do block are executed first.

Condition check: The while statement then evaluates the specified condition.

Loop repetition: If the condition is true, the code jumps back to the beginning of the do block and executes the statements again. This process repeats until the condition becomes false.

Loop termination: Once the condition becomes false, the loop terminates, and execution continues with the code following the while statement.

Program :

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

int main() {
  int i = 1;

  do {
    cout << i << " ";
    i++;
  } while (i <= 5);

  return 0;
}
```

Output:

```
PS E:\C++ PROGRAMMES> g++ main.cpp
PS E:\C++ PROGRAMMES> ./a.exe
1 2 3 4 5
```

Nested Loops:

Nested loops occur when one loop is placed inside another loop.

The inner loop runs to completion for each iteration of the outer loop.

They're used for tasks involving multi-dimensional data structures or repetitive operations within a larger loop.

Key points:

The inner loop completes all its iterations before the outer loop moves to its next iteration.

You can nest any loop type within any other loop type (for, while, do-while).

Nested loops can be visualized as a matrix, where each row represents an outer loop iteration and each column represents an inner loop iteration.

Syntax:

Syntax for Nested For loop:

```
for ( initialization; condition; increment ) {
        for ( initialization; condition; increment ) {
                // statement of inside loop
        }
        // statement of outer loop
}
```

Syntax for Nested While loop:

```
while(condition) {
        while(condition) {
                // statement of inside loop
        }
        // statement of outer loop
}
```

Syntax for Nested Do-While loop:

```
do{
        do{
         // statement of inside loop
        }while(condition);
         // statement of outer loop
}while(condition);
```

Note: There is no rule that a loop must be nested inside its own type. In fact, there can be any type of loop nested inside any type and to any level.

Syntax:

```
do{
        while(condition) {
                for ( initialization; condition; increment ) {
                        // statement of inside for loop
                 }
                // statement of inside while loop
        }
        // statement of outer do-while loop
}while(condition);
```

Program:

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

int main() {
    int n=5,m=5;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=m;j++){
            cout<<i<<" X "<<j<<" = "<<i*j<<endl;
        }
        cout<<endl;
    }
}
```

Output : PS E:\C++ PROGRAMMES> g++ main.cpp

```
PS E:\C++ PROGRAMMES> ./a.exe
1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
1 X 4 = 4
1 X 5 = 5

2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10

3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15

4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
4 X 5 = 20

5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
```

Home Work:

1. **What if we write the code as**

```
for(int i=0;i<=5;i++);{
        cout<<i;
}
```

The output is : the identifier i is not defined in this scope

If we write it as

```
        for(int i=0;i<=5;i++);{
        cout<<rohini;
}
```

The output is: rohini

2.

```
#include <iostream>

using namespace std;

int main() {
    int i;
    if(cin>>i){
        cout<<i;
    }
}
```

Output:

PS E:\C++ PROGRAMMES> g++ main.cpp

PS E:\C++ PROGRAMMES> ./a.exe

9

9

3.

```cpp
      #include <iostream>

using namespace std;

int main() {

    if(cout<<"hi"){
        cout<<" Rohini";
    }
}
```

Output :

PS E:\C++ PROGRAMMES> g++ main.cpp

PS E:\C++ PROGRAMMES> ./a.exe

hi Rohini