Lists of Lists
And Solving Everyday Problems with Lists
Two Types of Lists

Useful List Operations

Sorting out cats

Two Approaches to Filtering Data

Examples
Useful List Operations

```python
>>> L = [3, 10, 3, 5, 1, -1, 0, 6]
>>> L
[3, 10, 3, 5, 1, -1, 0, 6]
>>> L.sort()
>>> L
[-1, 0, 1, 3, 3, 5, 6, 10]
>>> L.reverse()
>>> L
[10, 6, 5, 3, 3, 1, 0, -1]
>>> L[0]
10
>>> L[0:3]
[10, 6, 5]
>>> L[-3:]
[1, 0, -1]
>>> S = set(L)
>>> S
{0, 1, 3, 5, 6, 10, -1}
>>> L = list(S)
>>> L
[0, 1, 3, 5, 6, 10, -1]
```
Useful List Operations

Sorting Tuples

```python
>>> L = [ (10, "Smith"), (1, "Amherst"), (3, "Umass"), (5, "Hampshire") ]

>>> L
[(10, 'Smith'), (1, 'Amherst'), (3, 'Umass'), (5, 'Hampshire')]

>>> L.sort()

>>> L
[(1, 'Amherst'), (3, 'Umass'), (5, 'Hampshire'), (10, 'Smith')]

>>> L2 = [ ("Smith", 10), ("Amherst", 1), ("Umass", 3), ("Hampshire", 5) ]

>>> L2.sort()

>>> L2
[('Amherst', 1), ('Hampshire', 5), ('Smith', 10), ('Umass', 3)]
```
Two Types of Lists

Useful List Operations

**Sorting out cats**

Two Approaches to Filtering Data

Examples
def __str__(self):
    if self.vaccinated == True:
        vacc = "vaccinated"
    else:
        vacc = "not vaccinated"
    return "\{0:20\}==\{1:1\}, \{2:1\}, \{3:1\} yrs old".format(
        self.name, self.breed, vacc, self.age )

def main():
    cats = []
cats.append( Cat( "Minou", 3, True, "stray" ) )
cats.append( Cat( "Max", 1, False, "Burmese" ) )
cats.append( Cat( "Gizmo", 2, True, "Bengal" ) )
cats.append( Cat( "Garfield", 2, False, "Orange Tabby" ) )

    print( "\nComplete list: " )
    for cat in cats:
        print( cat )

    print( "\nCats sorted by age: " )
cats.sort()
    for cat in cats:
        print( cat )

main()
Complete list:

- Minou: stray, vaccinated, 3 yrs old
- Max: Burmese, not vaccinated, 1 yrs old
- Gizmo: Bengal, vaccinated, 2 yrs old
- Garfield: Orange Tabby, not vaccinated, 2 yrs old

Cats sorted by age:

Traceback (most recent call last):
  File "~/Users/thiebaut/Desktop/Dropbox/111/sortingCats.py", line 58, in <module>
    main()
  File "~/Users/thiebaut/Desktop/Dropbox/111/sortingCats.py", line 55, in main
cats.sort()
TypeError: unorderable types: Cat() < Cat()
Default <> == != Operators

```python
# Cats.py
# D. Thiebaut
# Minou, 3, vac, stray
# Max, 1, not-vac, Burmese
# Gizmo, 2, vac, Bengal
# Garfield, 4, not-vac, Orange Tabby

class Cat:
    def __init__( self, na, ag, vacc, bre ):
        self.name = na
        self.age = ag
        self.vaccinated = vacc
        self.breed = bre
        return

    def __gt__( self, otherCat ):
        return self.age > otherCat.age

    def __lt__( self, otherCat ):
        return self.age < otherCat.age

    def getName( self ):
        return self.name

```

https://docs.python.org/3/reference/datamodel.html?highlight=__gt__#object.__gt__
Two Types of Lists

Useful List Operations

Sorting out cats

Filtering Data (Everyday Python)

Examples
Two Types of Filtering Problems…
The Problem at Hand

Textual Info.
The Problem at Hand

OPTION 1:
We are only interested in the red information, and only the smaller or larger items…
The Problem at Hand

[ ( , , ),
( , , ),
( , , ),
...
( , , ) ]
The Problem at Hand

[ ( , , ),
  ( , , ),
  ( , , ),
  ...
  ( , , ) ]

SORT

[ ( , , ),
  ( , , ),
  ( , , ),
  ...
  ( , , ) ]

Textual Info.
OPTION 2:
We are only interested in the lines that contain the red information
The Problem at Hand

Textual Info.

