	Course Name: Design Patterns/Thinking LAB		EXPERIMENT NO. 11	
	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.

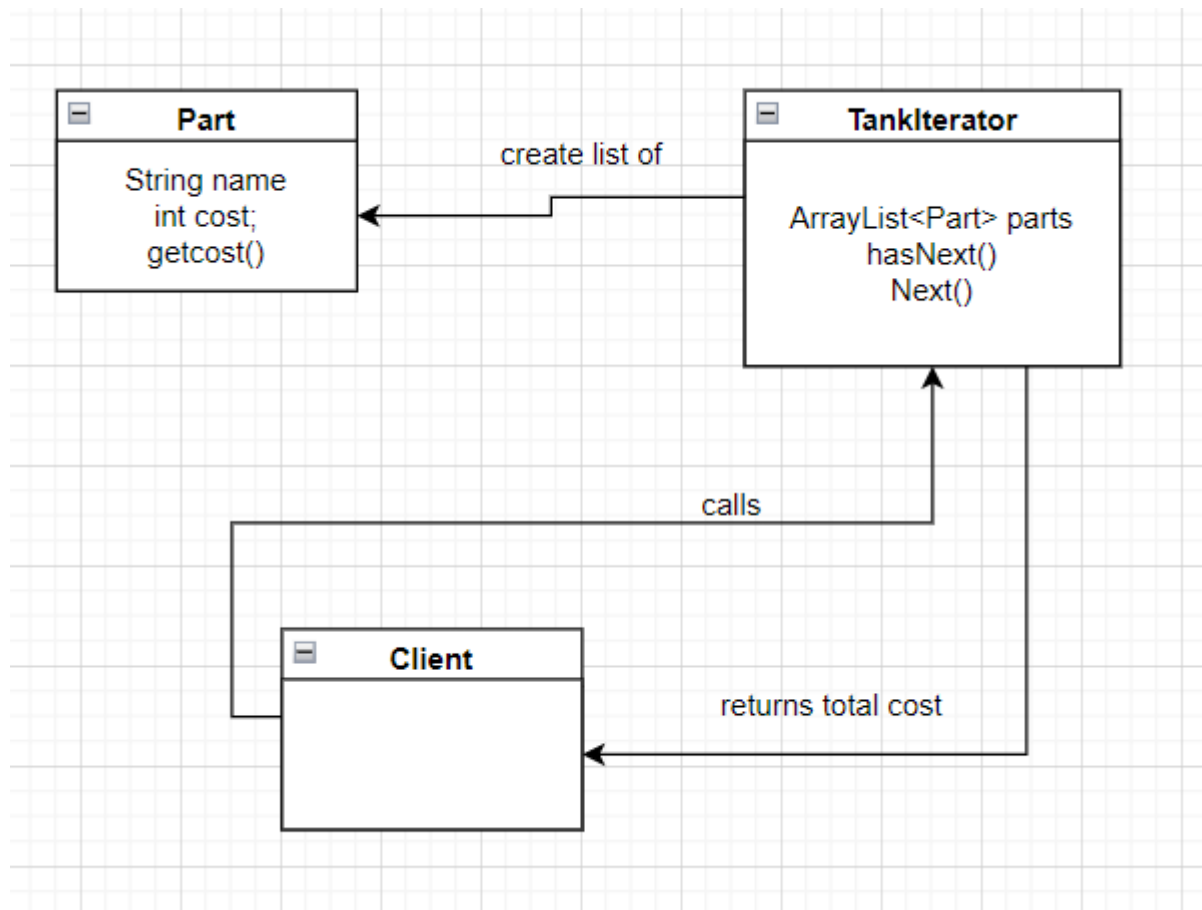
Experiment: Explain the Iterator design pattern and write a program using any object-oriented programming language to demonstrate the working of Iterator design pattern.

Theory: Imagine a Scenario where you have to calculate final price of a product but your product is made of n no of components which have their own price and some of them are directly purchased. Now to calculate the price you have to maintain a variable which will update price when new product is added to a tank now for 100 tanks 100 variable need to be maintained hence we use iterator. Iterator traverse through whole time tile using some logic and give you your final price

Problem Statement Explanation:

We have a Part class Method which is the part which goes in making of tank and cost of the part . We have a Tank Iterator which iterates through the whole journey of building tank and give the final cost of tank. It has method hasNext to check whether there exist next element and next to access next element.

Flowchart Explanation:



Code:

```
package Iterator;

import java.util.ArrayList;
import java.util.List;

class Part {
    String name;
    int cost;

    Part(String name, int cost) {
        this.name = name;
        this.cost = cost;
    }
}
```

```

        public double getcost() {
            return cost;
        }
    }
}

class TankIterator {
    int currentIndex = 0;
    List<Part> parts;

    public TankIterator(List<Part> part) {
        this.parts = part;
    }

    public boolean hasNext() {
        return currentIndex < parts.size();
    }

    public Part next() {
        if (!hasNext()) {
            System.out.println("No such Element found");
        }
        return parts.get(currentIndex++);
    }
}

public class iterator {
    public static void main(String[] args) {
        List<Part> parts = new ArrayList<>();
        parts.add(new Part("Manhole", 1000));
        parts.add(new Part("Airvent", 100));
        parts.add(new Part("Gasket", 400));
        parts.add(new Part("Sprayball", 6000));
        TankIterator t = new TankIterator(parts);
        int totalcost = 0;
        while (t.hasNext()) {
            totalcost += t.next().getcost();
        }

        System.out.println("Total Cost: " + totalcost);
    }
}

```

Output:

```
PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern> cd "c:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\Iterator\" ; if ($?) { javac iterator.java } ;  
Total Cost: 7500  
PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\Iterator>
```