## Python课后题答案

### Python概述

#### 填空题

1. 对象
2. 可移植
3. 字节码
4. UTF-8

#### 判断题

1. 对
2. 对
3. 对
4. 对
5. 错
6. 错
7. 对
8. 错
9. 对
10. 错

#### 选择题

1. C
2. ABCD
3. B
4. C
5. D

#### 简答题。

1． Python具有的特点如下：

* 简单易学
* 开源
* 高级语言
* 可移植性
* 解释性
* 面向对象
* 可扩展性
* 丰富的库
* 规范的代码

2. Python可以应用在下列领域：

* web应用开发
* 操作系统管理、服务器运维的自动化脚本
* 科学计算
* 桌面软件
* 服务器软件（网络软件）
* 游戏
* 构思实现，产品早期原型和迭代

3. Python2.x和Python3.x的区别如下：

（1）在python2.x中，输出数据使用的是print语句。但是在python3.x中， print语句没有了，取而代之的是print()函数

（2）python2有ASCII Str()类型，unicode()是单独的，不是byte类型。在python3.x版本的源代码中，默认使用的是UTF-8编码，从而可以很好的支持中文字符。

（3）在python 2.x中，使用运算符/进行除法运算，整数相除的结果是一个整数，浮点数除法会保留小数点的部分得到一个浮点数的结果。在python 3.x中使用运算符/进行除法，整数之间的相除，结果也会是浮点数。

（4）相比python2.x版本，python3.x版本在异常方面有很多改变：

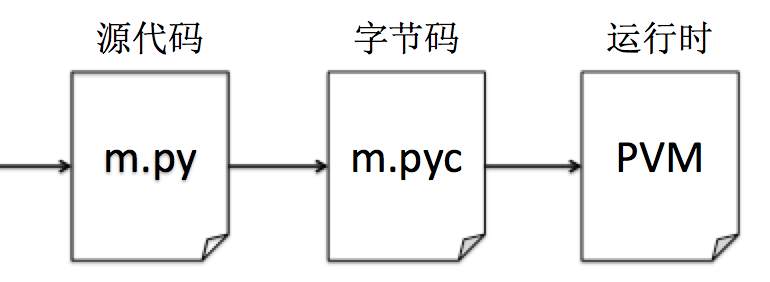
* 在python2.x版本，所有类型的对象都是直接被抛出的，但是，在python3.x版本中，只有继承自BaseException的对象才可以被抛出。
* 在python2.x版本中，捕获异常的语法是except exc，var。在python3.x版本中，引入了as关键字，捕获异常的语法变更为except exc as var。
* 在python2.x版本中，处理异常使用raise Exception(args)。在python3.x版本中，处理异常使用raiseException,args。
* python3.x取消了异常类的序列行为和.message属性。

（5）在Python 3.x中，表示八进制字面量的方式只有一种，并且必须写0o1000这样的方式，原来01000的方式不能使用了。

（6）Python 2.x中不等于有两种写法 != 和 <>；但是，Python 3.x中去掉了<>, 只有!=一种写法

（7）python3.x去除了long类型，现在只有一种整型int，但它的行为就像是python2.x版本的long。

4. Python程序的执行原理如图所示。



python解释器将源代码转换为字节码，然后把编译好的字节码转发到Python虚拟机（PVM）中进行执行

5. IPython的特点如下：

* IPython具有卓越的Python shell，其性能远远优于标准Python的shell。
* IPython支持变量自动补全，自动缩进，支持 bash shell 命令，内置了许多很有用的功能和函数。
* IPython提供了基于控制台命令环境的定制功能，可以十分轻松地将交互式Python shell包含在各种Python应用中，甚至可以当作系统级shell来使用。

#### 编程题。

1． 代码如下：

print("+++++++++++")  
 print("+ +")  
 print("+++++++++++")

2. 代码如下：

print("学Python，来传智播客黑马程序员")  
 print("学Python，来传智播客黑马程序员")  
 print("学Python，来传智播客黑马程序员")  
 print("学Python，来传智播客黑马程序员")  
 print("学Python，来传智播客黑马程序员")

