Khoi Duong

Prof. Yang

CS360L

6/18/2022

LAB#5

1.

Source code:

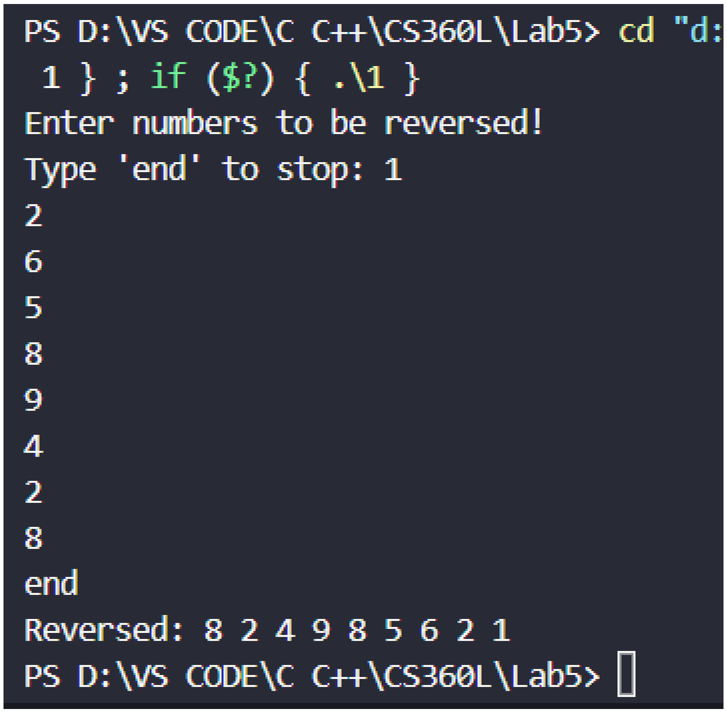
```
#include <iostream>
#include <vector>
using namespace std;

// create a function that takes a vector of integers and reverses its elements
void rvrs(vector<int>& vct)
{
    int size = vct.size();
    for (int i = 0; i < size / 2; i++)
    {
        int temp = vct[i];
        vct[i] = vct[size - i - 1];
        vct[size - i - 1] = temp;
    }
}

int main(){
    vector<int> vct;
    int num;
    cout << "Enter numbers to be reversed!\n";
    cout << "Type 'end' to stop: ";
    while (cin >> num)
    {
        vct.push_back(num);
    }
    rvrs(vct);
    cout << "Reversed: ";
