Python理论题

1. Python中的input()函数以字符串的形式返回用户输入的数据。（1分）

A. 对  
B. 错

**参考答案：**

A

1. Python代码的注释只有一种方式，那就是使用#符号。（1分）

A. 对  
B. 错

**参考答案：**

B

1. Python是一种用途广泛、解释型、面向对象的程序设计语言。（1分）

A. 对  
B. 错

**参考答案：**

A

1. eval()、int()以及float()函数都可用于将字符串转换为数值。（1分）

A. 对  
B. 错

**参考答案：**

A

1. Python程序是区分大小写的。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 算法是解决某个问题所需要的方法和步骤。

A. 对  
B. 错

**参考答案：**

A

1. Python使用缩进来体现代码之间的逻辑关系。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 不可以在同一台计算机上安装多个Python版本。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 程序设计语言包括机器语言、汇编语言和高级语言。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 解释程序对高级语言编写的程序是一边翻译，一边执行的。下次执行同样的程序时，还必须重新翻译。（1分）

A. 对  
B. 错

**参考答案：**

A

1. Python解释器的提示符是\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

>>>

1. 程序设计语言包括机器语言、汇编语言和\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

高级语言

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(5 // 4)

【来源】  
《Python程序设计基础》第1章思考与练习。**参考答案：**

1

1. 基于冯 · 诺依曼思想而设计的计算机硬件由\_\_\_\_\_\_\_\_、运算器、存储器、输入设备和输出设备5个基本组成部分构成。

**参考答案：**

控制器

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print((15 / 5) \* 2)

**参考答案：**

6.0

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print("hello", 5)

【来源】  
《Python程序设计基础》第1章思考与练习。**参考答案：**

hello 5

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(format(57.467657,".2f"))

【来源】  
《Python程序设计基础》第1章思考与练习。**参考答案：**

57.47

1. Python语言源程序文件的扩展名通常为\_\_\_\_\_\_\_\_。  
   【来源】  
   《Python程序设计基础》第1章思考与练习。

**参考答案：**

.py、py

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print('{},{}'.format('carmen',20))

**参考答案：**

carmen,20

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print('{0},{1}'.format('carmen',20))

**参考答案：**

carmen,20

1. Python 3.4及后续版本中，默认的安装和管理Python包（库）的工具是\_\_\_\_\_\_\_\_。  
   【来源】  
   《Python程序设计基础》第1章思考与练习。（2分）

**参考答案：**

pip

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print("{:.2f}".format(96.3213))

（2分）

**参考答案：**

96.32

1. Python内置的集成开发环境是\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

IDLE

1. 一个字节等于\_\_\_\_\_\_\_\_位（注：填写阿拉伯数字）。（2分）

**参考答案：**

8

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print("%.2f" % (57.467657))

【来源】  
《Python程序设计基础》第1章思考与练习。（2分）

**参考答案：**

57.47

1. 算法是程序设计的基础和精髓。一个有效的算法应该具有如下特点，其中错误的是（    ）。

（2分）

A. 零个或多个输入及一个或多个输出  
B. 无穷性  
C. 可行性  
D. 确定性

**参考答案：**

B

1. 下列Python程序，正确的是（    ）。

（2分）

A.

print("Programming is fun")

print("Python is fun")

B.

print("Programming is fun")

print("Python is fun")

C.

print("Programming is fun)

print("Python is fun")

D.

print("Programming is fun")

print("Python is fun")

**参考答案：**

B

1. Python的段落注释的语法格式是（    ）。

（2分）

A. // comments //  
B. /\* comments \*/  
C. """comments """  
D. /# comments #/

**参考答案：**

C

1. Python语言被称为高级程序设计语言，（   ）。

A. 因为它比低级语言功能强大  
B. 因为它是编译型语言  
C. 因为它是解释型语言  
D. 因为它比低级语言更贴近人类的思维

**参考答案：**

D

1. 下列Python程序，正确的是（    ）。

（2分）

A.

print("Programming is fun")

print("Python")

print("Computer Science")

B.

print("Programming is fun")

print("Python")

print("Computer Science")

C.

print("Programming is fun")

print("Python")

print("Computer Science")

D.

print("Programming is fun")

print("Python")

print("Computer Science")

**参考答案：**

D

1. 下列Python程序，正确的是（    ）。

（2分）

A.

print("Hello, I'm Tom.")

B.

print('Hello, ' print 'World!')

C.

print('Hello, this's Tom.')

D.

Print('Hello, World!')

**参考答案：**

A

1. 编写Python程序，输出“Hello World”，要求将这两个单词分两行输出，以下错误的是（    ）。

（2分）

A.

print("Hello\nWorld")

B.

print('Hello\nWorld')

C.

print('Hello

World')

D.

print('Hello')

print('World')

**参考答案：**

C

1. Python的行注释以（    ）开头。

A. //  
B. /\*  
C. #  
D. $$

**参考答案：**

C

1. 下列（   ）是对算法的正确描述。

（2分）

A. 解决一个问题只有一种算法  
B. 对于所有问题都可以找到最好的算法  
C. 算法所包含的语句数量越少，算法越先进  
D. 解决一个问题可以有多种算法

**参考答案：**

D

1. 关于Python程序的执行过程，正确的是（    ）。

（2分）

A. 由解释器一条语句一条语句地执行  
B. 由编译器将源程序转化为机器语言，然后执行  
C. 可以同时执行多条语句  
D. 执行过的语句将不会再被执行

**参考答案：**

A

1. Python不允许使用关键字（保留字）作为变量名，允许使用内置函数名作为变量名，但这会改变函数名的含义。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 内置函数id(object)返回对象object在其生命周期内的内存地址。（1分）

A. 对  
B. 错

**参考答案：**

A

1. Python其实不支持命名常量，只是约定在程序运行过程中值不会改变的变量为命名常量。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在Python中，"+"运算符可以用来连接字符串并生成新的字符串。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在Python中，同一个变量在当前作用域内可以先后赋予不同类型的值。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在Python中，列表只能包含相同类型的数据。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在Python中，字符串、列表、元组和字典都属于序列类型。（1分）

A. 对  
B. 错

**参考答案：**

B

1. Python的标识符必须以字母或下画线开头，不能以数字开头。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在Python中，可以不加声明就使用变量，但使用前需要初始化变量。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 执行语句print(len({}))，其输出结果是1。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(int("08"))

【来源】  
《Python程序设计基础》第2章思考与练习。（2分）

**参考答案：**

8

1. 使用math模块前，需要使用\_\_\_\_\_\_\_\_关键字导入该模块。（2分）

**参考答案：**

import

1. print(hex(16))的输出结果是（进制用小写字母表示）\_\_\_\_\_\_\_\_。**参考答案：**

0x10

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. s = "Hello"

print(len(s))

**参考答案：**

5

1. 下面程序的输出结果是。
2. x = 'a'
3. y = 'A'
4. print(chr(ord(y) + ord(x) - ord(y)))

【来源】  
《Python程序设计基础》第2章思考与练习。**参考答案：**

a

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_ 。

print(format(0.0033911, "6.3%"))

【来源】  
《Python程序设计基础》第2章思考与练习。（2分）

**参考答案：**

0.339%

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。

print(1, 2, 3, sep=':')

**参考答案：**

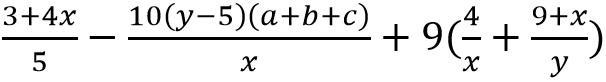
1:2:3

1. 下列表达式的值是\_\_\_\_\_\_\_\_。

3\*\*2\*\*3

**参考答案：**

6561

1. 下列数学式子转换为算术表达式是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。  
     
   【来源】  
   《Python程序设计基础》第2章思考与练习。（2分）

**参考答案：**

(3+4\*x)/5-10\*(y-5)\*(a+b+c)/x+9\*(4/x+(9+x)/y)

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. value = 8.6

print(int(value))

【来源】  
《Python程序设计基础》第2章思考与练习。

**参考答案：**

8

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. value = 8.6

print(round(value))

【来源】  
《Python程序设计基础》第2章思考与练习。（2分）

**参考答案：**

9

1. 下列表达式的值是\_\_\_\_\_\_\_\_。

1 + 2 + (3 + 4) \* ((5 + 6 % 7 \* 8) - 9) – 10

【来源】

《Python程序设计基础》第2章思考与练习。

（2分）

**参考答案：**

301

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(max(3, 5, 1, 7, 4))

**参考答案：**

7

1. 下面程序的输出结果是。
2. s = "Python" + str(3)

print(s.lower())

**参考答案：**

python3

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(float("08"))

【来源】  
《Python程序设计基础》第2章思考与练习。（2分）

**参考答案：**

8.0

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print([3, "dog", 9, "cat"][1])

**参考答案：**

dog

1. print(bin(10))的输出结果是（进制用小写字母表示）\_\_\_\_\_\_\_\_。

**参考答案：**

0b1010

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. d = {1:"a", 2:"b"}

print(d[1])

**参考答案：**

a

1. print(int('101', 2))的输出结果是\_\_\_\_\_\_\_\_。  
   （2分）

**参考答案：**

5

1. print("hello" 'world')的输出结果是\_\_\_\_\_\_\_\_。**参考答案：**

helloworld

1. 45 // 4的计算结果是（   ）。

A. 10  
B. 11  
C. 11.25  
D. 12

**参考答案：**

B

1. type((1,))的返回值是（    ）。（2分）

A. <class 'tuple'>  
B. <class 'list'>  
C. <class 'dict'>  
D. <class 'int'>

**参考答案：**

A

1. 下列（   ）是正确的。
2. （1） x -= x + 4
3. （2） x = x + 4 - x

（3） x = x - (x + 4)

A. （1）和（2)是相同的  
B. （1）和（3）是相同的  
C. （2）和（3）是相同的  
D. （1）、（2）和（3）是相同的

**参考答案：**

B

1. 表达式"Good " + 1 + 2 + 3的计算结果是（    ）。

（2分）

A. Good123  
B. Good6  
C. Good 123  
D. 非法的表达式

**参考答案：**

D

1. 对象5.6的类型是（    ）。

（2分）

A. int  
B. float  
C. str  
D. bool

**参考答案：**

B

1. 下列（   ）语句的输出结果是python\exam1\test.txt。

（2分）

A.

print("python\exam1\test.txt")

B.

print("python\\exam1\\test.txt")

C.

print("python\"exam1\"test.txt")

D.

print("python"\exam1"\test.txt")

**参考答案：**

B

1. 下列（    ）语句是错误的。（2分）

A. x = y = z = 1  
B. x = (y = z + 1)  
C. x, y = y, x  
D. x += y

**参考答案：**

B

1. 执行下列程序，输出的结果是（    ）。
2. x = 1
3. x = x + 2.5

print(x)

A. 1  
B. 2  
C. 3  
D. 3.5

**参考答案：**

D

1. 下列（   ）函数的返回值是4。

A. int(3.4)  
B. int(3.9)  
C. round(3.4)  
D. round(3.9)

**参考答案：**

D

1. 下列程序的输出结果是（   ）（注意：?表示空格）。
2. print(format("Welcome", "10s"), end = '#')
3. print(format(111,"4d"), end = '#')

print(format(924.656, "3.2f"))

A.

???Welcome#?111#924.66

B.

Welcome#111#924.66

C.

Welcome#111#.66

D.

Welcome???#?111#924.66

**参考答案：**

D

1. 执行下列程序，输出的结果是（    ）。
2. x = 1
3. y = x = x + 1

print(y)

（2分）

A. 0  
B. 1  
C. 2  
D. 3

**参考答案：**

C

1. 使用（        ）输出可以换行的字符串。

A. 转义符\\  
B. 转义符\n  
C. 空格  
D. 转义符\r

**参考答案：**

B

1. math.degrees(math.pi / 2)的返回值是（    ）。

A. 0.0  
B. 90.0  
C. 45.0  
D. 30.0

**参考答案：**

B

1. 假设x的值为345.3546，format(x, "10.3f")结果是（    ）（注意：b表示空格）。

A.

bb345.355

B.

bbb345.355

C.

bbbb345.355

D.

bbb345.354

E.

bbbb345.354

**参考答案：**

B

1. 45 / 4的计算结果是（   ）。

A. 10  
B. 11  
C. 11.25  
D. 12

**参考答案：**

C

1. 假设x是一个字符变量，其值为'b'。语句print(chr(ord(x) + 1))的输出结果是（    ）。

（2分）

A. a  
B. b  
C. c  
D. d

**参考答案：**

C

1. 使用下列（   ）函数读取一个字符串。

A.

input("输入一个字符串")

B.

eval(input("输入一个字符串"))

C.

enter("输入一个字符串")

D.

eval(enter("输入一个字符串"))

**参考答案：**

A

1. 下列表达式的计算结果是（   ）。

eval("1 + 3 \* 2")

A.

"1 + 3 \* 2"

B.

7

C.

8

D.

"1 + 6"

**参考答案：**

B

1. 执行下列程序，输出的结果是（    ）。
2. x, y = 1, 2
3. x, y = y, x

print(x, y)

（2分）

A. 1 1  
B. 2 2  
C. 1 2  
D. 2 1

**参考答案：**

D

1. 下列表达式计算结果为1的是（    ）。

A. 2 % 1  
B. 15 % 4  
C. 25 % 5  
D. 37 % 6

**参考答案：**

D

1. 下列程序的输出结果是（    ）。
2. print("输入三个数: ")
3. number1 = eval(input())
4. number2 = eval(input())
5. number3 = eval(input())
6. # 求平均值
7. average = (number1+ number2 + number3) / 3
8. # 显示结果

print(average)

程序运行时，输入（↙表示回车）：

1 2 3↙

A. 1.0  
B. 2.0  
C. 3.0  
D. 4.0  
E. 输入数据时，程序发生运行时错误

**参考答案：**

E

1. 将x的值格式化为小数点后3位小数，使用（    ）。

（2分）

A. format(x, "5.3f")  
B. format("5.3f", x)  
C. format(x, "5.3%")  
D. format("3d", x)

**参考答案：**

A

1. 下列程序的输出结果是（   ）。
2. print("A", end = ' ')
3. print("B", end = ' ')
4. print("C", end = ' ')

print("D", end = ' ')

（2分）

A.

ABCD

B.

A, B, C, D

C.

A B C D

D.

