	Course Name: Design Patterns/Thinking LAB	EXPERIMENT NO. 2	
	Course Code: 20CP210P Faculty: Dr. Ketan Sabale	Branch: CSE	Semester: IV
<p style="text-align: center;">Submitted by: Jangle Parth Roll no: 22BCP083</p>			

Objective: To familiarize students with standard Creational design patterns.

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

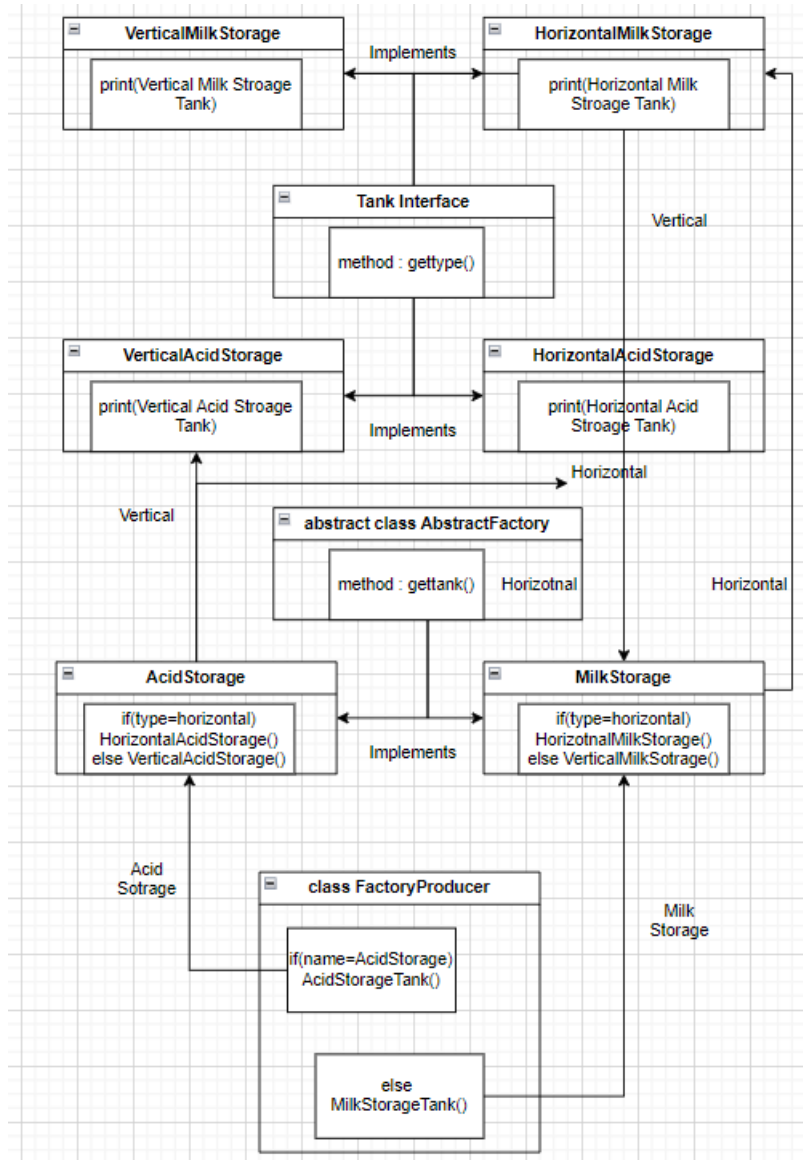
Theory:

Imagine a Scenario where you need to create different Kinds of Products. Example: Generally, in the market there is requirement of vertical tanks but as you scale order for horizontal tanks may also arise. Now Diff Types of Tanks can be managed by factory design pattern. But Different Category of Product can be Difficult Hence, we use Abstract Factory Design Pattern Wherein we have a Factory here AbstractFactory which is responsible for creating the Subfactory. Which extends AbstractFactory. And a Factory Producer which is responsible to Create Objects of Factory.

Problem Statement Explanation:

We will have an interface Tank Which is the Product of the Company and it is implemented by 4 Classes Vertical and Horizontal Acid Storage and Milk Storage Tank. There are 2 categories Horizontal and Vertical and Everyone has 2 types Acid Storage and Milk Storage.

Flowchart Explanation:



Code:

```

interface Tank{
    void gettype();
}

class VerticalAcidStorage implements Tank{
    public void gettype(){
        System.out.println("Vertical Acid Storage Tank");
    }
}

class HorizontalAcidStorage implements Tank{
    public void gettype(){
        System.out.println("Horizontal Acid Storage tank");
    }
}

class AcidStorage {
    if(type=horizontal)
        HorizontalAcidStorage()
    else VerticalAcidStorage()
}

class MilkStorage {
    if(type=horizontal)
        HorizontalMilkStorage()
    else VerticalMilkStorage()
}

class FactoryProducer {
    if(name=AcidStorage)
        AcidStorageTank()
    else
        MilkStorageTank()
}

```

```

class HorizontalMilkStorage implements Tank{
    public void gettype(){
        System.out.println("Horizontal Milk Storage Tank");
    }
}

class VerticalMilkStorage implements Tank{
    public void gettype(){
        System.out.println("Vertical Milk Storage Tank");
    }
}

abstract class Abstractfactory{
    abstract Tank getTank(String tanktype);
}

class AcidStorageTank extends Abstractfactory{
    public Tank getTank(String tanktype){
        if(tanktype.equalsIgnoreCase("horizontal")){
            return new HorizontalAcidStorage();
        }
        else if(tanktype.equalsIgnoreCase("vertical")){
            return new VerticalAcidStorage();
        }
        return null;
    }
}

class MilkStorageTank extends Abstractfactory{
    public Tank getTank(String tanktype){
        if(tanktype.equalsIgnoreCase("horizontal")){
            return new HorizontalMilkStorage();
        }
        else if(tanktype.equalsIgnoreCase("vertical")){
            return new VerticalMilkStorage();
        }
        return null;
    }
}

class FactoryProducer{
    public static Abstractfactory getFactory(String name){
        if(name.equalsIgnoreCase("Acid Storage")){
            return new AcidStorageTank();
        }
        else if(name.equalsIgnoreCase("Milk Storage")){
            return new MilkStorageTank();
        }
        else{
            return null;
        }
    }
}

```

```

}

public class abstractfactoryexample{
    public static void main(String[] args) {
        // Horizontal = True
        Abstractfactory tankfactory = FactoryProducer.getFactory("acid storage");
        Tank VBL = tankfactory.getTank("vertical");
        VBL.gettype();
        Tank Amul = tankfactory.getTank("horizontal");
        Amul.gettype();

        // Vertical = True
        Abstractfactory ATPL = FactoryProducer.getFactory("milk storage");
        Tank CocaCola = ATPL.getTank("vertical");
        Tank Nestle = ATPL.getTank("horizontal");
        CocaCola.gettype();
        Nestle.gettype();
    } }

```

Output:

```

● PS C:\Users\onlyf> cd "c:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\
Vertical Acid Storage Tank
Horizontal Acid Storage tank
Vertical Milk Storage Tank
Horizontal Milk Storage Tank
○ PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\Abstract Factory>

```