    for (int i = 0; i < vct.size(); i++)
    {
        cout << vct[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Run program & result:



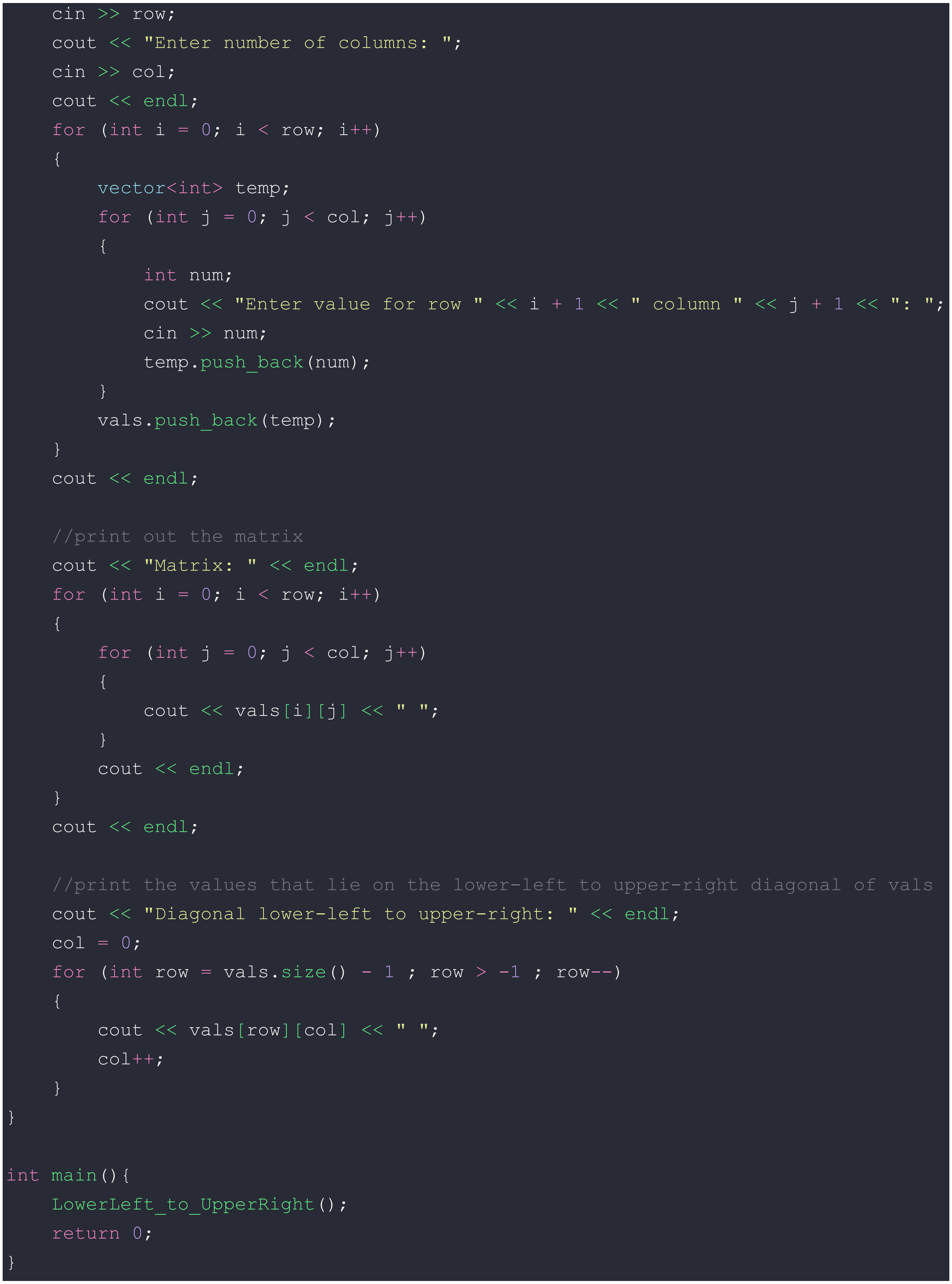
```
PS D:\VS CODE\C C++\CS360L\Lab5> cd "d:
 1 } ; if ($?) { .\1 }
Enter numbers to be reversed!
Type 'end' to stop: 1
2
6
5
8
9
4
2
8
end
Reversed: 8 2 4 9 8 5 6 2 1
PS D:\VS CODE\C C++\CS360L\Lab5> 
```

2.

Source code:

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

void LowerLeft_to_UpperRight()
{
    int row, col;
    //create a vector of vectors named vals
    vector<vector<int>> vals;
    //input the values into vals
    cout << "Enter number of rows: ";
```



