As you arrive:

- 1. Start up your computer and plug it in
- 2. Log into Angel and go to CSSE 120
- 3. Do the Attendance Widget the PIN is on the board
- 4. Go to the course Schedule Page
- 5. Open the **Slides** for today if you wish
- 6. Check out today's project: Session18 BlackJackWithClasses

Defining classes part 2

- Blackjack with classes
- Defining and exploring the card class
- Integrating other classes
- Object interaction

Session 18

Plus lots of in-class time to work on team project.

Project work:

Work in your team to complete next milestone

CSSE 120 – Introduction to Software Development

Checkout project: Session18_BlackJackWithClasses

Troubles gettingtoday's project?If so:

Are you in the Pydev perspective? If not:

• Window ~ Open Perspective ~ Other then Pydev

Messed up views? If so:

• Window ~ Reset Perspective

No SVN repositories view (tab)? If it is not there:

• Window ~ Show View ~ Other then SVN ~ SVN Repositories

In your SVN repositories view (tab), expand your repository (the top-level item) if not already expanded.

• If no repository, perhaps you are in the wrong Workspace. Get help as needed.

Right-click on today's project, then select **Checkout**. Press **OK** as needed.

The project shows up in the

Pydev Package Explorer to the right. Expand and browse the modules under **src** as desired.

Review of Key Ideas

□ Constructor:

Defined with special name _____init____

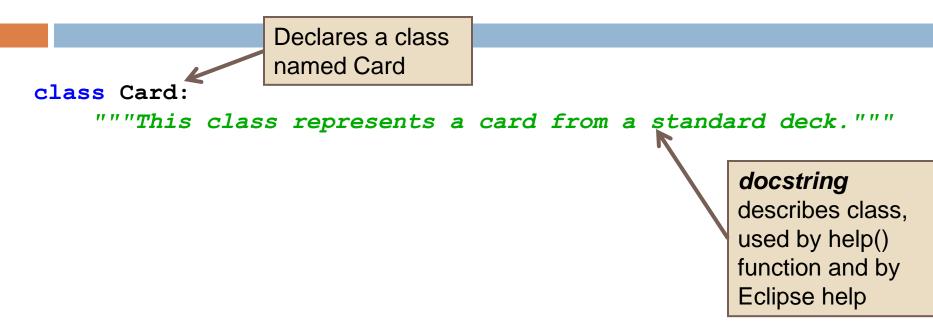
- Called like ClassName ()
- Instance variables:
 - Created when we assign to them
 - Live as long as the object lives
- self formal parameter:
 - Implicitly get the value before the dot in the call
 - Allows an object to "talk about itself" in a method

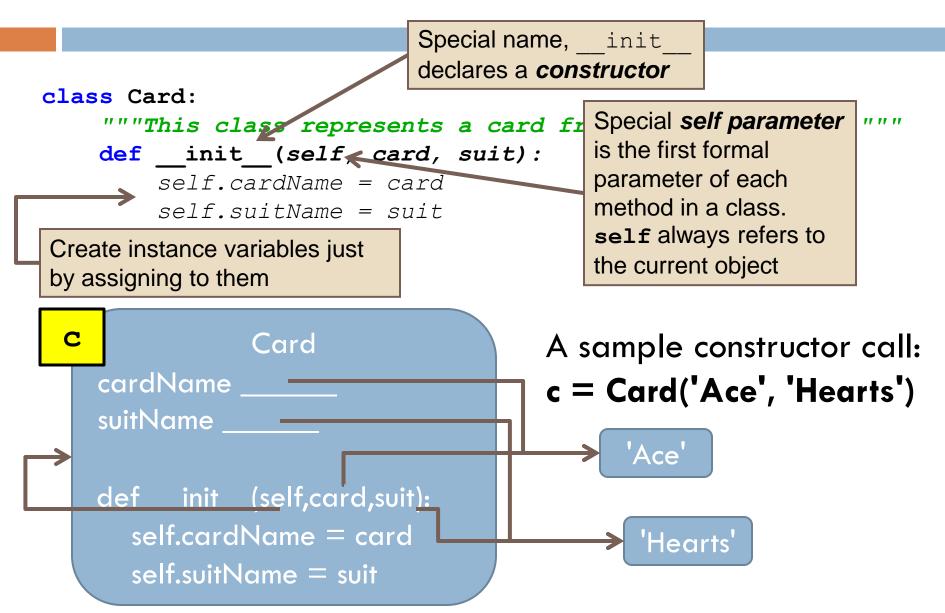
Creating Custom Objects: Defining Your Own Classes

