# Advanced JavaScript Sample Exam: Amazon

The goal of this sample exam is to practice the creation of a JavaScript structure, using the best practices of **quality JavaScript**, without any help of external libraries such as jQuery.

### The Amazon System

**The Amazon system** is a library for online shopping. The system should have

**Categories**,

**Items**,

**CustomerReviews**,

**Users** and

**ShoppingCart**.

Implement function constructors or objects for all types.

You can use Pseudo-classical or Prototypal OOP.

* Category – **name**, collection of **categories** (nested categories) and collection of **items**. Items and categories should be hidden. Add functions for:
  + addCategory
  + getCategories
  + addItem
  + getItems
* Item - **title**, **description**, **price** and collection of **customerReviews.** Customer reviews should be hidden. Add functions for:
  + addCustomerReview
  + getCustomerReviews
* UsedItem – **title**, **description**, **price**, **condition** and collection of **customerReviews**. Customer reviews should be hidden.
* User – **username**, **fullName**, **balance** and **shoppingCart.** Balance and shopping cart should be hidden. Add functions for:
  + addItemToCart
* ShoppingCart – Should have collection of **items.** Items should be hidden. Add functions for:
  + addItem
  + getTotalPrice
* customerReview = models.getCustomerReview(customer, content, rating, createdOn);

Separate all models into different files. Expose **functions** for **creating an instance** of the current model in a global object called "**models**".

|  |
| --- |
| **Source code** |
| scope.getUser = function getUser(username, fullName, balance) {  Return new User(username, fullName, balance);  } |

### The Source Code

You are given the following components:

* **index.html** - the HTML file of the test page
* **style.css** - the stylesheet
* **generator.js** – generates test data
* **html-loader.js** – loads the categories, users and attaches event handlers
* **app.js** – app.js is the start point of the application

Your task is to implement only empty models. If you implement them correctly, your application will run without errors.