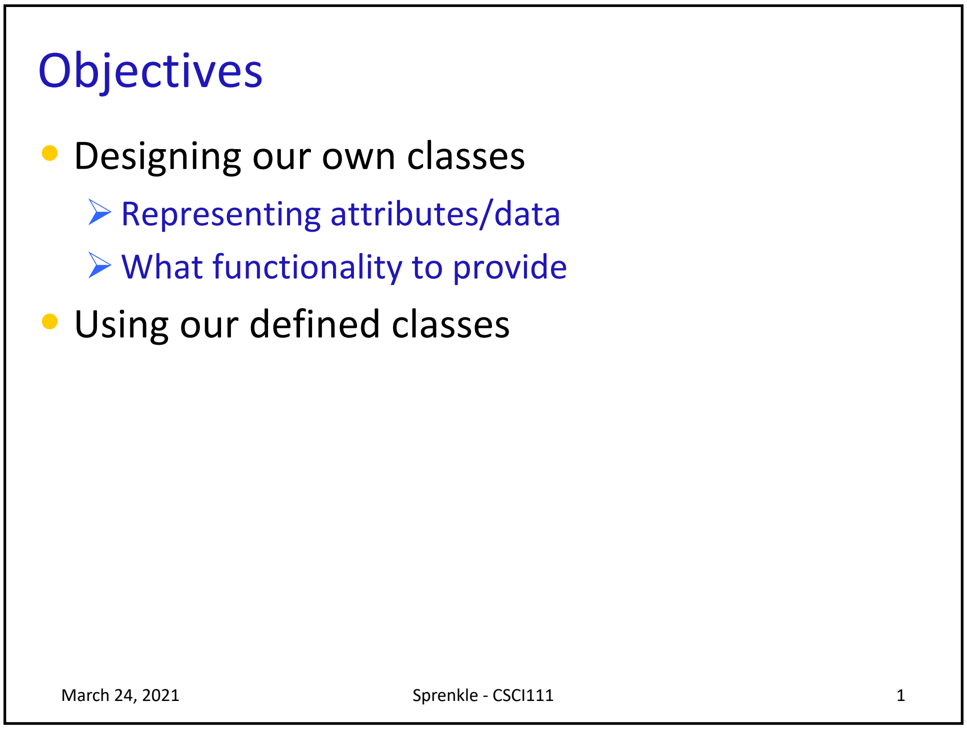
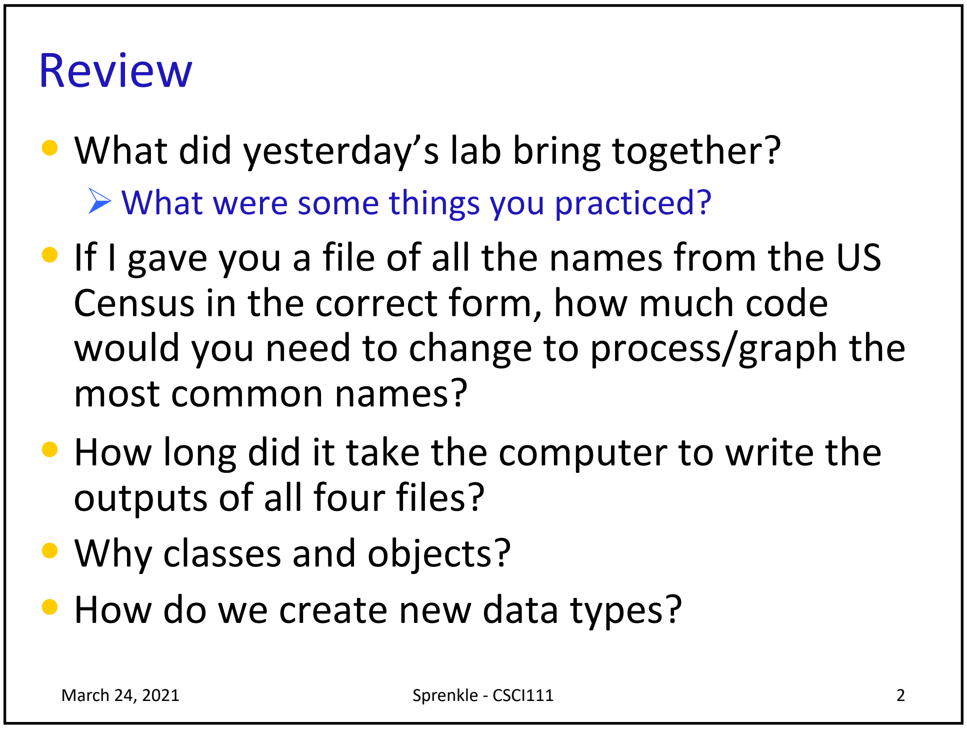


