

# CSSE 374: Exam 3

---

## Policy

This is an open-book, open-note, open-moodle, take-home exam. You are also allowed to refer to any documents on the internet provided that you are not communicating or collaborating with another individual. Asking questions in online-forums, newsgroups, or via chat applications and emails are also forms of collaboration and are not allowed in this exam.

This part of the exam contributes 80% to your Exam 3 grade and 8% to your overall course grade. The exam is due by 8:00 am, Sunday, 2/15/2015 on Moodle.

## Problem

Assume that you work for Amazon. Amazon has developed a library that a client can use to programmatically retrieve textbook listings. Currently, three API methods have been developed:

1. One, for listing all available books in the given catalog in no particular order.
2. Another for getting books sorted by price.
3. And yet another, for getting books sorted by relevance. (See the **src-amzn** folder)

The **src-client** folder has a sample usage of the Amazon's API. Amazon wants to refactor its API so that it is easy to maintain over a long period of time while still allowing addition of new features to the library.

## Analysis

Create **Exam3/docs/Answer.pdf** to answer the following questions.

**Q1.** Study the source code available in **src-amazon**. Identify, all of the design patterns used in the solution.

**Q2.** Identify all of the weaknesses in the solution.

## Features

You are one of the super-star software engineers at Amazon and they have asked you to take on the following features:

**F1.** Rather than supporting just three methods for retrieving books, Amazon wants its client to supply their own custom algorithms to retrieve books. Redesign the Amazon's library to support this feature.

You will use the **src-amazon-new** folder to implement your new solution. You may copy the code from **src-amazon** to **src-amazon-new** as you see necessary. While you are redesigning the solution for this feature, also fix all of the issues identified in **Q2**.

As a proof of concept, implement an additional **sort by relevance** algorithm that uses not only the **title**, but also the **name of authors** to evaluate relevance.

**F2.** Amazon and Ebay have reached an agreement to maximize their book selling profit. In this new agreement, Amazon and Ebay will both cross list each other's books in their store. For Amazon side, you have been tasked to support listing of the EBay's books through the Amazon's library that you developed in **F1**. You have the clear instruction that **src-ebay**

cannot be modified to support this feature. Implement this feature in the **src-amazon-new** folder. It is understood that **F1** is also applicable to the books added via EBay's library.

**F3.** The marketing department at Amazon decided that the support for EBay's books could be used to generate more revenue for Amazon. They have asked you to create two kinds of users for Amazon library: **AmazonGold** and **AmazonDiamond**. The AmazonGold users are allowed access to **only** Amazon's books, whereas AmazonDiamond users are allowed access to both Amazon's and Ebay's books. Implement this feature in the **src-amazon-new** folder.

Please create appropriate packages in **src-amazon-new** folder as you see fit.

## Grading

You will design, implement, and test all three features appropriately. Please document your design in **docs/Answer.pdf** with necessary explanation. Also, **identify the patterns** that you used for each feature. Here is the grading rubric:

- Q1 – 2.5 points
- Q2 – 2.5 points
- F1 – Design: 5 points, Implementation: 10 points, Testing: 5 points
- F2 – Design: 5 points, Implementation: 15 points, Testing: 5 points
- F3 – Design: 5 points, Implementation: 20 points, Testing: 5 points

## Deliverables

Please bundle your project [**zip**] and turn it in on Moodle.