	Course Name: Design Patterns/Thinking LAB	**EXPERIMENT NO. 13**	
	Course Code: 20CP210P **Faculty: Dr. Ketan Sabale**	**Branch: CSE**	**Semester: IV**
(To be filled by Student) **Submitted by: Jangle Parth** **Roll no: 22BCP083**			

Objective: To familiarize students with standard Behavioral design patterns.

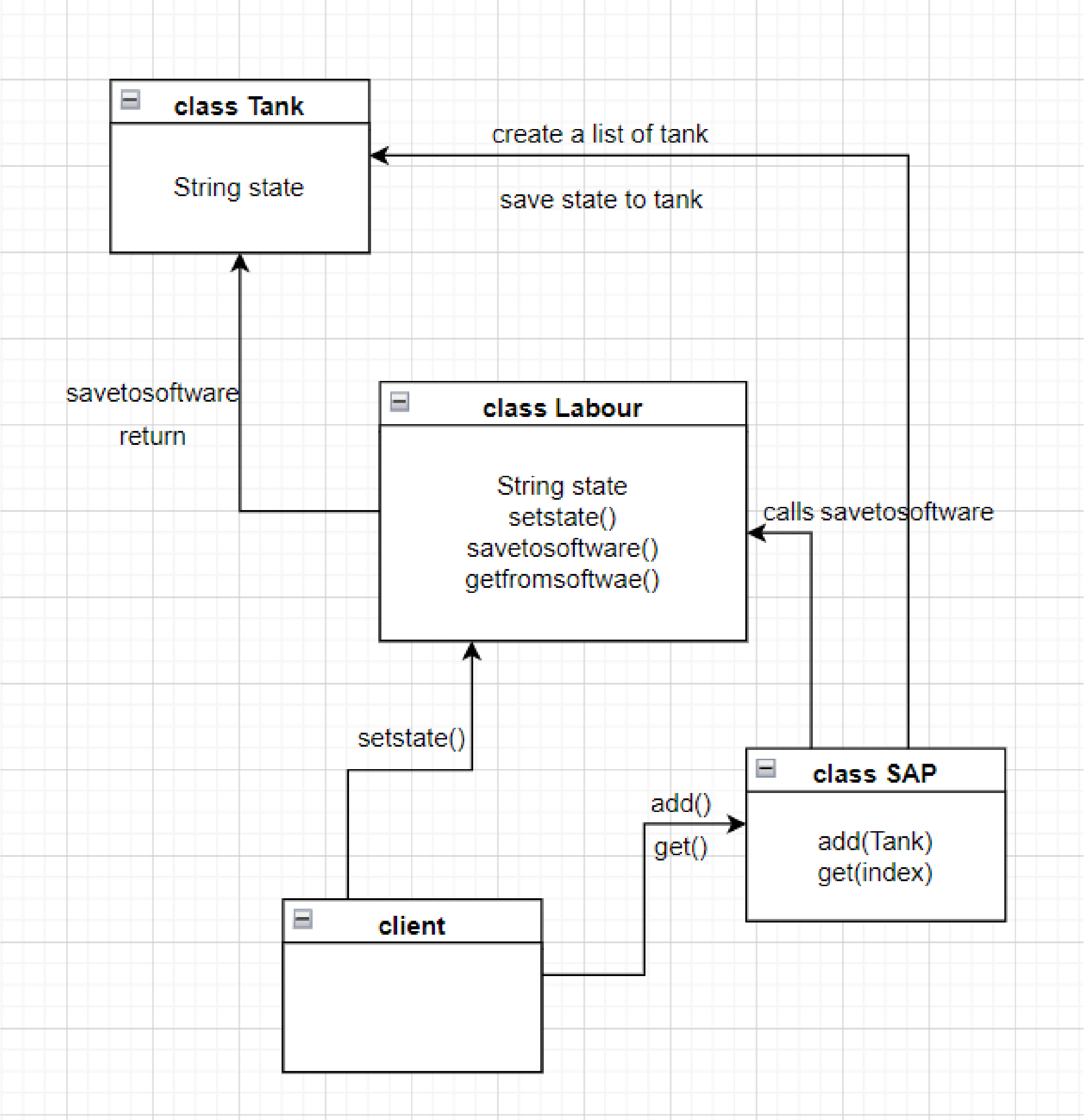
Experiment: Explain the Memento design pattern and write a program using any object-oriented programming language to demonstrate the working of Memento design pattern.