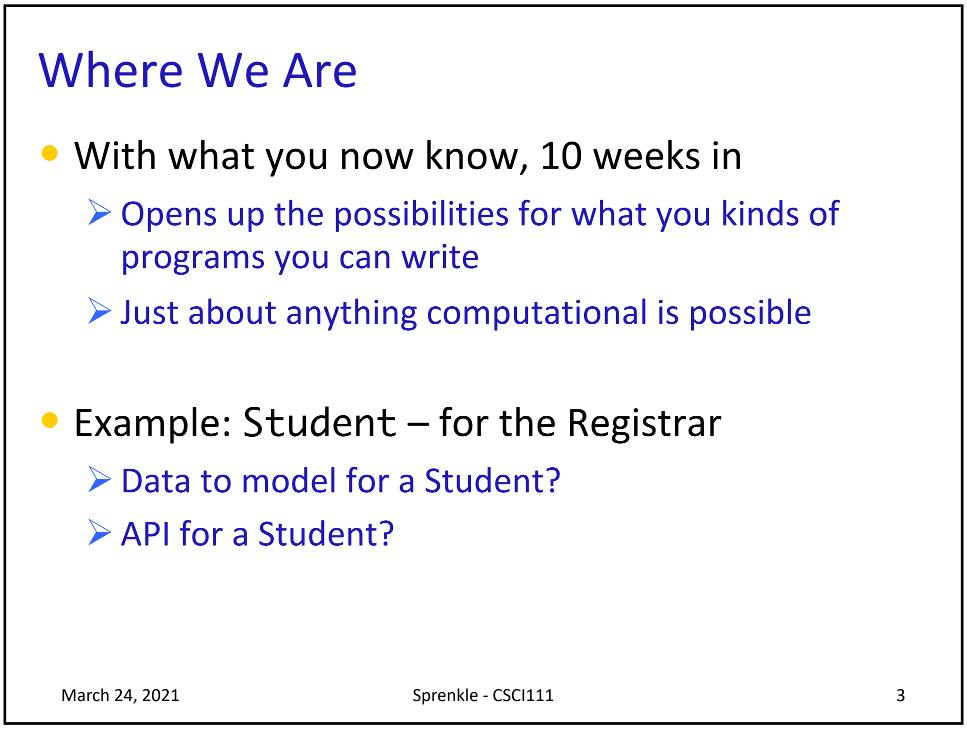
Objectives

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



Review

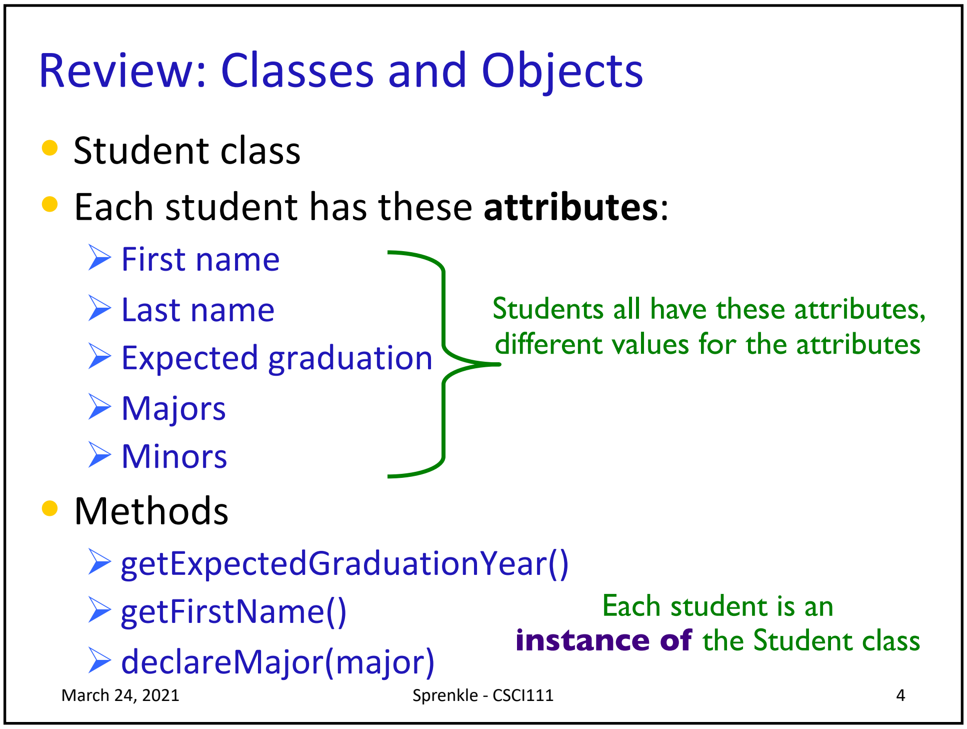
- What did yesterday's lab bring together?
  - What were some things you practiced?
- If I gave you a file of all the names from the US Census in the correct form, how much code would you need to change to process/graph the most common names?
- How long did it take the computer to write the outputs of all four files?
- Why classes and objects?
- How do we create new data types?



