Task 8: Introduction to Namespaces in C++

Objective:
To understand and demonstrate the use of namespaces in C++. This task defines two namespaces containing functions with the same name but different implementations, and shows how namespaces help avoid name conflicts.

1. Concept Overview

A namespace in C++ is used to group related identifiers such as variables and functions. Namespaces help avoid naming conflicts, especially in large programs where multiple libraries may define functions with the same name.

2. Program Code

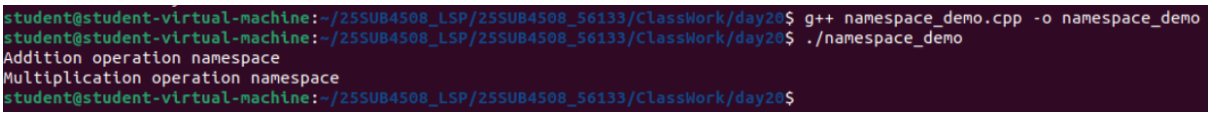
```
#include <iostream>
 using namespace std;

 namespace MathAdd {
        void operation() {
        cout << "Addition operation namespace" << endl;
        }
}

 namespace MathMultiply {
        void operation() {
        cout << "Multiplication operation namespace" << endl;
        }
}

 int main() {
        MathAdd::operation();
        MathMultiply::operation();
        return 0;
}
```

3. Compilation Instructions

Compile the program using a C++ compiler:

```
g++ namespace_demo.cpp -o namespace_demo
```

4. Sample Output



5. Observations & Explanation

1. Two namespaces are defined with the same function name 'operation'.
2. Each namespace has its own implementation of the function.
3. The scope resolution operator (::) is used to specify which namespace function to call.
4. Namespaces prevent naming conflicts and improve code organization.

6. Advantages of Namespaces

- Avoid name collisions
- Improve code readability
- Useful in large-scale applications

7. Conclusion

This task demonstrates how namespaces allow functions with the same name to coexist without conflict. Namespaces are an essential feature of C++ for writing modular and maintainable programs.