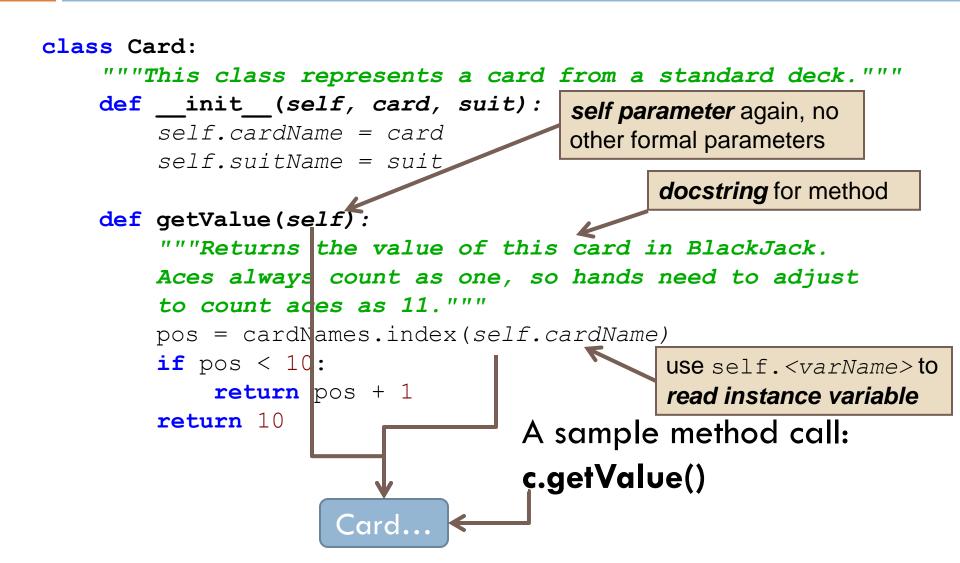
Custom objects:

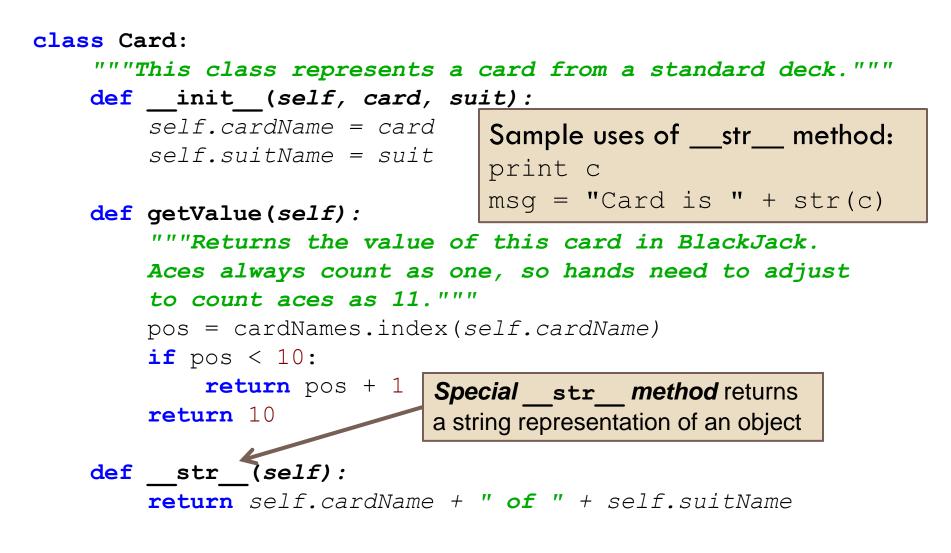
- Hide complexity
- Provide another way to break problems into pieces
- Make it easier to pass information around
- Example: Cards, Decks, and Hands

Recall from BlackJack:
suits = ['Clubs', 'Diamonds', 'Hearts', 'Spades']
cardNames = ['Ace', 'Deuce', '3', '4', '5', '6', '7',
'8', '9', '10', 'Jack','Queen', 'King']