Where We Are

- With what you now know, 10 weeks in
  - Opens up the possibilities for what you kinds of programs you can write
  - Just about anything computational is possible
- Example: `Student` – for the Registrar
  - Data to model for a Student?
  - API for a Student?

3



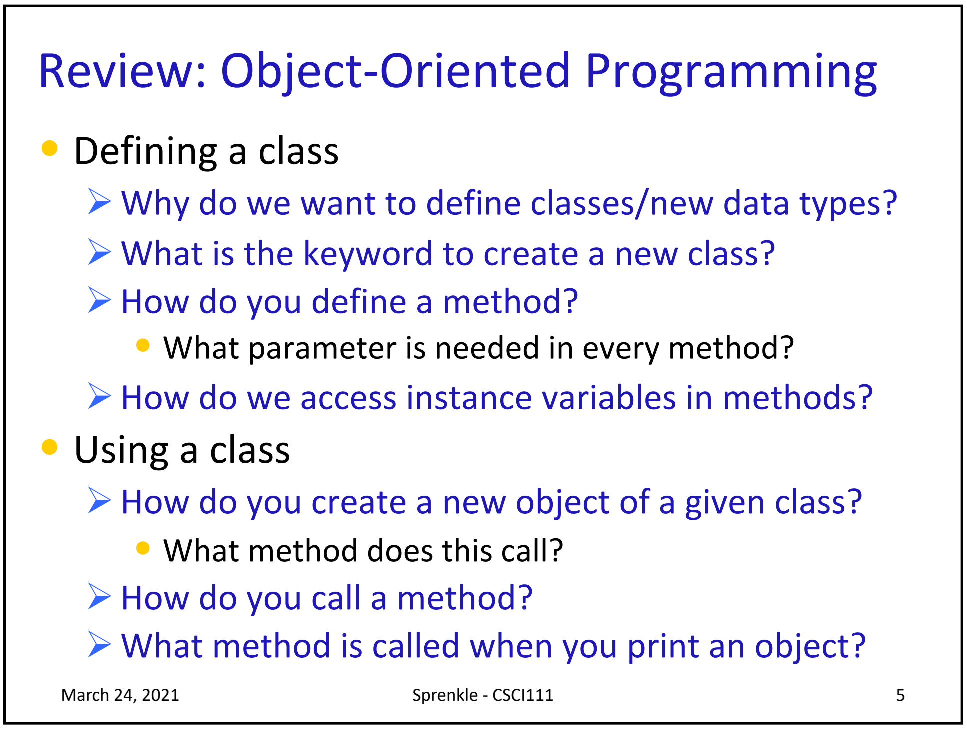
Review: Classes and Objects

- Student class
- Each student has these **attributes**:
  - First name
  - Last name
  - Expected graduation
  - Majors
  - Minors
- Methods
  - getExpectedGraduationYear()
  - getFirstName()
  - declareMajor(major)