A

B

C

D

**参考答案：**

C

1. 如下语句执行后，x的值是（   ）。
2. x = 2
3. y = 1

x \*= y + 1

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

**参考答案：**

D

1. 假设number的值为4，下列语句错误的是（    ）。

（2分）

A.

print(format(number, "2d"), format(number \*\* 1.5, "4.2d"))

B.

print(format(number, "2d"), format(number \*\* 1.5,"4.2f"))

C.

print(format(number, "2f"), format(number \*\* 1.5,"4.2f"))

D.

print(format(number, "2.1f"), format(number \*\* 1.5,"4.2f"))

**参考答案：**

A

1. 下面（   ）是一个有效的标识符。

（2分）

A. $343  
B. 9X  
C. import  
D. max\_radius  
E. "red"

**参考答案：**

D

1. 2 \*\* 3.0的计算结果是（   ）。

（2分）

A. 9  
B. 8  
C. 9.0  
D. 8.0

**参考答案：**

D

1. 执行下列程序，输出的结果是（    ）。
2. y = x = 1
3. x += y + y \* 5

print(x)

（2分）

A. 1  
B. 5  
C. 6  
D. 7

**参考答案：**

D

1. 执行下列程序，输出的结果是（    ）。
2. x = 7.0
3. y = 5

print(x % y)

A. 2.0  
B. 1.0  
C. 2  
D. 1

**参考答案：**

A

1. 若字典d={1:"a", 2:"b"}，则能够访问d的第一个元素的语句是（    ）。

（2分）

A. d['0']  
B. d[1]  
C. d['1']  
D. d[0]

**参考答案：**

B

1. 假设s = "Welcome"，type(s)结果是（   ）。

A. <class 'int'>  
B. <class 'float'>  
C. <class 'str'>  
D. <class 'complex'>

**参考答案：**

C

1. 下列程序的输出结果是（    ）。
2. number1, number2, number3 = eval(input("输入三个数: "))
3. # 求平均值
4. average = (number1 + number2 + number3) / 3
5. # 显示结果

print(average)

程序运行时，输入（↙表示回车）：

1, 2, 3↙

A. 1.0  
B. 2.0  
C. 3.0  
D. 4.0

**参考答案：**

B

1. 下列程序的输出结果是（    ）。
2. print("输入三个数: ")
3. number1 = eval(input())
4. number2 = eval(input())
5. number3 = eval(input())
6. # 求平均值
7. average = (number1+ number2 + number3) / 3
8. # 显示结果

print(average)

程序运行时，输入（↙表示回车）：

1↙

2↙

3↙

（2分）

A. 1.0  
B. 2.0  
C. 3.0  
D. 4.0

**参考答案：**

B

1. math.radians(30) \* 6的返回值是（    ）。

（2分）

A. 0.0  
B. 1.3434343  
C. 3.141592653589793  
D. 5.565656

**参考答案：**

C

1. 下列程序的输出结果是（   ）（注意：?表示空格）。
2. print(format("Welcome",">10s"), end = '#')
3. print(format(111,"<4d"), end = '#')

print(format(924.656, ">10.2f"))

（2分）

A.

???Welcome#?111#924.66

B.

???Welcome#?111#????924.66

C.

???Welcome#111?#????924.66

D.

Welcome???#111?#????924.66

**参考答案：**

C

1. 假设s = "Welcome"，s.upper()结果是（   ）。

A. welcome  
B. WELCOME  
C. Welcome

**参考答案：**

B

1. print(chr(65))的运行结果是（   ）。

A. 65  
B. 6  
C. 5  
D. A

**参考答案：**

D

1. 下列关于Python字符串的说法，错误的是（   ）。

A. 字符即长度为1的字符串  
B. 字符串以\0标志字符串的结束  
C. 既可以用单引号，也可以用双引号创建字符串  
D. 在三引号字符串中可以包含换行回车等特殊字符

**参考答案：**

B

1. 如下语句执行后，x的值是（   ）。
2. x = 1

x \*= x + 1

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

**参考答案：**

B

1. 2 + 2 \*\* 3 / 2的计算结果是（    ）。

A. 4  
B. 6  
C. 4.0  
D. 6.0

**参考答案：**

D

1. 执行语句print(bool(0))，其输出结果是False。

A. 对  
B. 错

**我的答案：**

A

1. 条件表达式5 if 5>6 else (6 if 3>2 else 5)的值为6。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 执行语句print(int(True))，其输出结果是1。

A. 对  
B. 错

**我的答案：**

A

1. 表达式3 and 5的值为True。（1分）

A. 对  
B. 错

**我的答案：**

B

1. 表达式3 not in [1, 2, 3]的值为False。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 下列程序的输出结果是"Small number."。
2. x = 5
3. if x >= 10:
4. print("Big number!")
5. elif:

print("Small number.")

（1分）

A. 对  
B. 错

**我的答案：**

B

1. 表达式3<5>2的值为True。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 表达式3 or 5 的值为True。（1分）

A. 对  
B. 错

**我的答案：**

B

1. 下列程序的输出结果是苹果：
2. if True:
3. if False:
4. print("香蕉")
5. else:
6. print("苹果")
7. else:
8. if True:
9. print("红枣")
10. else:

print("玉米")

（1分）

A. 对  
B. 错

**我的答案：**

A

1. 交互方式下，已知a = 257和b = 257，那么表达式a is b的值为True。（1分）

A. 对  
B. 错

**我的答案：**

B

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a, b, c = 1, 2, 6
3. if a <= b or c < 0 or b < c:
4. s = b + c
5. else:
6. s = a + b + c

print(s)

**我的答案：**

8

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a, b = 10, 20
3. ok1, ok2 = 5, 0
4. if a < b:
5. if b != 15:
6. if not ok1:
7. x = 1
8. elif ok2:
9. x = 10
10. else:
11. x = -1

print(x)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**我的答案：**

-1

1. 要想得到[0,100]范围内的随机数，random.randint(0, n)里的n应该是\_\_\_\_\_\_\_\_。  
   （2分）

**我的答案：**

100

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(int(True))

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**我的答案：**

1

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

0 and 1 or not 2 < True

（2分）

**我的答案：**

True

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. if 6 < 5:
3. print('one')
4. elif 7 == 9:
5. print('two')

print("three")

（2分）

**我的答案：**

three

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(bool(8))

【来源】

《Python程序设计基础》第3章思考与练习。

（2分）

**我的答案：**

True

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. x = 100
3. if x > 10:
4. x = x + 10
5. if x > 20:
6. x = x + 50

print(x)

（2分）

**我的答案：**

160

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. m, n, x = 1, 0, 2
3. if not n:
4. x -= 1
5. if m:
6. x -= 2
7. if x:
8. x -= 3

print(x)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**我的答案：**

-4

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a = 3
3. b = 2
4. x = a if a > b else b

print(x)

【来源】  
《Python程序设计基础》第3章思考与练习。

（2分）

**我的答案：**

3

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

((2 >= 2) or (2 < 2)) and 2

（2分）

**我的答案：**

2

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(bool(0))

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**我的答案：**

False

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a, b, c, d = 1, 3, 5, 4
3. if a < b:
4. if c < d:
5. x = 1
6. else:
7. if a < c:
8. if b < d:
9. x = 2
10. else:
11. x = 3
12. else:
13. x = 6
14. else:
15. x = 7

print(x)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**我的答案：**

2

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

3 and 0 and 5

（2分）

**我的答案：**

0

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

[2] in [1,2,3]

（2分）

**我的答案：**

False

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。

print(int(False))

【来源】  
《Python程序设计基础》第3章思考与练习。

（2分）

**我的答案：**

0

1. 下面条件表达式的值为\_\_\_\_\_\_\_\_。

5 if 5>6 else (6 if 3>2 else 5)

（2分）

**我的答案：**

6

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a = 14 // 3
3. b = 4 > 5

print(a and b)

（2分）

**我的答案：**

False

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

10 + 5 // 3 - True + False

（2分）

**我的答案：**

10

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. x = 5
3. if 8 % 4:
4. x = x - 1
5. elif 3 < 4 / 2:
6. x = x - 2
7. elif "t":
8. x = x - 3
9. else:
10. x = x - 4

print(x)

**我的答案：**

2

1. 下列（         ）不可能出现在and操作的真值表中。

（2分）

A. True and False，结果为True  
B. False and False，结果为False  
C. True and True，结果为True  
D. False and True，结果为False

**我的答案：**

A

1. 以下程序的输出结果是（    ）。
2. ch = 'F'
3. if ch >= 'A' and ch <= 'Z':

print(ch)

（2分）

A. F  
B. f  
C. 无输出  
D. F f

**我的答案：**

A

1. 能实现下面功能的语句是（   ）。  
   接收用户输入的一个整数。如果输入的是偶数，则输出"True"，否则输出"False"。

（2分）

A.

print(not bool(input() % 2))

B.

print(int(input()) % 2 != 0)

C.

print(int(input()) % 2 == 1)

D.

print(not bool(int(input()) % 2))

**我的答案：**

D

1. 对于以下程序：
2. even = False
3. if even = True:

print("It is even!")

（    ）是正确的

。

（2分）

A. 在行1（even = False），程序有语法错误  
B. 在行2，程序有语法错误，if even = True:不是一个正确的条件，它应该替换成if even == True:或if even:  
C. 程序正常运行，但不显示任何信息  
D. 程序正常运行，显示It is even!

**我的答案：**

B

1. 若年龄（age）小于16岁，显示"Cannot get a driver's license"；若年龄（age）大于或等于16岁，显示"Can get a driver's license"，以下代码段最佳的是（    ）。

（1）

if age < 16:

print("Cannot get a driver's license")

if age >= 16:

print("Can get a driver's license")

（2）

if age < 16:

print("Cannot get a driver's license")

else:

print("Can get a driver's license")

（3）

if age < 16:

print("Cannot get a driver's license")

elif age >= 16:

print("Can get a driver's license")

（4）

if age < 16:

print("Cannot get a driver's license")

elif age == 16:

print("Can get a driver's license")

elif age > 16:

print("Can get a driver's license")

（2分）

A. （1）  
B. （2）  
C. （3）  
D. （4）

**我的答案：**

B

1. 对于以下代码段：

Code1:

if number % 2 == 0:

even = True

else:

even = False

Code2:

even = number % 2 == 0

（    ）是正确的。

（2分）

A. Code1有错误  
B. Code2有错误  
C. Code1和Code2都有错误  
D. Code1和Code2都是正确的，但Code2更好

**我的答案：**

D

1. 运算符+、\*、and、or的优先级（从高到低）是（   ）。

（2分）

A. and、or、\*、+  
B. \*、+、or、and  
C. \*、+、and、or  
D. or、and、\*、+

**我的答案：**

C

1. "小于或等于"比较运算符是（    ）。

（2分）

A. <  
B. <=  
C. =<  
D. <<  
E. !=

**我的答案：**

B

1. "等于"比较运算符是（    ）。

（2分）

A. <>  
B. !=  
C. ==  
D. =

**我的答案：**

C

1. 假设x的值为4、y的值为5，以下表达式的值为真（True）的是（    ）。

（2分）

A. x < 5 and y < 5  
B. x < 5 or y < 5  
C. x > 5 and y > 5  
D. x > 5 or y > 5

**我的答案：**

B

1. 以下程序的输出结果是（    ）。
2. x = 1
3. y = -1
4. z = 1
5. if x > 0:
6. if y > 0:
7. print('AAA')
8. elif z > 0:

print('BBB')

（2分）

A. 语法错误  
B. BBB  
C. 无输出  
D. AAA

**我的答案：**

C

1. random.randint(0, 1)的返回值是（    ）。

（2分）

A. 0  
B. 1  
C. 0或1  
D. 2

**我的答案：**

C

1. random.random()的返回值是（    ）。

（2分）

A. 一个浮点数i，0 < i < 1.0  
B. 一个浮点数i，0 <= i < 1.0  
C. 一个浮点数i，0 <= i <= 1.0  
D. 一个浮点数i，0 < i < 2.0

**我的答案：**

C

|  |  |
| --- | --- |
| 题目得分 | 0 |

1. 下列（         ）不可能出现在or操作的真值表中。

（2分）

A. False or True，结果为True  
B. True or True，结果为True  
C. False or False，结果为True  
D. True or False，结果为True

**我的答案：**

C

1. 以下程序的输出结果是（    ）。
2. isCorrect = False

print("Correct" if isCorrect else "Incorrect")

（2分）

A. Correct  
B. Incorrect  
C. 无输出  
D. Correct Incorrect

**我的答案：**

B

1. 假设x的值为4、y的值为5，以下表达式的值为真（True）的是（    ）。

（2分）

A. not (x == 4)  
B. x != 4  
C. x == 5  
D. x != 5

**我的答案：**

D

1. 若半径是正数，下面（   ）代码段输出一个圆的面积。

（2分）

A.

if radius != 0:

print(radius \*radius \* 3.14159)

B.

if radius >= 0:

print(radius \*radius \* 3.14159)

C.

if radius > 0:

print(radius \*radius \* 3.14159)

D.

if radius <= 0:

print(radius \*radius \* 3.14159)

**我的答案：**

C

1. 以下程序的输出结果是（    ）。
2. temperature = 50
3. if temperature>= 100:
4. print("too hot")
5. elif temperature<= 40:
6. print("too cold")
7. else:

print("just right")

（2分）

A. too hot  
B. too cold  
C. just right  
D. too hot too cold just right

**我的答案：**

C

1. 以下程序的输出结果是（    ）。
2. x = 0
3. if x < 4:
4. x = x + 1

print(x)

（2分）

A. 0  
B. 1  
C. 2  
D. 3

**我的答案：**

B

1. 若年龄（age）小于16岁，显示"Cannot get a driver's license"；若年龄（age）大于或等于16岁，显示"Can get a driver's license"，以下代码段正确的是（   ）。

（1）

if age < 16:

print("Cannot get a driver's license")

if age >= 16:

print("Can get a driver's license")

（2）

if age < 16:

print("Cannot get a driver's license")

else:

print("Can get a driver's license")

（3）

if age < 16:

print("Cannot get a driver's license")

elif age >= 16:

print("Can get a driver's license")

（4）

if age < 16:

print("Cannot get a driver's license")

elif age == 16:

print("Can get a driver's license")

elif age > 16:

print("Can get a driver's license")

（2分）

A. （1）和（2）  
B. （2）和（3）  
C. （1）、（2）和（3）  
D. （3）和（4）  
E. 以上都是

**我的答案：**

E

1. 给定|x-2|≥4，以下表达式正确的是（   ）。

（2分）

A. x - 2 >= 4 and x - 2 <= -4  
B. x - 2 >= 4 or x - 2 <= -4  
C. x - 2 >= 4 and x - 2 < -4  
D. x - 2 >=4 or x - 2 < -4

**我的答案：**

B

1. 假设income的值为4001，以下代码段的输出结果是（   ）。
2. if income > 3000:
3. print("Income is greater than 3000")
4. elif income > 4000:

print("Income is greater than 4000")

（2分）

A. 无输出  
B. 输出Income is greater than 3000  
C. 先输出Income is greater than 3000，接着输出Income is greater than 4000  
D. 输出Income is greater than 4000  
E. 先输出Income is greater than 4000，接着输出Income is greater than 3000

**我的答案：**

B

1. 以下运算符右结合的是（   ）。

（2分）

A. \*  
B. \*\*  
C. %  
D. and  
E. =

**我的答案：**

E

1. 以下程序的输出结果是（    ）。
2. x = 1
3. y = -1
4. z = 1
5. if x > 0:
6. if y > 0:
7. print("x > 0 and y >0")
8. elif z > 0:

print("x < 0 and z > 0")

（2分）

A. x > 0 and y > 0  
B. x < 0 and z > 0  
C. x < 0 and z <0  
D. 无输出

**我的答案：**

D

1. 给定|x-2|≤4，以下表达式正确的是（    ）。

（2分）

A. x - 2 <= 4 and x - 2 >= 4  
B. x - 2 <= 4 and x - 2 > -4  
C. x - 2 <= 4 and x - 2 >= -4  
D. x - 2 <=4 or x - 2 >= -4

**我的答案：**

C

1. 以下程序的输出结果是（    ）。
2. number = 30
3. if number % 2 == 0:
4. print(number, 'is even')
5. elif number % 3 == 0:

print(number, 'is multiple of 3')

（2分）

A.

30 is even

B.

程序出错

C.

30 is multiple of 3

D.

