```cpp
#pragma once

#include <iostream>
#include <chrono>

#include "door.hpp"

/// CHECKED FOR THREAD SAFTEY //

enum FloorNum {
    FG = 0,
    F1,
    F2,
    F3
};

class invalid_floor_reached : public std::exception {
public:
    std::string what() {
        return "Umm, ya you killed people. Invalid floor reached";
    }
};

/// I know I know What am I doing here, Do I even know C++.
/// Ill fix it later
inline void inc(std::atomic<FloorNum>& a) {
    switch (a) {
        case FG: a = F1; return;
        case F1: a = F2; return;
        case F2: a = F3; return;
        default:         throw invalid_floor_reached();
    }
}

inline void dec(std::atomic<FloorNum>& a) {
    switch (a) {
        case F1: a = FG; return;
        case F2: a = F1; return;
        case F3: a = F2; return;
        default:         throw invalid_floor_reached();
    }
}

// inline void operator++ (FloorNum& a) {
    // switch (a) {
        // case FG: a = F1; return;
        // case F1: a = F2; return;
        // case F2: a = F3; return;
        // default:         throw invalid_floor_reached();
    // }
// }

// inline void operator++ (FloorNum& a, int) {
    // switch (a) {
        // case FG: a = F1; return;
        // case F1: a = F2; return;
        // case F2: a = F3; return;
        // default:         throw invalid_floor_reached();
    // }
// }

// inline void operator-- (FloorNum& a) {
    // switch (a) {
        // case F1: a = FG; return;
        // case F2: a = F1; return;
        // case F3: a = F2; return;
        // default:         throw invalid_floor_reached();
    // }
// }
```

```
70
71  // inline void operator-- (FloorNum& a, int) {
72      // switch (a) {
73          // case F1: a = FG; return;
74          // case F2: a = F1; return;
75          // case F3: a = F2; return;
76          // default:         throw invalid_floor_reached();
77      // }
78  // }
79
80  enum ElevState {
81      ES_WAIT = 0b00,
82      ES_DOWN = 0b01,
83      ES_UP = 0b10,
84      ES_DC = 0b11 // wont be used (dont care)
85  };
86
87  inline std::string pretty(ElevState s) {
88      switch (s) {
89          case ES_WAIT:
90              return "Waiting";
91          case ES_DOWN:
92              return "Down";
93          case ES_UP:
94              return "Up";
95          default:
96              return "Broken";
97      }
98  }
99
100 struct Elevator {
101     std::atomic<FloorNum> mFloor;
102     std::atomic<ElevState> mState;
103     Door mDoor;
104     std::thread mThread;
105
106     std::atomic_bool mStop;
107
108     Elevator();
109     ~Elevator();
110
111     void start();
112     void reset(FloorNum flr);
113
114 };
```