Homework 10 Steve Hulme (UNC Bootcamp Raleigh September 2017 cohort)

Overview of Challenge 1 Solution (bamazonCustomer.js)

(revised 28 Dec 2017)

Technologies

- bamazonCustomer is a "node.js" application. All user input comes from the command line. All output from the app goes to the console, via the bash shell.
  - Several node modules were acquired from npmjs.org:
    - mysql – used to connect to MySQL instance, retrieve data from the Products table in bamazon db, and to update the products table.
    - inquirer – used to query the user for Product ID and desired quantity in order to construct a transaction.
    - colors – used to colorize output to the console.
    - markdown-table – used to neatly organize and print the Products table to the console.

Database

- Created a MySQL database called "bamazon" and a table inside that database called "products".
- The "products" table contains these columns:
  - id (unique id for each product and the key).
  - product_name
  - department_name
  - price [this is the unit price]
  - stock_quantify
- bamazon.sql contains the SQL that
  - drops the database if it exists.
  - creates the database
  - builds the Products table
  - populates the Products table with 10 rows, each of which is a unique product. An eclectic mix of products in the application designer's home office was chosen in order to populate the table.

Application Flow

- The application begins by displaying all the items available for sale. While adding products to the product display object the app keeps track of the total number of products. This value is saved to a global called maxProducts.

- The user is prompted to supply the Product ID of the product the user would like to buy, as well as the number of units of this product.
  - The Product ID supplied by the user is validated as follows:
    - must be an integer. No character input is allowed for Product ID.
    - must be >= 0 and <= maxProducts. An error message is presented to user if the Product ID is < 0 or > maxProducts.
  - The Quantity Desired of the product is validated as follows:
    - Quantity Desired must be an integer. No character input is allowed for the Quantity Desired.
- Once a Product ID and Quantity Desired have passed initial validation (using inquirer's "validate" facility, the app retrieves the row whose id column value matches the Product ID.
- The app checks the value in the stock_quantity field against the user's input Quantity Desired.
  - If the Quantity Desired is greater than stock_quantity, the user is informed that the order cannot be fulfilled. No update is done to the Products table.
  - Otherwise, we know that this is a valid order.
  - The New Quantity of the product is set to the stock_quantity - the Quantity Desired and SQL Update is used to update the row in the DB.
  - The app calculates the total price of the order and displays it in the console.
- Finally, the app makes a recursive call to the shopping logic, which takes the user back to the (re) display of the Products. Note that the updated quantity for the product just purchased will be displayed when this happens.

Video

- An in-depth video demonstration of this project can be found at BAMAZON Customer Project Walkthrough - YouTube