BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 17 EXERCISE 17.7
ILLINOIS STATE UNIVERSITY

This exercise is a cautionary tale about one of the most common, and difficult to find, errors in Python. Write a definition for a class named Kangaroo with the following methods:

1. An __init__ method that initializes an attribute named pouch_contents to an empty list.

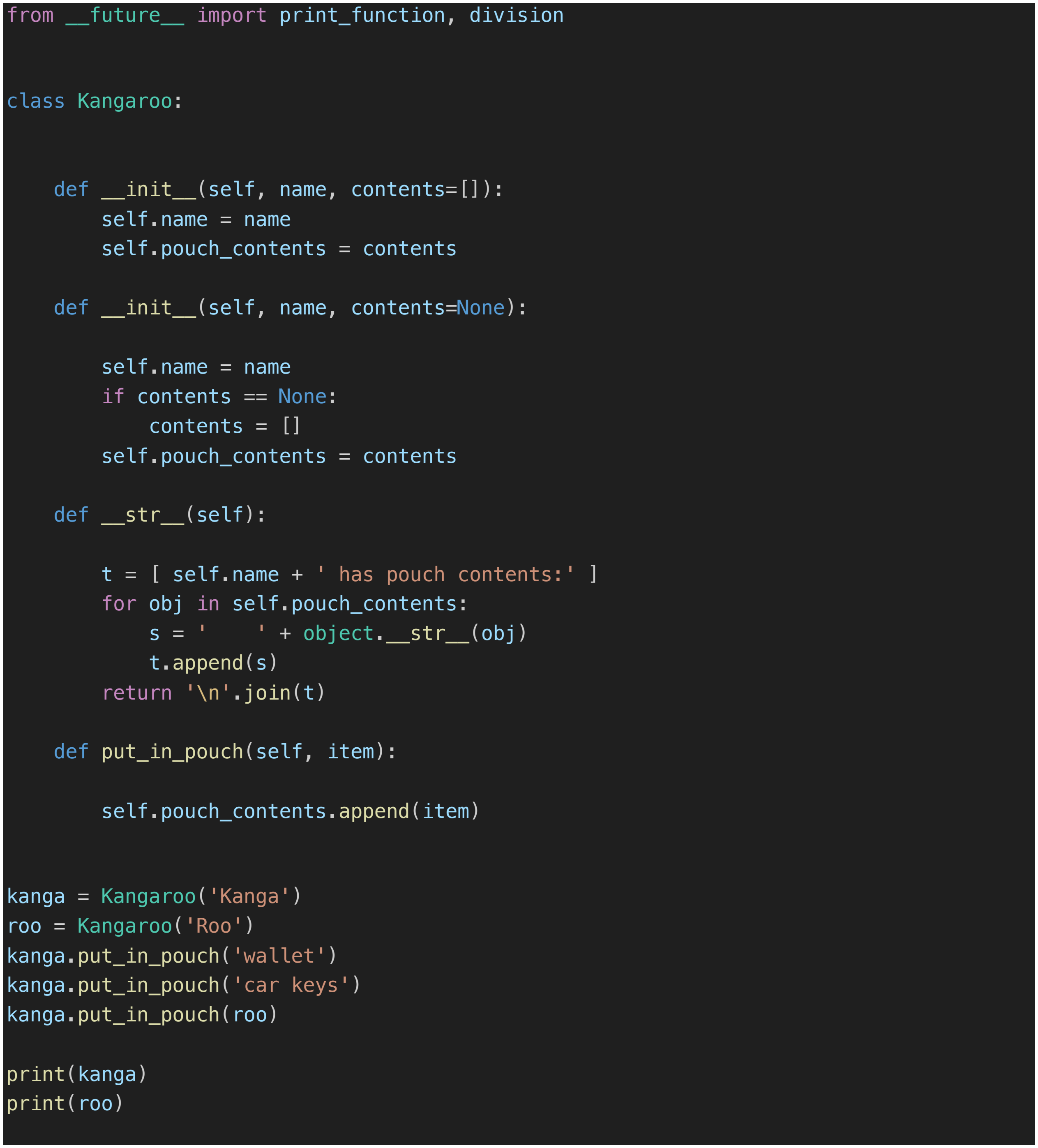
2. A method named put_in_pouch that takes an object of any type and adds it to pouch_contents.

3. A __str__ method that returns a string representation of the Kangaroo object and the contents of the pouch.

Test your code by creating two Kangaroo objects, assigning them to variables named kanga and roo, and then adding roo to the contents of kanga's pouch. Classes and methods. Download http: // thinkpython. com/ code/ BadKangaroo. py . It contains a solution to the previous problem with one big, nasty bug. Find and fix the bug.

If you get stuck, you can download http: // thinkpython. com/ code/ GoodKangaroo. py , which explains the problem and demonstrates a solution.

```
from __future__ import print_function, division


class Kangaroo:



    def __init__(self, name, contents=[]):
        self.name = name
        self.pouch_contents = contents


    def __init__(self, name, contents=None):
        self.name = name
        if contents == None:
            contents = []
        self.pouch_contents = contents

    def __str__(self):

        t = [ self.name + ' has pouch contents:' ]
        for obj in self.pouch_contents:
            s = '    ' + object.__str__(obj)
            t.append(s)
        return '\n'.join(t)

    def put_in_pouch(self, item):

        self.pouch_contents.append(item)

kanga = Kangaroo('Kanga')
roo = Kangaroo('Roo')
kanga.put_in_pouch('wallet')
kanga.put_in_pouch('car keys')
kanga.put_in_pouch(roo)

print(kanga)
print(roo)
```



```python
from __future__ import print_function, division


class Kangaroo:


    def __init__(self, name, contents=[]):
        self.name = name
        self.pouch_contents = contents

    def __init__(self, name, contents=None):

        self.name = name
        if contents == None:
            contents = []
        self.pouch_contents = contents

    def __str__(self):

        t = [ self.name + ' has pouch contents:' ]
        for obj in self.pouch_contents:
            s = '    ' + object.__str__(obj)
            t.append(s)
        return '\n'.join(t)

    def put_in_pouch(self, item):

        self.pouch_contents.append(item)


kanga = Kangaroo('Kanga')
roo = Kangaroo('Roo')
kanga.put_in_pouch('wallet')
kanga.put_in_pouch('car keys')
kanga.put_in_pouch(roo)

print(kanga)
print(roo)
```