### Name:\_\_\_\_\_\_\_\_\_\_\_\_\_

**Application Engineering and Development ISY 100**

# Exam II

## Friday Nov 21, 2008

1. For the model below, the Location class defines street address, city, and country of where the factory is located. Define all java changes you will make to this model to implement the getAllFactoriesByCountry(countryname) method on the FactoryDirectory class. You are responsible for defining all methods and attributes necessary to return the list of factories that are located in a certain country.

**Location**

**Business**

**Factory Directory**

**Supplier**

**Factor**

1. Continuing with the previous model from 2, suppose that on the business class we define a method called findNumberOfCountriesOfManufacturingBySupplierName which takes a string name as input and returns the number of countries where supplier has factories. Write detailed java implementation for this method.
2. Continuing with the previous model from 2, suppose that on the business class we define a method called findAllLocationsOfManufacturing which returns a list of all locations associated with all suppliers. Write detailed java implementation for this method.

Consider the following object model for an internet business doing buy and sell through customer self-service. Also, consider the class definitions given below and then answer the following set of questions

**Order**

**Master Order List**

Payment for the order

**Payment**

**Business**

**Payment History**

To

From

**Financial Account**

**Customer**

**Customer Directory**

Belongs to the business

**Site Financial Account**

**Payment**: Represents the invoice and actual money made for the order. It describes the payment method used, payment status, as well as if the payment is made by credit card then the personal info of the customer, etc. This is the same as you order over the web and choose to pay by credit card, or any other cash method. Payment here is an obligation to pay as well the fulfillment of the outstanding payment obligation.

**Financial Account:** This class is separated from the person (customer in this case) where responsibility for the Financial Account class is to do with financial matters only. It specifies the debt and credit in the customer account such as how much the business owes the customer which could happen in the case of product returns, for example. It also keeps track of all the payment transactions made on the account. Please note Financial Account is away for the web site to keep track of customer money.

**Site Financial account:** Is the same like Financial Account except it relates to the site or company’s bank account. All money charged are transferred to the site’s financial account.

**Bank Account:** An abstraction for a physical bank account, such as a Bank of America account with account number, routing info, etc. (not in the picture).

1. Suppose you want to calculate the total amount of accumulated money owed to a customer. Define all the java classes, methods, method parameters, and attributes to answer this question. You must be careful in the way you place your method and attributes.
2. Suppose the business (the site) wants to know if the total revenues generated from the orders match the money actually collected from customers. Define all java classes, methods, and attributes to solve this problem. Remember some orders might not be valid for some reason or another.
3. Unlike the model above, some businesses might invoice the user first then the user will have a number of days or weeks to pay. How will the design above change to accommodate this way of doing business? First: a) the customer orders products, b) company invoices the customer, c) the company ships ordered products, and then d) customer pays; e) company declares customer paid, or settled. How to change the design above to accommodate the different way of doing business.
4. Continuing with the previous example, suppose the company wants to check to make sure there are no money leaks: The money generated from orders is exactly the same as the money invoiced and is exactly the same like the money deposited in their account. How your design will change to help answer the following questions? What methods and attributes you will need to add to your design to support these requirements? Explain your answers by defining the java classes, methods, and attributes to accomplish the task. You might want to define methods on the business class, such as validateInvoicesVsIncome() and validateOrdersVsInvoices that will return minus, plus, or zero amounts depending on the calculations. You don’t need to redefine classes you already declared in the previous answers.
5. The Financial Account and Site Financial classes share many methods and attributes. Show how to use inheritance to combine the two to create a more efficient design leveraging inheritance and reuse. Write the java class definitions showing how inheritance will work in this case.