Theory: Memento design helps in saving each and every state of a particular object . example of memento design pattern is GitHub it is used to save changes in your code. It takes help of caretaker to keep track on memento object.

Problem Statement Explanation:

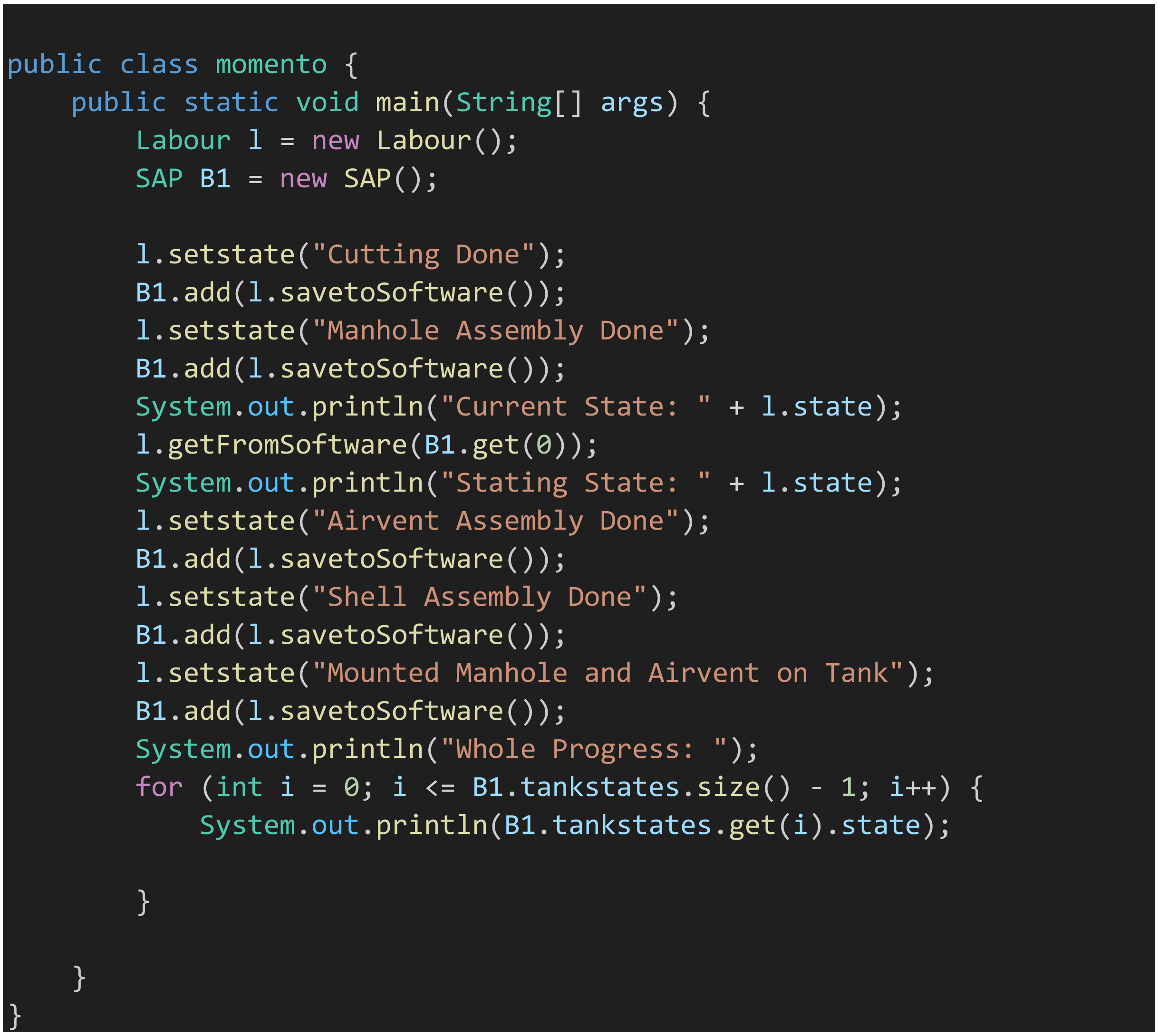
In making of a tank there are many states like material purchased ,shell cutting done, then cone cutting done etc. to save this state we will use memento design patten it has a caretaker class labor which is responsible for changing the state of object i.e. tank and SAP class which saves all the state changes in the code.

