	Course Name: Design Patterns/Thinking LAB	EXPERIMENT NO. 1	
	Course Code: 20CP210P Faculty: Dr. Ketan Sabale	Branch: CSE	Semester: IV
Submitted by: Jangle Parth Roll no: 22BCP083			

Objective: To familiarize students with standard Creational design patterns.

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

Theory:

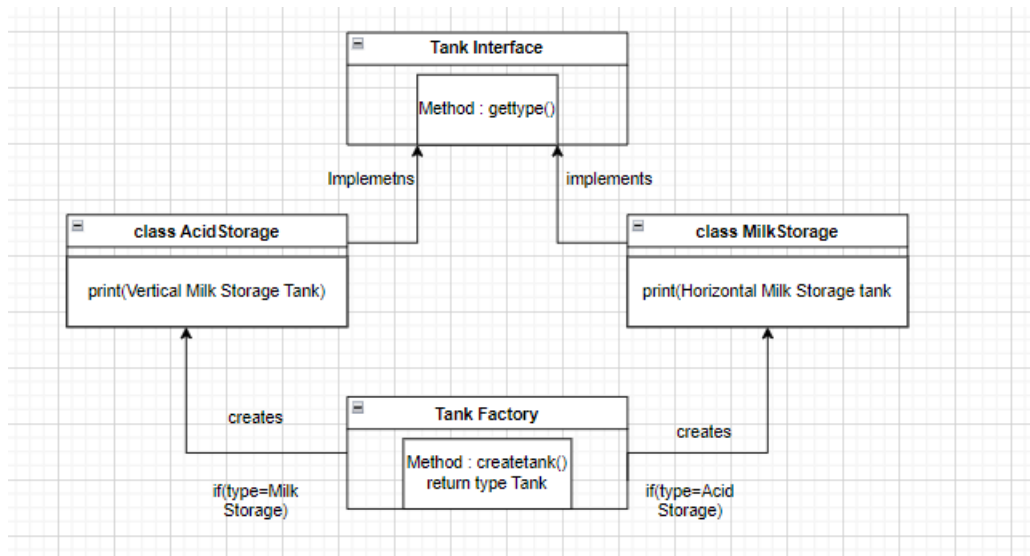
Imagine a Scenario where you need to create different types of tanks. There are many types of Tanks eg: Milk Storage, Acid Storage, Pulp Storage etc. And the No Increases as the Company Grows. Now every time it is not feasible to create object for new type of Tank in the main method (client side). Hence, we use Factory Design Pattern Wherein we have a Factory here TankFactory which is responsible for creating the objects.

In Factory Patten, we create object without exposing the creation logic to the client. It is a creational Pattern that provides and interface for creating object in superclass but allow subclass to alter the type of object.

Problem Statement Explanation:

We will have a interface Tank Which is the Product of the Company and it is implemented by two Classes Acid Storage and Milk Storage which are two types of Tank Available . We have a Tank Factory which has a method create tank and takes type of tank as parameter and creates a object of it

Flowchart Explanation:



Code:

```
interface Tank{
    void gettype();
}
class ACIDSTORAGE implements Tank{
    public void gettype(){
        System.out.println("This is a Acid Storage Tank");
    }
}
class MILKSTORAGE implements Tank{
    public void gettype(){
        System.out.println("This is a Milk Storage Tank");
    }
}

class TankFactory{
    public Tank createtank(String type){
        if(type==null){
            return null;
        }
        else if(type.equalsIgnoreCase("MILKSTORAGE")){
            return new MILKSTORAGE();
        }
        else if(type.equalsIgnoreCase("ACIDSTORAGE")){
            return new ACIDSTORAGE();
        }
        else{
            return null;
        }
    }
}
```

```

    }
}

public class tank_factory{
    public static void main(String[] args) {
        TankFactory atpl = new TankFactory();
        Tank VB1 = atpl.createtank("ACIDSTORAGE");
        VB1.gettype();
        Tank amul = atpl.createtank("MILKSTORAGE");
        amul.gettype();
    }
}

```

Output:

```

● PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern> cd "c:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\factory pattern\" ; if ($?) { javac tank_factory.java
{ java tank_factory }
This is a Acid Storage Tank
This is a Milk Storage Tank
○ PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\factory pattern>

```