Rubik's Cube Implementation in C++

Your task is to implement a Rubik's Cube using classes and pointers in C++. Your Rubik's Cube should be able to perform the following two functionalities:

1. **Swap Full Side:** You should be able to swap one full side of the Rubik's Cube with another full side. For instance, you could swap the front side with the right side or the top side with the bottom side.

2. **Rotate Left Column:** You should be able to rotate the left most column of the Rubik's Cube in the Rubik's Cube fashion. This means that the colors on the left most column should rotate in such a way that the top color moves to the left, the left color moves to the bottom, the bottom color moves to the right, and the right color moves to the top.

You should use classes and pointers to implement the Rubik's Cube. Your Rubik's Cube should have a `Cube` class that contains six `Face` objects, each of which represents one of the six sides of the Rubik's Cube. Each `Face` object should have a 3×3 array of `int` objects that represents the colors on that side of the Rubik's Cube. A mapping is provided which represents the six colors that can appear on a Rubik's Cube: red, green, blue, yellow, white, and orange.

Your `Cube` class should have the following member functions:

- `swap_front_top()`: This function should work on two `Face` objects and swap all the colors (int mapping) on those faces. This will effectively swap the two full sides of the Rubik's Cube. (Hint: Since this class composes the face class, so you can use the methods of that class.)

- `rotate_left_column()`: This function should rotate the left most column of the Rubik's Cube in the Rubik's Cube fashion. To do this, you should

swap the colors on the top, left, bottom, and right sides of the Rubik's Cube as described above.

Your `main` function should create a `Cube` object and test its functionality by swapping two full sides and rotating the left most column. You will need to implement `print` function(s) to validate the working of your Rubik's Cube.

Here's a skeleton code that you can use to get started:

```
#include <iostream>
#include <cstdlib>

using namespace std;

// Color Space
// 1: Red
// 2: Blue
// 3: Green
// 4: White
// 5: Yellow
// 6: Orange

const int colors = 6;

class Face {
        int** side;
        const int size = 3;
public:
        // TODO: Implement Face class

        Face() {}

        ~Face() {}

        void exposeSide() {}

        void swapSide(Face one) {}

};

class Cube {
        Face* front;
        Face* back;
```

```
        Face* left;
        Face* right;
        Face* top;
        Face* bottom;
public:
        // TODO: Implement Cube class

        Cube(){}

        ~Cube() {}

        void swap_front_top(){}

        void rotate_left_column() {}

};

int main() {
        srand(static_cast<unsigned int>(time(0)));

        // TODO: Test functionality of Cube class

        return 0;
}
```

In the `Face` class, you should implement the 3×3 array of `int` objects and any member functions that you think are necessary. In the `Cube` class, you should implement the two member functions provided above, as well as any member variables and other member functions that you think are necessary.

To swap two full sides, you may simply swap the `Face` objects themselves. To rotate the left most column, you should swap the appropriate `integers` that map out the colors on the top, left, bottom, and right sides of the Rubik's Cube.

Good luck and have fun!