### Python基本语法

#### 填空题

1. int
2. True False
3. 8
4. 1.2e5
5. 00000010
6. 0b10100
7. type()
8. 1
9. 20
10. 4.34×103

#### 判断题

1. 对
2. 错
3. 对
4. 对
5. 错
6. 错
7. 对
8. 对
9. 对
10. 对

#### 选择题

1. D
2. D
3. A
4. A
5. C
6. A
7. B
8. B
9. C
10. D

#### 简答题

1. Python中标识符的命名规则如下所示：

* 标示符由字母、下划线和数字组成，且数字不能开头
* python中的标识符是区分大小写的。
* python中的标示符不能使用关键字

2. 简述Python中的数字类型有：int（整型）、long（长整型）、float（浮点数）、complex（复数）

#### 编程题

1. 代码如下：

import math  
 a=float(input("请输入斜边 1 的长度")) #输入实数  
 b=float(input("请输入斜边 2 的长度")) #输入实数  
 c=a\*a+b\*b #计算,得到的是斜边的平方  
 c=sqrt(c) #开方，得到的是斜边长  
 print("斜边长为:",c) #显示，一项是字符串，一项是 c 表示的斜边长

2. 代码如下：

# 用户输入

x = input('输入 x 值: ')

y = input('输入 y 值: ')

# 不使用临时变量

x,y = y,x

print('交换后 x 的值为: {}'.format(x))

print('交换后 y 的值为: {}'.format(y))

### Python常用语句

#### 填空题

1. break
2. elif
3. continue
4. True
5. pass

#### 判断题

1. 错
2. 对
3. 对
4. 对
5. 对

#### 选择题

1. C
2. C
3. C
4. C
5. D

#### 简答题

1. Pass语句Python中的pass是空语句，它的出现是为了保持程序结构的完整性。pass 不做任何事情，一般用做占位语句。
2. break语句用于结束整个循环；continue的作用是用来结束本次循环，紧接着执行下一次的循环。

#### 编程题

1. 代码如下：

for i in range(1,11):  
 print(i)

1. 代码如下：

a=int(input("请输入一个数："))  
 if a>0:  
 print("a是一个正数")  
 elif a<0:  
 print("a是一个负数")  
 else:  
 print("a等于0")

1. 代码如下：

i=1  
 while i<10:  
 j=1  
 while j<=i:  
 print("%d\*%d=%-2d "%(i,j,i\*j),end='')  
 j+=1  
 print("\n")  
 i+=1

### 字符串

#### 单选题

1. A
2. D
3. B
4. C
5. C
6. B
7. A

#### 判断题

1. 对
2. 对
3. 错
4. 对
5. 错
6. 错
7. 错

#### 填空题

1. 文本
2. 转义
3. input
4. 切片
5. 开

#### 程序分析题

1. 答：不能编译通过。因为num1和num2属于字符串类型，不能执行取余操作。
2. 结果为：J
3. 答：由于没有在字符串中找到子串，index方法默认会抛出ValueError异常。
4. 结果为：Hello World（默认删除两头的空格）
5. 结果为：HelloPython

#### 编程题

1. 答案：

myStr = input("请输入任意字符串:")

num = 0

for s in myStr:

if s.isdigit():

num += 1

print(num)

1. 答案：

week = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

while True:

myStr = input("请输入任意一个字母:")

upMyStr = myStr.upper()

for weekS in week:

if upMyStr == weekS[0]:

if weekS.startswith("M"):

print("星期一")

elif weekS.startswith("W"):

print("星期三")

elif weekS.startswith("F"):

print("星期五")

elif weekS.startswith("T") or weekS.startswith("S"):

secondStr = input("请再输入任意一个字母:")

newStr = upMyStr+secondStr

print(newStr)

for weekStr in week:

if weekStr.find(newStr,0,2) != -1:

if newStr == "Tu":

print("星期二")

elif newStr == "Th":

print("星期四")

elif newStr == "Sa":

print("星期六")

elif newStr == "Su":

print("星期日")

break

break

1. 答案：

str = "Hello,welcome to itheima!!"