FILTER

\[
\begin{align*}
&\left[ (\quad, \quad, \quad),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left(\quad, \quad, \quad\right),
\left.\left(\quad, \quad, \quad\right)\right]\nonumber
\end{align*}
\]
Two Types of Lists

Useful List Operations

Two Approaches to Filtering Data

Examples
Example 1
10 **Rainiest** Months In **Cambridge**, U.K.?

http://cs.smith.edu/~dthiebaut/UKTemperatures/

https://www.metoffice.gov.uk/public/weather/climate-historic/#?tab=climateHistoric

- Mean daily maximum temperature (tmax)
- Mean daily minimum temperature (tmin)
- Days of air frost (af)
- Total rainfall (rain)
- Total sunshine duration (sun)
Example 2
List of email addresses for students enrolled in several classes.

Need a list of all Smith students without duplicates and a list of all 5-College students without duplicates
Who was president in 1939?

Presidency, President, Took office, Left office, Party, Home State
1. George Washington, 30/04/1789, 4/03/1797, Independent, Virginia
2. John Adams, 4/03/1797, 4/03/1801, Federalist, Massachusetts
3. Thomas Jefferson, 4/03/1801, 4/03/1809, Democratic-Republican, Virginia
4. James Madison, 4/03/1809, 4/03/1817, Democratic-Republican, Virginia
5. James Monroe, 4/03/1817, 4/03/1825, Democratic-Republican, Virginia
6. John Quincy Adams, 4/03/1825, 4/03/1829, Democratic-Republican/National Republican, Massachusetts
7. Andrew Jackson, 4/03/1829, 4/03/1837, Democratic, Tennessee
8. Martin Van Buren, 4/03/1837, 4/03/1841, Democratic, New York
9. William Henry Harrison, 4/03/1841, 4/04/1841, Whig, Ohio
10. John Tyler, 4/04/1841, 4/03/1845, Whig, Virginia
11. James K. Polk, 4/03/1845, 4/03/1849, Democratic, Tennessee
12. Zachary Taylor, 4/03/1849, 9/07/1850, Whig, Louisiana
14. Franklin Pierce, 4/03/1853, 4/03/1857, Democratic, New Hampshire
15. James Buchanan, 4/03/1857, 4/03/1861, Democratic, Pennsylvania
16. Abraham Lincoln, 4/03/1861, 15/04/1865, Republican/National Union, Illinois
17. Andrew Johnson, 15/04/1865, 4/03/1869, Democratic/National Union, Tennessee
18. Ulysses S. Grant, 4/03/1869, 4/03/1877, Republican, Ohio
19. Rutherford B. Hayes, 4/03/1877, 4/03/1881, Republican, Ohio
20. James A. Garfield, 4/03/1881, 19/09/1881, Republican, Ohio
22. Grover Cleveland, 4/03/1885, 4/03/1889, Democratic, New York
23. Benjamin Harrison, 4/03/1889, 4/03/1893, Republican, Indiana
24. Grover Cleveland, 4/03/1893, 4/03/1897, Democratic, New York
25. William McKinley, 4/03/1897, 14/9/1901, Republican, Ohio
27. William Howard Taft, 4/3/1909, 4/03/1913, Republican, Ohio
28. Woodrow Wilson, 4/03/1913, 4/03/1921, Democratic, New Jersey
29. Warren G. Harding, 4/03/1921, 2/8/1923, Republican, Ohio
30. Calvin Coolidge, 2/8/1923, 4/03/1929, Republican, Massachusetts
31. Herbert Hoover, 4/03/1929, 4/03/1933, Republican, Iowa
32. Franklin D. Roosevelt, 4/03/1933, 12/4/1945, Democratic, New York
33. Harry S. Truman, 12/4/1945, 20/01/1953, Democratic, Missouri
34. Dwight D. Eisenhower, 20/01/1953, 20/01/1961, Republican, Texas
40. Ronald Reagan, 20/01/1981, 20/01/1989, Republican, California
42. Bill Clinton, 20/01/1993, 20/01/2001, Democratic, Arkansas
43. George W. Bush, 20/01/2001, 20/01/2009, Republican, Texas
44. Barack Obama, 20/01/2009, 20/01/2017, Democratic, Illinois
45. Donald Trump, 20/01/2017, 6/1/2021, Republican, New York
46. Joe Biden, 6/1/2021, 6/1/2025, Democratic, Delaware
We stopped here last time...
Class Inheritance
(End of Chapter 12)
Class Inheritance
The Idea...
Car class
- constructor
- draw
- move
- undraw
- changeColor
- etc...
Car class
- constructor
- draw
- move
- undraw
- etc...

