```cpp
#include<iostream>
using namespace std;
// *** Base State (Abstract Class) ***
class State
{
        protected:
        State * const next_a, * const next_b; // Pointers to next state
        char output;
        public:
        State( State & a, State & b):next_a(&a),next_b(&b){}
        virtual State* transition(char)=0;
};
// *** State1 ***
class State1:public State
{
        public:
        State1( State & a, State & b):State(a,b){}
        State* transition(char);
};
// *** State2 ***
class State2:public State
{
        public:
        State2( State & a, State & b):State(a,b){}
        State* transition(char);
};
/*** State3 ***/
class State3:public State
{
        public:
        State3( State & a, State & b):State(a,b){}
        State* transition(char);
};
State* State1::transition(char input)
{
        cout << endl << "Current State: 1";
        switch(input){
        case 'a': output='y';
        cout << endl << "Output: "<< output;
        cout << endl << "Next State: 1";
        return next_a;
        case 'b': output='x';
        cout << endl << "Output: "<< output;
        cout << endl << "Next State: 2";
        return next_b;
        default : cout << endl << "Undefined input";
```

```
	cout << endl << "Next State: Unchanged";
	return this;
	}
}
State* State2::transition(char input)
{
	cout << endl << "Current State: 2";
	switch(input){
	case 'a': output='x';
	cout << endl << "Output: "<< output;
	cout << endl << "Next State: 3";
	return next_a;
	case 'b': output='y';
	cout << endl << "Output: "<< output;
	cout << endl << "Next State: 2";
	return next_b;
	default : cout << endl << "Undefined input";
	cout << endl << "Next State: Unchanged";
	return this;
	}
}
State* State3::transition(char input)
{
	cout << endl << "Current State: 3";
	switch(input){
	case 'a': output='y';
	cout << endl << "Output: "<< output;
	cout << endl << "Next State: State1";
	return next_a;
	case 'b': output='x';
	cout << endl << "Output: "<< output;
	cout << endl << "Next State: 2";
	return next_b;
	default : cout << endl << "Undefined input";
	cout << endl << "Next State: Unchanged";
	return this;
	}
}
// *** Finite State Machine ***
// This class has 3 State objects as members
class FSM
{
	State1 s1;
	State2 s2;
	State3 s3;
	State * current;
```

```
        public:
        FSM():s1(s1,s2),s2(s3,s2),s3(s1,s2),current(&s1) {}
        void run();
};
void FSM::run(){
        char in;
        cout << endl << "The finite state machine starts ...";
        do{
        cout << endl << "Give the input value (a or b; x:EXIT) "; cin >> in;
        if (in != 'x')
        current = current->transition(in);
        else
        current = 0; // EXIT
        }
        while(current);
        cout << endl << "The finite state machine stops ..." << endl;;
}

int main()
{
        FSM machine1;
        machine1.run();
        return 0;
}
```