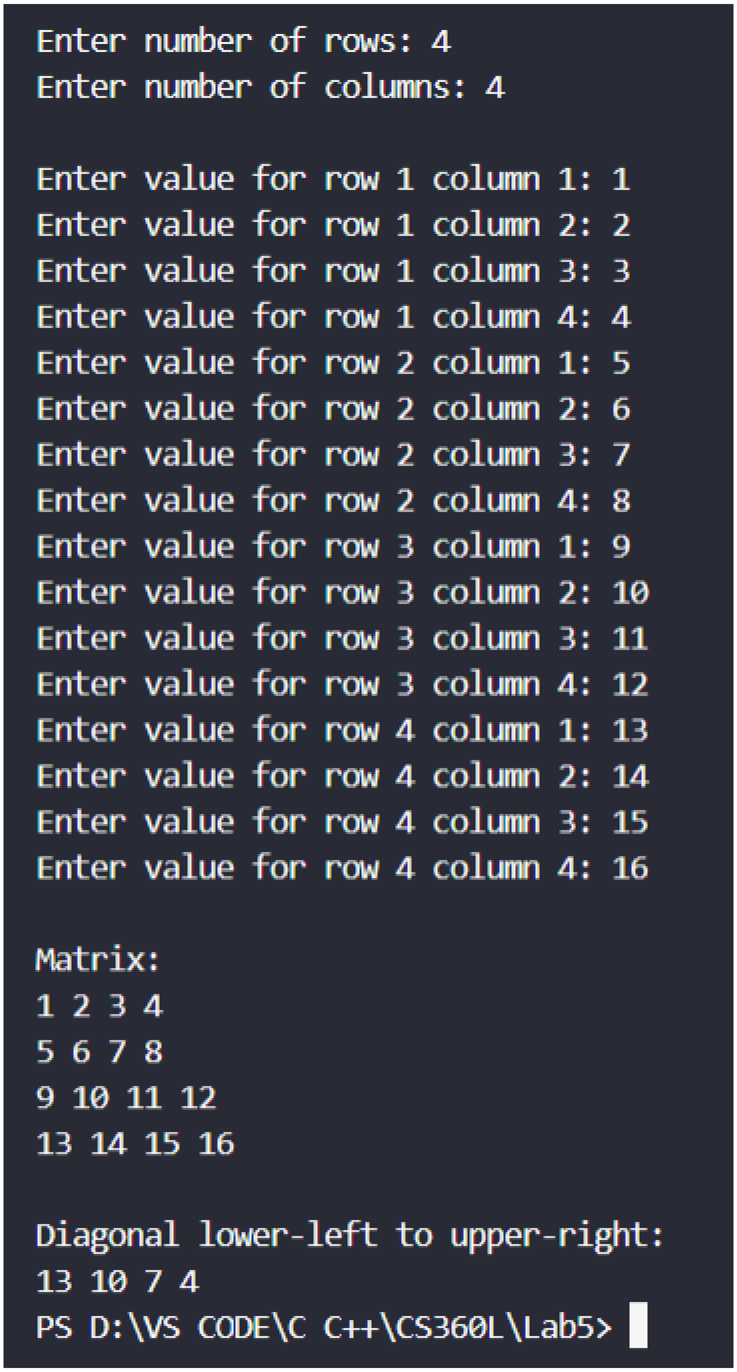
```
    cin >> row;
    cout << "Enter number of columns: ";
    cin >> col;
    cout << endl;
    for (int i = 0; i < row; i++)
    {
        vector<int> temp;
        for (int j = 0; j < col; j++)
        {
            int num;
            cout << "Enter value for row " << i + 1 << " column " << j + 1 << ": ";
            cin >> num;
            temp.push_back(num);
        }
        vals.push_back(temp);
    }
    cout << endl;

    //print out the matrix
    cout << "Matrix: " << endl;
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            cout << vals[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;

    //print the values that lie on the lower-left to upper-right diagonal of vals
    cout << "Diagonal lower-left to upper-right: " << endl;
    col = 0;
    for (int row = vals.size() - 1 ; row > -1 ; row--)
    {
        cout << vals[row][col] << " ";
        col++;
    }
}

int main(){
    LowerLeft_to_UpperRight();
    return 0;
}
```

