Python OO Tutorials

Complete each section with activity

Contents

[1: A class called Cat aka OO 1](#_Toc111536006)

[2: Good Cat, Bad Cat aka OO extending Class 3](#_Toc111536007)

[3: Very Bad Cat exceptions 4](#_Toc111536008)

[4: Cage for a Very Bad Cat aka decorators 5](#_Toc111536009)

# 1: A class called Cat aka OO

**Required Knowledge:**Watch Classes tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=ZDa-Z5JzLYM>   
  
Watch Class variables tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=BJ-VvGyQxho>  
  
Watch Class functions tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=rq8cL2XMM5M>

Read instructions on how classes work

<https://python.swaroopch.com/oop.html>

**Activities:**

Create a class called “Cat” with a function called “meow” when called it will print out “meow”  
  
Import this Cat class into a cat\_runner.py file with the following contents  
  
cat: Cat = Cat()  
cat.meow()  
# should print out meow

$ python cat\_runner.py  
meow

# 2: Good Cat, Bad Cat aka OO extending Class

**Required Knowledge:**  
Watch Class inheritance tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=RSl87lqOXDE>

Read instructions on how classes work

<https://python.swaroopch.com/oop.html>

**Activities:**

Extend your previous class “Cat” into two new Classes

* GoodCat which overrides “meow” function to print “purr purr meow”
* BadCat which overrides “meow” function to print “hiss”

Import these Cat classes into cat\_runner.py file with the following contents

print('good cats go')  
cat: Cat = GoodCat()  
cat.meow()  
# should print out purr purr meow

print('bad cats go')  
cat: Cat = BadCat ()  
cat.meow()  
# should print out hiss

$ python cat\_runner.py

good cats go  
purr purr meow  
bad cats go

hiss

# 3: Very Bad Cat exceptions

**Required Knowledge:**Watch Corey’s amazing try/catch explainer  
<https://www.youtube.com/watch?v=NIWwJbo-9_8>

Watch Magic setter/getter tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=3ohzBxoFHAY>

Read instructions on how try/catch work  
<https://python.swaroopch.com/exceptions.html>

**Activities:**  
1) create Very Bad Cat Class

Extend your previous class “Cat” into new Class

* VeryBadCat which overrides “meow” function to throw Exception("Nope, scratch scratch")  
  hint: raise Exception("Nope, scratch scratch")

Import the Cat class into cat\_runner.py file with the following contents

print('very bad cats need cages')  
cat: Cat = VeryBadCat()  
cat.meow()  
# should crash

$ python cat\_runner.py

very bad cats need cages  
<crash stack>

# 4: Cage for a Very Bad Cat aka decorators

**Required Knowledge:**Watch Magic decorator tutorial from the excellent Corey  
<https://www.youtube.com/watch?v=jCzT9XFZ5bw>

Read instructions on how decorator work  
<https://python.swaroopch.com/more.html>

**Activities:**  
  
1) create decorator/annotation cat\_cage

- create decorator, that runs given function in a try / except statement, if the function exceptions catch error passes it to this print function like so print("Cat was bad: {error}, no harm was done due to Cage")

Import the All the previous Cat classes and cat\_cage into cat\_runner.py file with the following contents

print('very bad cats need cages')

cat1: Cat = GoodCat()  
cat2: Cat = BadCat ()  
cat3: Cat = VeryBadCat()

@cat\_cage  
def run\_in\_cage ():  
 cat1.meow()

cat2.meow()

cat3.meow()  
run\_in\_cage()

$ python cat\_runner.py

purr purr meow

hiss  
Cat was bad: very bad cats need cages, no harm was done due to Cage