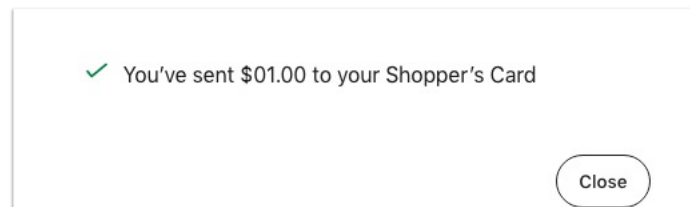


Objectives

- Reviewing creating our own classes
- Designing a Social Network
- Prep for Lab 10



March 29, 2021

Sprenkle - CSCI111

1

1

Review: Classes

- Defining Classes
 - What parameter needs to be the first parameter in every method?
 - What does that parameter represent?
 - Define *instance variable*
 - How do we create instance variables?
 - What is the special method name for the constructor?
 - What are the rules for defining the `__str__` method?
- Using classes
 - How do we create a new object of a given type?
 - How do we call methods on an object?
 - What method is automatically called when we print an object?
 - What is the API for a class? (How do we know what methods we can call on an object?)

March 29, 2021

Sprenkle - CSCI111

2

2

DESIGNING CLASSES

March 29, 2021

Sprenkle - CSCI111

3

3

Summary: Designing Classes

- What does the object/class represent?
- How to model/represent the class's *data*?
 - Instance variable
 - Data type
- What *functionality* should objects of the class have?
 - How will others want to use the class?
 - Put into methods for others to call (API)

General Class Design:

- **nouns** in a problem are **classes/objects** or **data**
- **verbs** are **methods**

March 29, 2021

Sprenkle - CSCI111

4

4

Top-Down Design

Break down larger problems into pieces that you can solve

- Smaller pieces: classes, methods, functions
 - Implement smallest pieces and build up
- We've been doing this most of the semester
 - Typically, program was 1) read input, 2) process input, 3) print result
 - Started putting Step 2 into ≥ 1 functions
 - Steps 1 and 3 were sometimes a function
 - Now: on larger scale

March 29, 2021

Sprenkle - CSCI111

5

5

Requirements for a Social Network Application

- Reads social network from two files
 - One file contains people
 - One file contains connections between people
- Adds connections between people
 - Symmetric relationship
- Provides a user interface to access/update a social network



March 29, 2021

Sprenkle - CSCI111

6

6

Designing a Social Network Application

- Break down into pieces
- What classes do we need?
 - What data needed to model those classes?
 - What functionality do each of those classes need?
- What does our user interface do?
- How should we implement those classes/program?

Recall: General Class Design:

- **nouns** in a problem are **classes/objects** or data
- **verbs** are **methods**

March 29, 2021

7

Designs

- For each of your classes
 - Data
 - API

March 29, 2021

Sprenkle - CSCI111

8

8

Social Network Classes & UI Data

- Person
 - User id
 - Name
 - Friends
- Social Network
 - People in network
- User Interface (UI)
 - Social network

What are the data types for each class's data?

March 29, 2021

Sprenkle - CSCI111

9

9

SN Classes & UI Functionality

- Person
 - Getters (accessors)
 - String rep
 - Setters
- Social Network
 - Getters
 - String rep
 - Add people to network
 - Add connections
 - Writing to a file
- User Interface
 - Getting user input to
 - Read people, connections files
 - Store social network to file
 - Add a person
 - Add connections
 - Summary: call appropriate methods on classes to do above

How should we test these?

March 29, 2021

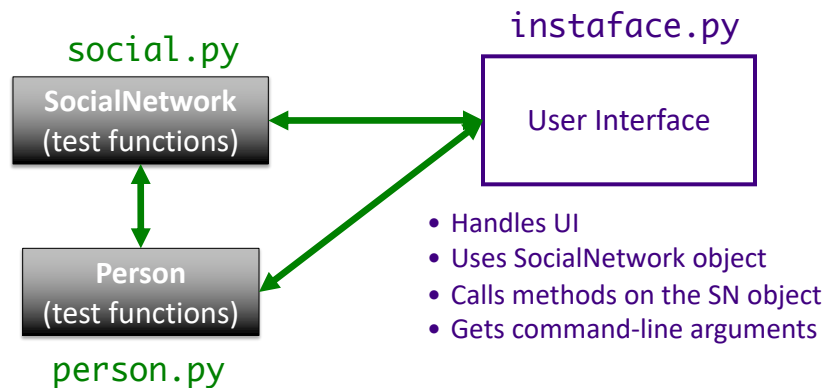
Sprenkle - CSCI111

10

10

Lab 10 Design

- 3 files: `person.py`, `social.py`, `instaface.py`



March 29, 2021

Sprenkle - CSCI111

11

11

Problem: People Files

- Given the name of people file that has the format

```
<num_users>
<user_id>
<name>
...
<user_id_n>
<name_n>
```

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

March 29, 2021

Sprenkle - CSCI111

12

12

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

March 29, 2021

Sprenkle - CSCI111

13

13

InstaFace UI Specification

- Checks if user entered command-line arguments
 - 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
 - Note how much of the functionality will be implemented in social network class. Just need to call appropriate method.

March 29, 2021

Sprenkle - CSCI111

14

14

InstaFace 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
```

March 29, 2021

Sprenkle - CSCI111

15

15

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. Complete implementation of user interface

March 29, 2021

Sprenkle - CSCI111

16

16

Plan for Implementing a Class

- Write the constructor and string representation/print methods first
- Write function to test them
 - See `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

March 29, 2021

Sprenkle - CSCI111

17

17

This Week

- Pre Lab 10:
 - Reviewing classes, some new stuff
 - Review dictionaries and lists if you're rusty
- Lab 10
 - Define your own classes
- No broader issue because no class on Friday

March 29, 2021

Sprenkle - CSCI111

18

18