Wanted with same features
Car class
- constructor
- draw
- move
- undraw
- changeColor
- etc…

Wanted with same features

Wanted with different features
• We *could* write complete new classes for the two new car shapes…

• But instead, we can *save code, save time, save debugging aggravation*, by reusing the original Car class.
• We *could* write complete new classes for the two new car shapes…

• But instead, we can **save code, save time, save debugging aggravation**, by reusing the original Car class.

• We will **derive** a new class from the Car class. The new class will be **derived** from it, and will **inherit** all its member variables and methods.

• The original Car class will become the **super** class.
Another Way of Looking at Inheritance

Class

value
Another Way of Looking at Inheritance
Coding Exercise

GenericCar

CarWithTop

Taxi
Coding Exercise

• Create a generic car class with 2 wheels and a body. Add methods to set the speed and to set the color. Add a move method.

• Derive a Taxi class from the generic class. The color will always be yellow, and the word "TAXI" will be on the body.

• Derive a car with a top from the generic car class.
# genericCar.py
# D. Thiebaut
# The definition for a generic Car class
from graphics import *
from random import *

class GenericCar:
    """Definition for a car with a body and two wheels""

    def __init__(self, win, topLeft, width, height):
        """constructs a car made of 1 rectangle with top-left
        point topLeft, dimension width x height, and two wheels
        away from left and right by 10 pixels""
        # save width and height of car
        self.width = width
        self.height = height

        # create bottom-right point
        x1 = topLeft.getX()
        y1 = topLeft.getY()
        P2 = Point( x1+width, y1+height )

        # body is a rectangle between topLeft and P2
        self.body = Rectangle( topLeft, P2 )
        self.body.setFill( "yellow" )

        # create wheel #1
The Main Program

```python
# useGenericCar.py
# D. Thiebaut

from genericCar import *
from graphics import *

def main():
    win = GraphWin( "Cars Cars Cars", 700, 500 )

    car = GenericCar( win, Point( 100, 100 ), 200, 50 )
    car.draw( win )
    car.setSpeed( -1.5 )
    car.setFill( "blue" )

    taxi = Taxi( win, Point( 150, 300 ), 200, 50 )
    taxi.setSpeed( +2.0 )
    taxi.setFill( "grey" )
    taxi.draw( win )

    car2 = CarWithTop( win, Point( 250, 320 ), 210, 50 )
    car2.setSpeed( -1.5 )
    car2.setFill( "pink" )
    car2.draw( win )

    while True:
        car.move( )
        taxi.move( )
```
Two Different Syntaxes

```python
def draw(self, win):
    super().draw(win)
    self.top.draw(win)

def draw(self, win):
    GenericCar.draw(self, win)
    self.top.draw(win)
```
Two Different Syntaxes

```
def draw( self, win ):  
    super().draw( win )  
    self.top.draw( win )
```

```
def draw( self, win ):  
    Car.draw( self, win )  
    self.top.draw( win )
```
Graphic Libraries Are Built on Inheritance: Hierarchy of Classes
The Essential Qt 3.0 Class Hierarchy
class inheritance hierarchy example

Google

All Images Maps Videos News More

employee  C++ composition java  uml class

Java Inheritance Hierarchy:

Base Class

Child Class 1

Child Class 2

………….. Child Class n

Fig: Hierarchical Inheritance
• Open the **graphics.py** library

• Locate the **Rectangle** class and observe its code

• In a new program (**RectLabelClass.py**), create a new class derived from Rectangle that supports a **text label** in the middle of the rectangle.