30 is even

30 is multiple of 3

**我的答案：**

A

1. 对于以下程序：
2. even = False
3. if even:

print("It is even!")

（    ）是正确的。

A. 程序正常运行，显示It is even!  
B. 程序正常运行，但不显示任何信息  
C. 程序有错误，if even:应该替换成if even == True:  
D. 程序有错误，if even:应该替换成if even = True:

**我的答案：**

B

1. 假设isPrime是一个布尔类型的变量，为了测试isPrime是否为真（True），下列（    ）语句是正确且最佳的。

A.

if isPrime = True:

B.

if isPrime == True:

C.

if isPrime:

D.

if not isPrime = False:

E.

if not isPrime == False:

**我的答案：**

C

1. 执行以下代码段后，y的值是（   ）。
2. x = 0

y = 10 if x > 0 else -10

（2分）

A. -10  
B. 0  
C. 10  
D. 20  
E. 非法的表达式

**我的答案：**

A

1. 对于以下代码段：

Code1:

if number % 2 == 0:

even = True

else:

even = False

Code2:

even = True if number % 2 == 0 else False

Code3:

even = number % 2 == 0

（    ）是正确的。

A. Code2有错误  
B. Code3有错误  
C. Code1、Code2和code3都是正确的，但Code1更好  
D. Code1、Code2和code3都是正确的，但Code2更好  
E. Code1、Code2和code3都是正确的，但Code3更好

**我的答案：**

E

1. 在Python中，可以使用 for 作为变量名。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在循环中，break语句的作用是跳过当前当次循环。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面程序的输出结果是2。
2. m = 0
3. x = 1
4. while x < 5:
5. y = 1
6. while y < 4:
7. m = m + y
8. y = y + 3
9. x = x + 2
10. print(m)

A. 对  
B. 错

**参考答案：**

A

1. 对于下面程序：
2. total = 0
3. for i in range(5):
4. total += i

print(i)

for循环结束后，循环变量i不存在，导致程序错误。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在循环中，continue语句的作用是结束当前循环。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面while循环执行的次数为6次。
2. k = 10
3. while k >= 5:

k = k - 1

A. 对  
B. 错

**参考答案：**

A

1. 对于带有else子句的for语句和while语句，当循环因循环条件不成立而自然结束时，会执行else子句中的代码。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在编写嵌套循环时，为了提高运行效率，应尽量减少内循环中不必要的计算。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 下面程序的输出结果是14。
2. count = 0
3. for x in range(2, 5):
4. count = count + x

print(count)

A. 对  
B. 错

**参考答案：**

B

1. 表达式len(range(1,10))的值为10。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面程序循环了\_\_\_\_\_\_\_\_次（注：填写阿拉伯数字）。
2. k = 1000
3. while k > 1:
4. print(k)

k = k // 2

**参考答案：**

9

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. number = 15292
3. if number == 0:
4. print("0")
5. else:
6. while number:
7. right\_digit = number % 10
8. print(right\_digit, end='')

number //= 10

**参考答案：**

29251

1. 下面程序执行后，输出结果的最后一行是\_\_\_\_\_\_\_\_。
2. for i in range(1, 3):
3. for j in range(2, 5):

print(i \* j)

（2分）

**参考答案：**

8

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

len(range(1,10))

**参考答案：**

9

1. 如果输入4、-1、6、9、8、3、0，下面程序的输出结果是\_\_\_\_\_\_\_\_ 。
2. number = int(input('输入一个整数: '))
3. max = number
4. while number != 0:
5. number = int(input('输入一个整数: '))
6. if number > max:
7. max = number

print(max)

**参考答案：**

9

1. 在程序中\_\_\_\_\_\_\_\_处填上恰当的成分（注：不要有任何空格），完成题目要求的功能。

从键盘输入20个整数，输出它们的和。

total = 0

for i in range(1, 21):

a= int(input())

total = \_\_\_\_\_\_\_\_

print("Total=", total)

**参考答案：**

total+a、a+total

1. 下面程序的输出结果是。
2. y = 0
3. for i in range(0, 10, 2):
4. y += i

print(y)

**参考答案：**

20

1. 下面程序执行的循环次数是\_\_\_\_\_\_\_\_ 。
2. for i in range(1, n + 1):

# 循环体

**参考答案：**

n

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. total = 0
3. for i in range(100):
4. if i % 10:
5. continue
6. total = total + i

print(total)

**参考答案：**

450

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. i=0
3. while i<10:
4. i += 1
5. if i == 7:
6. continue
7. else:

print(i,end=' ')

**参考答案：**

1 2 3 4 5 6 8 9 10

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. counter = 0
3. for i in range(10):
4. for j in range(10):
5. if i == j:
6. continue
7. counter += 1

print(counter)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**参考答案：**

90

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. x = 15
3. while 10 < x < 50:
4. x += 1
5. if x // 3:
6. x += 1
7. break
8. else:
9. continue
10. print(x)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**参考答案：**

17

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. a, b, c = 1, 2, 3
3. while a < b < c:
4. a, b = b, a
5. c -= 1

print(a, b, c)

【来源】  
《Python程序设计基础》第3章思考与练习。（2分）

**参考答案：**

2 1 2

1. 下面表达式的值为\_\_\_\_\_\_\_\_。

list(range(2, 10, 2))

**参考答案：**

[2, 4, 6, 8]

1. 在程序中\_\_\_\_\_\_\_\_处填上恰当的成分（注：不要有任何空格），完成题目要求的功能。

计算1-2+3-4+5 ... 99的所有数的和。

total1 = 0

total2 = 0

for i in range(1,100,2):

\_\_\_\_\_\_\_\_

for j in range(2,100,2):

total2 += j

total3 = total1 - total2

print(total3)

**参考答案：**

total1+=i、total1=total1+i、total1=i+total1

1. 以下程序，print语句执行了（         ）次。
2. for i in range(10):
3. for j in range(10):

print(i \* j)

（2分）

A. 100  
B. 20  
C. 10  
D. 45

**参考答案：**

A

1. 对于以下程序：
2. total = 0
3. for d in range(0,10, 0.1):

total += total + d

（    ）是正确的。  
  
（2分）

A. 程序有语法错误，因为range函数不能有三个参数  
B. 程序有语法错误，因为range函数的参数必须是整数  
C. 程序无限循环运行  
D. 程序运行正常

**参考答案：**

B

1. 给定以下四种图案：
2. 图案（1） 图案（2） 图案（3） 图案（4）
3. 1 1 2 3 4 5 6 1 1 2 3 4 5 6
4. 1 2 1 2 3 4 5 2 1 1 2 3 4 5
5. 1 2 3 1 2 3 4 3 2 1 1 2 3 4
6. 1 2 3 4 1 2 3 4 3 2 1 1 2 3
7. 1 2 3 4 5 1 2 5 4 3 2 1 1 2

1 2 3 4 5 6 1 6 5 4 3 2 1 1

（    ）是由如下程序生成的。

for i in range(1, 6 + 1):

for j in range(6, 0, -1):

print(j if j <= i else " ", end = " ")

print()

（2分）

A. 图案（1）  
B. 图案（2）  
C. 图案（3）  
D. 图案（4）

**参考答案：**

C

1. 以下程序的输出结果是（         ）。
2. y = 0
3. for i in range(10, 1, -2):
4. y += i

print(y)

（2分）

A. 10  
B. 40  
C. 30  
D. 20

**参考答案：**

C

1. 以下程序的输出结果是（         ）。
2. y = 0
3. for i in range(0, 10):
4. y += i

print(y)

（2分）

A. 10  
B. 11  
C. 12  
D. 13  
E. 45

**参考答案：**

E

1. 以下程序的输出结果是（         ）。
2. number = 6
3. while number > 0:
4. number -= 3

print(number, end = ' ')

（2分）

A. 6 3 0  
B. 6 3  
C. 3 0  
D. 3 0 -3  
E. 0 -3

**参考答案：**

C

1. 以下程序输出"Welcome to Python"（        ）次。
2. count = 0
3. while count <10:

print("Welcome to Python")

（2分）

A. 8  
B. 9  
C. 10  
D. 11  
E. 无数次

**参考答案：**

E

1. 以下程序输出"Welcome to Python"（        ）次。
2. count = 0
3. while count <10:
4. print("Welcome to Python")

count += 1

（2分）

A. 8  
B. 9  
C. 10  
D. 11  
E. 0

**参考答案：**

C

1. 对于以下程序：
2. total = d = 0
3. while d != 10.0:
4. d += 0.1

total += total + d

（         ）是正确的。

（2分）

A. 程序不能运行，因为total和d没有正确初始化  
B. 程序永远不会停止，因为循环中d的值始终是0.1  
C. 程序可能不会停止，原因是浮点数运算误差引起的数值错误  
D. 循环结束后，total的值是0 + 0.1 + 0.2 + 0.3 + ... + 1.9之和

**参考答案：**

C

1. 以下程序的输出结果是（         ）。
2. total = 0
3. item = 0
4. while item < 5:
5. item += 1
6. total += item
7. if total > 4:
8. continue

print(total)

（2分）

A. 15  
B. 16  
C. 17  
D. 18

**参考答案：**

A

1. 以下程序，print语句执行了（         ）次。
2. for i in range(10):
3. for j in range(i):

print(i \* j)

（2分）

A. 100  
B. 20  
C. 10  
D. 45

**参考答案：**

D

1. 以下程序的输出结果是（         ）。
2. x = 0
3. while x < 4:
4. x = x + 1

print(x)

A. 0  
B. 1  
C. 2  
D. 3  
E. 4

**参考答案：**

E

1. 以下程序的输出结果是（         ）。
2. for i in range(1, 11):

print(i, end = " ")

（2分）

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

**参考答案：**

B

1. 以下程序的输出结果是（         ）。
2. number = 25
3. isPrime = True
4. i = 2
5. while i < number and isPrime:
6. if number % i == 0:
7. isPrime = False
8. i += 1

print("i is", i, "isPrime is", isPrime)

A. i is 5 isPrime is True  
B. i is 5 isPrime is False  
C. i is 6 isPrime is True  
D. i is 6 isPrime is False

**参考答案：**

D

1. 以下（         ）程序正确计算了"1/2 + 2/3 + 3/4 + ... + 99/100"的值。  
   （1）
2. total = 0
3. for i in range(1, 99):
4. total += i / (i + 1)

print("Total is", total)

（2）

total = 0

for i in range(1, 100):

total += i / (i + 1)

print("Total is", total)

（3）

total = 0

for i in range(1.0, 99.0):

total += i / (i + 1)

print("Total is", total)

（4）

total = 0

for i in range(1.0, 100.0):

total += i / (i + 1)

print("Total is", total)

（2分）

A. （2）、（3）、（4）  
B. （1）、（2）、（3）、（4）  
C. （2）  
D. （3)、（4）

**参考答案：**

C

1. 以下程序的输出结果是（         ）。
2. y = 0
3. for i in range(0, 10, 2):
4. y += i

print(y)

A. 9  
B. 10  
C. 11  
D. 20

**参考答案：**

D

1. 以下程序的输出结果是（         ）。
2. number = 25
3. isPrime = True
4. for i in range(2,number):
5. if number % i == 0:
6. isPrime = False
7. break

print("i is", i, "isPrime is", isPrime)

（2分）

A. i is 5 isPrime is True  
B. i is 5 isPrime is False  
C. i is 6 isPrime is True  
D. i is 6 isPrime is False

**参考答案：**

B

1. 以下（         ）程序输出"Welcome to Python"10次。

（1）

for count in range(1, 10):

print("Welcome to Python")

（2）

for count in range(0, 10):

print("Welcome to Python")

（3）

for count in range(1, 11):

print("Welcome to Python")

（4）

for count in range(1, 12):

print("Welcome to Python")

（2分）

A. （2）、（4）  
B. （1）、（2）、（3）  
C. （1）、（3)  
D. （2）、（3）  
E. （1）、（2）

**参考答案：**

D

1. list(range(5))返回序列（         ）。（2分）

A. [1, 2, 3, 4, 5]  
B. [0, 1, 2, 3, 4, 5]  
C. [1, 2, 3, 4]  
D. [0, 1, 2, 3, 4]

**参考答案：**

D

1. 以下程序的输出结果是（         ）。
2. total = 0
3. item = 0
4. while item < 5:
5. item += 1
6. total += item
7. if total > 4:
8. break

print(total)

A. 5  
B. 6  
C. 7  
D. 8

**参考答案：**

B

1. 在自定义函数内部没有办法定义全局变量。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在同一个作用域内，局部变量会隐藏同名的全局变量。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 如果函数中没有return语句或者return语句不带任何返回值，那么该函数的返回值为None。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 自定义函数时，即使该函数不需要接收任何参数，也必须在函数名后保留一对空的圆括号来表示这是一个函数。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在自定义函数时，某个参数名字前面带有\*符号表示可变长度参数，可以接收任意多个关键字参数并将其存放于一个字典之中。

（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在调用函数时，可以通过关键字参数的形式进行传值，从而避免必须记住函数形参顺序的麻烦。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 自定义函数时，若一个参数带有默认值，那么所有其他参数都必须设置默认值。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在Python中，自定义函数的关键字是def。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 形参可以看做是函数内部的局部变量，函数运行结束之后形参就不可访问了。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 函数是代码重用的一种方式。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 若匿名函数f = [lambda x = 3: x \* 3, lambda x: x \*\* 3]，则f[1](f[0]())返回的结果是\_\_\_\_\_\_\_\_（若有错误，则填写Error）。（2分）

**参考答案：**

729

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. lst = [1]
3. def scope():
4. lst.append(6)
5. print(\*lst)

scope()

**参考答案：**

1 6

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def scope():
3. n = 4
4. m = 5
5. print(m, n, end = ' ')
6. n = 5
7. t = 8
8. scope()

print(n, t)

**参考答案：**

5 4 5 8

1. 下列程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def f(x, y):
3. return x + y, x - y, x \* y, x / y
4. t1, t2, t3, t4 = f(9, 5)

print(t1, t2, t3, t4)

**参考答案：**

14 4 45 1.8

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_ 。
2. def fun(x, y=2, z=3):
3. return x + y + z

print(fun(1, z = 4))

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

7

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a = "first"
3. def second(a):
4. a = "second"
5. def third():
6. global a
7. a = "third"
8. third()
9. print(a, end=',')
10. second("fourth")

print(a)

**参考答案：**

third,third

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. import functools
3. lst = ["Welcome ", "to ", "Python", ", ", "Programming ", "is ", "fun", "."]

print(functools.reduce(lambda x, y : x + y, lst))

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

Welcome to Python, Programming is fun.