Run program & result:



```
Enter number of rows: 4
Enter number of columns: 4

Enter value for row 1 column 1: 1
Enter value for row 1 column 2: 2
Enter value for row 1 column 3: 3
Enter value for row 1 column 4: 4
Enter value for row 2 column 1: 5
Enter value for row 2 column 2: 6
Enter value for row 2 column 3: 7
Enter value for row 2 column 4: 8
Enter value for row 3 column 1: 9
Enter value for row 3 column 2: 10
Enter value for row 3 column 3: 11
Enter value for row 3 column 4: 12
Enter value for row 4 column 1: 13
Enter value for row 4 column 2: 14
Enter value for row 4 column 3: 15
Enter value for row 4 column 4: 16

Matrix:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

Diagonal lower-left to upper-right:
13 10 7 4
PS D:\VS CODE\C C++\CS360L\Lab5>
```

3.

Source code:

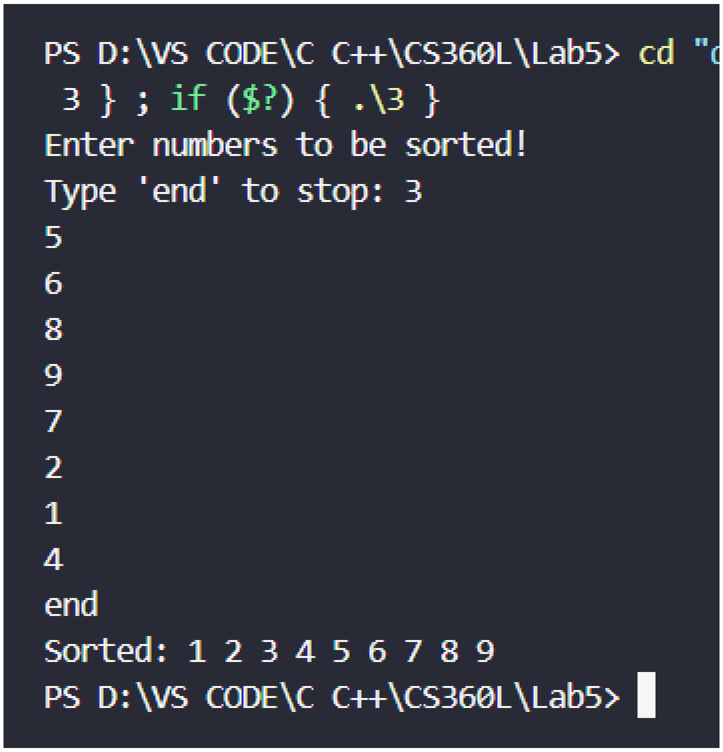
```
#include <iostream>
#include <vector>
using namespace std;
```

```cpp
//create a class Tensor with a method sort to sort a vector input argument and print it out
class Tensor
{
public:
    void sort(vector<int>& vct)
    {
        int size = vct.size();
        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j < size - 1; j++)
            {
                if (vct[j] > vct[j + 1])
                {
                    int temp = vct[j];
                    vct[j] = vct[j + 1];
                    vct[j + 1] = temp;
                }
            }
        }
    }
};

int main(){
    vector<int> vct;
    int num;
    cout << "Enter numbers to be sorted!\n";
    cout << "Type 'end' to stop: ";
    while (cin >> num)
    {
        vct.push_back(num);
    }
    Tensor tensor;
    tensor.sort(vct);
    cout << "Sorted: ";
    for (int i = 0; i < vct.size(); i++)
    {
        cout << vct[i] << " ";
    }
    cout << endl;

    return 0;
}
```

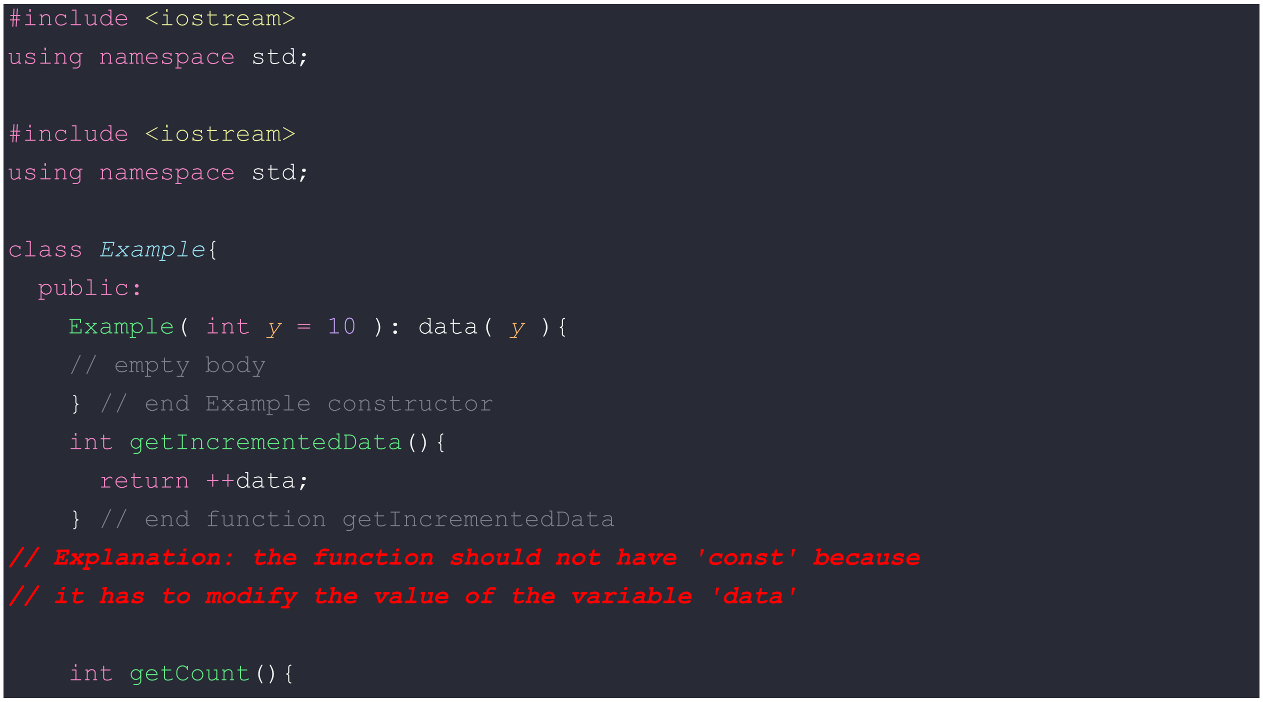
Run program & result:



```
PS D:\VS CODE\C C++\CS360L\Lab5> cd "
 3 } ; if ($?) { .\3 }
Enter numbers to be sorted!
Type 'end' to stop: 3
5
6
8
9
7
2
1
4
end
Sorted: 1 2 3 4 5 6 7 8 9
PS D:\VS CODE\C C++\CS360L\Lab5>
```