print(str.count("o"))

### 列表、元组和字典

#### 单选题

1. D
2. B
3. C
4. D
5. D
6. C
7. A
8. B
9. C

#### 判断题

1. 对
2. 对
3. 对
4. 对
5. 错
6. 错
7. 错

#### 填空题

1. 字典
2. 字典，列表
3. not in
4. sort
5. 小括号

#### 简答题

1．答案：

（1）外形：列表是中括号括起来的数据；元组是圆括号括起来的数据；字典是花括号括起来的数据

（2）存储结构：列表可以存储多个不同类型的数据，以逗号分隔；元组同样能存储多个不同类型的数据，以逗号分隔；字典能存储多个键值对，以逗号分隔，键是唯一的，值是任何类型的。

（3）访问方式：列表可以通过下标索引访问元素，索引从0开始；元组可以通过下标索引访问值，索引从0开始；字典通过键来访问值。

（4）是否可变类型：列表是可变类型，列表是不可变类型，字典是可变类型。

#### 程序分析题

1. 答案：程序运行错误，元组不能使用下标增加元素。
2. 结果为：3
3. 结果为：1，26

#### 编程题

1. 答案：

def message():  
 myStr = input("请输入密码，只能为字母和数字：")  
 if myStr.isalnum():  
 print("符合要求")  
 else:  
 print("不符合要求")

1. 答案：

arr = []

evenStr = ""

oddStr = ""

message = input("请输入任意字符串：")

for string in message:

arr.append(string)

for eStr in (arr[::2]):

evenStr = evenStr+eStr

for oStr in (arr[1::2]):

oddStr = oddStr+oStr

print(evenStr+oddStr)

1. 答案：

arr = []

def count():

myStr = input("请输入只包含字母的字符串：")

if myStr.isalpha():

newStr = myStr.lower()

for string in newStr:

arr.append(string)

a = {}

for i in arr:

if arr.count(i) >= 1:

a[i] = arr.count(i)

print(a)

else:

print("输入的内容有误")

count()

1. 答案：

arr = []

result = ""

myStr = input("请输入字符串：")

for string in myStr:

arr.append(string)

last = arr[-1]

arr.remove(last)

arr.insert(0,last)

for new in arr:

result = result + new

print(result)

1. 答案：

arr = []

length = int(input("请要输入数字的总个数（必须为奇数）:"))

i = 0

while i < length:

num = int(input("输入第%d个数字:"%(i+1)))

arr.append(num)

i+=1

arr.sort()

index = int(length/2)

print(arr[index])

1. 答案：

第一种：

arr = [1,2,3,4,5]

arr.reverse()

print(arr)

第二种：

arr = [1,2,3,4,5]

arr.sort(reverse=True)

print(arr)

1. 答案：

arr = []

length = int(input("请输入数字的总个数:"))

i = 0

while i < length:

num = int(input("输入第%d个数字:"%(i+1)))

arr.append(num)

i+=1

print("排序前：%s"%arr)

# 对列表排序

for i in range(length):

flag = 0

for j in range(1,length):

if arr[j-1] > arr[j]:

arr[j-1],arr[j] = arr[j],arr[j-1]

flag = 1

if not flag:

break

print("排序后：%s"%arr)

1. 答案：

arr = []

length = int(input("请输入列表的总个数:"))

i = 0

while i < length:

element = input("输入第%d个元素:"%(i+1))

arr.append(element)

i+=1

# 列表转为集合

newList = set(arr)

print(newList)

1. 答案：

def add(a, b):

# 下面两行保证输入的a和b是维数相同的矩阵，根据实际情况不要也可以

assert (len(a) == len(b))

assert (all([len(a[i]) == len(b[i]) == len(a[0]) for i in range(len(a))]))

i, j = len(a), len(a[0])

c = [[0] \* j] \* i

for m in range(i):

for n in range(j):

c[m][n] = a[m][n] + b[m][n]

return c

a = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

b = [[9, 8, 7], [6, 5, 4], [3, 2, 1]]

c = add(a, b)

print(c)