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def fun(\*x):
3. print(x)

fun(1, 2, 3)

**参考答案：**

(1, 2, 3)

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def fib(n):
3. f1, f2 = 0, 1
4. while f2 < n:
5. print(f2, end = ' ')
6. f1, f2 = f2, f1 + f2

fib(10)

**参考答案：**

1 1 2 3 5 8

1. 若匿名函数f = lambda x, y: x + y，则f({1:2}, {3:4})返回的结果是\_\_\_\_\_\_\_\_（若有错误，则填写Error）。（2分）

**参考答案：**

Error

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. a = 10
3. def fun():
4. global a
5. a = 20
6. print(a, end = ' ')
7. fun()

print(a)

**参考答案：**

20 20

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. def fun(x, y):
3. x = x + y
4. y = x - y
5. x = x - y
6. def main():
7. x, y = 2, 3
8. fun(x, y);
9. print("%d#%d" % (x, y))

main()

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

2#3

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. a, b = 1, 2
3. def fun():
4. a = 100
5. b = 200
6. def main():
7. a, b = 5, 7
8. fun()
9. print("%d#%d" % (a, b))

main()

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

5#7

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. def main():
3. print(min(5, 6))
4. def min(n1, n2):
5. smallest = n1
6. if n2 < smallest:
7. smallest = n2

main()

**参考答案：**

None

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. b, c = 2, 4
3. def fun(d):
4. global a
5. a = d \* c
6. fun(b)

print(a)

（2分）

**参考答案：**

8

1. ﻿下面程序的输出结果是\_\_\_\_\_\_\_\_ 。
2. def fun(x, y, z):
3. return x + y + z

print(fun(y = 4, z = 5, x = 2))

【来源】  
《Python程序设计基础》第4章思考与练习。  
（2分）

**参考答案：**

11

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def nprint(message, n):
3. for i in range(n):
4. print(message, end = ' ')
5. nprint('a', 3)

nprint(n = 5, message = "good")

（2分）

**参考答案：**

a a a good good good good good

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。

print(list(map(lambda x : x \*\* 2, range(1, 11))))

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。

print(list(filter(lambda x : x % 2 != 0 and x % 3 != 0, range(2, 25))))

【来源】  
《Python程序设计基础》第4章思考与练习。（2分）

**参考答案：**

[5, 7, 11, 13, 17, 19, 23]

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. def fun(\*\*x):
3. print(x)

fun(a = 1, b = 2, c = 3)

（2分）

**参考答案：**

{'a': 1, 'b': 2, 'c': 3}

1. 下列程序的输出结果是（         ）。
2. x = 1
3. def foo():
4. y = x + 2
5. print(y, end = ' ')
6. foo()

print(x)

（2分）

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为x未定义  
D. 1 1  
E. 3 3

**参考答案：**

B

1. 下列程序的输出结果是（         ）。
2. def foo(x = 1, y =2):
3. x = x + y
4. y += 1
5. print(x, y)

foo(2, 1)

（2分）

A. 1 3  
B. 2 3  
C. 程序有运行时错误，因为x和y未定义  
D. 3 2  
E. 3 3

**参考答案：**

D

1. 若一个函数没有返回值，默认情况下，该函数返回（         ）。

（2分）

A. None  
B. int  
C. double  
D. public  
E. Null

**参考答案：**

A

1. 下列程序的输出结果是（         ）。
2. def foo(x = 1, y = 2):
3. x = x + y
4. y += 1
5. print(x, y)

foo(y = 2, x = 1)

（2分）

A. 1 3  
B. 2 3  
C. 程序有运行时错误，因为x和y未定义  
D. 3 2  
E. 3 3

**参考答案：**

E

1. print(type(lambda:3))的输出结果是（    ）。  
   （2分）

A. <class 'function'>  
B. <class 'int'>  
C. <class 'NoneType'>  
D. <class 'float'>

**参考答案：**

A

1. 函数的参数总是出现在（        ）中。

（2分）

A. 方括号  
B. 双引号  
C. 圆括号  
D. 花括号

**参考答案：**

C

1. 下面（        ）应该被定义为一个无返回值函数。

（2分）

A. 编写一个函数，打印从1到100的整数  
B. 编写一个函数，返回一个1到100之间的随机整数  
C. 编写一个函数，检查当前秒数是否是一个1到100之间的整数  
D. 编写一个函数，将一个大写字母转换为小写字母

**参考答案：**

A

1. 函数可以改变（    ）数据类型的实参。  
   （2分）

A. int  
B. string  
C. list  
D. float

**参考答案：**

C

1. 对于下列函数：
2. def n\_print(message, n):
3. while n > 0:
4. print(message, end = '')

n -= 1

调用n\_print('a', 4)，输出结果是（         ）。

（2分）

A. aaaaa  
B. aaaa  
C. aaa  
D. 无效的函数调用  
E. 无限循环

**参考答案：**

B

1. 下列程序的输出结果是（         ）。
2. def foo(x = 1, y =2):
3. return x + y, x - y
4. x, y = foo(y = 2, x= 1)

print(x, y)

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为函数返回多个值  
D. 3 -1  
E. -1 3

**参考答案：**

D

1. 在一个函数中如局部变量和全局变量同名，则（    ）。

A. 局部变量屏蔽全局变量  
B. 全局变量屏蔽局部变量  
C. 全局变量和局部变量都不可用  
D. 程序错误

**参考答案：**

A

1. 对于下列函数，错误的函数调用是（         ）。
2. def foo(arg1, arg2 = 'test', arg3 = 100):

print(arg1, arg2, arg3)

（2分）

A. foo('where', 'what')  
B. foo(arg1 = 'where', arg2 = 'what')  
C. foo(arg = 'where')  
D. foo('where')

**参考答案：**

C

1. 下列函数头，正确的是（        ）。

（2分）

A. def f(a, b == 0):   
B. def f(a, b, \*):  
C. def f(a=0, b):  
D. def f(a, \*b):

**参考答案：**

D

1. 对于下列函数：
2. def n\_print(message, n):
3. while n > 0:
4. print(message, end = '')

n -= 1

调用n\_print("A message", k)后，k的值是（         ）。

k = 2

n\_print("A message", k)

（2分）

A. 0  
B. 1  
C. 2  
D. 3

**参考答案：**

C

1. 下列程序的输出结果是（         ）。
2. x = 1
3. def foo():
4. x = x + 2
5. print(x)
6. foo()

print(x)

（2分）

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为x未定义  
D. 1 1  
E. 3 3

**参考答案：**

C

1. 在Python中，对于函数定义代码的理解，正确的是（    ）。  
     
   （2分）

A. 必须存在形参  
B. 必须存在return语句  
C. 形参和return语句都是可有可无的  
D. 形参和return语句要么都存在，要么都不存在

**参考答案：**

C

1. 下列程序的输出结果是（         ）。
2. x = 1
3. def foo():
4. x = 3
5. print(x)
6. foo()

print(x)

（2分）

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为x未定义  
D. 1 1  
E. 3 3

**参考答案：**

B

1. 对于下面不完整的程序：
2. def f(number):
3. # 缺失的函数体

print(f(5))

缺失的函数体应该是（

）。

（2分）

A. return "number"  
B. print(number)  
C. print("number")  
D. return number

**参考答案：**

D

1. 有关函数的说法，（         ）是错误的。

（2分）

A. 函数是对一段代码的命名  
B. 函数是代码功能的一种抽象  
C. 函数是计算机对代码执行优化的要求  
D. 函数是代码逻辑的封装

**参考答案：**

C

1. 下列程序的输出结果是（         ）。
2. def foo(x = 1, y =2):
3. x = x + y
4. y += 1
5. print(x, y)

foo()

（2分）

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为x和y未定义  
D. 1 1  
E. 3 3

**参考答案：**

E

1. 对于下列函数：
2. def n\_print(message, n):
3. while n > 0:
4. print(message, end = '')

n -= 1

调用n\_print(n = k, message = "A message")后，k的值是（        ）。

k = 2

n\_print(n = k, message = "A message")

（2分）

A. 0  
B. 1  
C. 2  
D. 3

**参考答案：**

C

1. 下列程序的输出结果是（         ）。
2. x = 1
3. def foo():
4. global x
5. x = x + 2
6. print(x)
7. foo()

print(x)

（2分）

A. 1 3  
B. 3 1  
C. 程序有运行时错误，因为x未定义  
D. 1 1  
E. 3 3

**参考答案：**

E

1. 下列函数头，正确的是（        ）。

（2分）

A. def f(a = 1, b):  
B. def f(a = 1, b, c = 2):  
C. def f(a = 1, b =1, c = 2):  
D. def f(a = 1, b =1, c = 2, d):

**参考答案：**

C

1. 函数头由（        ）组成。

（2分）

A. def关键字、函数名和冒号  
B. def关键字、函数名、参数表和冒号  
C. 函数名和参数表  
D. 函数名、参数表和冒号

**参考答案：**

B

1. 一个函数（        ）。

（2分）

A. 必须至少有一个参数  
B. 可以没有参数  
C. 必须有一个return语句返回一个值  
D. 必须有一个return语句返回多个值

**参考答案：**

B

1. 在turtle中，画布绘图坐标系的绝对0度方向默认指向画布正右方。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 海龟图起源于LOGO编程语言。这是一个很经典的专门用来教小孩子编程的图形模块。Python本身实现并内置了海龟图模块。

（1分）

A. 对  
B. 错

**参考答案：**

A

1. turtle.circle(50)沿顺时针方向绘制了一个半径为50像素的圆，圆心在在海龟左侧。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在turtle中，画布（Canvas）是用于绘制图形的矩形区域。（1分）

A. 对  
B. 错

**参考答案：**

A

1. turtle.circle(-90，90)沿顺时针方向绘制了一个半径为90像素的弧形，圆心在在海龟右侧。（1分）

A. 对  
B. 错

**参考答案：**

A

1. turtle.dot(100, "red")绘制了一个半径为100像素的红色实心点。（1分）

A. 对  
B. 错

**参考答案：**

B

1. mainloop方法或done方法开始事件循环，必须是海龟图形程序中的最后一条语句。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在turtle中，画布绘图坐标系的原点默认在屏幕左上角。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在turtle中，setheading(to\_angle)或seth(to\_angle)方法，将海龟的方向设置为to\_angle，to\_angle为角度。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在turtle中，clear方法用于清除画布，并将海龟状态和位置重置为起始默认值。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法可以绘制圆、弧形和正多边形（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

circle、circle()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法朝着海龟箭头指向的反方向移动画笔（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

backward、backward()、back、back()、bk、bk()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法朝着海龟箭头指向的方向移动画笔（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

forward、forward()、fd、fd()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法设置画布窗口大小和位置（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

setup、setup()

1. 在Python中，为了使用turtle图形模块，必须使用\_\_\_\_\_\_\_\_来导入该模块（注意：不要有任何多余的空格）。（2分）

**参考答案：**

import turtle、import

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法抬起画笔，画笔移动时不绘制图形（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

penup、penup()、pu、pu()、up、up()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法设置填充颜色（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

fillcolor、fillcolor()、color、color()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法设置画笔颜色（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

pencolor、pencolor()、color、color()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法绘制文本（注意：只需填写方法或函数名即可）。**参考答案：**

write、write()

1. 在turtle中，使用\_\_\_\_\_\_\_\_方法设置画笔宽度（注意：只需填写方法或函数名即可）。（2分）

**参考答案：**

pensize、pensize()、width、width()

1. 可重复撤消最后一次海龟操作，使用（         ）。（2分）

A. turtle.rollback()  
B. turtle.redo()  
C. turtle.undo()  
D. turtle.remove()

**参考答案：**

C

1. 放下画笔，移动海龟时绘制图形，使用（         ）。（2分）

A. turtle.penDown()  
B. turtle.pendown()  
C. turtle.putDown()  
D. turtle.Down()

**参考答案：**

B

1. 显示海龟，使用（         ）。（2分）

A. turtle.show()  
B. turtle.showLocation()  
C. turtle.showDirection()  
D. turtle.showturtle()  
E. turtle.showTurtle()

**参考答案：**

D

1. 隐藏海龟，使用（         ）。（2分）

A. turtle.home()  
B. turtle.hideturtle()  
C. turtle.hideTurtle()  
D. turtle.clear()  
E. turtle.reset()

**参考答案：**

B

1. 下面程序的执行结果是（         ）。
2. import turtle
3. t = turtle.Turtle()
4. t.circle(40)
5. t.circle(60)
6. t.circle(80)
7. t.circle(100)

turtle.mainloop()

A. 同心圆  
B. 太极图  
C. 同切圆  
D. 笛卡尔心形

**参考答案：**

C

1. 下面程序的执行结果是（         ）。
2. import turtle
3. t = turtle.Turtle()
4. t.circle(100, steps=6)

turtle.mainloop()

（2分）

A. 圆  
B. 椭圆  
C. 弧形  
D. 正六边形

**参考答案：**

D

1. 下列（         ）不能改变海龟画笔的移动方向。

A. bk()  
B. seth()  
C. left()  
D. right()

**参考答案：**

A

1. 将海龟移动到坐标点(4、5)，不改变海龟的方向，使用（        ）。（2分）

A. turtle.move(4, 5)  
B. turtle.moveTo(4, 5)  
C. turtle.moveto(4, 5)  
D. turtle.go(4, 5)  
E. turtle.goto(4, 5)

**参考答案：**

E

1. 下面程序的执行结果是（         ）。
2. import turtle
3. t = turtle.Turtle()
4. for i in range(3):
5. t.seth(i \* 120)
6. t.fd(200)

turtle.mainloop()

（2分）

A. 直角三角形  
B. 等边三角形  
C. 等腰三角形  
D. 锐角三角形

**参考答案：**

B

1. 设置海龟移动速度为5，使用（        ）。（2分）

A. turtle.speed(5)  
B. turtle.setSpeed(5)  
C. turtle.setspeed(5)  
D. turtle.velocity(5)

**参考答案：**

A

1. "HZNU"[::-1]返回的结果是"HZNU"。（1分）

A. 对  
B. 错

**我的答案：**

B

1. 在Python中，字符串字面量可以表示为以单引号'或双引号"括起来的一个字符序列。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 字符串是不可变对象，不可以直接修改字符串中的内容。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 表达式'ab' in 'acbed'的值为True。（1分）

A. 对  
B. 错

**我的答案：**

B

1. len("HZNU")返回的结果是4，len("杭州师大")返回的结果也是4。（1分）

A. 对  
B. 错

**我的答案：**

A

1. Unicode编码是目前最广泛使用的字符编码。UTF-8是在互联网上使用最广泛的一种Unicode的实现方式。Python 3默认使用UTF-8编码。  
   （1分）

A. 对  
B. 错

**我的答案：**

A

1. 已知x为非空字符串，那么表达式 ''.join(x.split())==x 的值一定为True。（1分）

A. 对  
B. 错

**我的答案：**

B

1. 假设s1="Python"，s2="Python"，则表达式s1 is s2的值为True。（1分）

A. 对  
B. 错

**我的答案：**

A

1. "HHZZNNUU"[::2]返回的结果是"HZNU"。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 字符串属于Python序列类型，支持下标操作。（1分）

A. 对  
B. 错

**我的答案：**

A

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print("oops too".strip('o'))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

ps t

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. x = "university"

print(x[0:4])

【来源】

《Python程序设计基础》第5章思考与练习。

（2分）

**我的答案：**

univ

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print("hello".isalpha(), "CSE-1001".isalnum())

【来源】  
《Python程序设计基础》第5章思考与练习。

（2分）

**我的答案：**

True False

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print("Computer science".split ('e'))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

['Comput', 'r sci', 'nc', '']

1. 当输入为'cbabc'时，下列程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. s1 = input()
3. index = 0
4. s2 = ''
5. while index < len(s1) - 1:
6. if s1[index] > s1[index + 1]:
7. s2 += s1[index]
8. else:
9. s2 = s2 \* 2
10. index += 1

print(s2)

（2分）

**我的答案：**

cbcbcbcb

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("this is a test".count('s', 5))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

2

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print("Mississippi".replace('i', 'z', 2))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

