1. Please draw the sequence diagram for the code shown in the Appendix. A class diagram is also shown in the Appendix for your reference.

*(the initial arrow, invoking order’s member functions, was skipped below)*

2. What is the most suitable architecture style for a compiler (e.g., gcc) that pre-processes program source code, conducts error checking and optimization to the code, transforms the source code to assembly code, and finally transforms the assembly code to machine code.
   a. layered
   b. repository (i.e., data-centered)
3. Which one of the following non-functional requirements benefit the most from the “layered” architecture style?
   a. usability
   b. performance
   c. maintainability
   d. reliability
Appendix
You do NOT need to turn in this sheet

```c
float Order::calculatePrice () {
    //suppose there is only one line in this order
    float basePrice = line.calculatePrice ();
    return buyer.customizedPrice (basePrice);
}

float OrderLine::calculatePrice ( ) {
    return quantity * item.getPrice ();
}

float Customer :: customizedPrice (float base) {
    return base;
}

float PrimeCustomer::customizedPrice ( float base ) {
    return base * discountRate;
}
```