1. 答案：

dic = {}

i=0

while i<3:

number = input("输入学生学号：")

name = input("输入学生姓名：")

dic.\_\_setitem\_\_(number,name)

i+=1

print("排序前：%s"%dic)

def dict2list(dic:dict):

''' 将字典转化为列表 '''

keys = dic.keys()

vals = dic.values()

lst = [(key, val) for key, val in zip(keys, vals)]

return lst

new = sorted(dict2list(dic), key=lambda x:x[0], reverse=False)

print("排序后：%s"%new)

1. 答案：

def delMale(dict):

keys = []

values = []

for (key,value) in dict.items():

keys.append(key)

values.append(value)

for value in values:

if value == 0:

index = values.index(value)

arrKey = keys[index]

del dict[arrKey]

values[index] = 3

keys[index] = "占位"

dic = {"小明":0, "小红":1,"小兰":1,"小白":0}

print("删除前：%s"%dic)

delMale(dic)

print("删除后：%s"%dic)

1. 答案：

arr = [12,3,37,7,91,67,27,45,6]

def delPrime(arr):

for element in arr:

# 质数大于 1

if element > 1:

# 查看因子

for i in range(2, element):

if (element % i) == 0:

break

else:

arr.remove(element)

delPrime(arr)

print(arr)

### 函数

#### 单选题

1. C
2. D
3. C
4. A
5. C
6. D

#### 判断题

1. 错
2. 对
3. 对
4. 错
5. 对
6. 对
7. 错
8. 对

#### 填空题

1. 逗号
2. return
3. return
4. 不定长
5. 嵌套
6. 局部
7. 整个程序
8. global

#### 简答题

1. 答案：

局部变量：

（1）函数内部定义的变量；

（2）作用域是函数的内部。

全局变量：

（1）函数外部定义的变量；

（2）作用域是整个程序。

1. 答案：

（1）以def关键字开头，后面接函数标识符名称和圆括号；

（2）给函数起名字的时候，规则跟变量的名字是一样的；

（3）任何传入参数和自变量必须放在圆括号中间；

（4）函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明；

（5）函数内容以冒号起始，并且缩进；

（6）return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的return相当于返回 None。

#### 程序分析题

1．结果为100

2．结果为：200

3．结果为：200，100

#### 编程题

1. 答案：

def getMax():

num1 = int(input("请输入第1个整数："))

num2 = int(input("请输入第2个整数："))

num3 = int(input("请输入第3个整数："))

temp = 0 # 临时变量，存放中间产生的第2大数值

if num1 > num2:

temp = num1

else:

temp = num2

if temp>num3:

return "其中最大值为："+ str(temp)

else:

return "其中最大值为："+ str(num3)

maxValue = getMax()

print(maxValue)

1. 答案：

def sum(n):

res = 0

while n >= 1:

if n%2 == 0:

res -= 1.0/(n\*(n+1))

else:

res += 1.0/(n\*(n+1))

n-=1

return res

num = int(input("请输入一个整数："))

print(sum(num))

1. 答案：

def is\_palindrome(n):

n=str(n)

m=n[::-1]

return n==m

result = is\_palindrome(1247321)

print(result)

1. 答案

def sjx(a, b, c):

if (a + b > c and a + c > b and b + c > a):

return "能构成三角形"

else:

return "不能构成三角形"

result1 = sjx(3, 4, 5)

print(result1)

result2 = sjx(1, 1, 2)

print(result2)

1. 答案：

# 定义函数

def lcm(x, y):

# 获取最大的数

if x > y:

greater = x

else:

greater = y

while(True):

if((greater % x == 0) and (greater % y == 0)):

lcm = greater

break

greater += 1

return lcm

# 获取用户输入

num1 = int(input("输入第一个数字: "))

num2 = int(input("输入第二个数字: "))

print( num1,"和", num2,"的最小公倍数为", lcm(num1, num2))

