Task 1: Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store.

An Invoice should include four pieces of information as instance variables: a item number(type String),a item description(type String),a quantity of the item being purchased (type int) and a price per item  (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoice Amount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.

Task 2: Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has ondeposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value.

Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month’s interest and print the new balances for both savers.

Task 3: Shape Area and Perimeter Classes - Create an abstract class called Shape and then inherit from it other shapes like diamond, rectangle, circle, triangle etc. Then have each class override the area and perimeter functionality to handle each shape type. Write a test application with a list/array of shape objects that will be filled by data entered by the user of the application. Firstly the user will enter choice (diamond, rectangle,...) for what shape to be created and then will enter the needed data for that shape. This flow will be endless until the user decides to see info (area and perimeter) of all entered shapes.

A little advnanced but do not hesitate to try :)

Task 4: Library Catalog – Create a book class with a title, page count, ISBN and whether or not it is checked out or not. Manage a collection of various books and allow the user to check out books or return books. For added complexity generate a report of those books overdue and any fees. Also allow users to put books on reserve.