Mzsszssippi

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("test".center(10, 'x'))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

xxxtestxxx

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("abcdef".find('e'))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

4

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("test".join(['A', 'B', 'C']))

【来源】  
《Python程序设计基础》第5章思考与练习。  
（2分）

**我的答案：**

AtestBtestC

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print("knickknack".rfind('k', 3, -2))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

5

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("university".index("iv"))

【来源】  
《Python程序设计基础》第5章思考与练习。

（2分）

**我的答案：**

2

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。

print("bird".endswith("ir"))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

False

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. my\_str = "Hello"

print(my\_str.lower(), my\_str.upper())

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**我的答案：**

hello HELLO

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. my\_str = "Hello"

print(my\_str.islower(), my\_str.isupper())

【来源】  
《Python程序设计基础》第5章思考与练习。

（2分）

**我的答案：**

False False

1. 对于如下字符串：  
   s= "Programming is fun"  
     
   s.find('rom')的计算结果是（        ）。

（2分）

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

**我的答案：**

E

1. "Programming is fun"[:2]的计算结果是（        ）。

（2分）

A. Pr  
B. p  
C. Pro  
D. Programming  
E. Programming is

**我的答案：**

A

1. 字符串s长度为奇数，则显示中间字符的表达式为（         ）。

（2分）

A. s[len(s) // 2 + 1]  
B. s[(len(s) - 1) // 2]  
C. s[len(s) // 2 - 1]  
D. s[(len(s) + 1) // 2]

**我的答案：**

B

1. 字符串"abc"的长度是3，字符串"老师好"的长度是（         ）。

（2分）

A. 1  
B. 3  
C. 6  
D. 9

**我的答案：**

B

1. len("Good")的计算结果是（         ）。

（2分）

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

**我的答案：**

D

1. "Programming is fun"[4:6]的计算结果是（         ）。

（2分）

A. ram  
B. ra  
C. r  
D. pr  
E. pro

**我的答案：**

B

1. "Programming is fun"[-3:-1]的计算结果是（        ）。

（2分）

A. Pr  
B. p  
C. fun  
D. un  
E. fu

**我的答案：**

E

1. "Programming is fun"[1:1]的计算结果是（        ）。

（2分）

A. Pr  
B. r  
C. Pr  
D. ''  
E. 非法的表达式

**我的答案：**

D

1. min("Programmingis fun")的计算结果是（        ）。

（2分）

A. P  
B. r  
C. 空格字符  
D. u  
E. n

**我的答案：**

C

1. 对于如下字符串：  
   s ="Welcome"  
     
   s.count('e')的计算结果是（        ）。

（2分）

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

**我的答案：**

B

1. max("Programming is fun")的计算结果是（         ）。

（2分）

A. P  
B. r  
C. 空格字符  
D. u  
E. n

**我的答案：**

D

1. "Programming is fun"[:-1]的计算结果是（        ）。

（2分）

A. Programming  
B. Programming is fun  
C. Programming is f  
D. Programming is fu  
E. Programming is

**我的答案：**

D

1. "Programming is fun"[-1]的计算结果是（        ）。

（2分）

A. Pr  
B. p  
C. fun  
D. n  
E. un

**我的答案：**

D

1. 对于如下字符串：  
   s ="Welcome"  
     
   下列（         ）是错误的。

（2分）

A. print(s[0])  
B. print(s.lower())  
C. s[1] = 'r'  
D. print(s.strip())

**我的答案：**

C

1. 对于如下字符串：  
   s= "Programming is fun"  
     
   s.find('ram')的计算结果是（        ）。

（2分）

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

**我的答案：**

D

1. 对于如下字符串：  
   s= "Programming is fun"  
     
   s.find('m')的计算结果是（        ）。

（2分）

A. 8  
B. 7  
C. 6  
D. 5  
E. -1

**我的答案：**

C

1. 对于如下字符串：  
   s= "Programming is fun"  
     
   s.rfind('m')的计算结果是（        ）。

（2分）

A. 8  
B. 7  
C. 6  
D. 5  
E. -1

**我的答案：**

B

1. "Good".replace('o', 'e')的计算结果是（         ）。

（2分）

A. God  
B. Good  
C. Geed  
D. Ged

**我的答案：**

C

1. 假设s = "\t\tWelcome\n"，s.strip()结果是（   ）。

（2分）

A. welcome  
B. \t\tWelocme\n  
C. \t\tWELCOME\n  
D. Welcome

**我的答案：**

D

1. 字符串的strip方法的作用是（         ）。

A. 按照指定字符分割字符串为列表  
B. 连接两个字符串序列  
C. 去掉字符串两侧空格或指定字符  
D. 替换字符串中特定字符

**我的答案：**

C

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[0:100]的结果是错误。  
   （1分）

A. 对  
B. 错

**参考答案：**

B

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[3:8:2]的结果是[-10, 35, 49]。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[::2]的结果是[12, -22, -26, 0, 3]。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 只能通过切片访问元组中的元素，不能使用切片修改元组中的元素。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 元组的访问速度比列表要快一些，如果定义了一系列常量值，并且主要用途仅仅是对其进行遍历而不需要进行任何修改，建议使用元组而不使用列表。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 将列表中的元素顺序打乱的shuffle()方法不是random模块中的方法。  
   （1分）

A. 对  
B. 错

**参考答案：**

B

1. 表达式tuple(zip(\*[[1, 2, 3], [4, 5, 6]]))的值为((1, 4), (2, 5), (3, 6))。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 已知x为非空列表，那么表达式 sorted(x,reverse=True) == list(reversed(x)) 的值一定是True。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 元组是不可变对象，但如果元组中的元素是可变对象，这样的可变对象是可以修改的。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 同一个列表中的元素类型可以各不相同。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[100:]的结果是[]。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 表达式 (1,) + (2,) 的值为3。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[100]的结果是运行错误。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 列表是可变对象，可以直接修改列表中的元素值。

（1分）

A. 对  
B. 错

**参考答案：**

A

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[::]的结果是[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 只能通过切片访问列表中的元素，不能使用切片修改列表中的元素。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[1::2]的结果是[12, -22, -26, 0, 3]。  
   （1分）

A. 对  
B. 错

**参考答案：**

B

1. 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]，lst[::-1]的结果是[-21, 3, 49, 0, 35, -26, -10, -22, -5, 12]。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 对于列表而言，在尾部追加元素比在中间位置插入元素速度更快一些，尤其是对于包含大量元素的列表。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 列表可以用find()方法来搜索数据是否在列表中。  
   （1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面表达式的值是\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

[x for x in [1,2,3,4,5] if x < 3]

（2分）

**参考答案：**

[1, 2]

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. a = 8
3. b = 3
4. c = 2
5. my\_tuple = a, b, c

print(my\_tuple)

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

(8, 3, 2)

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。（注意：不要有任何多余的空格）
2. my\_list = [3, "dog", 9, "cat"]

print(my\_list.index('cat'))

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

3

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. my\_list = ["Python", 3.4, 54, "Java", 82, 'C', 7.4]

print(my\_list[2::3])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

[54, 'C']

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（若有错误，则填写Error）。
2. my\_list = [5, "old", "new", 8, "time", 2]

print(my\_list[6])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

Error

1. 语句x = (3,)执行后，x的值为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。（2分）

**参考答案：**

(3,)

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。（注意：不要有任何多余的空格）
2. x = [5, 'dog']
3. y = ['cat', 3.5]

print(3 \* x)

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

[5, 'dog', 5, 'dog', 5, 'dog']

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. my\_list = ["Python", 3.4, 54, "Java", 82, 'C', 7.4]

print(my\_list[6:0:-2])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

[7.4, 82, 54]

1. 语句x = (3)执行后，x的值为\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

3

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_。
2. my\_list = [5, "old", "new", 8, "time", 2]

print(my\_list[-1])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

2

1. lst=[3,4,5,6,5,4,3]，执行lst.remove(3)后，lst[0]的值是\_\_\_\_\_\_\_\_。

**参考答案：**

4

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。（注意：不要有任何多余的空格）
2. x = [5, 'dog']
3. y = ['cat', 3.5]

print(x + y)

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

[5, 'dog', 'cat', 3.5]

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. my\_list = ["pet", 12, "cat", 4.3, "dog", 46]

print(my\_list[3:-3])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

[]

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注：不要有任何多余的空格）。
2. my\_list = ["pet", 12, "cat", 4.3, "dog", 46]

print(my\_list[0:1])

【来源】  
《Python程序设计基础》第5章思考与练习。（2分）

**参考答案：**

['pet']

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. my\_list = [3, "dog", 9, "cat"]
3. my\_list.append("dog")
4. my\_list.insert(1, [6, 9])
5. my\_list.remove(3)
6. my\_list.pop()
7. del my\_list[0]
8. my\_list.extend([7,8])

print(my\_list)

【来源】  
《Python程序设计基础》第5章思考与练习。  
**参考答案：**

['dog', 9, 'cat', 7, 8]

1. 下列程序的输出结果为（         ）。
2. myList = [1, 2, 3, 4, 5, 6]
3. for i in range(1, 6):
4. myList[i - 1] = myList[i]
5. for i in range(0, 6):

print(myList[i], end = " ")

（2分）

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

**参考答案：**

C

1. 将5添加到list1列表的尾部，使用（         ）。（2分）

A. list1.add(5)  
B. list1.append(5)  
C. list1.addLast(5)  
D. list1.addEnd(5)

**参考答案：**

B

1. 假设t = (1, 2, 4, 3)，t[1:3]的计算结果是（        ）。（2分）

A. (1, 2)  
B. (1, 2, 4)  
C. (2, 4)  
D. (2, 4, 3)  
E. (1, 2, 4, 3)

**参考答案：**

C

1. 假设list1 = [1, 3, 2, 4, 5, 2, 1, 0]，list1[:-1]的计算结果是（        ）。（2分）

A. 0  
B. [1, 3, 2, 4, 5, 2, 1]  
C. [1, 3, 2, 4, 5, 2]  
D. [1, 3, 2, 4, 5, 2, 1, 0]

**参考答案：**

B

1. 从list1列表中删除字符串“red”，使用（        ）。（2分）

A. list1.remove("red")  
B. list1.remove(red)  
C. list1.removeAll("red")  
D. list1.removeOne("red")

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. def f(value,values):
3. value = 1
4. values[0] = 44
5. t = 3
6. v = [1, 2, 3]
7. f(t, v)

print(t, v[0])

（2分）

A. 1 1  
B. 1 44  
C. 3 1  
D. 3 44

**参考答案：**

D

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.extend([34, 5])后，list1为（         ）。（2分）

A. [3, 4, 5, 20, 5, 25, 1, 3, 34, 5]  
B. [1, 3, 3, 4, 5, 5, 20, 25, 34, 5]  
C. [25, 20, 5, 5, 4,3, 3, 1, 34, 5]  
D. [1, 3, 4, 5, 20, 5, 25, 3, 34, 5]  
E. [3, 1, 25, 5, 20, 5, 4, 3, 34, 5]

**参考答案：**

A

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.index(5)的计算结果是（        ）。（2分）

A. 0  
B. 4  
C. 1  
D. 2

**参考答案：**

D

1. 将list1的所有元素随机排列，使用（         ）。（2分）

A. list1.shuffle()  
B. shuffleList(list1)  
C. random.shuffle(list1)  
D. random.shuffleList(list1)

**参考答案：**

C

1. 假设list1 = [1, 3, 2, 4, 5, 2, 1, 0]，list1[-1]的计算结果是（        ）。（2分）

A. 3  
B. 5  
C. 1  
D. 0

**参考答案：**

D

1. 假设list1 = [11, 2, 23]、list2 = [2, 11, 23]，list1 == list2的计算结果是（         ）。（2分）

A. True  
B. False  
C. 非法的表达式

**参考答案：**

B

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，max(list1)的计算结果是（ ）。（2分）

A. 5  
B. 4  
C. 8  
D. 25  
E. 1

**参考答案：**

D

1. 假设t = (1, 2)，2 \* t的计算结果是（        ）。（2分）

A. (1, 2, 1, 2)  
B. [1, 2, 1, 2]  
C. (1, 1, 2, 2)  
D. [1, 1, 2, 2]  
E. 非法的表达式

**参考答案：**

A

1. 假设list1 = [0.5 \* x for x in range(0, 4)]，list1的计算结果是（        ）。（2分）

A. [0, 1, 2, 3]  
B. [0, 1, 2, 3, 4]  
C. [0.0, 0.5, 1.0,1.5]  
D. [0.0, 0.5, 1.0,1.5, 2.0]

**参考答案：**

C

1. 下列（        ）语句创建一个列表。（2分）

A. list1 = list()  
B. list1 = []  
C. list1 = list([12, 4, 4])  
D. list1 = [12, 4, 4]  
E. list1 = [1,"3", "red"]  
F. 以上都是

**参考答案：**

F

1. 假设list1 = [11, 2, 23]、list2 = [11, 2, 2]，list1 < list2的计算结果是（        ）。（2分）

A. True  
B. False  
C. 非法的表达式

**参考答案：**

B

1. 假设x = [[1, 2], [3, 4, 5], [5, 6, 5, 9]]，len(x[0])、len(x[1])和len(x[2])的值为（         ）。

（2分）

A. 2、3和3  
B. 2、3和4  
C. 3、3和3  
D. 3、3和4

**参考答案：**

B

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.pop()后，list1为（         ）。（2分）

A. [3, 4, 5, 20, 5, 25, 1]  
B. [1, 3, 3, 4, 5, 5, 20, 25]  
C. [3, 5, 20, 5, 25, 1, 3]  
D. [1, 3, 4, 5, 20, 5, 25]  
E. [3, 1, 25, 5, 20, 5, 4]

**参考答案：**

A

1. 假设t = (1, 2, 4, 3)，t[1:-1]的计算结果是（        ）。（2分）

A. (1, 2)  
B.  (1, 2, 4)  
C. (2, 4)  
D. (2, 4, 3)  
E. (1, 2, 4, 3)

**参考答案：**

C

1. list("a#b#c#d".split('#'))的计算结果是（         ）。（2分）

A. ['a', 'b', 'c', 'd']  
B. ['a b c d']  
C. ['a#b#c#d']  
D. ['abcd']

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. list1 = [1, 3]
3. list2 = list1
4. list1[0] = 4

print(list2)

（2分）

A. [1, 3]  
B. [4, 3]  
C. [1, 4]  
D. [1, 3, 4]

**参考答案：**

B

1. 下列（        ）是元组（tuple）。

（2分）

A. [1, 2, 3]  
B. (1, 2, 3)  
C. {1, 2, 3}  
D. {}

**参考答案：**

B

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.pop(1)后，list1为（         ）。（2分）

A. [3, 4, 5, 20, 5, 25, 1, 3]  
B. [1, 3, 3, 4, 5, 5, 20, 25]  
C. [3, 5, 20, 5, 25, 1, 3]  
D. [1, 3, 4, 5, 20, 5, 25]  
E. [3, 1, 25, 5, 20, 5, 4]

**参考答案：**

C

1. "Welcome to Python".split()的计算结果是（         ）。（2分）

A. ["Welcome", "to", "Python"]  
B. ("Welcome","to", "Python")  
C. {"Welcome","to", "Python"}  
D. "Welcome","to", "Python"