Students all have these attributes, different values for the attributes

Each student is an **instance of** the Student class

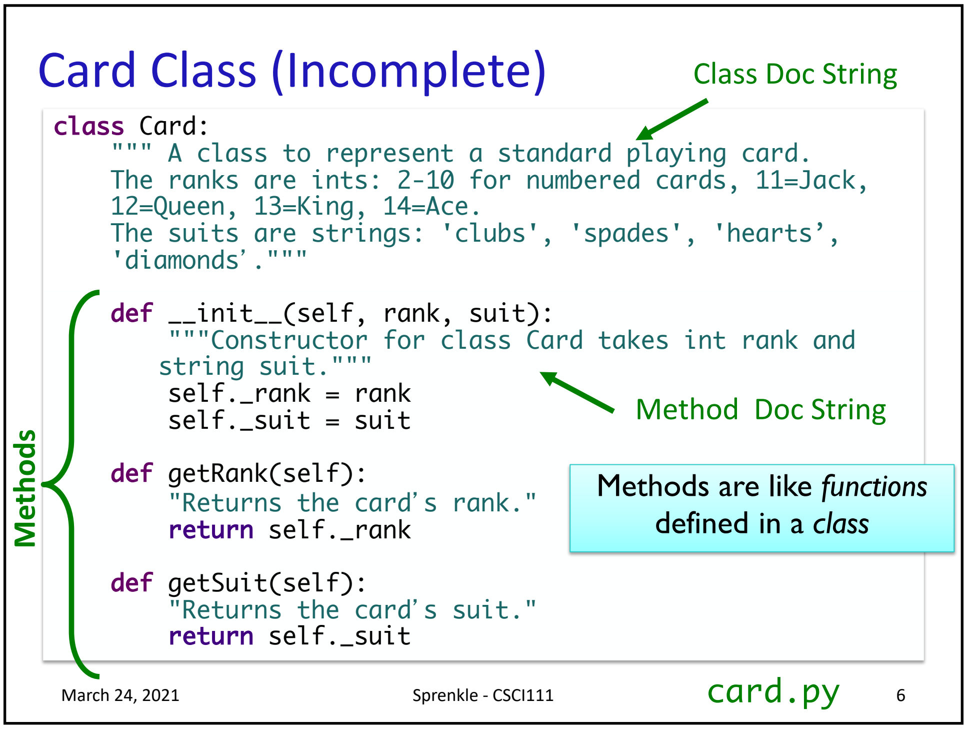
4



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?
  - How do we access instance variables in methods?
- 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?

5



Card Class (Incomplete)

