

---

## PROGRAMMING EXERCISES

1. a. Create an application named **BookDemo** that declares and demonstrates objects of the `Book` class and its descendents. The `Book` class includes auto-implemented properties for the International Standard Book Number (ISBN), title, author, and price. (An ISBN is a unique number assigned to each published book.) Create a child class named `TextBook` that includes a grade level and a `CoffeeTableBook` child class that contains no additional fields or properties. In the child classes, override the accessor that sets a `Book`'s price so that `TextBooks` must be priced between \$20.00 and \$80.00, inclusive, and `CoffeeTableBooks` must be priced between \$35.00 and \$100.00, inclusive. Be sure to use valid and invalid values when testing the child class properties.  
  
b. In the `Book` class you created in Exercise 1a, overload the `Object` class `Equals()` method to consider two `Books` equal if they have the same ISBN. Create a program that declares three `Books`; two should have the same ISBN and one should have a different one. Demonstrate that the `Equals()` method works correctly to compare the `Books`. Save the program as **BookDemo2**.
  
2. a. Create an application named **CustomerDemo** that prompts a user for data for five `Customer` objects, sorts them in ID number order, and displays all their data as well as a grand total of the amounts owed by all customers. The `Customer` class includes auto-implemented properties for a customer ID number, name, and balance due. Override the `ToString()` method to return all the details for a customer.  
  
b. Using the `Customer` class as a base, derive a **CreditCustomer** class. A `CreditCustomer` contains all the data of a `Customer`, plus a field to hold a monthly interest rate. Create a program named **CustomerDemo2** that contains an array of five `CreditCustomer` objects. Prompt the user for all the necessary data. Override the parent class `ToString()` method to include the child class's additional data. Sort all the records in ID number order and display them with a total amount owed from all `CreditCustomers`.