**参考答案：**

A

1. max((3, 5, 1, 7, 4))的结果是（        ）。

A. 1  
B. 3  
C. 5  
D. 7

**参考答案：**

D

1. 假设list1 = [1, 3, 2]，list1 \* 2的计算结果是（         ）。（2分）

A. [2, 6, 4]  
B. [1, 3, 2, 1, 3]  
C. [1, 3, 2, 1, 3, 2]  
D. [1, 3, 2, 3, 2,1]

**参考答案：**

C

1. list("abcd")的计算结果是（         ）。（2分）

A. ['a', 'b', 'c', 'd']  
B. ['ab']  
C. ['cd']  
D. ['abcd']

**参考答案：**

A

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.count(5)的计算结果是（        ）。（2分）

A. 0  
B. 4  
C. 1  
D. 2

**参考答案：**

D

1. 下列（        ）不是创建元组的正确方式。（2分）

A. (1,)  
B. (1, 2)  
C. (1, 2, (3, 4))  
D. (1)

**参考答案：**

D

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.sort()后，list1为（         ）。（2分）

A. [3, 4, 5, 20, 5, 25, 1, 3]  
B. [1, 3, 3, 4, 5, 5, 20, 25]  
C. [25, 20, 5, 5, 4, 3, 3, 1]  
D. [1, 3, 4, 5, 20, 5, 25, 3]

**参考答案：**

B

1. 下列（        ）是列表（list)。

（2分）

A. [1, 2, 3]  
B. (1, 2, 3)  
C. {1, 2, 3}  
D. {}

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. myList = [1, 5, 5, 5, 5, 1]
3. max = myList[0]
4. indexOfMax = 0
5. for i in range(1,len(myList)):
6. if myList[i] > max:
7. max = myList[i]
8. indexOfMax = i

print(indexOfMax)

（2分）

A. 0  
B. 1  
C. 2  
D. 3  
E. 4

**参考答案：**

B

1. 假设list1 = [1, 3, 2]，sum(list1)的计算结果是（         ）。（2分）

A. 5  
B. 4  
C. 6  
D. 2  
E. 1

**参考答案：**

C

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，list1.reverse()后，list1为（         ）。（2分）

A. [3, 4, 5, 20, 5, 25, 1, 3]  
B. [1, 3, 3, 4, 5, 5, 20, 25]  
C. [25, 20, 5, 5, 4, 3, 3, 1]  
D. [1, 3, 4, 5, 20, 5, 25, 3]  
E. [3, 1, 25, 5, 20, 5, 4, 3]

**参考答案：**

E

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，min(list1)的计算结果是（        ）。（2分）

A. 5  
B. 4  
C. 8  
D. 25  
E. 1

**参考答案：**

E

1. 假设t = (1, 2, 4, 3, 8, 9)，[t[i] for i in range(0, len(t), 2)]的计算结果是（        ）。（2分）

A. [2, 3, 9]  
B. [1, 2, 4, 3, 8, 9]  
C. [1, 4, 8]  
D. (1, 4, 8)  
E. (2, 3, 9)

**参考答案：**

C

1. 假设list1 = [3, 4, 5, 20, 5, 25, 1, 3]，len(list1)的计算结果是（        ）。（2分）

A. 6  
B. 7  
C. 8  
D. 5  
E. 4

**参考答案：**

C

1. 下列程序的输出结果为（         ）。
2. def f(values):
3. values[0] = 44
4. v = [1, 2, 3]
5. f(v)

print(v)

（2分）

A. [1, 44]  
B. [1, 2, 3, 44]  
C. [44, 2, 3]  
D. [1, 2, 3]

**参考答案：**

C

1. 将5插入到list1列表的第三个位置，使用（        ）。（2分）

A. list1.insert(3, 5)  
B. list1.insert(2, 5)  
C. list1.add(3, 5)  
D. list1.append(3, 5)

**参考答案：**

B

1. 假设t1 = (1, 2, 4, 3)、t2 = (1, 2, 3, 4)，t1 < t2 的计算结果是（         ）。（2分）

A. True  
B. False  
C. 非法的表达式

**参考答案：**

B

1. 当以指定键为下标给字典对象赋值时，若该键存在则表示修改该键对应的值，若不存在则表示为字典对象添加一个新的键/值对。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 字典是可变对象，可以直接修改字典中的元素值。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 字典中的值不允许重复。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 字典中的键可以是元组。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 可以使用空的花括号{}或内置函数dict()创建一个空字典。

（1分）

A. 对  
B. 错

**参考答案：**

A

1. 执行语句d ={[1, 2]:3, [3, 4]:7}，创建了字典d。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 字典中的元素之间存在顺序关系。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 字典中的键可以对应多个值。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 字典中的键不允许重复。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 字典中的键可以是列表。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. d = {"uno": "one","dos": "two"}
3. d.setdefault('one', 'dog')

print(d["one"])

**参考答案：**

dog

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. def main():
3. d = {}
4. d["susan"] = 50
5. d["jim"] = 45
6. d["joan"] = 54
7. d["susan"] = 51
8. d["john"] = 53
9. print(len(d))

main()

**参考答案：**

4

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. numbers = {"one": 1, "two": 2, "three": 3}

print (numbers["two"])

**参考答案：**

2

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格） 。
2. d = {"uno": ["one", 1], "dos": ["two", 2], 3: ["tres","three"]}

print(d.get(3, 'cat'))

**参考答案：**

['tres', 'three']

1. 使用字典对象的\_\_\_\_\_\_\_\_方法（注意：不要有任何多余的空格）获取字典的所有值。**参考答案：**

values、values()

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. x = {1: 1}
3. x[2] = 2

print(len(x))

**参考答案：**

2

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print(dict(zip([1, 2], [3, 4])))

**参考答案：**

{1: 3, 2: 4}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格） 。
2. d = {"uno": ["one", 1], "dos": ["two", 2], 3: ["tres","three"]}

print(d.get('two', 'cat'))

**参考答案：**

cat

1. 下列程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格） 。
2. numbers = {"one":1, "two":[4, 6, 3], "three":3}
3. x = numbers["two"]
4. x.sort()

print(x)

**参考答案：**

[3, 4, 6]

1. 使用字典对象的\_\_\_\_\_\_\_\_方法（注意：不要有任何多余的空格）获取字典的所有键/值对。（2分）

**参考答案：**

items、items()

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. d = {"uno": "one","dos": "two"}
3. d.setdefault('uno', 'cat')

print(d["uno"])

**参考答案：**

one

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. numbers = {"one": 1, "two": 2, "three": 3}

print ("two" in numbers)

**参考答案：**

True

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格） 。
2. numbers = {1:2, 3:4}
3. numbers.pop(3)

print(numbers)

**参考答案：**

{1: 2}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. x = {1: 2, 3: 4, 5: 6}

print(sum(x))

**参考答案：**

9

1. 使用字典对象的\_\_\_\_\_\_\_\_方法（注意：不要有任何多余的空格）获取字典的所有键 。**参考答案：**

keys、keys()

1. 下列会返回错误的语句是（        ）。

（2分）

A. d = dict([0, 1], [2, 3])  
B. d = {}  
C. d = dict(([0, 1], [2, 3]))  
D. d = {0:1}

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. d ={"john":40, "peter":45}

print(list(d.keys()))

A. ['john', 'peter']  
B. [john': 40, 'peter': 45]  
C. ('john', 'peter')  
D. (john': 40, 'peter': 45)

**参考答案：**

A

1. 下面（        ）字典不能被正确创建。（2分）

A. d = {3:[1, 2], 7:[3, 4]}  
B. d = {[1, 2]:3, [3, 4]:7}  
C. d = {(1, 2):3, (3, 4):7}  
D. d = {"apple":1, "banana":3, "pear":3}  
E. d = {False:0, True:1}

**参考答案：**

B

1. 给定字典d，下列对d.values()的描述正确的是（        ）。

A. 返回一个字符串类型，包括字典d中所有值  
B. 返回一个元组类型，包括字典d中所有值  
C. 返回一种dict\_values类型，包括字典d中所有值  
D. 返回一个列表类型，包括字典d中所有值

**参考答案：**

C

1. 假设d = {"john":40, "peter":45}，使用的d["susan"]获取一个值时将发生（         ）。

A. 因为"susan"不是字典中的一个值，抛出KeyError异常  
B. 正常执行，没有异常发生，返回None  
C. 因为"susan"不是字典中的一个键，抛出KeyError异常  
D. 因为"susan"不是字典中的一个键，抛出语法错误

**参考答案：**

C

1. 若字典dict\_color = {"seashell":"海贝色", "gold":"金色", "pink":"粉红色", "brown":"棕色", "purple":"紫色", "tomato":"西红柿色"}，则能够输出"海贝色"的是（        ）。

A. print(dict\_color.keys())  
B. print(dict\_color["海贝色"])  
C. print(dict\_color.values())  
D. print(dict\_color["seashell"])

**参考答案：**

D

1. 给定字典d，下列（        ）对x in d的描述是正确的。

（2分）

A. 判断x是否是字典d中的键  
B. x是一个二元元组，判断x是否是字典d中的键值对  
C. 判断x是否是在字典d中以键或值方式存在  
D. 判断x是否是字典d中的值

**参考答案：**

A

1. 给定字典d，下列对d.keys()的描述正确的是（        ）。

A. 返回一种dict\_keys类型，包括字典d中所有键  
B. 返回一个元组类型，包括字典d中所有键  
C. 返回一个列表类型，包括字典d中所有键  
D. 返回一个字符串类型，包括字典d中所有键

**参考答案：**

A

1. 假设d = {"john":40, "peter":45}，使用的d.get("susan")获取一个值时将发生（         ）。

（2分）

A. 因为"susan"不是字典中的一个值，抛出KeyError异常  
B. 正常执行，没有异常发生，返回None  
C. 因为"susan"不是字典中的一个键，抛出KeyError异常  
D. 因为"susan"不是字典中的一个键，抛出语法错误

**参考答案：**

B

1. 假设d = {"john":40, "peter":45}，"john" in d的计算结果是（         ）。

A. True  
B. False  
C. 非法的表达式

**参考答案：**

A

1. 假设d = {"john":40, "peter":45}，键（keys）是（        ）。

A. "john"、40、45和"peter"  
B. "john"和"peter"  
C. 40和45  
D. d = (40:"john", 45:"peter")

**参考答案：**

B

1. 假设d = {"john":40, "peter":45}，删除"john":40，使用（         ）。

A. d.delete("john":40)  
B. d.delete("john")  
C. del d["john"]  
D. del d("john":40)

**参考答案：**

C

1. 下面程序的结果是（        ）。
2. d= {'a': 1, 'b': 2, 'b': '3'}

print(d['b'])

A. {'b':2}  
B. 3  
C. 2  
D. 1

**参考答案：**

B

1. 假设d1 = {"john":40, "peter":45}、d2 = {"john":466, "peter":45}，d1 > d2 的计算结果是（        ）。

A. True  
B. False  
C. 非法的表达式

**参考答案：**

C

1. 对于字典d={'abc':1, 'qwe':2, 'zxc':3}，len(d)的结果为（       ）。

A. 9  
B. 3  
C. 6  
D. 12

**参考答案：**

B

1. 假设d = {"john":40, "peter":45}，d["john"]的计算结果是（         ）。

A. 40  
B. 45  
C. "john"  
D. "peter"

**参考答案：**

A

1. 下列（        ）是字典（dictionary)。

（2分）

A. [1, 2, 3]  
B. (1, 2, 3)  
C. {1, 2, 3}  
D. {}

**参考答案：**

D

1. 假设d = {"john":40, "peter":45}，获取字典中的条目（键/值对）数量，使用（        ）。

A. d.size()  
B. len(d)  
C. size(d)  
D. d.len()

**参考答案：**

B

1. 关于花括号{}，下列描述正确的是（        ）。

A. 直接使用{}将生成一个字符串类型  
B. 直接使用{}将生成一个列表类型  
C. 直接使用{}将生成一个字典类型  
D. 直接使用{}将生成一个元组类型

**参考答案：**

C

1. 假设d1 = {"john":40, "peter":45}、d2 = {"john":466, "peter":45}，d1 == d2 的计算结果是（        ）。

A. True  
B. False  
C. 非法的表达式

**参考答案：**

B

1. 集合中的元素可以是列表。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 集合中的元素可以是元组。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 集合中的元素之间存在顺序关系。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 已知s1和s2是两个集合，则执行s1.union(s2)后，s1中的元素是原来s1和s2两个集合中的元素的并集。  
   （1分）

A. 对  
B. 错

**参考答案：**

B

1. 下面程序的运行结果一定是：1 2 3 4。
2. set1={1,2,3,4}
3. for i in set1:

print(i, end=" ")

A. 对  
B. 错

**参考答案：**

B

1. 集合中的元素不允许重复。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 集合中的元素可以是字典。  
   【来源】  
   《Python程序设计基础》第6章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 假设set1= {1, 2, 3}，set2={1, 2, 3}，则表达式set1 is set2的值为True。

A. 对  
B. 错

**参考答案：**

B

1. 集合分为可变集合和不可变集合。可以直接修改不可变集合中的元素值。

A. 对  
B. 错

**参考答案：**

B

1. 可以通过下标来访问集合中的元素。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 可以使用空的花括号{}或内置函数set()创建一个空集合。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 若不关心元素的顺序，使用集合来存储数据比使用列表效率更高。

A. 对  
B. 错

**参考答案：**

A

1. len(set([0,4,5,6,0,7,8])的结果是7。

A. 对  
B. 错

**参考答案：**

B

1. 集合的元素可以是任意数据类型。

A. 对  
B. 错

**参考答案：**

B

1. a={}，type(a)结果是<class 'set'>。

A. 对  
B. 错

**参考答案：**

B

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. x = {1, 2, 3}
3. x.add(3)

print(x)

**参考答案：**

{1, 2, 3}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3}.symmetric\_difference({2, 3, 4}))

**参考答案：**

{1, 4}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print({1, 2, 3} < {5, 10, 3, 1, -111, 2})

**参考答案：**

True

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3}.intersection({2, 3, 4}))

**参考答案：**

{2, 3}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print({4, 1, 2, 3, 8}.issuperset({2, 3, 4}))

**参考答案：**

True

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3} | {5, 3, 4})

**参考答案：**

{1, 2, 3, 4, 5}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3, 4, 5} ^ {4, 5, 6, 7})

**参考答案：**

{1, 2, 3, 6, 7}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3} - {5, 3, 4})

**参考答案：**

{1, 2}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3}.difference({2, 3, 4}))

**参考答案：**

{1}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. students = {"peter", "john"}
3. students.add("john")
4. students.add("peterson")
5. students.remove("peter")

print(students)

**参考答案：**

{'peterson', 'john'}、{'john', 'peterson'}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print({1, 2, 3} == {1, 3, 2})

**参考答案：**

True

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. set1 = {1, 2, 3}
3. set2 = {6, 2}
4. set1.update(set2)

print(set1)

**参考答案：**

{1, 2, 3, 6}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3}.union({2, 3, 4}))

{1, 2, 3, 4}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。

print({1, 2, 3} & {5, 3, 4})

（2分）

**参考答案：**

{3}

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。

print({1, 2, 3}.issubset({2, 3, 4}))

