

Objectives

- Designing our own classes
 - Representing attributes/data
 - What functionality to provide
- Using our defined classes

March 22, 2023

Sprenkle - CSCI111

1

1

Review

- What did yesterday's lab bring together?
 - What were some things you practiced?
- If I gave you a file of all the names from the US Census in the correct form, how much code would you need to change to process/graph the most common names?

March 22, 2023

Sprenkle - CSCI111

2

2

Parts of an Algorithm

- Input, Output
- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

March 22, 2023

Sprenkle - CSC1111

3

3

Where We Are: 10 weeks in

- With what you now know, opens up the possibilities for the programs you can write
 - Just about anything computational is possible
- Just need to
 - Break it up into smaller pieces
 - Iterate!

March 22, 2023

Sprenkle - CSC1111

4

4

Parts of an Algorithm

- Input, Output
 - Primitive operations
 - What data you have, what you can do to the data
 - Naming
 - Identify things we're using
 - Sequence of operations
 - Conditionals
 - Handle special cases
 - Repetition/Loops
 - Subroutines
 - Call, reuse similar techniques
- ← Going beyond the primitive data to making our own structures

March 22, 2023

Sprenkle - CSC1111

5

5

Review: Object-Oriented Programming

- Defining a class
 - Why do we want to define classes/new data types?
 - What is the keyword to create a new class?
 - How do you define a method?
 - What parameter is needed in every method?
 - What does that parameter represent?
 - Define **instance variable**
 - How do we create instance variables?
 - How do we access them?
- Using a class
 - How do you create a new object of a given class?
 - What method does this call?
 - How do you call a method?
 - What method is called when you print an object?
 - What is that method's signature?

March 22, 2023

Sprenkle - CSC1111

6

6

Card Class (Incomplete)

```
class Card:
    """ A class to represent a standard playing card.
        The ranks are ints: 2-10 for numbered cards, 11=Jack,
        12=Queen, 13=King, 14=Ace.
        The suits are strings: 'clubs', 'spades', 'hearts',
        'diamonds'."""
    def __init__(self, rank, suit):
        """Constructor for class Card takes int rank and
        string suit."""
        self._rank = rank
        self._suit = suit
    def getRank(self):
        """Returns the card's rank."""
        return self._rank
    def getSuit(self):
        """Returns the card's suit."""
        return self._suit
```

Class Doc String

Methods

Initializing instance variables

Access instance variables

March 22, 2023

Sprenkle - CSC1111

card.py

7

7

Card API

- Based on what we've seen/done so far, what is the Card class's API?
 - (Review: What is an API?)

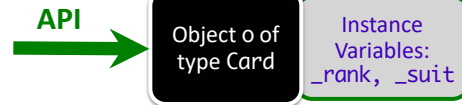
March 22, 2023

Sprenkle - CSC1111

8

8

Card API



- `Card(<rank>, <suit>)` Implementation of methods is hidden
- `getRank()`
- `getSuit()`
- `getCardColor()`
- `getRummyValue()`
- `__str__()` or `str(card)`

March 22, 2023

Sprenkle - CSC1111

9

9

Algorithm for Creating Classes

1. Identify need for a class
2. Identify state or attributes of a class/object in that class → these are the *instance variables*
3. Write the constructor (`__init__`)
 - Initialize the instance variables
4. Identify methods the class should provide
 - Implement the `__str__` method
 - Test the `__str__` method
 - How will a user call those methods (parameters, return values)?
 - Develop API
 - Implement and test methods one at a time

March 22, 2023

Sprenkle - CSC1111

10

10

Using the Card class

- Having the Card class means that we can represent a Card in code

Now that we have the Card class,
how can we **use** it?

March 22, 2023

Sprenkle - CSC1111

11

11

Review

```
from graphics import *  
  
win = GraphWin("Picture")  
win.setBackground("black")
```

```
from card import *  
  
c = Card(7, "diamonds")  
print(c.getRank())
```

- Same programming as before
- Just defining our own classes

March 22, 2023

Sprenkle - CSC1111

12

12

Using the Card class

Now that we have the Card class, how can we **use** it?

- Can make a **Deck** class
 - What data should a Deck contain?
 - How can we represent that data?
- To start: write methods **__init__** and **__str__**
 - What do the method headers look like?

March 22, 2023

Sprenkle - CSC1111

14

14

Creating a Deck Class (Partial)

- List of Card objects

```
from card import *
class Deck:
    def __init__(self):
        self._listOfCards = []
        for suit in ["clubs", "hearts", "diamonds", "spades"]:
            for rank in range(2,15):
                self._listOfCards.append(Card(rank, suit))
```

Initialize *instance variable*,
self._listOfCards

How would we want to display a deck?

Actual code should have doc strings

March 22, 2023

Sprenkle - CSC1111

15

15

Creating a Deck Class (Partial)

- List of Card objects

```
from card import *

class Deck:
    def __init__(self):
        self._listOfCards = []
        for suit in ["clubs", "hearts", "diamonds", "spades"]:
            for rank in range(2,15):
                self._listOfCards.append(Card(rank, suit))

    def __str__(self): Creates and returns a string
        deckRep= ""
        for c in self._listOfCards: ← Represents cards on separate lines
            deckRep += str(c) + "\n"
        return deckRep Actual code should have doc strings
```

March 22, 2023

Sprenkle - CSCI111

16

16

Lab 9: Staggered Extension

- Submit the programming Friday
- Graphs and web page: due before next Tuesday's lab

March 22, 2023

Sprenkle - CSCI111

17

17

Exam 2 Questions

- Content
 - Everything up through dictionaries
 - (Not creating our own classes)
 - Necessarily cumulative but focus is on second half
- What types of questions are you expecting?

March 22, 2023

Sprenkle - CSCI111

18

18

Exam 2

- What operations/methods can you do on strings? Lists? Dictionaries? Files?
- What is the syntax for writing conditions? Conditional statements? while loops?
- Problems:
 - Modify your lab 9 code to find the x^{th} most popular name
 - Modify the first lab 9 problem to take a string (a sentence) and, for each word in the string, maps the first letter to the word (like a children's book)
 - Other modifications: read from a file; make a function that returns the dictionary

March 22, 2023

Sprenkle - CSCI111

19

19