### 高级函数

#### 单选题

1. B
2. C
3. D
4. A

#### 判断题

1. 错
2. 错
3. 错
4. 对
5. 对
6. 错
7. 错
8. 对
9. 对
10. 错
11. 对

#### 填空题

1. 闭包
2. 函数
3. 被修饰的函数
4. @
5. 内嵌
6. map
7. None
8. filter
9. 布尔
10. 两

#### 简答题

1. 答案：

（1）存在于嵌套关系的函数中；

（2）嵌套的内部函数引用了外部函数的变量；

（3）嵌套的外部函数会将内部函数名作为返回值返回。

1. 答案：

（1）引入日志；

（2）函数执行时间统计；

（3）执行函数前预备处理；

（4）执行函数后清理功能；

（5）权限校验；

（6）缓存。

1. 答案：

map函数会根据提供的函数对指定的序列做映射。

filter函数会对指定序列执行过滤操作。

reduce函数会对参数序列中的元素进行累积。

#### 程序分析题

1．能，结果为：

6

7

8

2．能，结果为：

6

6

6

#### 编程题

答案：

result = filter(lambda x:(x+1)%2, [1, 2, 3, 4, 5])

print(result)

### 文件

#### 单选题

1. C
2. A
3. C
4. B
5. D
6. D
7. C

#### 判断题

1. 对
2. 对
3. 对
4. 对
5. 错

#### 填空题

1. close()
2. offset
3. 列表
4. 文件夹
5. tell

#### 简答题

1. 答案：

文本文件存储的是常规字符串，由若干文本行组成，通常每行以换行符“\n”结尾。二进制文件把对象内容以字节串进行存储，无法用记事本或其他普通字处理软件直接进行编辑，无法被人类直接阅读和理解，需要使用专门的软件进行解码后读取、显示、修改或执行。

1. 答案：
2. 使用read(size)方法可以指定读取的字节数，或者读取整个文件；
3. 使用readlines方法可以把整个文件的内容进行一次性读取；
4. 使用readline方法一行一行读数据。

#### 程序题

1. 答案：

new = []

f = open('itheima.txt', 'r')

contents = f.readlines()

for ele in contents:

if ele.startswith('#') == False:

new.append(ele)

print(new)

f.close()

1. 答案：

# 密码薄

passwordBook = {}

def add(password, value):

if password not in passwordBook.keys():

passwordBook[password] = value

save()

else:

print("该密码已存在")

def delete(password):

if password in passwordBook.keys():

del passwordBook[password]

print(passwordBook)

save()

else:

print("该密码不存在")

def modify(password, newValue):

if password in passwordBook.keys():

passwordBook[password] = newValue

print(passwordBook)

save()

else:

print("该密码不存在")

def find(value):

if value in passwordBook.values():

print("该网址存在")

else:

print("该网址不存在")

def save():

f = open("password.txt","w")

f.write(str(passwordBook))

f.close()

def printInfo():

print("密码簿的功能如下：")

print("1.添加")

print("2.删除")

print("3.修改")

print("4.查找")

i = 0

while i<5:

printInfo()

num = int(input("请输入要选择的功能："))

if num==1:

web = input("请输入存入的网址：")

password1 = input("请输入密码：")

add(password1, web)

print(passwordBook)

elif num==2:

password2 = input("请输入密码：")

delete(password2)

elif num==3:

password3 = input("请输入密码：")

value3 = input("请输入新的网址：")

modify(password3,value3)

elif num==4:

value4 = input("请输入要查找的网址：")

find(value4)

i+=1

1. 答案：

f = open("itheima.txt","r")

content = list(eval(f.read()))

content.sort()

print(content)

1. 答案：

f = open("itheima.txt","r")

content = f.read()

newStr = ""

for string in content:

temp = ord(string)

if temp in range(65,91):

if temp == 90:

char1 = chr(temp-25)

newStr += char1

else:

char2 = chr(temp+1)

newStr += char2

elif temp in range(97,123):

if temp == 122:

char3 = chr(temp-25)

newStr += char3

else:

char4 = chr(temp + 1)

newStr += char4

else:

newStr = newStr+string

f.close()

f2 = open("itheima-加密后.txt","w")

f2.write(newStr)

f2.close()

