Lab 8.4.1 Operators: representing values and values order

Objectives

Familiarize the student with:

- representing values with enums;
- writing overload operator code;
- using operators to operate on class content.

Scenario

Write a finite-state machine which has four or more states, and use enums to encapsulate the states. Add an overloaded operator to move into the next state. The next state depends on the value passed to the FSM (with the overload operator). A finite-state machine represents a model of computation. It's conceived as an abstract machine that can be in one of a finite number of states. An FSM is in only one state at a time; this state is called the current state. The machine can change from one state to another when initiated by a triggering condition; this operation is called a transition. Our exemplary FSM is defined by a list of six states; one state for start and stop; and four indirect states in two layers. You can define your own transition rules. If you have no idea how to define it, then you should just move to one state when the value is lower than 5, and go to another state when the value is equal to or greater than 5.

Example input

```
fsm<<3;
fsm<<6;
fsm<<4;
```

Example output

```
Stop state reached
States visited:
1(Start), 3, 4, 6(Stop)
```