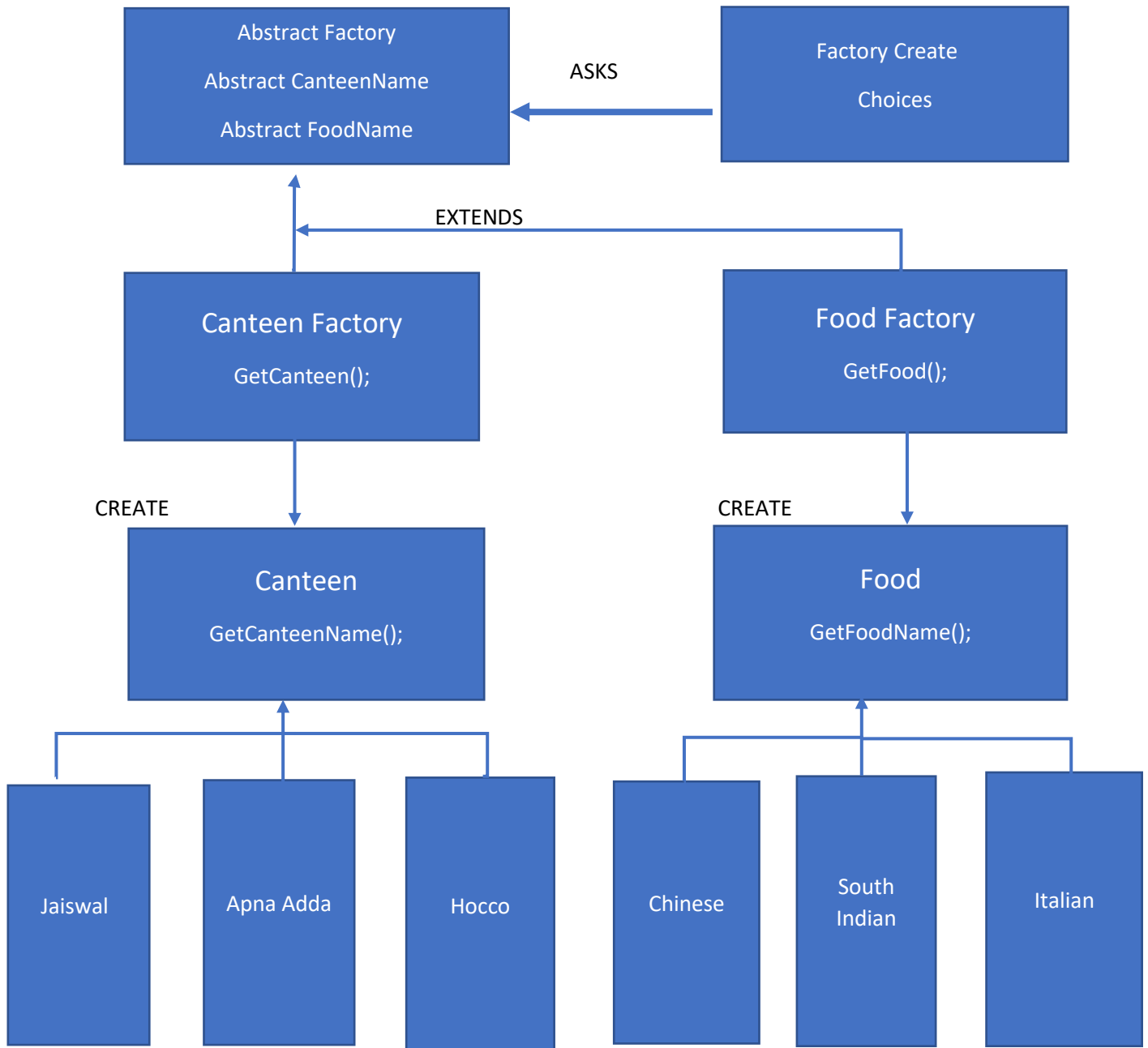


Design Pattern Lab Assignment – 5



Name : Jay Lapani
Roll No. : 21BCP150
Div - 3 , G - 5

Diagram of abstract factory method(food canteen):



