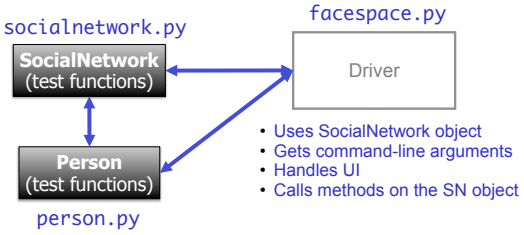
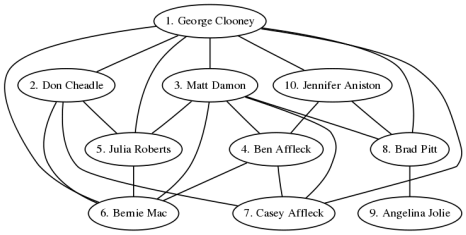
Lab 10 Design

- 3 files: `person.py`, `socialnetwork.py`, `facespace.py`



Goal Output

- You will create two graphs that look something like this and put them on a new web page for Lab 10



Problem: People Files

- Given a people file that has the format

```
<num_users>
<user_id>
<name>
<network>
…
<user_id_n>
<name_n>
<network_n>
```

- Write algorithm to create `Person` objects to represent each person, add to `SocialNetwork` object

Problem: Connection Files

- Given a connection file that has the format

```
<user_id> <user_id>
<user_id> <user_id>
…
<user_id> <user_id>
```

- Each line represents a friend/connection
  - Symmetric relationship
  - Each is a friend of the other
- Update `SocialNetwork` object

UI Specification

- Checks if user entered command-line argument
  - Default files otherwise
- Read people, connections from files
- Repeatedly gets selected options from the user, until user quits
- Repeatedly prompts for new selection if invalid option
- Executes the appropriate code for the selection
- Stops when user quits
- Stores the social network into the file

UI Pseudocode

```
Use default files if only one command-line argument
Read people, connections from files
while True:
    display menu options
    prompt for selection
    while invalid option
        print error message
        prompt for selection
    break if selected quit
    otherwise, do selected option
Store social network to designated file
```

Implementation Plan

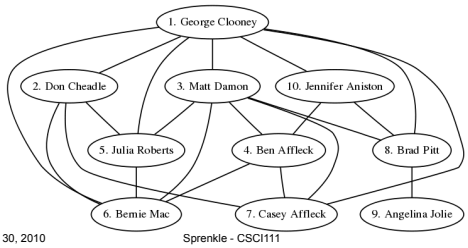
1. Implement `Person` class
   - Test (write test functions, e.g., `testPerson()`)
2. Implement `SocialNetwork` class
   - Example runs in lab write up
   - Note: Methods for classes will **not** prompt for input; Use **input parameters**
   - Test
3. Implement driver program

Plan for Implementing a Class

- Write the constructor and string representation/print methods first
- Write function to test them
  - See counter.py and card.py for example test functions
- While more methods to implement …
  - Write method
  - Test
  - REMINDER: methods should not be using input function but getting the input as parameters to the method

Writing Data To Dot File

- I will provide method that prints your social network to a file in a particular format (dot)
- Can display network graphically using `dot` program, e.g.,



Using dot

- Syntax:
  - `dot –Tfiletype –o outputname dotfile`
- Examples:

```
dot –Tpng –o graphs/network.png graphs/network.dot
dot –Tpdf –o graphs/network.pdf graphs/network.dot
```


Export SocialNetwork to Files

- I provide method to write connections to a file
  - Because only want connection once
- You handle writing to people file
  - Must be in **same format** that you read in
  - Just "undoing" the read
- Good test: if you read in a people file, export it → should look the same
  - If you read in that exported file, should see same social network

Test Data

- SocialNetwork requires: People file, Connections file
- Social Networks:
  - Simple
  - Hollywood
  - Randomly generated files
    - From W&L first and last names, randomly combined, connected
- Could combine different ones

Notes

- Advice:
  - Read through entire lab to see where you are going
- A `Person's` *network* is different from the `SocialNetwork`
  - `Person's` network is just a `string`
    - Will do more with that later…
- Can have people in different networks in the same `SocialNetwork`