CSCI 2320
Spring 2018
Exam 1

Last Name: _____

First Name: _____

Write your answers on the answer sheets provided. You may reference printouts of any source code you have written. Other resources are not allowed. NOTE: All code you write on the exam should follow Java programming conventions.

1. (3 pts each) Briefly define each of the following terms:
   (a) object
   (b) repository
2. (2 pts) Name one benefit to defining a "toString" method for a class.
3. (2 pts) What is the purpose of a constructor?
4. NOTE: This question is focused on non-OOP use of classes. Suppose you have been asked to write a program that tracks an individual's stock portfolio.

```
class Stock
{
   String ticker;        // ticker symbol for the stock
   int sharesOwned;      // number of shares owned
   double pricePerShare; // current price of 1 share of this stock
}
```

   (a) (4 pts) Suppose a person owns 20 shares of Google stock. The stock ticker symbol for Google is "GOOGL" and at the time I made this exam one share of their stock sells for $1,056.41. Write a section of code to create a `Stock` object and initialize it to have these values.
   (b) (6 pts) Write a method that accepts an array of `Stock` objects and an integer indicating the number of stocks in the array as parameters. The method should be called `averagePrice` and should calculate and return the average price per share of all the stocks. NOTE: The method should not read values and should not print values.

5. In this problem we will continue working with the `Stock` class but will modify it to be more in line with a typical object-oriented approach.
   (a) (2 pts) Rewrite the class so that its attributes are private.
   (b) (4 pts) Write a constructor for the class that accepts as parameters a ticker symbol, the number of shares owned, and the price per share and uses those values to initialize the object.

(c) (4 pts) Write a toString method that returns the stock ticker followed by a space followed by a dollar sign followed by the value of the current holding. The holding's value is calculated by multiplying the pricerPerShare by the number of shared owned.

(d) (4 pts) Write a getter method for the ticker attribute.

(e) (6 pts) Write a method named `sellShares` that is to be called whenever the stock owner decides to sell some shares of the stock. The method will accept the number of shares to sell as a parameter and will verify that the number of shares currently owned is at least as big. If the number of shared specified to sell is more than currently owned then no shares should be sold. Otherwise, subtract the number of shares sold from the shares owned and return the value of the shares sold (i.e., shares sold multiplied by pricePerShare).

(f) (4 pts) Show the JavaDoc comments you would create for the `sellShares()` method you just wrote.

(g) (4 pts) Suppose, in your workspace, you had just modified the `Stock` class as described above. Write the sequence of `git` statements you would use to record your work and post it to your bitbucket.org homework account.

(h) (4 pts) Draw a UML diagram that depicts the new design of the `Stock` class.

(i) (4 pts) Companies that buy and sell stocks typically charge a flat rate transaction fee (for example $10.00) for any transaction (regardless of how many shares to buy or sell). Suppose you were going to add an attribute to the `Stock` class called `transactionFee`. Under what circumstances would it make sense to declare this new attribute as static. And when would you make it non-static?

(j) Write code (presumably in `main()`) that will:

i. (4 pts) Create an array of 100 Stock objects whose names are "Stock 1", "Stock 2", etc. The prices and shares owned can be whatever you want.

ii. (2 pts) Write code to display the first member in the array.

iii. (6 pts) Write code to save all data in the array to a file named `stocks.txt`.