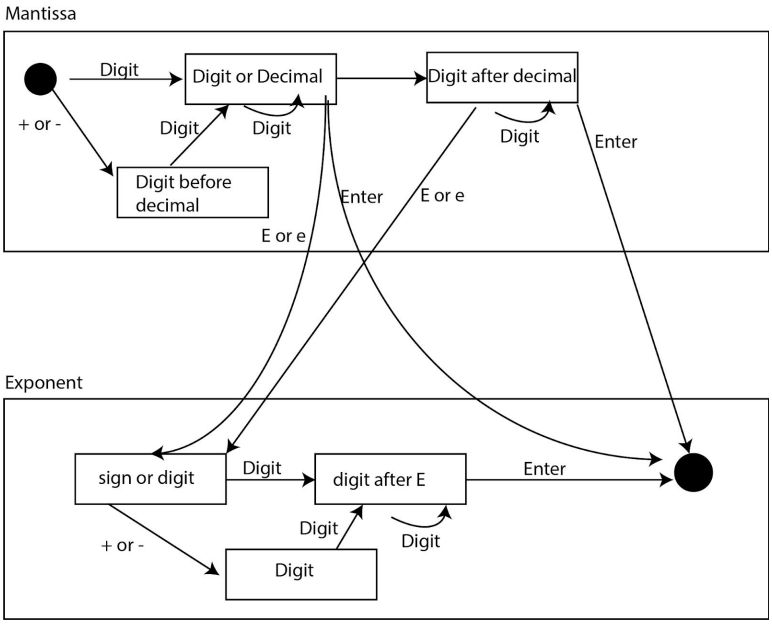
Cherrell Finister
Professor BJ
CMSI 402: Senior Project Lab
20 February 2019