4.

Source code after fix (including explanation)

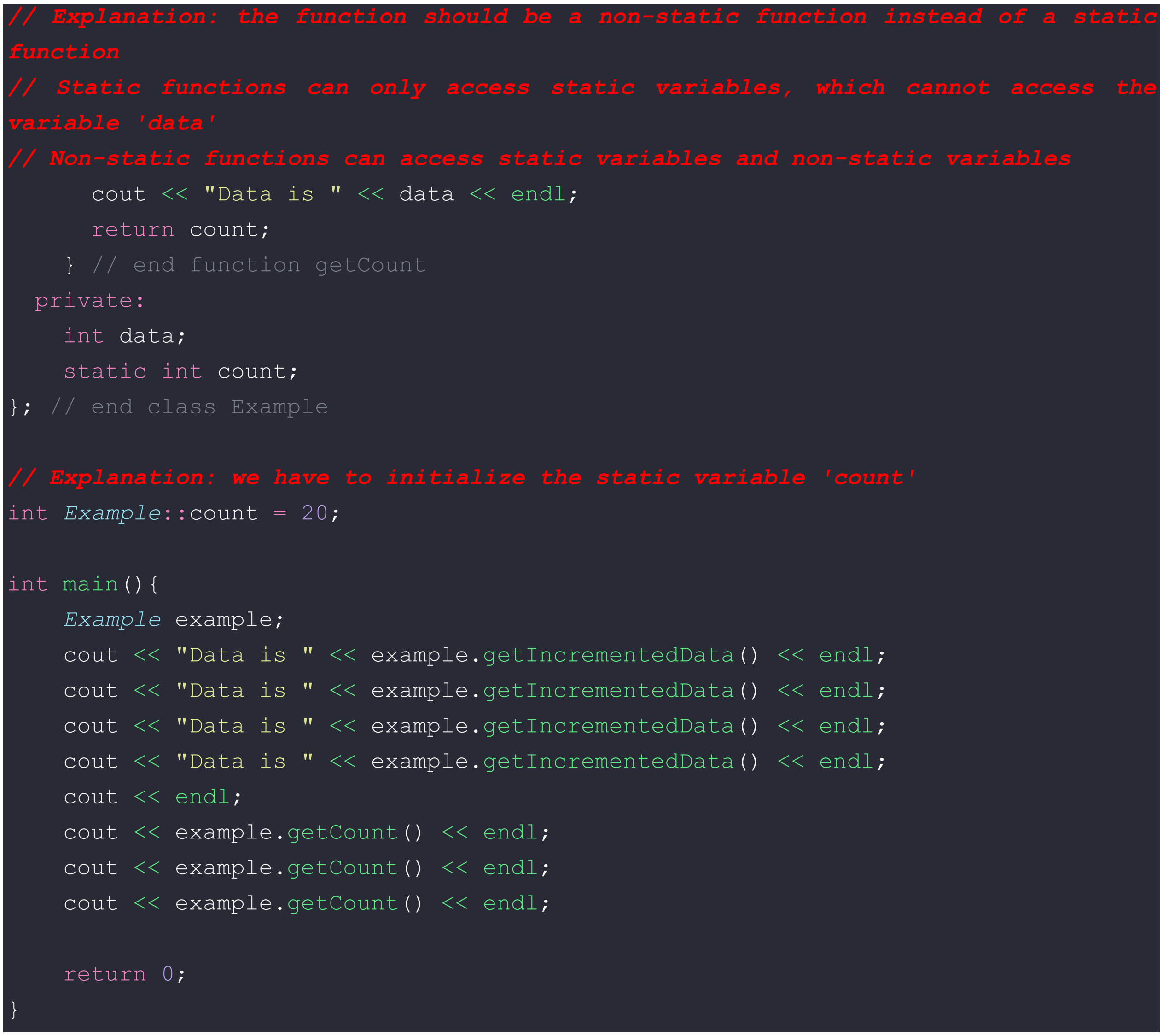


```
#include <iostream>
using namespace std;

#include <iostream>
using namespace std;

class Example{
  public:
    Example( int y = 10 ): data( y ){
    // empty body
    } // end Example constructor
    int getIncrementedData(){
      return ++data;
    } // end function getIncrementedData
// Explanation: the function should not have 'const' because
// it has to modify the value of the variable 'data'

    int getCount(){
```



