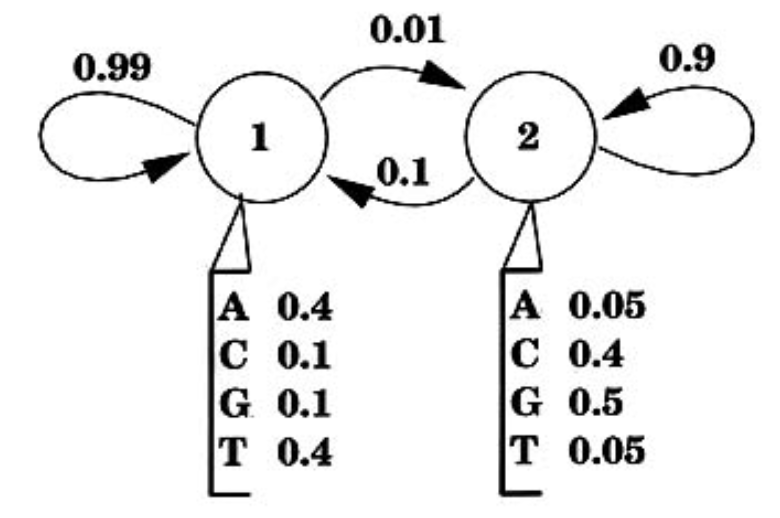
**Logitech HW08**

1. Please implement the following hidden Markov model using the state design pattern:



Hidden state: state1 and state2

Transition prob.: state1->state1: 0.99; state1->state2: 0.01, etc.

Emission prob.: in state1: A:0.4, C:0.1, G:0.1, T: 0.4; in state2: A: 0.05, c: 0.4, etc.

Int main()

{

string symbol = “ATCGTACG”;

string state = “11221121”;

//calculate the probability of getting the symbol sequence by the given states using the HMM above. The answer should be the product of that as follows: 1(initial) \* 0.4(state1 emits A) \* 0.99 (state1->state1) \* 0.4 (state1 emits T) \* 0.01 (state1->state2) \* 0.4 (state2 emits C) …; Please just show the results without the calculation process. You might want to add the begin and end states for the start and end of the symbol string (with prob. = 1).

}

1. Visitor

Test1

{

std::vector<Vistable\_Item\*> items;

items.push\_back (new Square(10));

item.push\_back(new Circle(5));

visitor\* v1 = new get\_area\_vistor;

EXPECTEQ(items[0].accept(v),100);

EXPECTEQ(items[0].accept(v), 78.5);

}

Test2

{

std::vector<Vistable\_Item\*> items;

items.push\_back (new Square(10));

item.push\_back(new Circle(5));

visitor\* v1 = new get\_perimeter\_vistor;

EXPECTEQ(items[0].accept(v),40);

EXPECTEQ(items[0].accept(v), 31.4);

}