| Constructors and Copying
Deep and Shallow Copies | Complete the US State class below so that we can create states in the following way - |

Your Notes:

```
State s1 = new State();

// 52% are democ. voters, 45% are repub. votes. 3% other-

State michigan = new State("Michigan", 0.52, 0.45);

State copyOfMichigan = new State(s2);


class State {
   private String name
   private double dem; // likelihood of democratic result 0..1
   private double repub; // likelihood of republican result 0..1
   double getDem()    {                             }
   double getRepub() {                             }
   String getName()   {                             }
   String toString() {
     return getName() + ":" + getDem() + "," + getRepub();
   }
```

Rewrite the following code to use constructors instead of the *set___* and *copy* methods.

Ghost g1 = new Ghost();
g1.setX(10); g1.setY(20);
g1.setEdible(true);

Ghost g2 = g1.copy();

… and write the two new Ghost constructors for the Ghost class: i) A Constructor that takes 3 parameters and ii) a copy-constructor that takes a reference to another ghost.

```
class Ghost {
  int x,y;
  boolean edible;
```

Make a list of US States.
```
    StateList list = new StateList();
    State michigan = new State("Michigan",0.52, 0.45);
    list.add(michigan);
    ...
```
```
public class StateList {
    private State[] array = new State[0]; // empty array of pointers.
```
Note Each time *add* is called we'll make a larger array.
```
    public State getState(int i) { return this.array[i]; }
    public int getSize() { return this.array.length; }

    public void add(State s) {
      State[] temp = new State[this.array.length + 1];
      for (int i = 0; i < array.length; i++) temp[i] = array[i];

      temp[temp.length - 1] = s;
      this.array = temp; // array pointer now looks at new array
    }

    public void addAll(StateList other) { // Spot the error :-)
      for (int x = 0 ; x < other.length ; x ++)
          add(other.getState(x));
    }
    // returns states where state.repub > 0.5
    public StateList getRepublicanStates() {
        StateList result = new StateList();
        for (int x = 0 ; x < array.length x`; x ++) {
            State state = getState(x);
            if (state.getRepub() > 0.5)
                result.add(state);
        }
        return result;
    }
    // ---- CONSTRUCTORS ----
    public StateList() { // do nothin'
    }
    public StateList(StateList other) {
        array = new State[other.getSize()];
        for(int x = 0 ; x < array.length ; x ++) {
            array[x] = other.getState(x); // SHALLOW COPY

      or      array[x] = _____// DEEP
        }
    }
}
```