# 

# Purpose

This project is an extension of the previous project which allowed one to process one credit card at a time. In this project, you will add a **CreditCardList** class to those developed in Project 1. The **CreditCardList** class will contain a **private** attribute representing an **ArrayList** of **CreditCard** objects. The new class will provide additional functionality that allows the user to do the following things:

* **Add** a **CreditCard** to the list
* **Retrieve** the **CreditCard** currently in position **n** in the list
* **Remove** a **CreditCard** from the list
* **Find** a **CreditCard** with a particular **number** in the list
* Find all **CreditCard** objects for which the **credit** **card** **holder** is a specified person
* Find all **CreditCard** objects that have not expired at the time the program is run
* Sort the **CreditCard** objects in the list by credit card number
* Sort the **CreditCard** objects in the list by card holder’s name (as it appears on the card)

The **CreditCardList** class should have methods that support the required functionality of the class. The visibility and return types of the methods should suit the purpose. For example, the method that **finds** a **CreditCard** with a particular number should return a **CreditCard** object (which may be null if no card was found), while the **Sort** methods will return nothing. The “find all” methods may return an **ArrayList** of **CreditCards**.

# Additional Specifications

The find, retrieve, and remove methods in the **CreditCardList** class should function properly even if the item requested is not in the list.

The **driver** program should be menu-driven and allow the users choices that correspond to the capabilities of the **CreditCardList** class itemized above. Include an option that allows the user to display all credit card information about every credit card in the list. For the “Find” and “Retrieve” options, display the results to verify they are correct.

The **driver** must do **all I/O**, but, for this project, the I/O may use JOptionPane dialogs or it may use console window I/O or it may use a combination of the two as long as the end result “feels” friendly to the user.

Test the program with at least 8 credit card objects at one time. Also test all the options when the list is empty to verify that they handle such situations.

# Deliverables

The due dates for the deliverables are posted on the website for the course. Turn in your initial design document first. Then develop, test, and debug your program, update the design as needed, and submit your answer to the ethical issues question along with your code and your final design.

Remember **program documentation is NOT optional**. See the documentation standards on the course website.