Class Doc String

```
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
```

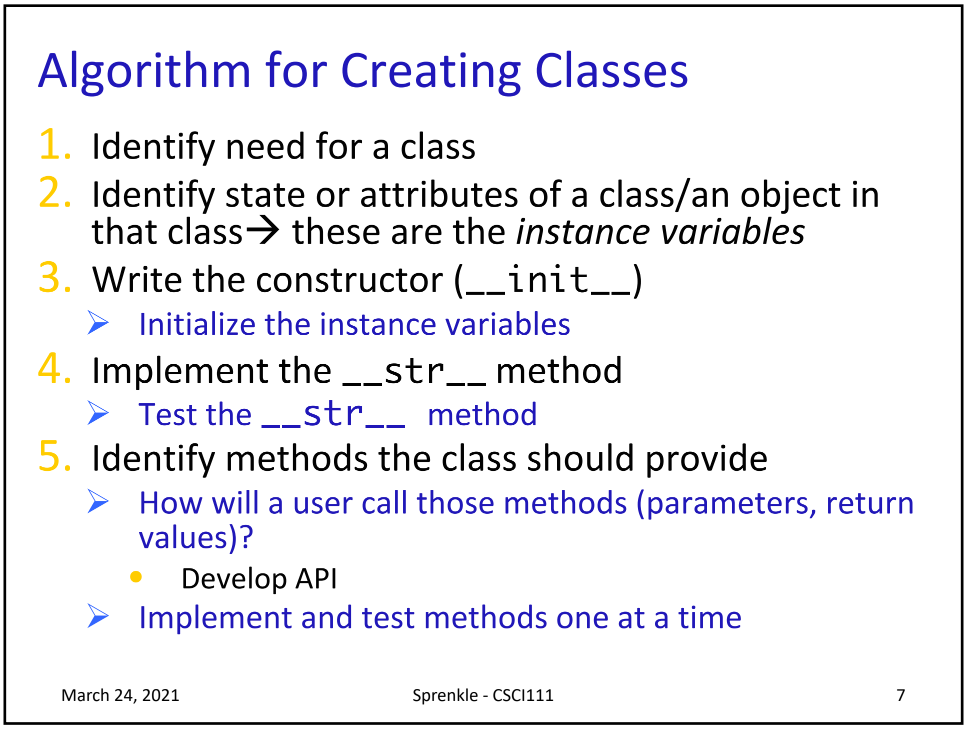
Method Doc String

Methods

Methods are like *functions* defined in a *class*

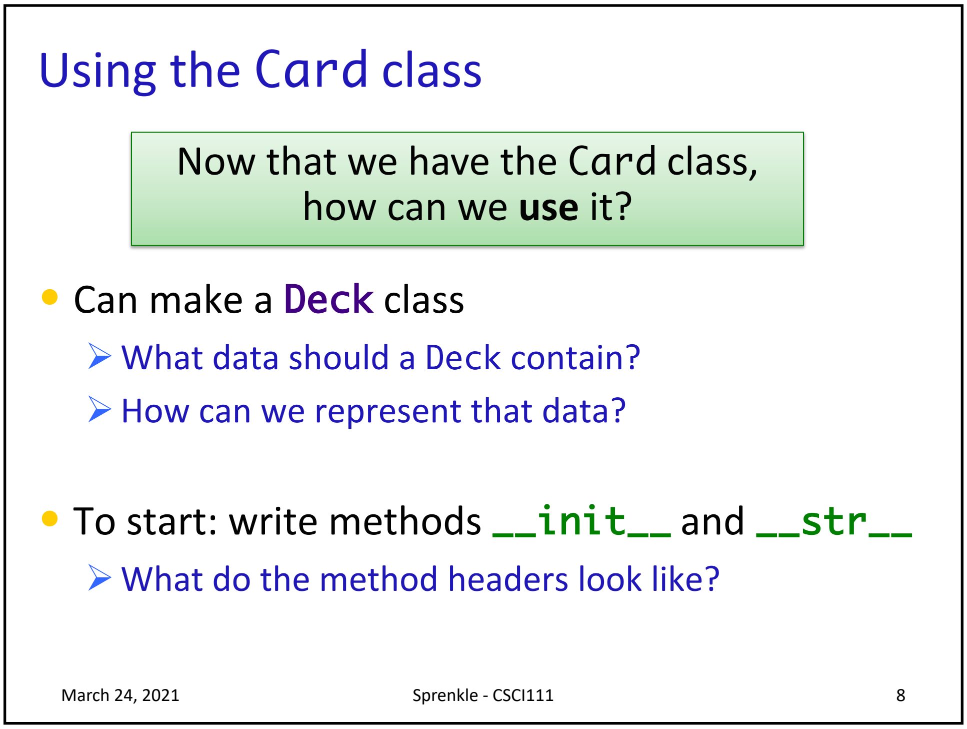
card.py

6



Algorithm for Creating Classes

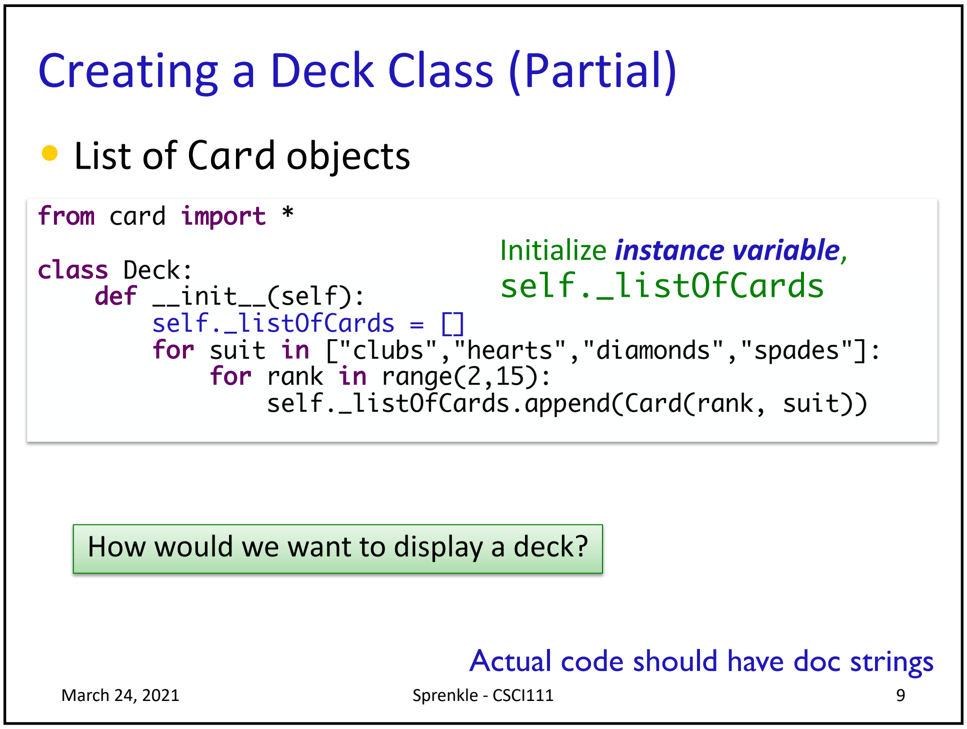
1. Identify need for a class
2. Identify state or attributes of a class/an object in that class→ these are the *instance variables*
3. Write the constructor (`__init__`)
   - Initialize the instance variables
4. Implement the `__str__` method
   - Test the `__str__` method
5. Identify methods the class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



