Subsets in C++

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

class Subsets {
public:
   vector<vector<int>> subsets(vector<int>& nums) {
      int n = nums.size();
      int totalno = (1 << n);
      vector<vector<int>> ans;

      for (int i = 0; i < totalno; i++) {
         vector<int> temp;
         for (int j = 0; j < n; j++) {
            if (checkBit(i, j)) {
               temp.push_back(nums[j]);
            }
         }
         ans.push_back(temp);
      }

      return ans;
   }

private:
   // Helper function to check if the i-th bit in n is set
   bool checkBit(int n, int i) {
      return (n & (1 << i)) != 0;
   }
};

int main() {
   // Create an instance of the Subsets class
   Subsets solution;

   // Hardcoded input array
   vector<int> nums = {1, 2, 3}; // Example input

   // Calling subsets to generate all subsets of the
array
   vector<vector<int>> subsets =
solution.subsets(nums);

   // Printing all subsets
   for (auto& subset : subsets) {
      cout << "[";
      for (size_t i = 0; i < subset.size(); i++) {
         cout << subset[i];
         if (i < subset.size() - 1) {
            cout << ", ";
         }
      }
      cout << "]" << endl;
   }

   return 0;
}
```

Detailed Table:

i (Binary)	Subset Indexes	Subset Elements	Subset
0 (000)	None	None	[]
1 (001)	0	{1}	[1]
2 (010)	1	{2}	[2]
3 (011)	0, 1	{1, 2}	[1, 2]
4 (100)	2	{3}	[3]
5 (101)	0, 2	{1, 3}	[1, 3]
6 (110)	1, 2	{2, 3}	[2, 3]
7 (111)	0, 1, 2	{1, 2, 3}	[1, 2, 3]

Explanation of Each Iteration:

1. **Iteration 1 (i = 0 / Binary 000)**:
   - No bits are set, so the subset is empty: [].
2. **Iteration 2 (i = 1 / Binary 001)**:
   - Only the least significant bit is set, so the subset includes only the element 1: [1].
3. **Iteration 3 (i = 2 / Binary 010)**:
   - The second bit is set, so the subset includes only the element 2: [2].
4. **Iteration 4 (i = 3 / Binary 011)**:
   - The first and second bits are set, so the subset includes the elements 1 and 2: [1, 2].
5. **Iteration 5 (i = 4 / Binary 100)**:
   - The third bit is set, so the subset includes only the element 3: [3].
6. **Iteration 6 (i = 5 / Binary 101)**:
   - The first and third bits are set, so the subset includes the elements 1 and 3: [1, 3].
7. **Iteration 7 (i = 6 / Binary 110)**:
   - The second and third bits are set, so the subset includes the elements 2 and 3: [2, 3].
8. **Iteration 8 (i = 7 / Binary 111)**:
   - All bits are set, so the subset includes all elements: [1, 2, 3].

Final Output:

The final list of subsets is:

```
 [ ]
[1]
[2]
[1, 2]
[3]
```

	[1, 3] [2, 3] [1, 2, 3]
[] [1] [2] [1, 2] [3] [1, 3] [2, 3] [1, 2, 3]	