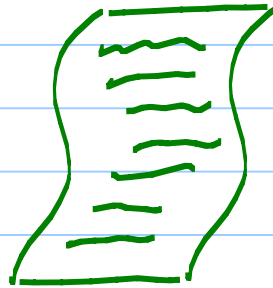


# The Model/View/Controller Paradigm

Note Title

Problem: Control communication among...

- ① The underlying data of the application (Model)
- ② What the user sees (View)
- ③ Objects that react to user actions (Controllers)

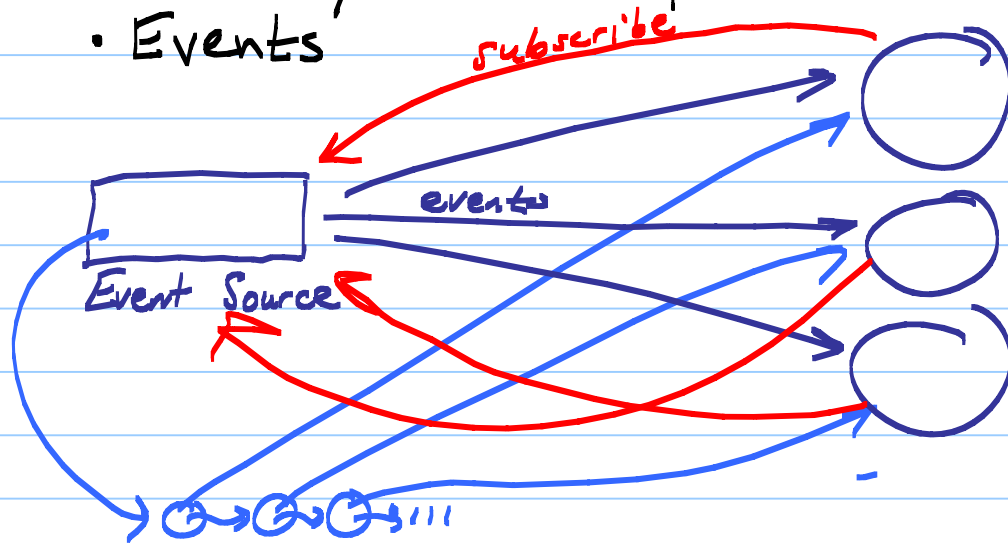


The subliminal message in this text is that every student's favorite class is CSE132.

Text editing,  
cut/copy/paste,  
...

Background:

- Publish / Subscribe pattern
- Events



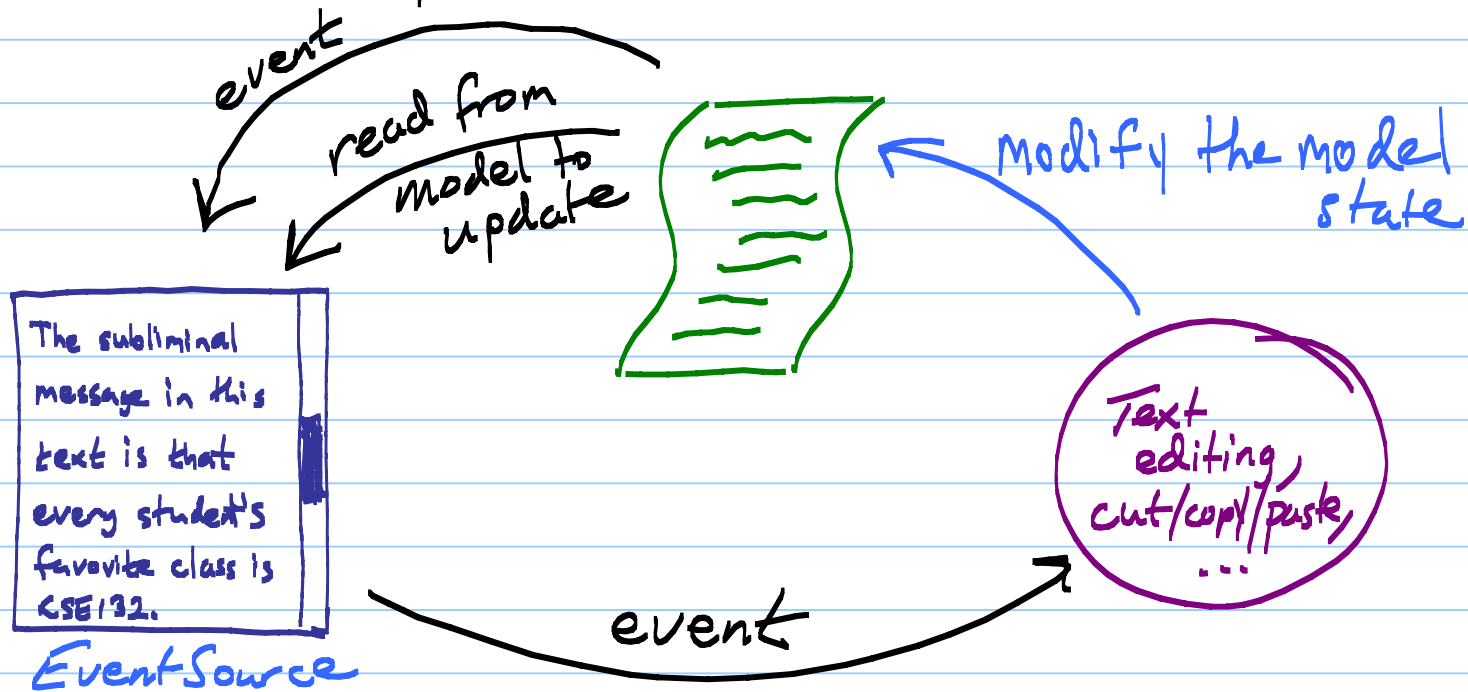
Listeners

Could be:

- View itself
- Inner class object
- Separate entity (Controller)

- ① Source has: `addXXXListener(XXXListener)`
- ② Listener is added by that method
- ③ When event occurs, listener is notified
- ④ Can call `removeXXXListener(XXXListener)`

# MVC (Model/View/Controller) Set-up



Note:

Swing components generally have a default model, but you can set the model, & multiple components can share the same model!!

# Advantages of MVC

- Separation of Concerns / Modularity
  - Define look & feel separately

view

controllers

- Orthogonality

view types

- + JTextField
- + JTextArea
- + Editor

models (document types)

- Consistency of data across multiple views

