

Write your answers on the answer sheets provided. You may refer to a printout of the command sheets (without commentary) handed out in class.

1. (3 pts each) Briefly define each of these terms in the context of object-oriented programming:
  - (a) this
  - (b) private
2. Consider the (somewhat problematic) class given here and then answer the questions that follow.

```
/*
 * A class for timing code. It's awesome. Check out the awesome methods.
 * @author Terry Sergeant
 * <p>NOTE: Some of the methods have been removed to make it shorter.</p>
 */
public class CodeTimer
{
    private long starttime, stoptime;

    /**
     * @return the number of seconds elapsed ...
     */
    public static double seconds() { return (stoptime-starttime) / 1000.0; }

    /**
     * Sets the timer to the specified value.
     * @param firstparam the value to set the timer to in seconds
     */
    public void set(double seconds)
    {
        stoptime= System.currentTimeMillis();
        starttime= stoptime-(int)(seconds*1000.0);
    }
}
```

- (a) (3 pts) Does this class adhere to the “pure object model”? Explain.
- (b) (4 pts) Name two places where JavaDoc conventions were not correctly followed (there are more than two).
- (c) (2 pts) Make a suggestion as to how the documentation could be improved that is not related to JavaDoc conventions.
- (d) The class will not compile as specified. In particular the `seconds()` method is declared to be static.
  - i. (2 pts) Why does having the “static” modifier prevent compilation?
  - ii. (2 pts) One possible solution to this error is to leave the `seconds()` method as static and make the attributes of the class static as well. How would this affect how the class could be used?

3. NOTE: This question is focused on non-OOP use of classes. Suppose you have been asked by a cable company to create a program to track customer status for various regions. In doing so you have created the following class to represent a customer:

```
public class Customer
{
    String email; // customer email address
    int cablePackage; // 1=basic; 2=supreme; 3=everything
    boolean wantsPromotions; // false= don't send promotional emails
}
```

- (a) (6 pts) Write a method call `countEmails` that accepts an array of `Customer` objects along with the number of elements in the array as parameters. The method should calculate and return the number of customers who have indicated they want to receive promotional email.
  - (b) (4 pts) Write a section of code that will insert a new customer assuming the integer variable `n` contains the number of customers currently in the variable named `cust` which is an array of `Customer` objects. Set the customer's email address to "fun@example.com" and leave the other attributes uninitialized.
4. In this problem we will continue working with the `Customer` 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 `Customer` class which will set the email address of the customer to the value of the parameter. New customers will automatically be assigned the basic package and will be included in promotional emails.
  - (c) (6 pts) Write a method called `getCost()` that will determine the monthly cost a customer incurs assuming the basic package is \$30 per month, the supreme package is \$60 per month and the everything package is \$100 per month.
  - (d) (6 pts) Write a getter method and a setter method for the attribute named `cablePackage`.
  - (e) (4 pts) Write a `toString` method that returns the customer's email address followed by the number of the package followed by a + or - symbol indicating they whether or not they want to receive promotional emails.
  - (f) (6 pts) Draw a UML diagram depicting this class. When depicting attributes you should specify their type. When depicting methods you should specify their return type, name, and parameters.
  - (g) (4 pts) Suppose, in your workspace, you had just modified the `Customer` 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) Suppose the `Customer` class has been correctly implemented as described in this problem and your are ready to test it in `main()`:
    - i. (2 pts) Create a new `Customer` object with the email address "cust1@example.com".
    - ii. (2 pts) Modify the new customer so they are using the "everything package".

- iii. (6 pts) Suppose we have an array called `cust` of `Customer` objects that is completely full with customers. Write a section of code that will calculate and display the total revenue we expect to be generated by these customers in a given month.