C++ Programming Assignment: Operations on Sets

Subject: Object-Oriented Programming

Topic: Operations on Sets using C++ STL

Level: Intermediate

Format: Console-based Application

Objective:

To practice implementation and usage of set data structures using C++ STL and understand the basic set operations like union, intersection, difference, and subset checking.

Instructions:

1. Use std::set or arrays to represent sets.
2. Implement a menu-driven program.
3. Ensure your program can handle input validation.
4. Include appropriate comments and follow best practices.

Features to Implement:

1. Create Sets

Allow the user to create two sets (A and B) of integers.

2. Display Sets

Print the current elements in both sets.

3. Union

Show the union of Set A and Set B.

4. Intersection

Show the intersection of Set A and Set B.

5. Difference

- A - B

- B - A

6. Check Subset

Check if one set is a subset of the other.

7. Exit Program

Set Input Note:

You may take input for each set using arrays and then perform set operations (such as union, intersection, difference, and subset checks) by manipulating these arrays. Input values should be integers, and duplicate elements should be ignored to follow set properties.

Example Input:

Enter number of elements in Set A: 5

Enter elements of Set A: 1 2 2 3 4

Enter number of elements in Set B: 4

Enter elements of Set B: 3 4 5 6

Processed Sets (after removing duplicates):

Set A = {1, 2, 3, 4}

Set B = {3, 4, 5, 6}

Expected Output:

```
Union of A and B = {1, 2, 3, 4, 5, 6}
Intersection of A and B = {3, 4}
A - B = {1, 2}
B - A = {5, 6}
Is A a subset of B? No
Is B a subset of A? No
```

Sample Menu Display:

```
----- Set Operations Menu -----
1. Input Sets
2. Display Sets
3. Union
4. Intersection
5. Difference (A - B)
6. Difference (B - A)
7. Check Subset
8. Exit
Enter your choice:
```