1. 答案：

 f = open(r'd:\1.txt','r')

s = f.readlines()

f.close()

r = [i.swapcase() for i in s]

 f = open(r'd:\2.txt','w')

f.writelines(r)

f.close()

1. 答案：

import pickle

d = {'张三':98,'李四':90,'王五':100}

print(d)

f = open('score.dat','wb')

pickle.dump(1,f)

pickle.dump(d,f)

f.close

 f = open('score.dat','rb')

pickle.load(f)

d = pickle.load(f)

f.close()

print(d)

### 异常

#### 单选题

1. B
2. A
3. C
4. D
5. A
6. C
7. D

#### 判断题

1. 对
2. 错
3. 错
4. 对
5. 对
6. 对

#### 填空题

1．Exception

2．索引

3．finally

4．assert

5．except

#### 简答题

1．答案：

在Python中，程序在执行的过程中产生的错误称为异常，比如列表索引越界、打开不存在的文件等。

2．答案：

try/except/else/finally语句

#### 编程题

1．答案：

class HeightException(Exception):

pass

try:

height = int(input("请输入您的身高："))

if height<30 or height>250:

raise HeightException

weight = int(input("请输入您的体重："))

weightS = height-100

if weight > weightS and weight-weightS < 0.05\*weightS:

print("体重达标")

elif weight < weightS and weightS-weight < 0.05\*weightS:

print("体重达标")

else:

print("体重不达标")

except HeightException:

print("您输入的身高有误")

2．答案：

try:

score = int(input("请输入学生的成绩："))

if score>=90 and score<=100:

print("A:优秀")

elif score>=80 and score<90:

print("B:良好")

elif score>=60 and score<80:

print("C:合格")

else:

assert score>60,"D:不及格"

except Exception as result:

print("低于60分：\n",result)

### Python模块

#### 单选题

1. C
2. D
3. A

#### 判断题

1. 错
2. 对
3. 对
4. 对
5. 错

#### 填空题

1. random. randint
2. 文件
3. \_\_name\_\_
4. 包
5. 安装
6. path

#### 简答题

1. 答案：

每个Python脚本在运行时都有一个“\_\_name\_\_”属性。如果脚本作为模块被导入，则其“\_\_name\_\_”属性的值被自动设置为模块名；如果脚本独立运行，则其“\_\_name\_\_”属性值被自动设置为“\_\_name\_\_”属性。利用“\_\_name\_\_”属性即可控制Python程序的运行方式。

1. 答案：

（1）搜索当前目录，如果不在当前目录，Python则搜索在shell变量PYTHONPATH下的每个目录。

（2）如果都找不到。Python会继续查看默认路径。

1. 答案：

在Python中有一个概念叫做模块（module），这个和C语言中的头文件以及Java中的包很类似，比如在Python中要调用sqrt函数，必须用import关键字引入math这个模块。

1. 答案：

在Python中用关键字import来引入某个模块：

1. 导入模块，使用“import 模块”引入；
2. 导入模块中的某个函数，使用“from 模块名 import 函数名”引入；
3. 导入模块的全部内容，使用“from 模块 import \*”。

#### 编程题

答案：

互换值.py

def changeNum1AndNum2(num1, num2):

temp = num1

num1 = num2

num2 = temp

return (num1, num2)

### 面向对象编程（上）

#### 单选题

1. B
2. D
3. C
4. C
5. A
6. A
7. B

#### 判断题

1. 错
2. 错
3. 错
4. 对
5. 错
6. 对

#### 填空题

1. class
2. 对象
3. self
4. \_\_init\_\_
5. 参数

#### 简答题

1. 不用实例化对象就能够在本类中访问自身的属性或方法。
2. 类名，属性，方法
3. 分别用于初始化对象的属性和释放类所占用的资源。

#### 程序分析题

1. 能，结果为：我的名字是小明
2. 能，结果为：

--del--

--end--

#### 编程题

1. 答案：

class Circle:

def \_\_init\_\_(self,tup, radius, color):

self.center = tup

self.radius = radius

self.color = color

def perimeter(self):

return 3.14 \* 2 \* self.radius

def area(self):

return 3.14 \* self.radius \* self.radius

circle = Circle((0,0),5,"蓝色")

print(circle.perimeter())

print(circle.area())