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?



Creating a Deck Class (Partial)

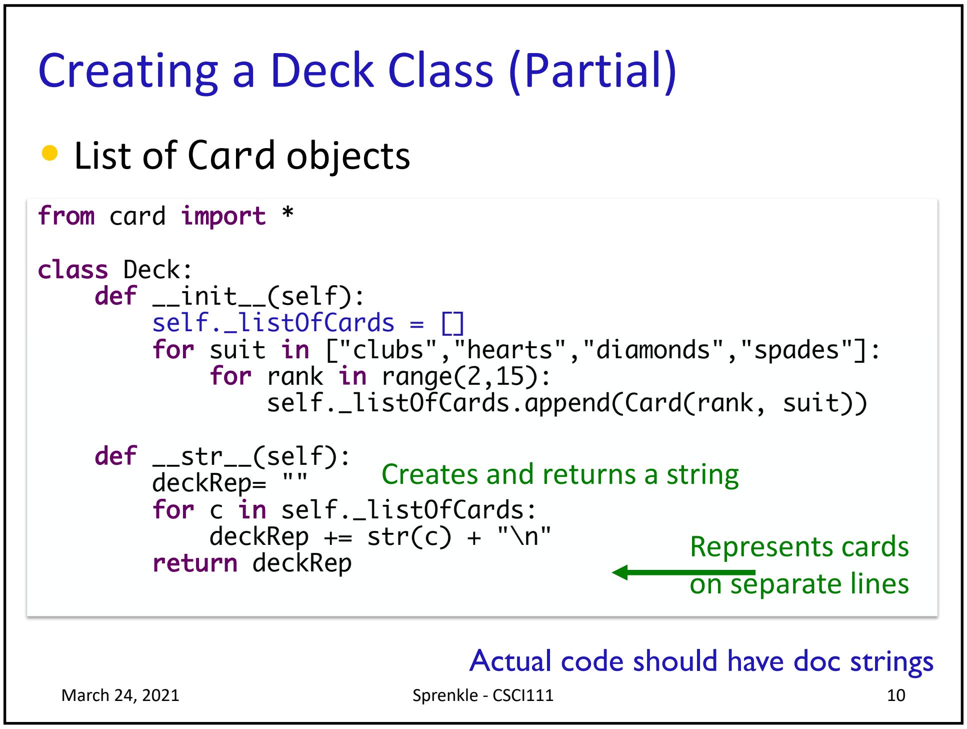
- List of Card objects

```
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



Creating a Deck Class (Partial)

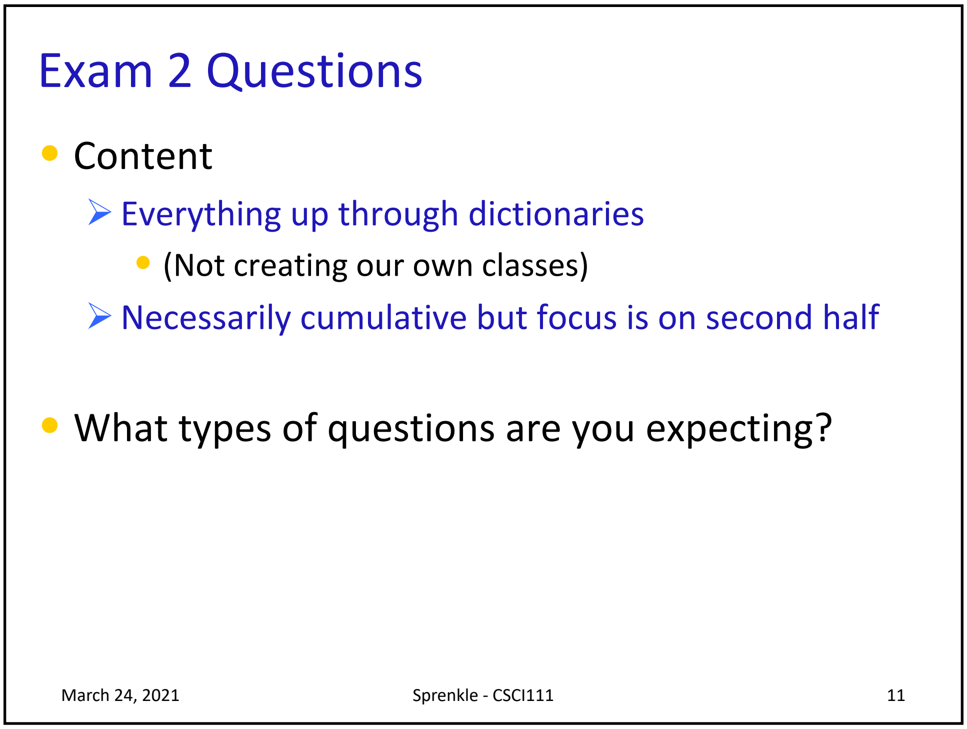
- List of Card objects

