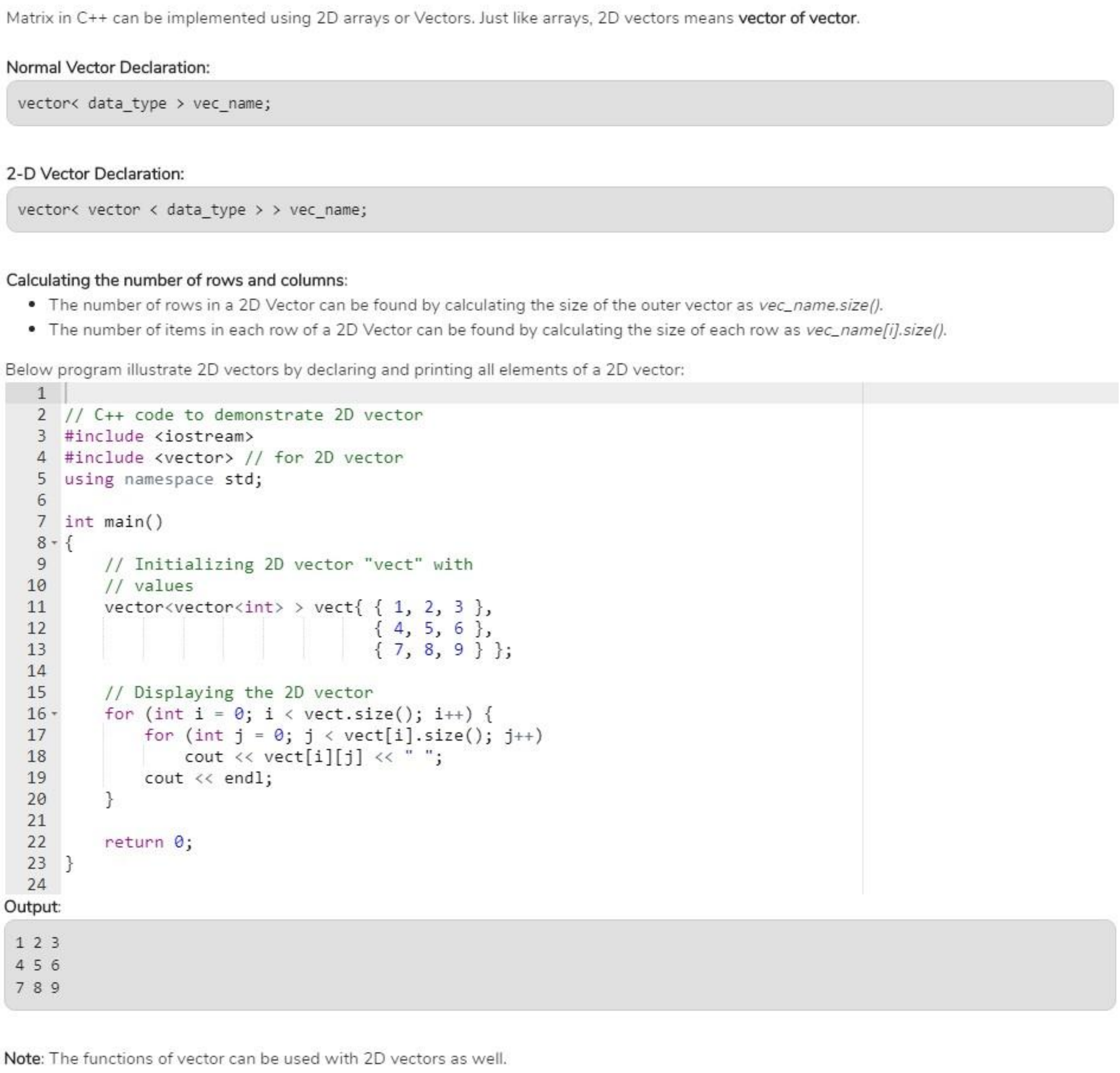
2D vector in C++



Matrix in C++ can be implemented using 2D arrays or Vectors. Just like arrays, 2D vectors means **vector of vector**.

Normal Vector Declaration:

```
vector< data_type > vec_name;
```

2-D Vector Declaration:

```
vector< vector < data_type > > vec_name;
```

Calculating the number of rows and columns:

- The number of rows in a 2D Vector can be found by calculating the size of the outer vector as *vec_name.size()*.
- The number of items in each row of a 2D Vector can be found by calculating the size of each row as *vec_name[i].size()*.

Below program illustrate 2D vectors by declaring and printing all elements of a 2D vector:

```

// C++ code to demonstrate 2D vector
#include <iostream>
#include <vector> // for 2D vector
using namespace std;

int main()
{
    // Initializing 2D vector "vect" with
    // values
    vector<vector<int> > vect{ { 1, 2, 3 },
                               { 4, 5, 6 },
                               { 7, 8, 9 } };

    // Displaying the 2D vector
    for (int i = 0; i < vect.size(); i++) {
        for (int j = 0; j < vect[i].size(); j++)
            cout << vect[i][j] << " ";
        cout << endl;
    }

    return 0;
}

```

Output:

```
1 2 3
4 5 6
7 8 9
```

Note: The functions of vector can be used with 2D vectors as well.