**Question 1: Design Decisions (3 points each)**

This is an essay question.  You will need to write an essay explaining your design decisions.

 Decision 1:

While fixing some existing code, you find two methods doing almost the same thing.  They both do the following:

* Establish a connection to a tax site
* Handle any exceptions encountered
* Send a request
* Handle the response
* Close the connection
* Record the response

The difference is in what request is sent to the tax service.   In one case, the request is looking for the sales tax on an item.  In the other method, it is looking for the meals tax on a restaurant bill.

 Do you refactor the code?  Why or why not? Does it matter to you if the two methods are in different classes?

***Ans:***I would definitely refactor the code. In the code there are two different methods performing the common task and the code is getting reused. In order to reduce that, I would create a class with constructor that will create a connection. Also I will create 2 methods in different that will send request for sales and meals tax.

Decision 2:

You’ve written a new class.  As you are writing the unit tests for the class, you realize that you either need to use PowerMockito or change your code to be easier to test.  Which choice do you make and why?

I would prefer using Mockito since we cannot give real time values, Mockito testing will be more effective than just changing the code.

Decision 3:

Congratulations!  You are now the expert on the part of the system that reaches out to the tax service from Decision 1, above.

Now there are a number of new requests that need to be made to the tax service:

* Get sales tax on an item
* Get meals tax on a meal
* Record sales tax paid for an item on a merchant’s behalf
* Record meals tax paid on a meal, on a restaurant’s behalf
* Request total tax paid by each merchant or restaurant for a a specified date
* Request total paid by each merchant for the year

In addition, there are now two tax services to connect to, so at the very least there are two URLs to connect to (one URL for each tax service).

Name one new class you would create in this situation.  Draw a UML diagram for the new class (yes, one single box), and include at least one method call in the diagram.

<<constructor>>+ (sales,sales,sales)

Getsalestax(Sales): Sales

Getsalespaid(Sales): Sales

Getsalestotalpaid(Sales): Sales

salestax: Sales

salestaxpaid: Sales

salestotaltax: Sales

Sales

<<constructor>>+ (meals,meals,meals)

Getmealstax(Meals): Meals

Getmealspaid(Meals): Meals

Getmealstotalpaid(Meals): Meals

Meals

mealtax: Meals

mealtaxpaid: Meals

mealtaxtotalpaid: Meals

Salesunion(): Boolean

Mealunion(): Boolean

Getsalestax(Sales): Sales

Getsalespaid(Sales): Sales

Getsalestotalpaid(Sales): Sales

Getmealstax(Meals): Meals

Getmealspaid(Meals): Meals

Getmealstotalpaid(Meals): Meals

Union

salestax: Sales

salestaxpaid: Sales

salestotaltax: Sales

mealtax: Meals

mealtaxpaid: Meals

mealtaxtotalpaid: Meals