**CSCI165 Computer Science II  
Lab Exercise  
Object Oriented Composition**

**This is part one of this lab. Expect more later this week**

**In UML Composition is represented by:**

**Lab Task One:** Define properly encapsulated Java classes to represent the following composition diagram. Use the Date class that was included in the code for this week.

**Privacy Leaks must be prevented**

A screenshot of a social media post

Description automatically generated

**Domain Validation:**

* **Customer:**
  + **Email:** 
    - must contain an ***@*** and only one ***@***
    - a top-level domain of either ***2 or 3 characters***
    - The top-level domain must be at the end of the email
    - @ must occur **before** the domain
* **Address:**
  + **City, State:** These fields should be populated automatically by specifying a zip code. I have included an offline version of ***zip\_code\_database.csv*** or you could research and figure out how to talk to the USPS Web Tools API: <https://www.usps.com/business/web-tools-apis/welcome.htm>   
    Validate the zip code also . . . if it is invalid decide on a way to handle this. I can help you brainstorm. I am interested in your design choices.
* **Account:**
  + **Credit Limit:** Cannot be more than 200% of the balance. If it is, make it 200% of the balance.
  + **Balance, Credit Limit and Discount Level:** Cannot be negative
  + **Discount Level:** For every year the account has been active it gets a 2% discount.
* **Product:**
  + **SKU:** Must be 10-character length string starting with one of the following
    - 001, 002, 003, 004, 110
  + **Price:** Cannot be negative

**Unit Test:** Define unit tests for each of the requirements above. Be sure to include full coverage. If you are unsure of this, talk to me.

**Driver:** Define a Driver class that demonstrates that you can create instances of the Invoice class. Demonstrate that you can stack toString and equals calls. Ensure that there are no null pointers.