Stepping Through Some Code

Sample use:

card = Card('7', 'Clubs')

print card.getValue()

class Card:

"""This class represents a card print card

def init (self, card, suit):

self.cardName = card
self.suitName = suit

```
def getValue(self):
    """Returns the value of this card in BlackJack.
    Aces always count as one, so hands need to adjust
    to count aces as 11."""
    pos = cardNames.index(self.cardName)
    if pos < 10:
        return pos + 1
    return 10</pre>
```

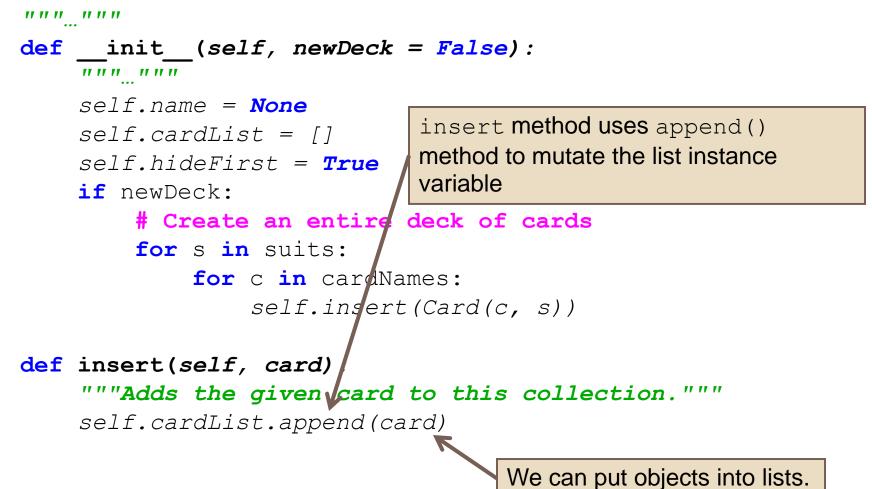
def __str__(self):
 return self.cardName + " of " + self.suitName

Another Example: Lists of Objects, Lists in Object

class CardCollection: """This class represents a collection of cards, either a single hand or the whole deck.""" def init (self, newDeck = False): """Creates a new collect Optional formal parameter, can it's empty, but if newbe construct a CardCollection three ways: **collection is a full sta** deck = CardCollection(True) self.name = None hand = CardCollection(False) self.cardList = [] hand = CardCollection() self.hideFirst = True if newDeck: # Create an entire deck of cards for s in suits: for c in cardNames: Instance variable self.insert(Card(c, s)) containing a list . . . Self call, constructor calls another method of the CardCollection class

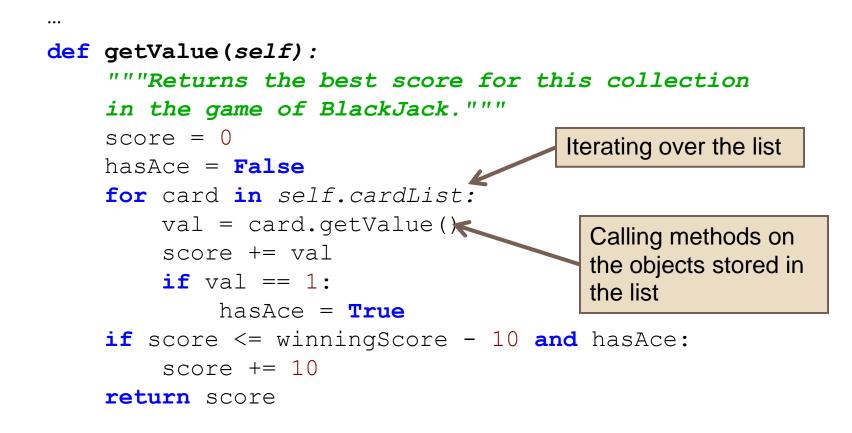
Another Example (continued): Lists of Objects, Lists in Object

class CardCollection:



Why not just use a list of cards?

class CardCollection:



Work on your team project

- Meet with your project team
 - Finish up what is due for session 18 milestone
 - Continue working on next milestone
 - Decide on time/venue for next meeting