**Interface Segregation**

Problem :

interface RestaurantEmployee

{

buyGroceries();

takeOrder();

cookFood();

serveOrder();

cleanTable();

}

class Waiter implements RestaurantEmployee

{

takeOrder()

{

Sop("Take order");

}

cleanTable()

{

Sop("Clean Table");

}

cookFood(){}

serveOrder(){}

buyGroceries(){}

}

Solution :

interface RestaurantEmployee{}

public interface manager extends RestaurantEmployee {

void buyGroceries();

}

public interface Steward extends RestaurantEmployee {

void takeOrder();

}

public interface Chef extends RestaurantEmployee {

void cookFood();

}

public interface Waiter extends RestaurantEmployee {

void serveOrder();

void cleanTable();

}

class WaiterImpl implements Waiter

{

void serveOrder(){...}

void cleanTable(){...}

}

Example 2

public interface Bird{}  
  
public interface FlyingBird extends Bird{  
 public void fly(){}  
}  
  
public interface WalkingBird extends Bird{  
 public void walk(){}  
}  
  
public class Parrot implements FlyingBird, WalkingBird {  
 public void fly(){ // to do}  
 public void walk(){ // to do }  
}  
  
public class Penguin implements WalkingBird{  
 public void walk(){ // to do }  
}