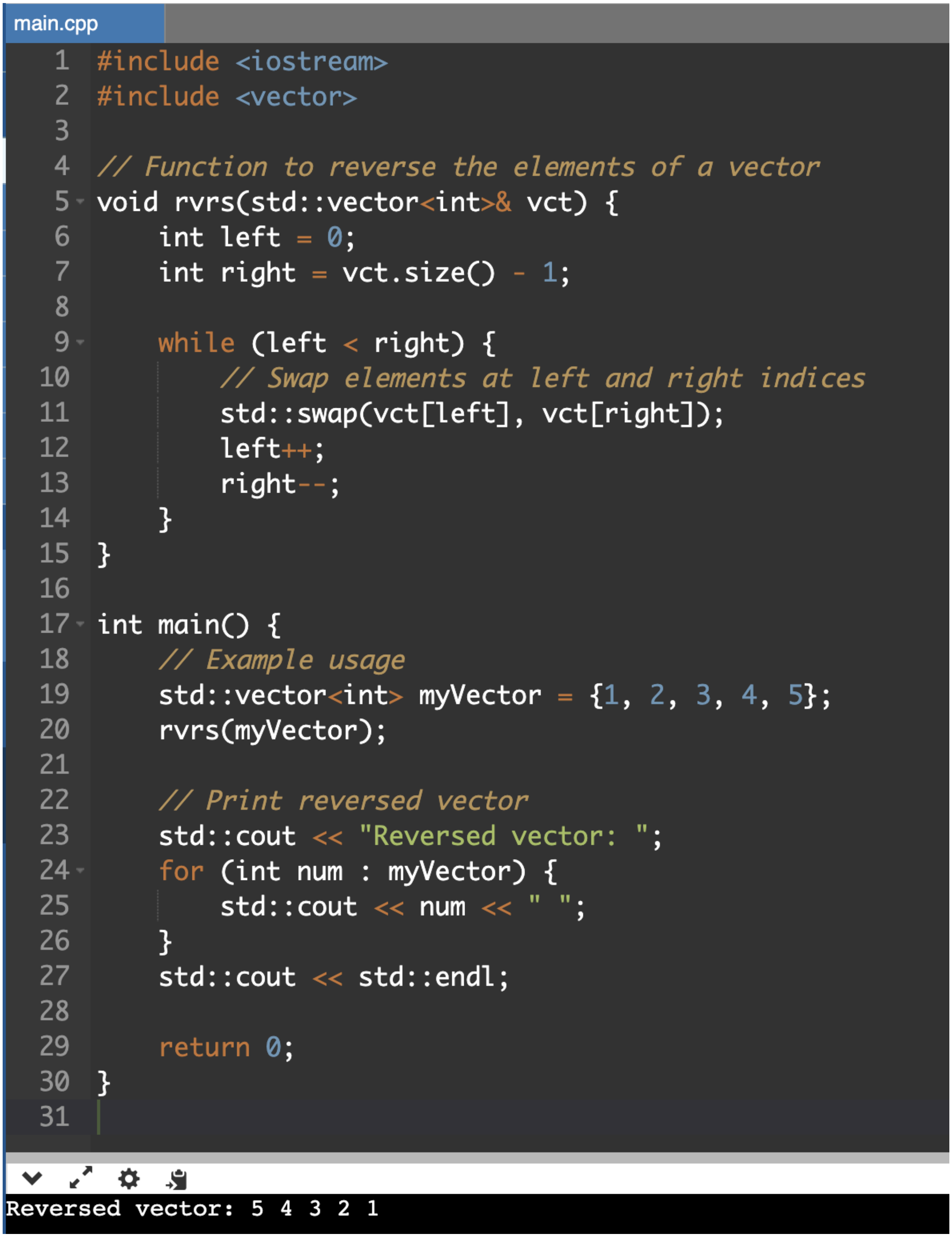
Samir Khadka
CS360L - Programming in C and C++ Lab
Lab Assignment #5

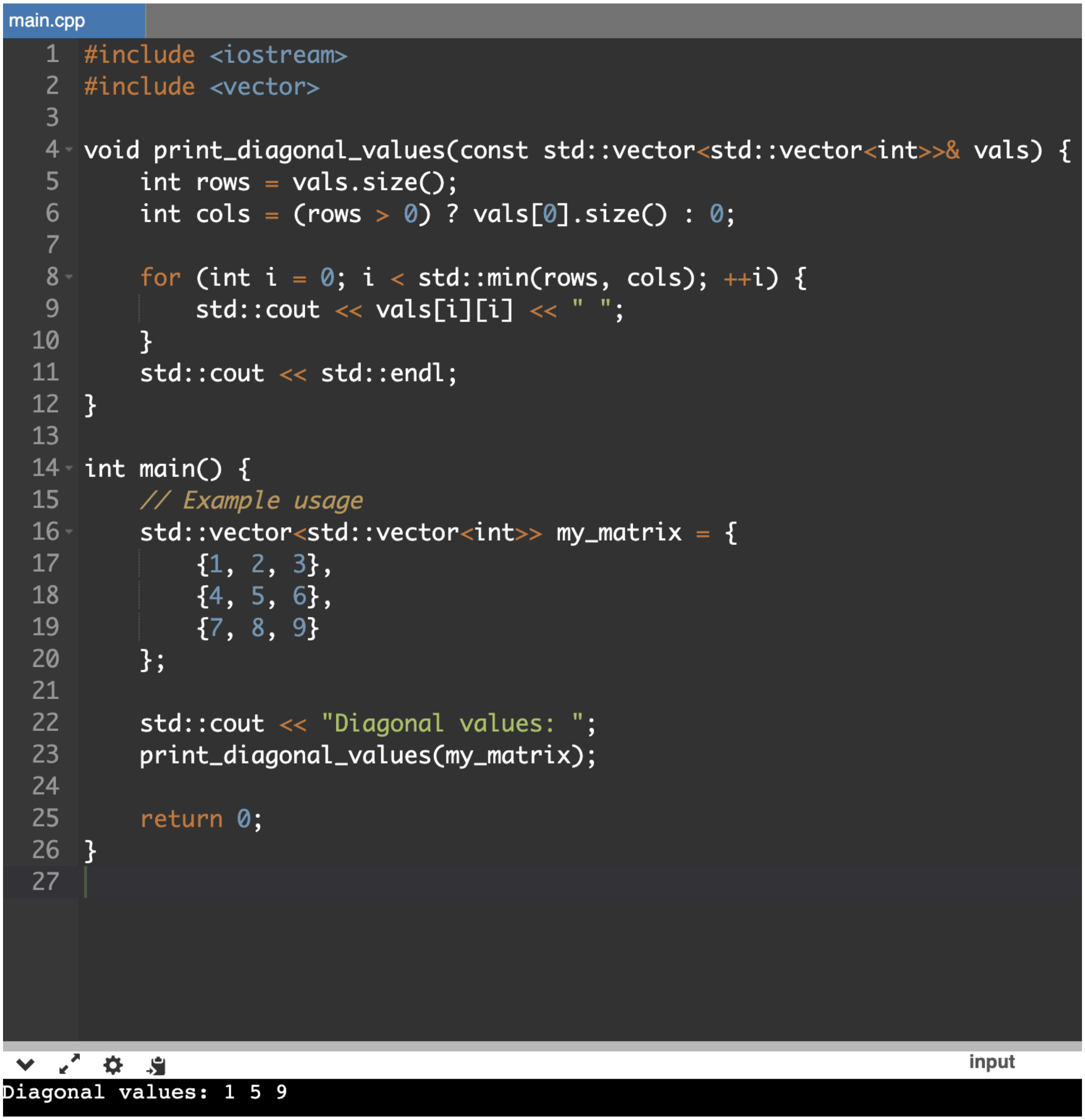
Question 1:



```cpp
#include <iostream>
#include <vector>

// Function to reverse the elements of a vector
void rvrs(std::vector<int>& vct) {
    int left = 0;
    int right = vct.size() - 1;

    while (left < right) {
        // Swap elements at left and right indices
        std::swap(vct[left], vct[right]);
        left++;
        right--;
    }
}

int main() {
    // Example usage
    std::vector<int> myVector = {1, 2, 3, 4, 5};
    rvrs(myVector);

    // Print reversed vector
    std::cout << "Reversed vector: ";
    for (int num : myVector) {
        std::cout << num << " ";
    }
    std::cout << std::endl;

    return 0;
}

```

```
Reversed vector: 5 4 3 2 1
```

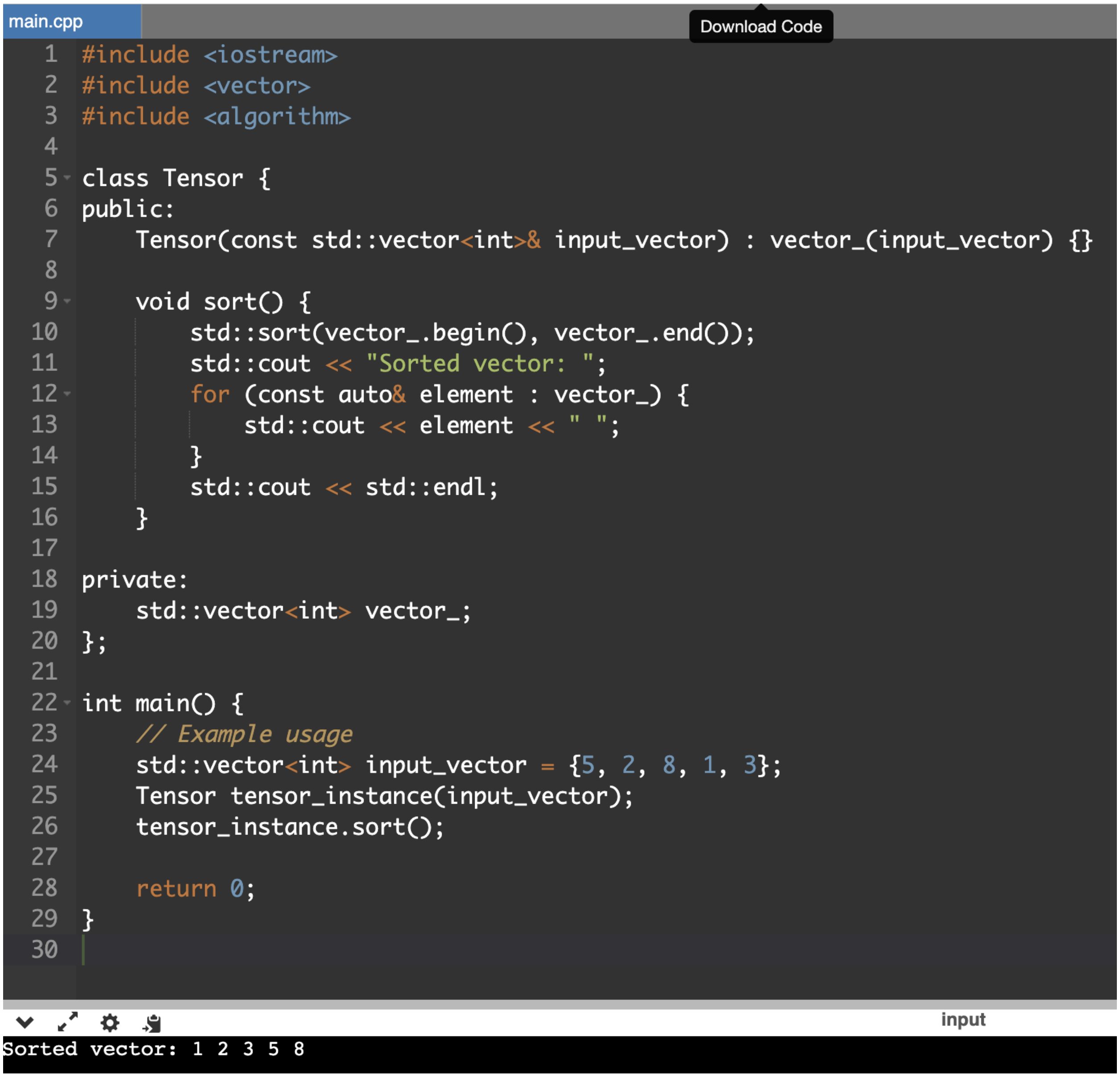
Question 2:



```cpp
#include <iostream>
#include <vector>

void print_diagonal_values(const std::vector<std::vector<int>>& vals) {
    int rows = vals.size();
    int cols = (rows > 0) ? vals[0].size() : 0;

    for (int i = 0; i < std::min(rows, cols); ++i) {
        std::cout << vals[i][i] << " ";
    }
    std::cout << std::endl;
}

int main() {
    // Example usage
    std::vector<std::vector<int>> my_matrix = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    std::cout << "Diagonal values: ";
    print_diagonal_values(my_matrix);

    return 0;
}

```

```
Diagonal values: 1 5 9
```

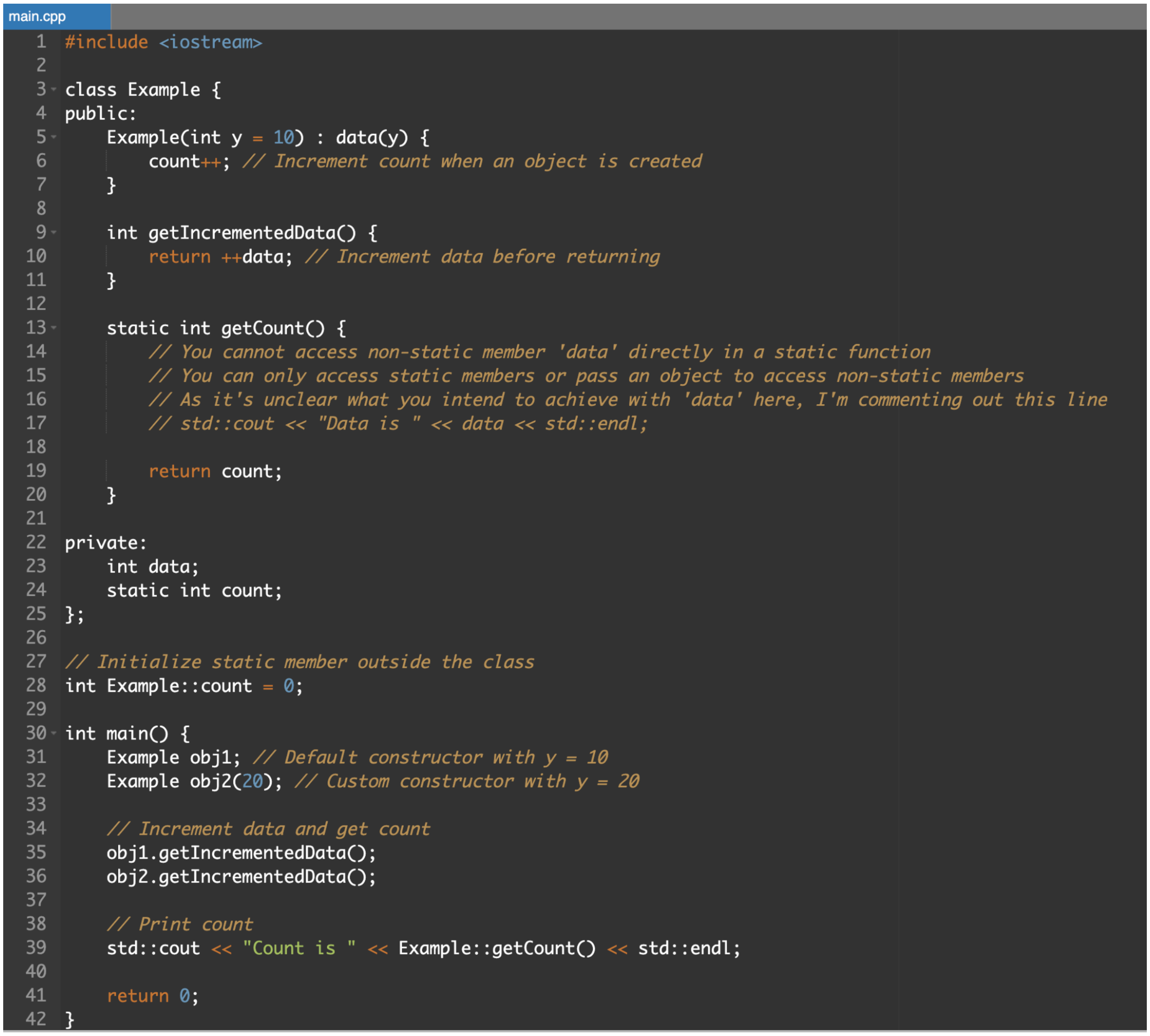
Question 3:



```cpp
#include <iostream>
#include <vector>
#include <algorithm>

class Tensor {
public:
