CSCE 317 Spring 2016 Assignment 2
Due: Wednesday, April 12, 2017

You are asked write a C++ program to implement a finite state machine (FSM) simulator. The objective is to enable system designers to model and simulate their systems' behavior while still in the design phase. The simulator reads an input file which contains the FSM description. The FSM description file contains three main sections: (1) machine name; (2) variables section; (3) states section; and (4) a transitions section (see example below).

The sections are defined as follows:

1. The machine name section contain the machine name (which must be the same as the file name that contains the machine (a machine called x would be stored in a file called x.fsm).
2. The variable section which starts with the keyword VAR followed by a list of machine variable.
3. The states section starts with the keyword STATES: (on a separate line) followed with state descriptions each on a separate line. Each state description starts with a state name (e.g., a:) followed by a list of actions to be performed while the machine is in this particular state. The state actions can be one of the following:
   a. **`<variable> = <expression> + <expression>`**. Where **`<expression>`** can either be a constant or a machine variable name.
   b. **`out <expression>`** which prints out the **`<expression>`** on the display.
   c. **`out <string>`** which prints out the **`<string>`** on the display.
   d. **`sleep <amount>`** which causes the simulator to pause for **`<amount>`** seconds.
   e. **`wait`** which causes the simulator to wait for the next transition to take place.
   f. **`run <machine>`** which causes the simulator to run machine called <machine>.fsm and the return to continue the rest of the state actions (if any).
   g. **end** which ends the execution of the state machine.

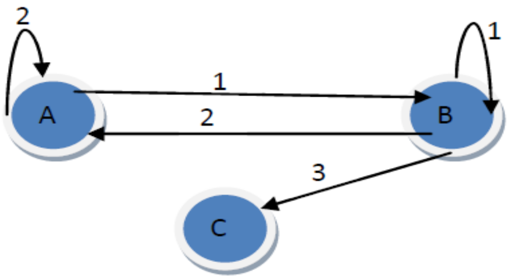
4. The transitions section starts with the keyword TRANSITIONS: (on a separate line) followed with transition descriptions each on a separate line. Each transition is a list of three elements: a source state, a destination state, and the input value that causes this transition to take place.

You should submit the following items:

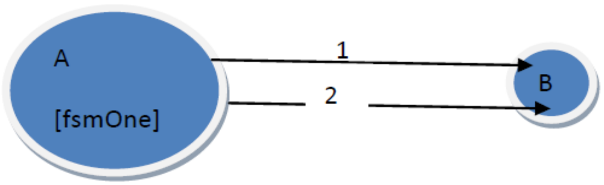
(a) **Class diagram showing the structure of the classes in your simulator;**
(b) **The documented C++ source code;**
(c) **Sample test FSMs to be run on your simulator; and**
(d) **A short users' guide explaining how to compile and use your simulator.**

Example FSMs:

```
FSM fsmOne
VAR X, Y
States:
a: out “state A”, X=X+1, sleep 10, wait
b: out “state B”, Y=Y+1, sleep 10, wait
c: out “thank you for using fsm1”, out X, out Y, end

Transitions:

a, b, 1
b, a, 2
a, a, 2
b, b, 1
b, c, 3
```



```
FSM fsmTwo
States:
a: out “state A machine 2”, sleep 10, run fsmOne, wait
b: out “thank you for using fsmTwo”, end

Transitions:
a, b, 1
a, b, 2
```


A
[fsmOne]
1
2
B