

# Homework #2

---

## 5.1

A component-based architecture decouples the pieces of code, but the pieces are contained within the same executable program so they communicate directly across a network. With a service oriented architecture, it is a self contained program that runs on its own and provides some kind of service for its clients

---

## 5.2

For an application like this, monolithic and event-driven architectures would be the best. Monolithic would be good since all the logic, and storage are on the device itself, not stored in a database or anything. This means there isn't a need to split the app into multiple tiers. Event driven architecture would be good since it has specific events like tapping the square, and there's a lot of time waiting around for an input.

---

## 5.4

For an application like this where the users play against each other over an internet connection, it would need event driven, distributed, and client/server architectures. It would need to be event driven since it's waiting for the user to do an event (click one of the pieces, move it, etc.). It would also need distributed architecture since it is on different computers at the same time. It would need client/server architecture because it's over an internet connection. Since multiple users cannot use the same data, you need an architecture where the client is separated from the server. This allows the two users to play against each other online.

---

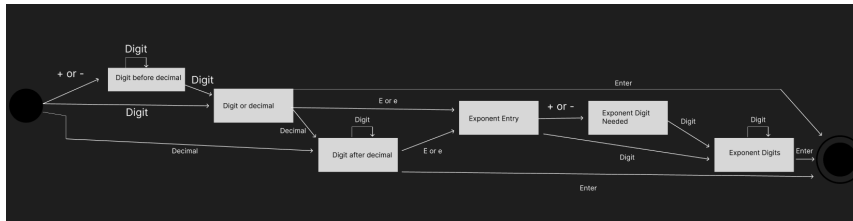
## 5.6

The ClassyDraw application should not use a traditional database, since it stores drawings in files. It can store the data inside the program with arrays, lists, and other data structures, meaning the database structure would be file based. This also means it wouldn't need complex

maintenance as that is for larger databases. Instead it can use simple backups with something like Github, or like the text suggests, copying a data file to a DVD every once in a while.

---

## 5.8



## 6.1

- These classes are all able to be selected and modified, they can all have the visual style (such as colors and line style) changed, and they can all be dragged and resized.
  - Each class has its own distinct shape and properties for that specific one. For instance a star requires properties for its points where an ellipse doesn't. The text class also has other characteristic properties that the shapes would not have.
  - There are many properties not shared by some classes. For instance classes like rectangles and ellipses would have their color be area based and could have different outline colors, where a line wouldn't have those properties.
  - The shared properties should all exist in the parent class, but the non shared properties should be in the specific child classes.
- 

## 6.2