```
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):
        deckRep= ""
        for c in self._listOfCards:
            deckRep += str(c) + "\n"
        return deckRep
```

Creates and returns a string

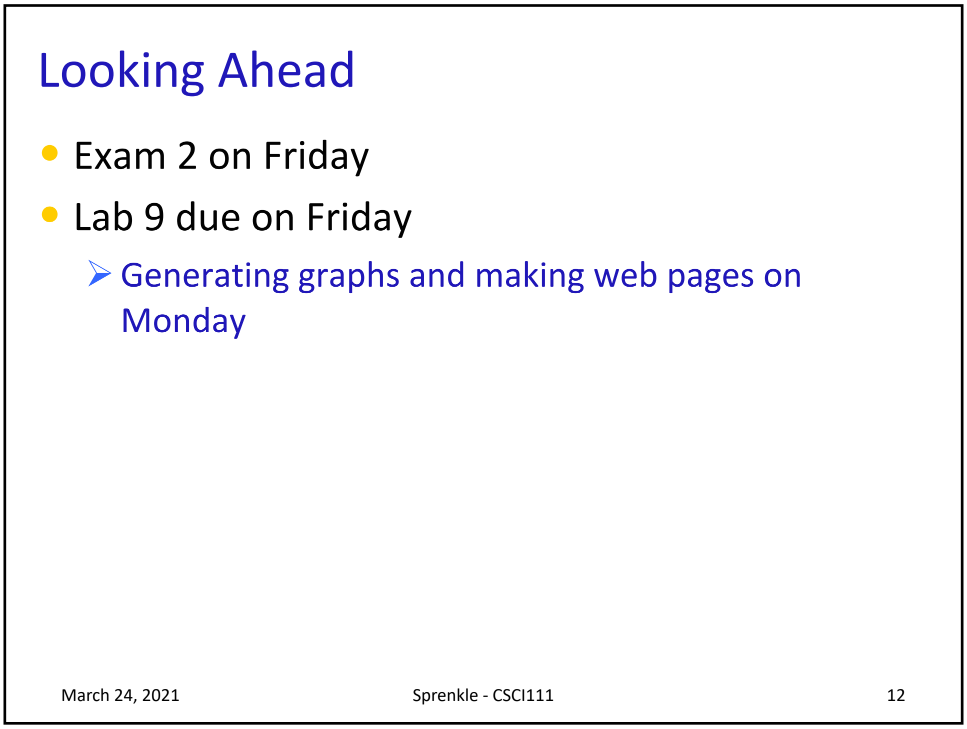
Represents cards on separate lines

Actual code should have doc strings



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?



Looking Ahead

- Exam 2 on Friday
- Lab 9 due on Friday
  - Generating graphs and making web pages on Monday