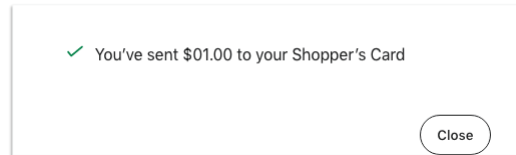


## Objectives

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



March 27, 2023

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, in general? What is the API for the Card class?

March 27, 2023

Sprenkle - CSCI111

2

2

## Review: Classes

- Defining Classes
  - Define *instance variable*
    - How do we create instance variables?
    - How do we access instance variables?
- Using classes
  - What is the API for a class, in general? What is the API for the `Card` class?

3

## DESIGNING CLASSES

4

## 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 27, 2023

5

5

## 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  $\geq 1$  functions
      - Steps 1 and 3 were sometimes functions
  - Now: on larger scale

March 27, 2023

Sprenkle - CSC1111

6

6

## Requirements for a Social Network Application

- Reads social network from two files
  - One file contains *people*
    - Their id/username, first and last names
  - One file contains *connections* between people
- Adds connections between people (makes them friends)
  - Symmetric relationship
- Provides a user interface to access/update a social network



March 27, 2023

Sprenkle - CSC1111

7

7

## 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 27, 2023

Sprenkle - CSC1111

8

8

## Designs

- For each of your classes
  - Data
  - API

March 27, 2023

Sprenkle - CSC1111

9

9

## Social Network Classes & UI Data

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

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

March 27, 2023

Sprenkle - CSC1111

10

10

## 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

March 27, 2023

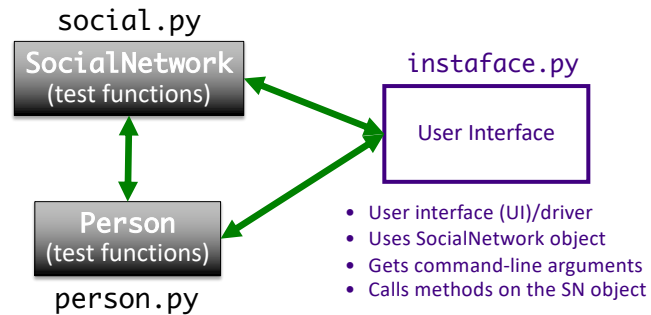
Sprenkle - CSC1111

11

11

## Lab 10 Social Network Design

- 2 classes: Person and SocialNetwork



- User interface (UI)/driver
- Uses SocialNetwork object
- Gets command-line arguments
- Calls methods on the SN object

March 27, 2023

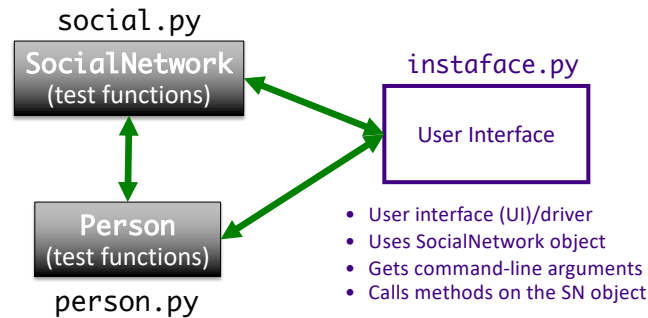
Sprenkle - CSC1111

12

12

## Lab 10 Social Network Design

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



March 27, 2023

Sprenkle - CSC1111

13

13

## 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 27, 2023

Sprenkle - CSC1111

14

14

## 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 27, 2023

Sprenkle - CSC1111

15

15

## 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 27, 2023

Sprenkle - CSC1111

16

16



## 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 27, 2023

Sprenkle - CSC1111

17

17

## 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 27, 2023

Sprenkle - CSC1111

18

18

## 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 27, 2023

Sprenkle - CSC1111

19

19

## 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

March 27, 2023

Sprenkle - CSC1111

20

20