QUIZ # 2

NAME :	ROLL NO. #

```
public interface ChineseFood {

    void c

    void c
}
```

```
public class ChinesePack implements Chin

    @Override
    public void chineseRice() {
        System.out.println("Chinese Rice

    }

    @Override
    public void chineseNoodles() {
        System.out.println("Chinese Noodels with Red Sauce ordered"

    }

}
```

```
public interface MexicanFood {

    void mexicanNachos();
}
```

```
public class MexicanPack implements MexicanFood {

    @Override
    public void mexicanNachos() {
        System.out.println("Mexican Nachos Crispy ordered");

    }

}
```

```
public interface Kitchen {

    public ChineseFood getChineseFood();

    public MexicanFood getMexicanFood();

}
```

```java
public class ChefA implements Kitchen {

    @Override
    public ChineseFood getChineseFood() {
        System.out.println("Chinese Food Order for ChefA");
        return new ChinesePack();
    }

    @Override
    public MexicanFood getMexicanFood() {
        System.out.println("Mexican Food Order for ChefA");
        return new MexicanPack();
    }

}
```

```java
public class ChefB implements Kitchen {

    @Override
    public ChineseFood getChineseFood() {
        System.out.println("Chinese Food Order for ChefB");
        return new ChinesePack();
    }

    @Override
    public MexicanFood getMexicanFood() {
        System.out.println("Mexican Food Order for ChefB");
        return new MexicanPack();
    }

}
```

Q1: Identify the design pattern used in above coded solution. Mention Entity – Class pairing. (5)

Q2: Draw complete Diagram for the above mentioned solution based on the used design pattern. (5)

Solution

Q1:

Entity-Class Pairing:

Abstract Factory → Kitchen

Concrete Factory → ChefA, ChefB

Abstract Product → ChineseFood, MexicanFood

Concrete Product → ChinesePackByChefA, ChinesePackByChefB, MexicanPackByChefA, MexicanPackByChefB

Q2: