Task 13: Lambda Expressions in C++

Objective:
To understand and demonstrate the use of lambda expressions in C++. This task uses a lambda expression with std::sort to sort a vector of strings in descending (reverse alphabetical) order.

1. Concept Overview

A lambda expression is an anonymous function defined at the place where it is used. Lambdas are commonly used with STL algorithms to provide custom behavior such as sorting, filtering, and transformation without writing separate functions.

2. Program Code

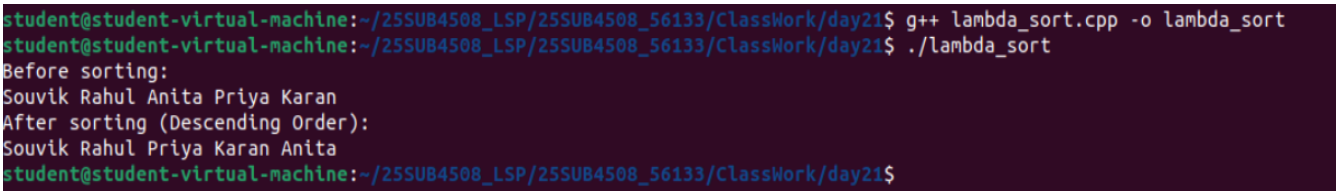
```
#include <iostream>
 #include <vector>
 #include <algorithm>
 #include <string>
 using namespace std;

 int main() {
        vector<string> names = {"Souvik", "Rahul", "Anita", "Priya", "Karan"};

        cout << "Before sorting:" << endl;
        for (const auto& name : names)
        cout << name << " ";
        cout << endl;

        sort(names.begin(), names.end(), [](const string& a, const string& b) {
        return a > b; // Descending order
        });

        cout << "After sorting (Descending Order):" << endl;
        for (const auto& name : names)
        cout << name << " ";
        cout << endl;

        return 0;
 }
```

3. Compilation Instructions

Compile the program using a C++ compiler:

```
g++ lambda_sort.cpp -o lambda_sort
```

4. Sample Output



5. Observations & Explanation

1. A vector<string> is created with unsorted names.
2. std::sort is used to sort the vector.
3. A lambda expression is passed as the third argument to std::sort.
4. The lambda compares two strings and returns true if the first is greater than the second.
5. This results in sorting the names in descending alphabetical order.

6. Advantages of Lambda Expressions

- Reduces the need for separate helper functions
- Improves code readability
- Works seamlessly with STL algorithms

7. Conclusion

This task demonstrates how lambda expressions can be used with STL algorithms to customize behavior efficiently. Lambdas make C++ code more concise, expressive, and suitable for modern programming practices.