    Tensor(const std::vector<int>& input_vector) : vector_(input_vector) {}

    void sort() {
        std::sort(vector_.begin(), vector_.end());
        std::cout << "Sorted vector: ";
        for (const auto& element : vector_) {
            std::cout << element << " ";
        }
        std::cout << std::endl;
    }

private:
    std::vector<int> vector_;
};

int main() {
    // Example usage
    std::vector<int> input_vector = {5, 2, 8, 1, 3};
    Tensor tensor_instance(input_vector);
    tensor_instance.sort();

    return 0;
}
```

```
Sorted vector: 1 2 3 5 8
```

Question 4:



```cpp
#include <iostream>

class Example {
public:
    Example(int y = 10) : data(y) {
        count++; // Increment count when an object is created
    }

    int getIncrementedData() {
        return ++data; // Increment data before returning
    }

    static int getCount() {
        // You cannot access non-static member 'data' directly in a static function
        // You can only access static members or pass an object to access non-static members
        // As it's unclear what you intend to achieve with 'data' here, I'm commenting out this line
        // std::cout << "Data is " << data << std::endl;

        return count;
    }

private:
    int data;
    static int count;
};

// Initialize static member outside the class
int Example::count = 0;

int main() {
    Example obj1; // Default constructor with y = 10
    Example obj2(20); // Custom constructor with y = 20

    // Increment data and get count
    obj1.getIncrementedData();
    obj2.getIncrementedData();

    // Print count
    std::cout << "Count is " << Example::getCount() << std::endl;

    return 0;
}
```