```
#include <iostream>

enum Direction { UP, DOWN, STOP };

class Lift {
private:
    int currentFloor;
    Direction direction;
    bool doorOpen;
    bool emergencyStop;
    bool overloaded;

public:
    Lift() : currentFloor(0), direction(STOP), doorOpen(false), emergencyStop(false), overloaded(false) {}

    void goToFloor(int floor) {
        if (emergencyStop) {
            std::cout << "Emergency stop activated. Lift is halted." << std::endl;
            return;
        }

        if (overloaded) {
            std::cout << "Lift is overloaded. Cannot move until excess weight is removed." << std::endl;
            return;
        }

        if (floor == currentFloor) {
            std::cout << "Already on floor " << floor << std::endl;
            return;
        }

        direction = floor < currentFloor ? DOWN : UP;

        while (currentFloor != floor) {
            std::cout << (direction == UP ? "Moving UP. " : "Moving DOWN. ") << "Now on floor " <<
currentFloor << std::endl;
            direction == UP ? currentFloor++ : currentFloor--;
        }


        std::cout << "Door opened" << std::endl;
        std::cout << "Door will close shortly..." << std::endl;

        std::cout << "Door closed" << std::endl;

        notifyFloorArrival(floor);
    }

    void closeDoor() {
        doorOpen = false;
        std::cout << "Door closed" << std::endl;
    }
```

```
    void activateEmergencyStop() {
        emergencyStop = true;
        std::cout << "Emergency stop activated." << std::endl;
    }

    void deactivateEmergencyStop() {
        emergencyStop = false;
        std::cout << "Emergency stop deactivated." << std::endl;
    }

    void reportOverload() {
        overloaded = true;
        std::cout << "Overload detected. Lift is halted until excess weight is removed." << std::endl;
    }

    void removeOverload() {
        overloaded = false;
        std::cout << "Excess weight removed. Lift is now operational." << std::endl;
    }

    void communicateWithEmergencyServices() {
        std::cout << "Communicating with emergency services..." << std::endl;
        std::cout << "Emergency services notified." << std::endl;
    }

    void notifyFloorArrival(int floor) {
        std::cout << "Arrived at floor " << floor << std::endl;
    }
};

int main() {
    Lift lift;

    lift.goToFloor(5);
    lift.activateEmergencyStop();
    lift.goToFloor(3);
    lift.deactivateEmergencyStop();
    lift.reportOverload();
    lift.goToFloor(3);
    lift.removeOverload();
    lift.communicateWithEmergencyServices();

    return 0;
}
```