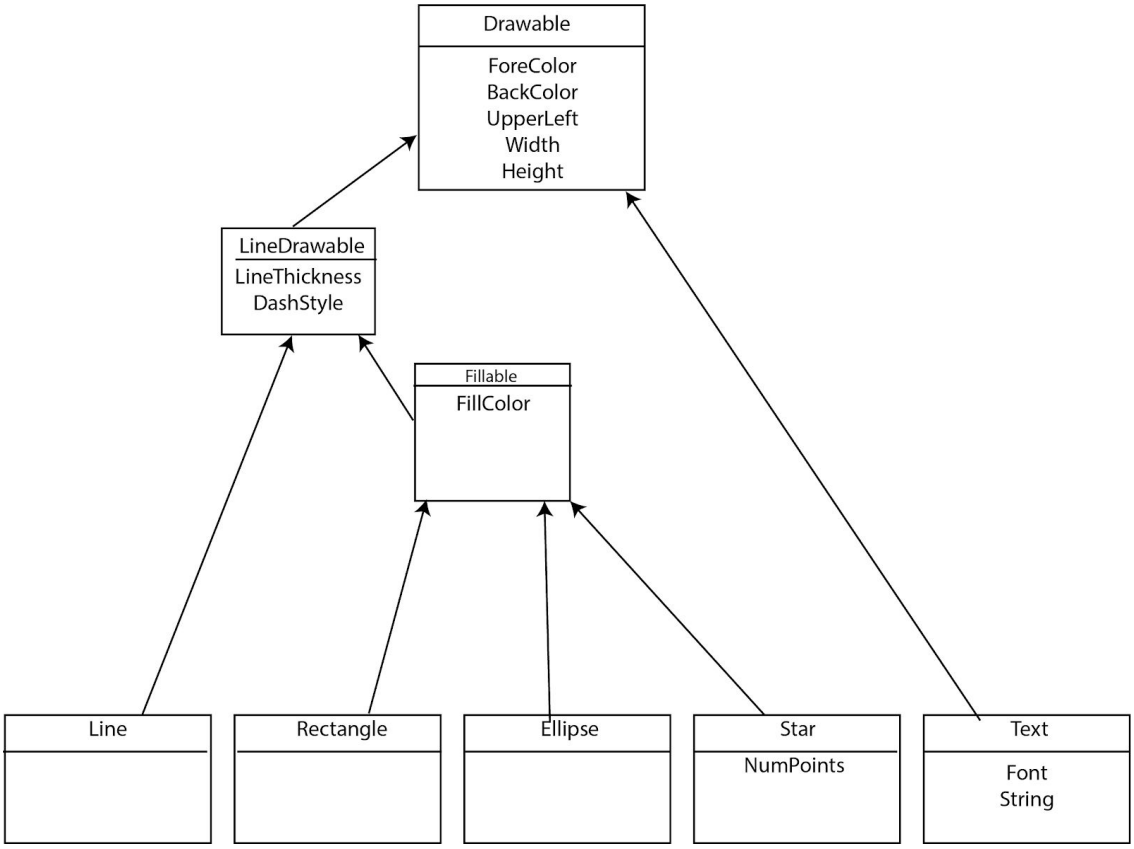
Homework #2

1. What's the difference between a component-based architecture and a service-oriented architecture?
    a. A component-based architecture is parts of the system as coupled components that provide services to each other while a service-oriented architecture using the parts of the system as implemented services, often running on separate computers communicating across a network.
2. Suppose you're building a phone application that lets you play tic-tac-toe against a simple computer opponent. It will display high scores stored on the phone, not in an external database. Which architectures would be most appropriate and why?
    a. For this application a monolithic architecture would be best because the application is self-contained within the phone. Using anything else would be too heavy duty for this program.
3. Repeat question 3 [after thinking about it; it repeats question 2 for a chess game] assuming the chess program lets two users play against each other over an Internet connection.
    a. The chess application is using an internet connection to connect two players against one another, the best architecture for this would be monolithic, data-centric, service-oriented. This program could use web services to communicate over the web with each other.
4. What kind of database structure and maintenance should the **ClassyDraw** application use?
    a. ClassyDraw Application does not need a large database because it is just storing users drawings. Using an operating system to help manage user files will allow users to delete their old files and make backups. In terms of maintenance, the program could have files that only last while the user is editing the drawing and then write over after every draw but if the program crashes it could reload the previous temporary file.
5. Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +37 or -12.3e+17 (which means -12.3 x 1017). Allow both E and e for the exponent symbol.