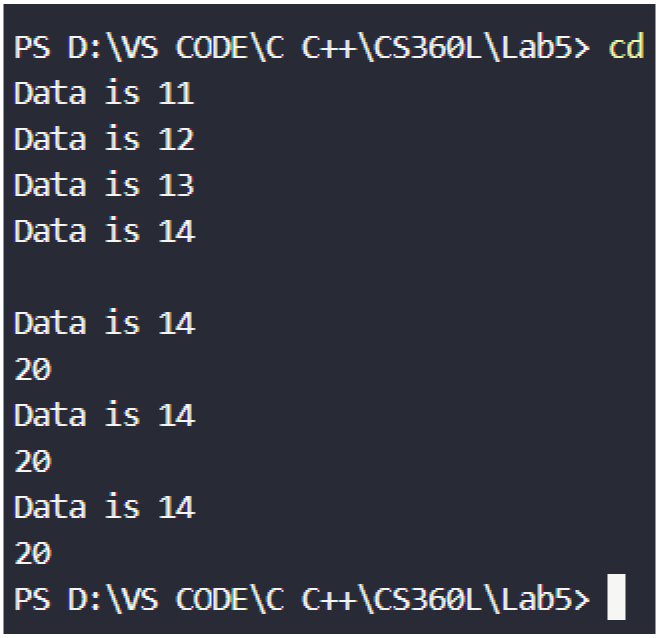
```
// Explanation: the function should be a non-static function instead of a static
function
// Static functions can only access static variables, which cannot access the
variable 'data'
// Non-static functions can access static variables and non-static variables
      cout << "Data is " << data << endl;
      return count;
    } // end function getCount
  private:
    int data;
    static int count;
}; // end class Example

// Explanation: we have to initialize the static variable 'count'
int Example::count = 20;

int main(){
    Example example;
    cout << "Data is " << example.getIncrementedData() << endl;
    cout << "Data is " << example.getIncrementedData() << endl;
    cout << "Data is " << example.getIncrementedData() << endl;
    cout << "Data is " << example.getIncrementedData() << endl;
    cout << endl;
    cout << example.getCount() << endl;
    cout << example.getCount() << endl;
    cout << example.getCount() << endl;

    return 0;
}
```

Run program & result:



```
PS D:\VS CODE\C C++\CS360L\Lab5> cd
Data is 11
Data is 12
Data is 13
Data is 14

Data is 14
20
Data is 14
20
Data is 14
20
PS D:\VS CODE\C C++\CS360L\Lab5>
```