**Due Date:** Nov 11, 2018 23:59:59       **Max Points:**150  
  
**Details:**

This is a Collaborative Learning Community (CLC) assignment.

**Objective:**Create the module that displays the content of the commerce site and manages customer interaction, including purchases via a shopping cart.

**Activity**: Browse the Internet and review several sites to develop a feel for what the catalogs and the display of a product (or several products) look like. Discuss the customer experience and make a list of all the elements and functions your list of products and catalog search and display would do and show. Evaluate several shopping carts on popular ecommerce sites and discuss what features you may want to implement. Discuss the interface, specific functions, and necessary backend support. Discuss how the customer actions, the search/display of catalog items, and the shopping cart will visually fit with the rest of the ecommerce project. Discuss the various types of front-end and back-end functions, their relationships, and the reasoning for implementing your selected features. Discuss how the new functionality might work with existing functionality. Modify prior modules as needed. Discuss any limitations you might impose on the system. Discuss what type of statistical information might be useful for the site owner/administrator to know.

**Build:**

In MySQL, build and/or update the necessary tables to store and display the filtered or searched information required during the catalog presentation and shopping activities, to reflect new information required by the list of products and cart display.

1. Modify the schema, the E-R diagrams, the tables, the keys, and the relationships accordingly. You might need to add new fields to existing tables to support new functionality.
2. In PHP, write a Catalog Display Class, a Product Search Class, a Shopping Cart Class, and a Transaction Processing Class. Keep the features to the minimum necessary for a complete customer experience that includes browsing for products, searching, rating products, adding products to the shopping cart, and making a purchase. Revisit the inventory management related classes you created in previous modules and use the same class (or a subclass of it) to implement similar functionality in the shopping cart. The shopping cart is akin to a small inventory, i.e., a subset of the catalog, with the added functionality of transactions.
3. In HTML, build or update the minimally functional form to display the catalog, the associated content, relevant information about the customer, shopping cart content, payment information, for a complete shopping experience. Add all GUI elements and connect with the PHP backend functionality.
4. In PHP, perform the necessary verification of compliance with constraints you decided upon, with a special emphasis on content validity and correctness of information displayed. Before you code, discuss and decide every detail of the user experience such as, how information will be displayed, what happens after an action is taken, what if a wrong action is taken, recovery from errors, etc.
5. Based on the desired interaction and data captured, build the MySQL tables to store, display, and manage this information and enable the desired functionality. Connect this module to previous modules of your database and ensure cohesiveness. Discuss any revisions that might be necessary to code, tables, schema, or user interface created in previous module(s).
6. In PHP, implement a simulation of the completion of a transaction. An external software application (JavaScript) should simulate the authentication and authorization of the user credentials and payment method, confirm the transaction, shipment information. Upon completing the transaction, generate a confirmation email to the user.
7. Revisit all current and previously created classes and assess the potential for code reusability, a major purpose of object-oriented programming. Improve and clean your code for increased reusability.

**Deliverables**:

1. The schema and E-R diagram
2. All necessary SQL tables
3. All necessary PHP classes
4. All necessary HTML
5. All necessary Python code, JavaScript, or other language
6. A fully functional New Product Page

**What to submit**:

1. A document describing the project, a list of all modules, files, and a user guide. In the document, explain how desired functionality, features, and constraints have been implemented in code.
2. Upload all necessary files to the GCU Cloud Hosting Solution.
3. In each file, include a commented header with the following information: Project name and version, Module name and version, Programmer(s) name(s), Date, Short synopsis of the module, and References.
4. Comments within the code, explaining non-obvious sections
5. Revise previous project and code files as needed and document revisions made.
6. In LoudCloud, submit the URL to your project.