Task 12: STL Maps in C++

Objective:
To understand and demonstrate the use of the STL map container in C++. This task maps students' names to their grades using std::map and prints the stored data.

1. Concept Overview

An STL map is an associative container that stores elements in key-value pairs. Each key in a map is unique and elements are stored in sorted order by key. Maps are commonly used for fast lookup, insertion, and deletion.

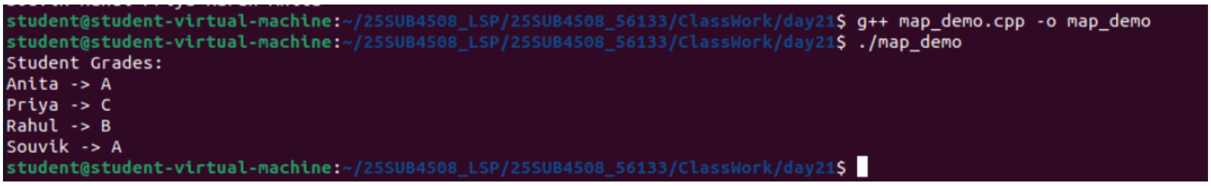
2. Program Code

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

 int main() {
        map<string, char> studentGrades;

        studentGrades["Souvik"] = 'A';
        studentGrades["Rahul"] = 'B';
        studentGrades["Anita"] = 'A';
        studentGrades["Priya"] = 'C';

        cout << "Student Grades:" << endl;
        for (const auto& entry : studentGrades) {
        cout << entry.first << " -> " << entry.second << endl;
        }

        return 0;
 }
```

3. Compilation Instructions

```
 g++ map_demo.cpp -o map_demo
```

4. Sample Output



```
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day21$ g++ map_demo.cpp -o map_demo
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day21$ ./map_demo
Student Grades:
Anita -> A
Priya -> C
Rahul -> B
Souvik -> A
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day21$ 
```

5. Observations & Explanation

1. std::map is used to store student names as keys and grades as values.
2. Elements are inserted using the subscript operator [].
3. The map automatically sorts elements based on the key.
4. A range-based for loop is used to iterate through the map.
5. Each key-value pair is accessed using entry.first and entry.second.

6. Advantages of STL Map

- Stores data in key-value pairs
- Automatically sorted by key
- Fast lookup and insertion
- Useful for dictionaries and mappings

7. Conclusion

This task demonstrates how STL maps can be used to efficiently associate keys with values. Using std::map makes data organization, retrieval, and display simple and structured.