Flowchart Explanation:



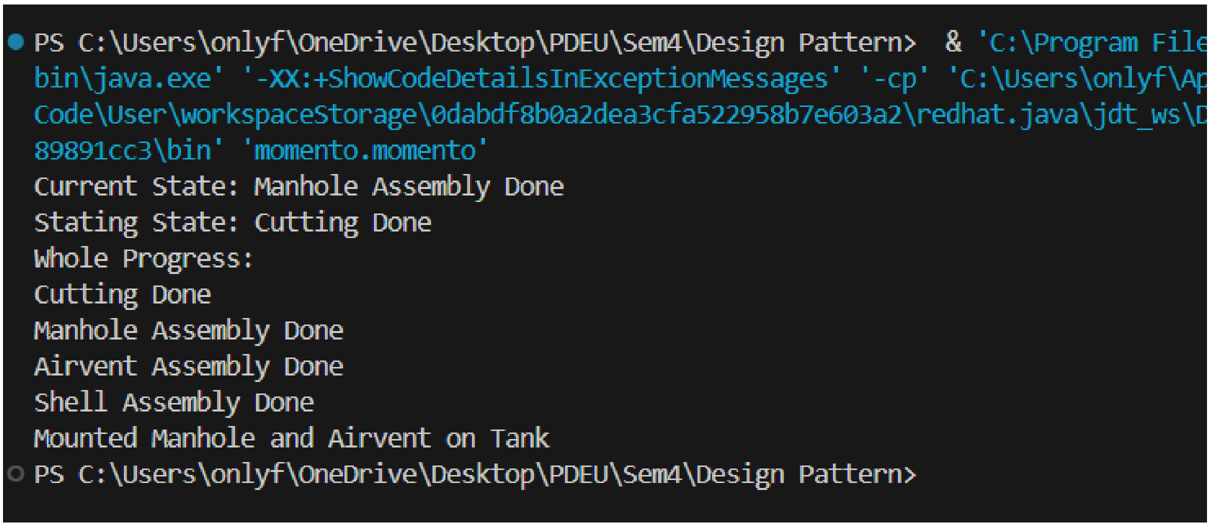
Code:

```java
package momento;

import java.util.ArrayList;

class Tank {
    String state;

    public Tank(String state) {
        this.state = state;
    }
}

class Labour {
    String state;

    public void setstate(String state) {
        this.state = state;
    }

    public Tank savetoSoftware() {
        return new Tank(state);
    }

    public void getFromSoftware(Tank tank) {
        state = tank.state;
    }

}

class SAP {
    ArrayList<Tank> tankstates = new ArrayList<Tank>();

    public void add(Tank state) {
        tankstates.add(state);
    }

    public Tank get(int index) {
        return tankstates.get(index);
    }
}
```



```java
public class momento {
    public static void main(String[] args) {
        Labour l = new Labour();
        SAP B1 = new SAP();

        l.setstate("Cutting Done");
        B1.add(l.savetoSoftware());
        l.setstate("Manhole Assembly Done");
        B1.add(l.savetoSoftware());
        System.out.println("Current State: " + l.state);
        l.getFromSoftware(B1.get(0));
        System.out.println("Stating State: " + l.state);
        l.setstate("Airvent Assembly Done");
        B1.add(l.savetoSoftware());
        l.setstate("Shell Assembly Done");
        B1.add(l.savetoSoftware());
        l.setstate("Mounted Manhole and Airvent on Tank");
        B1.add(l.savetoSoftware());
        System.out.println("Whole Progress: ");
        for (int i = 0; i <= B1.tankstates.size() - 1; i++) {
            System.out.println(B1.tankstates.get(i).state);

        }

    }
}
```

Output:



```
● PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern>  & 'C:\Program File
  bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\onlyf\Ap
  Code\User\workspaceStorage\0dabdf8b0a2dea3cfa522958b7e603a2\redhat.java\jdt_ws\D
  89891cc3\bin' 'momento.momento'
  Current State: Manhole Assembly Done
  Stating State: Cutting Done
  Whole Progress:
  Cutting Done
  Manhole Assembly Done
  Airvent Assembly Done
  Shell Assembly Done
  Mounted Manhole and Airvent on Tank
○ PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern>
```