Program for Food Canteen System:

```
import java.io.*;
import java.util.Scanner;

interface Canteen {
    String getCanteenName();
}

class JAISWAL implements Canteen {
    private final String CNAME;

    public JAISWAL() {
        CNAME = "JAISWAL CANTEEN";
    }

    public String getCanteenName() {
        return CNAME;
    }
}

class APNAADDA implements Canteen {
    private final String CNAME;

    APNAADDA() {
        CNAME = "APNAADDA CANTEEN";
    }
}
```

```
}
```

```
public String getCanteenName() {
```

```
    return CNAME;
```

```
}
```

```
}
```

```
class HOCCO implements Canteen {
```

```
    private final String CNAME;
```

```
public HOCCO() {
```

```
    CNAME = "HOCCO CANTEEN";
```

```
}
```

```
public String getCanteenName() {
```

```
    return CNAME;
```

```
}
```

```
}
```

```
abstract class Food {
```

```
    protected double rate;
```

```
    abstract void getFoodRate(double rate);
```

```
public void calculatePayment(double foodamount, int quntity) {
```

```
    double rate = foodamount * quntity;
```

```
    System.out.println("Your Total Payment is " + foodamount + "X" + quntity + "=" + rate);
```

```
}
```

```
}
```

```
class Chinese extends Food {  
    public void getFoodRate(double r) {  
        rate = r;  
    }  
}
```

```
class SouthIndian extends Food {  
    public void getFoodRate(double r) {  
        rate = r;  
    }  
}
```

```
}
```

```
class Italian extends Food {  
    public void getFoodRate(double r) {  
        rate = r;  
    }  
}
```

```
}
```

```
abstract class AbstractFactory {  
    public abstract Canteen getCanteen(String canteen);  
  
    public abstract Food getFood(String food);  
}
```

```
class CanteenFactory extends AbstractFactory {  
    public Canteen getCanteen(String canteen) {
```

```
    if (canteen == null) {  
        return null;  
    }  
    if (canteen.equalsIgnoreCase("JAISWAL")) {  
        return new JAISWAL();  
    } else if (canteen.equalsIgnoreCase("APNAADDA")) {  
        return new APNAADDA();  
    } else if (canteen.equalsIgnoreCase("HOCCO")) {  
        return new HOCCO();  
    }  
    return null;  
}
```

```
public Food getFood(String food) {  
    return null;  
}
```

}// End of the CanteenFactory class.

```
class FoodFactory extends AbstractFactory {  
    public Canteen getCanteen(String canteen) {  
        return null;  
    }  
}
```

```
public Food getFood(String food) {  
    if (food == null) {  
        return null;  
    }  
    if (food.equalsIgnoreCase("Chinese")) {  
        return new Chinese();  
    }  
}
```

```

    } else if (food.equalsIgnoreCase("SouthIndian")) {
        return new SouthIndian();
    } else if (food.equalsIgnoreCase("Italian")) {
        return new Italian();
    }
    return null;
}

}

```

```

class FactoryCreator {
    public static AbstractFactory getFactory(String choice) {
        if (choice.equalsIgnoreCase("Canteen")) {
            return new CanteenFactory();
        } else if (choice.equalsIgnoreCase("Food")) {
            return new FoodFactory();
        }
        return null;
    }
}

```

```

class AbstractFactoryPatternExample {
    public static void main(String args[]) throws IOException {

```

```

        Scanner sc = new Scanner(System.in);

```

```

        System.out.print("Enter the name of Canteen from where you want to take food
amount: ");

```

```

        String d = sc.nextLine();

```

```

        String canteenName = d;

```

```
System.out.print("\n");

System.out.print("Enter the type of food e.g. Chinese food or SouthIndian food or
Italian food : ");

String x = sc.next();

String foodName = x;


AbstractFactory canteenFactory = FactoryCreator.getFactory("Canteen");
Canteen b = canteenFactory.getCanteen(canteenName);


System.out.print("\n");


System.out.print("\n");
System.out.print("Enter the number of quantity: ");


Integer s = sc.nextInt();


int quntity = s;


System.out.print("\n");
System.out.println("you are taking the food from " + b.getCanteenName());


AbstractFactory foodFactory = FactoryCreator.getFactory("Food");
Food l = foodFactory.getFood(foodName);
double rate=20 ;
if(x.equalsIgnoreCase("Chinese")){
    rate=80;
}
else if(x.equalsIgnoreCase("SouthIndian")){
```



```

        rate=100;
    }
    else if(x.equalsIgnoreCase("Italian")){
        rate=150;
    }
    else{
        System.out.println("Service Charge:"+rate);
    }

    double foodAmount = rate;

    l.calculatePayment(foodAmount, quntity);
}
}

```

Output:



```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
PS D:\DAA> cd "d:\DAA\" ; if ($?) { javac AbstractFactoryPatternExample.java } ; if ($?) { java AbstractFactoryPatternExample }

Enter the name of Canteen from where you want to take food amount:
jaishwal

Enter the type of food e.g. Chinese food or SouthIndian food or Italian food :
italian

Enter the number of quantity:
3

you are taking the food from JAISWAL CANTEEN
Your Total Payment is 150.0X3=450.0
PS D:\DAA> 

```