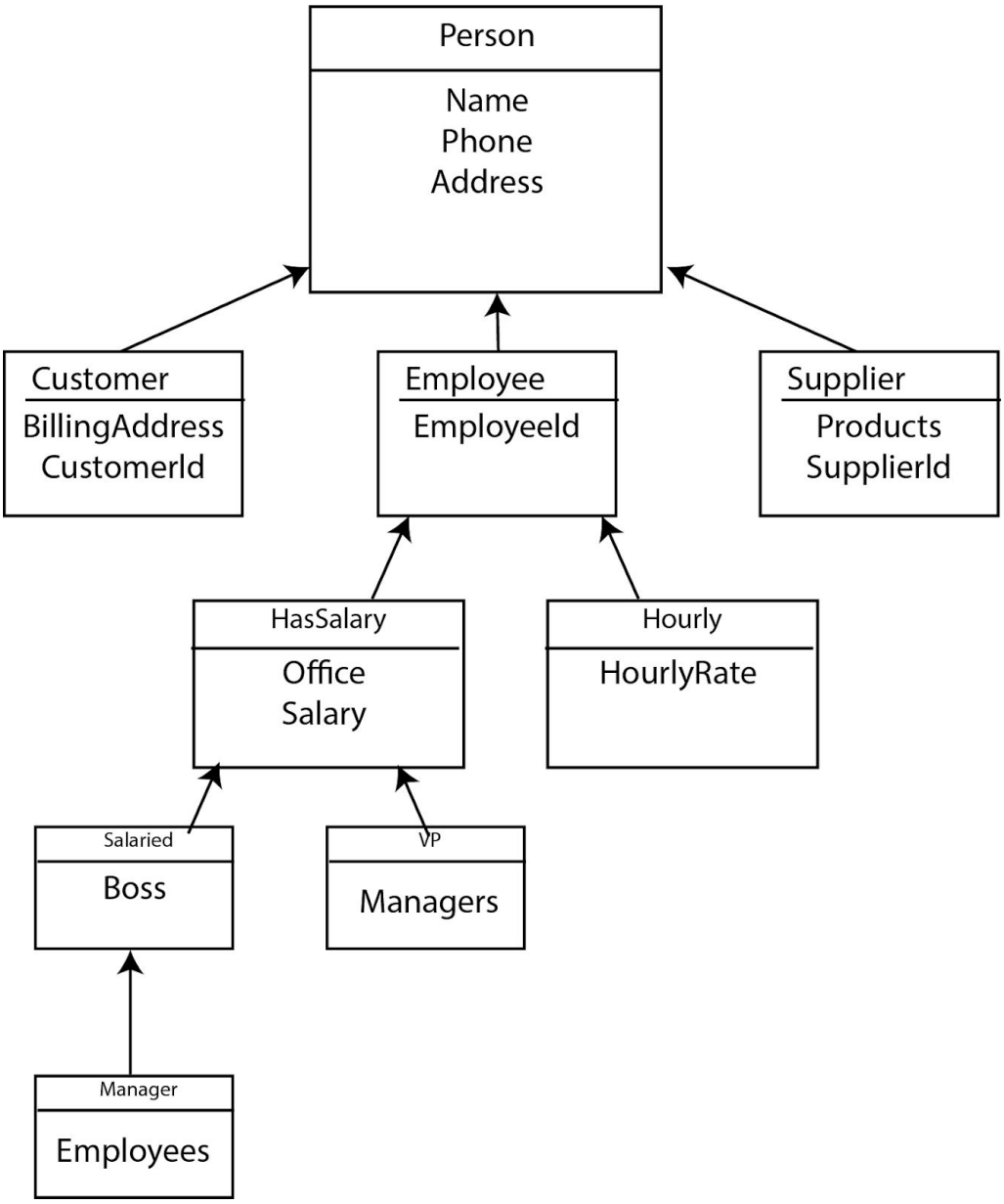
6. Consider the **ClassyDraw** classes **Line**, **Rectangle**, **Ellipse**, **Star**, and **Text**. What properties do these classes all share? What properties do they not share? Are there any properties shared by some classes and not others? Where should the shared and nonshared properties be implemented?
    a. They all share properties needed for drawing, which is the foreground and background color. They all also define drawing position by storing upper-left corner, width, and height. They won't share data were they need extra data to draw their particular type of shape. For example, the star class needs to know how many points to give a star. The rectangle, ellipse, and stat can be filled with color, while classes that draw lines (line, rectangle, ellipse, and star) also need to help with line style.
7. Draw an inheritance diagram showing the properties you identified for Exercise 1. (Create parent classes as needed, and don't forget the **Drawable** class at the top.)

a.



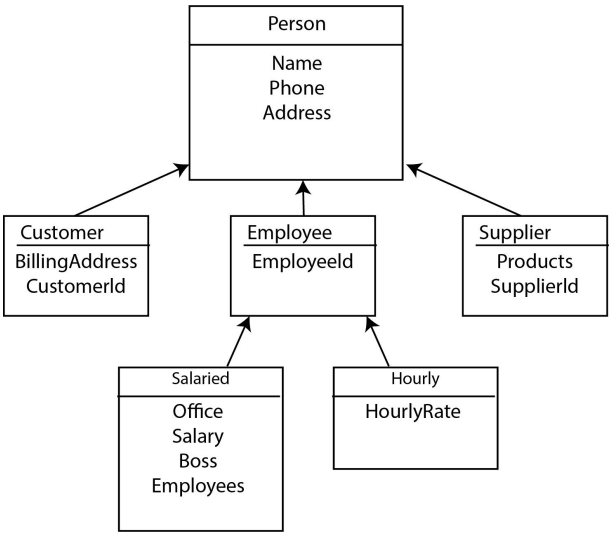
8. Assuming a **Supplier** is someone who supplies products for your business, draw an inheritance diagram showing the relationships among these classes. (Hint: Add extra classes if necessary.)

a.



9. Suppose your company has many managerial types such as department manager, project manager, and division manager. You also have multiple levels of vice president, some of whom report to other manager types. How could you combine the **Salaried**, **Manager**, and **VicePresident** types you used in Exercise 3? Draw the new inheritance hierarchy.

    a.


Person
Name
Phone
Address
Customer
BillingAddress
CustomerId
Employee
EmployeeId
Supplier
Products
SupplierId
Salaried
Office
Salary
Boss
Employees
Hourly
HourlyRate