**参考答案：**

False

1. 假设s = {1, 2}，2 \* s的计算结果是（        ）。（2分）

A. (1, 2, 1, 2)  
B. [1, 2, 1, 2]  
C. (1, 1, 2, 2)  
D. [1, 1, 2, 2]  
E. 非法的表达式

**参考答案：**

E

1. 在（        ）中，不能有重复的元素。（2分）

A. 列表（list）  
B. 元组（tuple）  
C. 集合（set）

**参考答案：**

C

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 5, 4, 13}，s1 & s2的计算结果是（        ）。（2分）

A. {2, 3, 5, 13}  
B. {1, 2, 4, 3, 5,13}  
C. {1, 4}  
D. {1, 5, 4, 13}  
E. {2, 3}

**参考答案：**

C

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 5, 4, 13}，s1 | s2的计算结果是（         ）。（2分）

A. {1, 2, 4, 3, 1, 5, 4, 13}  
B. {1, 2, 3, 4, 5,13}  
C. {1, 2, 4, 3}  
D. {1, 5, 4, 13}

**参考答案：**

B

1. 使用下列（        ）创建一个空集合。（2分）

A. {}  
B. ()  
C. []  
D. set()

**参考答案：**

D

1. 假设s = {1, 2, 4, 3}，调用s.remove(12)，将发生（         ）。

A. 集合对象没有remove方法  
B. 正常执行，没有异常发生  
C. 因为集合中12不存在，抛出KeyError异常  
D. 不能对一个集合删除元素

**参考答案：**

C

1. 下列（        ）语句生成集合{'a', 'b', 'c'}。（2分）

A. list("abac")  
B. tuple("abac")  
C. set("abac")  
D. 以上都不是

**参考答案：**

C

1. 下列（        ）是集合（set)。（2分）

A. [1, 2, 3]  
B. (1, 2, 3)  
C. {1, 2, 3}  
D. {}

**参考答案：**

C

1. 假设s1 = {1, 2, 4, 3}、s2 = {0, 1, 5, 3, 4, 2, 13}，下列（         ）的计算结果是True。（2分）

A. s2.issubset(s1)  
B. s1.issuperset(s2)  
C. s2.issubset(s1)  
D. s1.issubset(s2)

**参考答案：**

D

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 3, 4, 2}，下列（         ）的计算结果是True。（2分）

A. s1 == s2  
B. s1 != s2  
C. s1 > s2  
D. s1 < s2

**参考答案：**

A

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 5, 4, 13}，s1 ^ s2的计算结果是（         ）。（2分）

A. {2, 3, 5, 13}  
B. {4, 3, 5, 13}  
C. {1, 4}  
D. {1, 5, 4, 13}  
E. {2, 3}

**参考答案：**

A

1. 下列（        ）说法是错误的。（2分）

A. 列表（list）是可变对象  
B. 元组（tuple）是不可变对象  
C. 集合（set）是不可变对象  
D. 字典（dictionary）是可变对象

**参考答案：**

C

1. 假设s = {1, 2, 4, 3}，下列（        ）是错误的。（2分）

A. print(s[3])  
B. 3 in s  
C. print(max(s))  
D. print(len(s))  
E. print(sum(s))

**参考答案：**

A

1. 在（        ）中，元素不是按任何特定顺序存放的。（2分）

A. 列表（list）  
B. 元组（tuple）  
C. 集合（set）

**参考答案：**

C

1. 下列（        ）集合等于{1, 2, 3}。（2分）

A. {1, 2, 3}  
B. {2, 1, 3}  
C. {3, 2, 1}  
D. {2, 3, 1}  
E. {1, 2, 3, 2, 1, 3}  
F. 以上都是

**参考答案：**

F

1. 假设有集合s1和s2，s1 < s2的计算结果是（        ）。（2分）

A. 若len(s1)小于len(s2)，结果为True  
B. 若s1中的元素比较小于s2中的元素，结果为True  
C. 若s1是s2的真子集，结果为True  
D. 若s1是s2的真超集，结果为True  
E. 非法的表达式

**参考答案：**

C

1. 假设s = {1, 2, 5, 3}，下列（        ）的计算结果是4。（2分）

A. sum(s)  
B. len(s)  
C. min(s)  
D. max(s)  
E. None

**参考答案：**

B

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 5, 4, 13}，s1 + s2的计算结果是（         ）。（2分）

A. {1, 2, 4, 3, 1, 5, 4, 13}  
B. {1, 2, 4, 3, 5, 13}  
C. {1, 2, 4, 3}  
D. {1, 5, 4, 13}  
E. 非法的表达式

**参考答案：**

E

1. 假设s1 = {1, 2, 4, 3}、s2 = {1, 5, 4,13}，s1 - s2的计算结果是（         ）。（2分）

A. {2, 3, 5, 13}  
B. {1, 2, 4, 3, 5,13}  
C. {1, 2, 4, 3}  
D. {1, 5, 4, 13}  
E. {2, 3}

**参考答案：**

E

1. 假设s = {1, 2, 4, 3}，调用s.add(4)，将发生（         ）。（2分）

A. 集合对象没有add方法  
B. 正常执行，4被添加到集合中  
C. 因为集合中4已经存在，抛出KeyError异常  
D. 不能对一个集合添加新元素  
E. 正常执行，但4没有被添加到集合中，因为集合中4已经存在

**参考答案：**

E

1. 在面向对象程序设计中，函数和方法是完全一样的。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 类的构造方法是\_\_init\_\_()。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 通过对象不能调用类方法和静态方法。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 自定义类时，在方法前面使用@classmethod进行修饰，则该方法属于类方法。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在类的外部，没有任何办法可以访问对象的私有成员。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 属性可以像数据域一样进行访问，但赋值时具有方法的优点，可以对新值进行检查。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 自定义类时，运算符重载是通过重定义特殊方法来实现的。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 自定义类时，实例方法的第一个参数名称必须是self。  
   【来源】

《Python程序设计基础》第7章思考与练习。

A. 对  
B. 错

**参考答案：**

B

1. 自定义类时，在一个方法前面使用@staticmethod进行修饰，则该方法属于静态方法。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 自定义类时，实例方法的第一个参数名称不管是什么，都表示对象自身。  
   【来源】  
   《Python程序设计基础》第7章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. class Person:
3. def \_\_init\_\_(self, name, age):
4. self.name = name
5. self.age = age
6. def main():
7. p = Person("Peter", 18)
8. print(p.name + " is " + str(p.age) + " years old!")

main()

**参考答案：**

Peter is 18 years old!

1. 在Python中，自定义类时，与运算符"//"对应的特殊方法（魔法方法）名为\_\_\_\_\_\_\_\_。

**参考答案：**

\_\_floordiv\_\_()、\_\_floordiv\_\_

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. class A:
3. def \_\_init\_\_(self, count = 0):
4. self.\_\_count = count
5. a1 = A(2)
6. a2 = A(2)

print(id(a1) == id(a2))

【来源】  
《Python程序设计基础》第7章思考与练习。

**参考答案：**

False

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. class A:
3. def \_\_init\_\_(self):
4. self.count = 0
5. def increase(a, n):
6. a.count += 1
7. n += 1
8. def main():
9. a = A()
10. n = 0
11. for i in range(100):
12. increase(a, n)
13. print(a.count, n)

main()

【来源】  
《Python程序设计基础》第7章思考与练习。

**参考答案：**

100 0

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. class A:
3. def \_\_init\_\_(self, count = 0):
4. self.count = count
5. def modify(a, n):
6. a = A(8)
7. n = 5
8. def main():
9. a = A()
10. n = 1
11. modify(a, n)
12. print(a.count, n)

main()

【来源】  
《Python程序设计基础》第7章思考与练习。

（2分）

**参考答案：**

0 1

1. 若有以下程序：
2. 1 class Triangle:
3. 2 def \_\_init\_\_(base, height):
4. 3 self.base = base

4 self.height = height

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。

**参考答案：**

2

1. 若有以下程序：
2. 1 class Triangle:
3. 2 def \_\_init\_\_(self, base = 1, height):
4. 3 self.base = base

4 self.height = height

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。（2分）

**参考答案：**

2

1. 若有以下程序：
2. 1 class A:
3. 2 def \_\_init\_\_(self, x):
4. 3 self.x = x;
5. 4
6. 5 def main():
7. 6 a = A()
8. 7 print(a.x)
9. 8

9 main()

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。  
【来源】  
《Python程序设计基础》第7章思考与练习。

**参考答案：**

6

1. 结构化程序的基本构造单位是函数，而面向对象程序的基本构造单位是\_\_\_\_\_\_\_\_（注：填写中文）。

**参考答案：**

类

1. 若有以下程序：
2. 1 class A:
3. 2 def \_\_init\_\_(self, s):
4. 3 self.s = s
5. 4 def print(self):
6. 5 print(s)
7. 6
8. 7 a = A("Welcome")

8 a.print()

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。  
【来源】  
《Python程序设计基础》第7章思考与练习。

**参考答案：**

5

1. 目前主要有结构化程序设计和\_\_\_\_\_\_\_\_（注：填写中文）程序设计这两种重要的程序设计方法。

**参考答案：**

面向对象

1. 在Python中，自定义类时，与运算符"\*\*"对应的特殊方法（魔法方法）名为\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

\_\_pow\_\_()、\_\_pow\_\_

1. 若有以下程序：
2. 1 class A:
3. 2 def \_\_init\_\_(self, x):
4. 3 self.\_\_x = x;
5. 4
6. 5 def main():
7. 6 a = A(8)
8. 7 print(a.\_\_x)
9. 8

9 main()

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。  
【来源】  
《Python程序设计基础》第7章思考与练习。

**参考答案：**

7

1. Python使用\_\_\_\_\_\_\_\_（注：填写英文单词）关键字来定义类。  
   （2分）

**参考答案：**

class

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. class A:
3. def \_\_init\_\_(self, s = "Welcome"):
4. self.\_\_s = s
5. def print(self):
6. print(self.\_\_s)
7. def main():
8. a = A()
9. a.print()

main()

**参考答案：**

Welcome

1. 下列程序的输出结果是（        ）。
2. class Name:
3. def \_\_init\_\_(self, first\_name, mi, last\_name):
4. self.first\_name = first\_name
5. self.mi = mi
6. self.last\_name = last\_name
7. first\_name ="John"
8. name = Name(first\_name, 'F', "Smith")
9. first\_name ="Peter"
10. name.last\_name = "Pan"

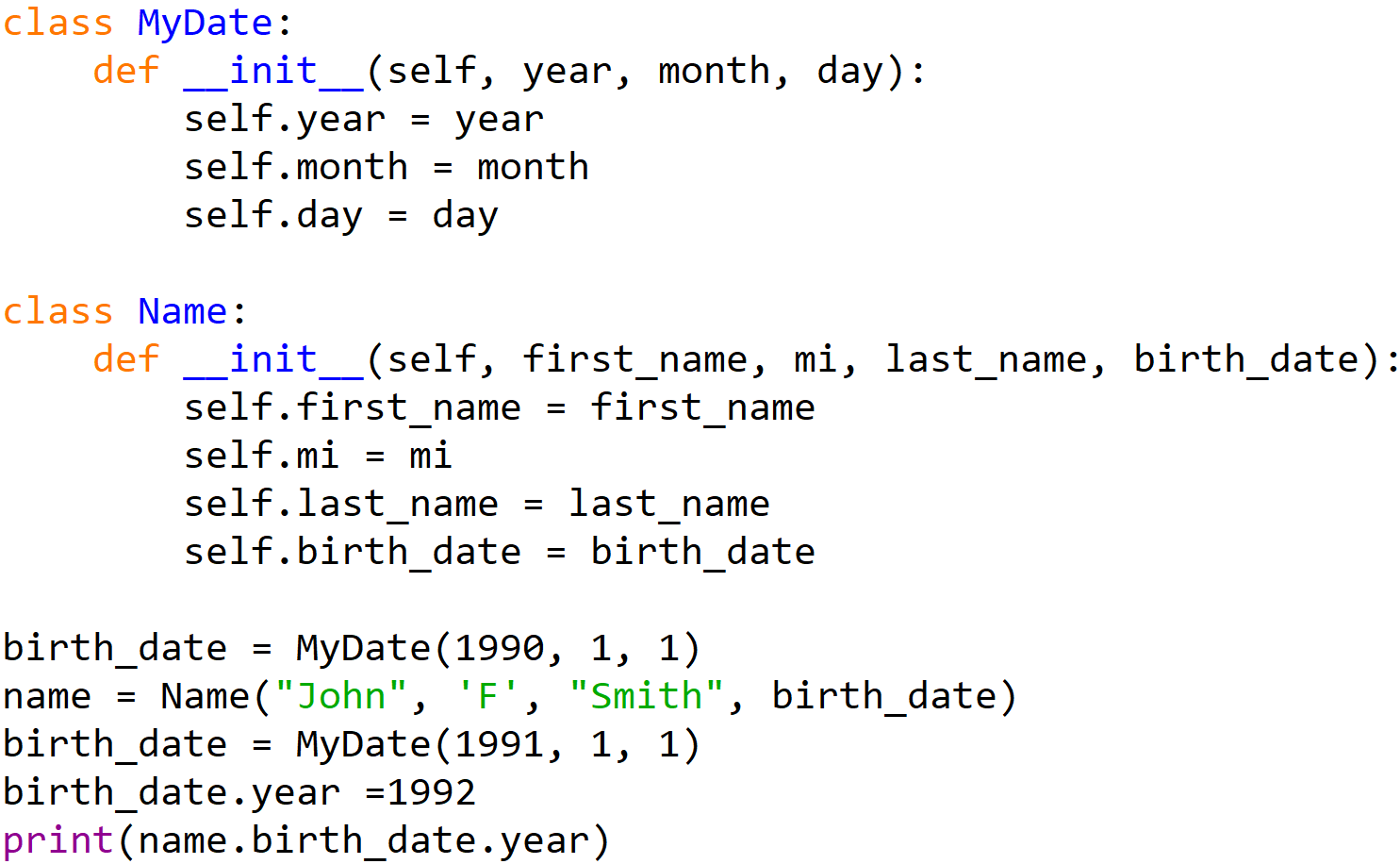
print(name.first\_name, name.last\_name)

A. Peter Pan  
B. John Pan  
C. Peter Smith  
D. John Smith

**参考答案：**

B

1. 下列程序的输出结果是（        ）。



（2分）

A. 1990  
B. 1991  
C. 1992  
D. 没有任何输出

**参考答案：**

A

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.x = 1
5. self.\_\_y = 1
6. def getY(self):
7. return self.\_\_y
8. a = A()

print(a.\_\_y)

（

）是正确的。

A. 程序有错误，因为x是私有的，不能在类的外部访问它  
B. 程序有错误，因为y是私有的，不能在类的外部访问它  
C. 程序有错误，因为不能使用\_\_y来命名变量  
D. 程序正常运行，输出1  
E. 程序正常运行，输出0

**参考答案：**

B

1. 对于Circle类，若定义对象：x = Circle();  
   下列（         ）说法是最准确的。

A. x包含一个整数值  
B. x包含一个Circle类型的对象  
C. x包含一个指向Circle对象的引用  
D. 可以给x赋一个整数值

**参考答案：**

C

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. \_\_a = 1
5. self.\_\_b = 1
6. self.\_\_c\_\_ = 1
7. \_\_d\_\_ = 1

# Other methods omitted

（

）是私有数据域。

A. \_\_a  
B. \_\_b  
C. \_\_c\_\_  
D. \_\_d\_\_

**参考答案：**

B

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.x = 1
5. self.\_\_y = 1
6. def get\_y(self):
7. return self.\_\_y
8. a = A()
9. a.\_\_y = 45

print(a.get\_y())