1. 答案：

class Curriculum:  
 def \_\_init\_\_(self):  
 self.number = 1001  
 self.name = "语文"  
 self.teacher = "小明"  
 self.\_\_address = "2号教学楼3层305室"  
 def \_\_str\_\_(self):  
 return """  
 课程编号：%d  
 课程名称：%s  
 任课教师：%s  
 上课地点：%s  
 """%(self.number,self.name,self.teacher,self.\_\_address)  
curri = Curriculum()  
print(curri)

### 面向对象编程（下）

#### 单选题

1. D
2. A
3. C
4. C
5. B
6. C
7. A

#### 判断题

1. 对
2. 错
3. 对
4. 错
5. 对
6. 错
7. 对
8. 对

#### 填空题

1. 下划线
2. 父类
3. 私有
4. 多
5. 重写
6. super()
7. 类
8. @classmethod

#### 简答题

1. 答案：

（1）把属性定义为私有属性，即在属性名的前面加上两个下划线；

（2）添加用于设置或者获取属性值的两个方法供外界调用。

1. 答案：

类的继承是指在一个现有类的基础上构建一个新的类，构建出来的新类被称作子类，现有类被称作父类，子类会自动拥有父类的属性和方法。

1. 答案：

当在一个类的内部定义了私有方法或者私有属性的时候，Python在运行的过程中，把属性或者方法的名字进行了修改，即在属性或者方法名称的前面加上“\_类名”，导致原有的方法无法访问到。

1. 答案：

在Python中，多态是指在不考虑对象类型的情况下使用对象。

1. 答案：

类方法需要使用@classmethod进行标识，该方法可以访问类属性，无法访问实例属性，可以通过类实例和类进行调用。

静态方法使用@staticmethod进行标识，该方法无法访问实例属性和类属性，起到类似于函数的作用，使用类或者类实例进行调用。

实例方法直接在类中使用def进行定义，可以访问其实例属性和类属性，使用类实例进行调用。

如果要修改实例属性的值，就直接使用实例方法；如果要修改类属性的值，就直接使用类方法；如果是辅助功能，比如打印菜单，这时可以考虑使用静态方法，可以在不创建对象的前提下使用。

1. 答案：

在Python中，以下划线开头的变量名有着特殊的含义，尤其是在类的定义中。用下划线作为变量前缀和后缀来表示类的特殊成员：

（1）\_xx：这样的对象叫做保护变量，不能用from module import \*导入，只有类对象和子类对象能够访问这些变量。

（2）\_\_xx\_\_：系统定义的特殊成员名字。

（3）\_\_xx：类中的私有成员，只有类对象自己能访问，子类对象也不能访问到这个成员，但在对象外部可以通过“对象名.\_类名\_\_ xx”这样特殊的方式来访问。Python中没有纯粹的C++意义上的私有成员。

#### 编程题

1. 答案：

class Student(object):

# 构造方法

def \_\_init\_\_(self, name, age, scores):

self.\_\_name = name # 姓名

self.\_\_age = age # 年龄

self.\_\_scores = scores # 分数

def get\_name(self):

return self.\_\_name

def get\_age(self):

return self.\_\_age

def get\_course(self):

return max(self.\_\_scores)

stu = Student('小丸子', 18, [89, 90, 91])

print("姓名：%s"%(stu.get\_name()))

print("年龄：%s"%(stu.get\_age()))

print("最高分：%s"%(stu.get\_course()))

1. 答案：

class Animal(object):

def \_\_init\_\_(self, color):

self.color = color #颜色

def call(self):

print("动物叫")

class Fish(Animal):

def \_\_init\_\_(self, color):

super().\_\_init\_\_(color)

self.tail = True

def call(self):

print("-%s的鱼在吐泡泡-"%self.color)

fish = Fish("蓝色")

fish.call()