（

）是正确的。

A. 程序有错误，因为x是私有的，不能在类的外部访问它  
B. 程序有错误，因为y是私有的，不能在类的外部访问它  
C. 程序有错误，因为不能使用\_\_y来命名变量  
D. 程序正常运行，输出1  
E. 程序正常运行，输出45

**参考答案：**

B

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self, s = "Welcome"):
4. self.s = s
5. def print(self):
6. print(self.s)
7. a = A()

a.print()

（

）是正确的。

A. 程序有错误，因为类A没有构造方法  
B. 程序有错误，因为在print(self)中s未定义  
C. 程序正常运行，但没有任何输出  
D. 程序有错误，因为调用了未定义的无参构造方法  
E. 程序正常运行，输出Welcome

**参考答案：**

E

1. 下列程序的输出结果是（         ）。
2. class Count:
3. def \_\_init\_\_(self,count = 0):
4. self.\_\_count =count
5. c1 = Count(2)
6. c2 = Count(2)
7. print(id(c1) == id(c2), end = " ")
8. s1 ="Good"
9. s2 ="Good"

print(id(s1) == id(s2))

（2分）

A. True False  
B. True True  
C. False True  
D. False False

**参考答案：**

C

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.x = 1
5. self.\_\_y = 1
6. def getY(self):
7. return self.\_\_y
8. a = A()

print(a.x)

（

）是正确的。

A. 程序有错误，因为x是私有的，不能在类的外部访问它  
B. 程序有错误，因为y是私有的，不能在类的外部访问它  
C. 程序有错误，因为不能使用\_\_y来命名变量  
D. 程序正常运行，输出1  
E. 程序正常运行，输出0

**参考答案：**

D

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_init\_\_(self, i = 2, j = 3):
4. self.i = i
5. self.j = j
6. def \_\_str\_\_(self):
7. return "A"
8. def \_\_eq\_\_(self, other):
9. return self.i \* self.j == other.i \* other.j
10. def main():
11. x = A(1, 2)
12. y = A(2, 1)
13. print(x == y)

main()

（2分）

A. True  
B. False  
C. 2  
D. 1

**参考答案：**

A

1. 定义一个类时，关键字（        ）是必需的。

A. def  
B. return  
C. class  
D. 以上都需要

**参考答案：**

C

1. 一个对象是一个（        ）的实例。

（2分）

A. 程序  
B. 类  
C. 方法  
D. 数据

**参考答案：**

B

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self, s):
4. self.s = s
5. def print(self):
6. print(s)
7. a =A("Welcome")

a.print()

（

）是正确的。

A. 程序有错误，因为类A没有构造方法  
B. 程序有错误，因为类A应该有一个签名为print(self, s)的print方法  
C. 程序有错误，因为类A应该有一个签名为print(s)的print方法  
D. 若将print(s)改为print(self.s)，程序将正常运行

**参考答案：**

D

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self, s):
4. self.s = s
5. def print(self):
6. print(self.s)
7. a = A()

a.print()

（

）是正确的。

（2分）

A. 程序有错误，因为类A没有构造方法  
B. 程序有错误，因为在print(self)中s未定义  
C. 程序正常运行，但没有任何输出  
D. 程序有错误，因为调用了未定义的无参构造方法

**参考答案：**

D

1. （        ）可以表示现实世界中能被明确标识的一个实体。

A. 一个类  
B. 一个对象  
C. 一个方法  
D. 一个数据域

**参考答案：**

B

1. 假设x的值为2、y的值为4，x + y与（         ）是等价的。

（2分）

A. x.\_\_add(y)  
B. x.\_\_add\_\_(y)  
C. x.\_\_Add(y)  
D. x.\_\_ADD(y)

**参考答案：**

B

1. 面向对象程序设计把数据和对数据的处理过程作为一个相互依存、不可分割的整体来看待，将数据和对数据的处理过程组织在一起的机制被称为（        ）。（2分）

A. 继承  
B. 封装  
C. 多态  
D. 抽象

**参考答案：**

B

1. （        ）是定义具有相同类型对象的一个模板、蓝图或合同。

A. 一个类  
B. 一个对象  
C. 一个方法  
D. 一个数据域

**参考答案：**

A

1. 创建对象时，会自动调用（        ）。

（2分）

A. 构造方法  
B. 类  
C. 带返回值的方法  
D. 无返回值的方法

**参考答案：**

A

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.x = 1
5. self.\_\_y = 1
6. def getY(self):
7. return self.\_\_y
8. a = A()
9. a.x = 45

print(a.x)

（

）是正确的。

A. 程序有错误，因为x是私有的，不能在类的外部访问它  
B. 程序有错误，因为y是私有的，不能在类的外部访问它  
C. 程序有错误，因为不能使用\_\_y来命名变量  
D. 程序正常运行，输出1  
E. 程序正常运行，输出45

**参考答案：**

E

1. mro表示类继承的顺序。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. python是一种多态语言，崇尚鸭子类型。在鸭子类型中，并不关心对象是什么类型，只关心对象的行为。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 子类是父类的子集。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在子类中可以覆盖其父类中定义的非私有方法。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 创建子类对象时，会自动调用其父类中定义的\_\_init\_\_方法。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在子类中可以覆盖其父类中定义的\_\_init\_\_方法。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 如果定义一个类时没有指定继承哪个类，则默认继承object类。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 所有类都派生自object类。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 在子类中可以覆盖其父类中定义的私有方法。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 在子类中可以通过“父类名.方法名”的方式显式调用父类中的方法。  
   【来源】  
   《Python程序设计基础》第8章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 面向对象程序设计的三大特性是封装、\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_（注：填写中文，注意顺序）。（2分）

**参考答案：**

继承，多态

1. Python使用\_\_\_\_\_\_\_\_（注：填写英文）表标记继承层次中查找父类的顺序。（2分）

**参考答案：**

mro

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. class Shape:
3. def \_\_init\_\_(self, width = 1, height = 1):
4. self.width = width
5. self.height = height
6. def setWidth(self, width):
7. self.width = width
8. def setHeight(self, height):
9. self.height = height
10. class Rectangle(Shape):
11. def name(self):
12. return "rectangle"
13. def area(self):
14. return self.width \* self.height
15. def main():
16. r = Rectangle(5)
17. r.setHeight(6)
18. print("The {0} has area {1:.2f}.".format(r.name(), r.area()))

main()

**参考答案：**

The rectangle has area 30.00.

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格）。
2. class Animal:
3. def msg(self):
4. return "Can Move."
5. class Vertebrate:
6. def msg(self):
7. return "Has a backbone."
8. def main():
9. creature = Vertebrate()
10. print(creature.msg(), isinstance(creature, Animal))

main()

**参考答案：**

Has abackbone. False

1. 表达式isinstance(3, object)的值为\_\_\_\_\_\_\_\_。（2分）

**参考答案：**

True

1. 如果自定义一个类时没有指定继承哪个类，则默认继承\_\_\_\_\_\_\_\_（注：填写英文单词）类。

**参考答案：**

object

1. 通过\_\_\_\_\_\_\_\_（注：填写中文）将数据和对数据的处理过程封装为一个有机的整体。

**参考答案：**

类

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. class Animal:
3. def \_\_init\_\_(self, feature = "I can move."):
4. self.\_\_feature = feature
5. def\_\_str\_\_(self):
6. return self.\_\_feature
7. class Vertebrate(Animal):
8. def \_\_init\_\_(self, feature = "I have a backbone."):
9. self.\_\_feature = feature
10. def \_\_str\_\_(self):
11. return self.\_\_feature
12. def main():
13. creature = Vertebrate()
14. print(creature)

main()

**参考答案：**

I have a backbone.

1. 子类（派生类）和父类（基类）之间必须存在\_\_\_\_\_\_\_\_（注：填写英文）关系。

**参考答案：**

is-a、is a

1. 为了覆盖父类（基类）中的方法，子类（派生类）中的方法必须与父类（基类）中的方法具有相同的\_\_\_\_\_\_\_\_（注：填写中文）。

**参考答案：**

方法头

1. 下面程序的下画线处为调用B的父类的构造方法，正确的语句是\_\_\_\_\_\_\_\_（注意：不要有任何多余的空格） 。
2. class A:
3. def \_\_init\_\_(self, i = 1):
4. self.i = i
5. class B(A):
6. def \_\_init\_\_(self, j = 2):
7. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. self.j = j
9. def main():
10. b = B()
11. print(b.i, b.j)

main()

**参考答案：**

super().\_\_init\_\_()、A.\_\_init\_\_(self)

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. class A:
3. def \_\_init\_\_(self, x = 0):
4. self.\_\_x = x
5. def show(self):
6. print(self.\_\_x, end = '#')
7. class B(A):
8. def \_\_init\_\_(self, x, y):
9. super().\_\_init\_\_(x)
10. self.\_\_y = y
11. def show(self):
12. print(self.\_\_y)
13. def display\_object(obj):
14. obj.show()
15. a = A(10)
16. b = B(20, 30)
17. display\_object(a)

display\_object(b)

【来源】  
《Python程序设计基础》第8章思考与练习。

**参考答案：**

10#30

1. 一个子类（派生类）只有一个直接父类（基类）的情况称为\_\_\_\_\_\_\_\_（注：填写中文）。（2分）

**参考答案：**

单继承

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. class Animal:
3. def msg(self):
4. return "Can Move."
5. class Vertebrate(Animal):
6. def msg(self):
7. return "Has a backbone."
8. def main():
9. creature = Vertebrate()
10. print(creature.msg(), isinstance(creature, Animal))

main()

**参考答案：**

Has a backbone. True

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. class A:
3. def \_\_init\_\_(self):
4. print('A', end = '#')
5. class B(A):
6. def \_\_init\_\_(self, x):
7. super().\_\_init\_\_()
8. self.\_\_x = x
9. print('B')

b = B(3)

【来源】  
《Python程序设计基础》第8章思考与练习。  
**参考答案：**

A#B

1. 下列程序的输出结果为（         ） 。
2. class A:
3. def \_\_new\_\_(cls):
4. print("A's \_\_new\_\_() invoked")
5. def \_\_init\_\_(self):
6. print("A's \_\_init\_\_() invoked")
7. class B(A):
8. def \_\_new\_\_(cls):
9. print("B's \_\_new\_\_() invoked")
10. def \_\_init\_\_(self):
11. print("B's \_\_init\_\_() invoked")
12. def main():
13. b = B()
14. a = A()
15. main()

A. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_init\_\_() invoked  
B. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_new\_\_() invoked  
C. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_init\_\_() invoked，最后输出A's\_\_new\_\_() invoked  
D. 先输出A's \_\_init\_\_() invoked，接着输出A's \_\_new\_\_() invoked

**参考答案：**

B

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.set\_i(20)
5. print("i from A is", self.i)
6. def set\_i(self, i):
7. self.i = 2 \* i;
8. class B(A):
9. def \_\_init\_\_(self):
10. super().\_\_init\_\_()
11. def set\_i(self, i):
12. self.i = 3 \* i;

b = B()

（         ）是正确的。

A. 类A的\_\_init\_\_方法没有被调用  
B. 类A的\_\_init\_\_方法被调用，输出i from A is 0  
C. 类A的\_\_init\_\_方法被调用，输出i from A is 40  
D. 类A的\_\_init\_\_方法被调用，输出i from A is 60

**参考答案：**

D

1. 下列程序的输出结果为（         ）。
2. class ParentClass:
3. def \_\_init\_\_(self):
4. self.\_\_x = 1
5. self.y = 10
6. def print(self):
7. print(self.\_\_x, self.y)
8. class ChildClass(ParentClass):
9. def \_\_init\_\_(self):
10. super().\_\_init\_\_()
11. self.\_\_x = 2
12. self.y = 20
14. c = ChildClass()

c.print()

A. 1 10  
B. 1 20  
C. 2 10  
D. 2 20

**参考答案：**

B

1. 给定如下类及其对象：
2. class C1:
3. pass
4. class C2(C1):
5. pass
6. class C3(C2):
7. pass
8. class C4(C1):
9. pass
10. c1 = C1()
11. c2 = C2()
12. c3 = C3()

c4 = C4()

下列表达式计算结果为假（False）的是（    ）。

A. isinstance(c1, C1)  
B. isinstance(c2, C1)  
C. isinstance(c3, C1)  
D. isinstance(c4, C2)

**参考答案：**

D

1. 面向对象程序设计允许从一个已经存在的类派生出一个新类，这被称为（         ）。

A. 封装  
B. 继承  
C. 抽象  
D. 多态

**参考答案：**

B

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_init\_\_(self, i = 1):
4. self.i = i
5. class B(A):
6. def \_\_init\_\_(self, j = 2):
7. super().\_\_init\_\_()
8. self.j = j
9. def main():
10. b = B()
11. print(b.i, b.j)

main()

A. 0 0  
B. 0 1  
C. 1 2  
D. 0 2  
E. 2 1

**参考答案：**

C

1. 类的声明如下：
2. class A：

pass

则类A的父类是（    ）。

A. 没有父类  
B. 本身  
C. object  
D. 以上都不对

**参考答案：**

C

1. 下列说法，错误的是（        ）。

A. 子类是父类的一个子集  
B. 不可以覆盖父类中定义的私有方法  
C. 可以覆盖父类中定义的非私有方法  
D. 可以覆盖父类中定义的构造（\_\_init\_\_）方法

**参考答案：**

A

1. 在Python中，下面说法正确的是（    ）。

（2分）

A. 一个子类可以有多个父类，一个父类也可以有多个子类  
B. 一个子类可以有多个父类，但一个父类只可以有一个子类  
C. 一个子类只可以有一个父类，但一个父类可以有多个子类  
D. 上述说法都不对

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_new\_\_(cls):
4. print("A's \_\_new\_\_() invoked")
5. A.\_\_init\_\_(cls)
6. def \_\_init\_\_(self):
7. print("A's \_\_init\_\_() invoked")
8. class B(A):
9. def \_\_new\_\_(cls):
10. print("B's \_\_new\_\_() invoked")
11. def \_\_init\_\_(self):
12. print("B's \_\_init\_\_() invoked")
13. def main():
14. b = B()
15. a = A()

main()

A. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_init\_\_() invoked  
B. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_new\_\_() invoked  
C. 先输出B's \_\_new\_\_() invoked，接着输出A's\_\_new\_\_() invoked，最后输出A's \_\_init\_\_() invoked  
D. 先输出A's \_\_init\_\_() invoked，接着输出A's \_\_new\_\_() invoked

**参考答案：**

C

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self):
4. self.set\_i(20)
5. def set\_i(self, i):
6. self.i = 2 \* i
7. class B(A):
8. def \_\_init\_\_(self):
9. super().\_\_init\_\_()
10. print("i from B is", self.i)
11. def set\_i(self, i):
12. self.i = 3 \* i

b = B()

（         ）是正确的。

A. 类A的\_\_init\_\_方法没有被调用  
B. 类A的\_\_init\_\_方法被调用，输出i from B is 0  
C. 类A的\_\_init\_\_方法被调用，输出i from B is 40  
D. 类A的\_\_init\_\_方法被调用，输出i from B is 60

**参考答案：**

D

1. 下列程序的输出结果为（         ）。
2. class Person:
3. def \_\_get\_info(self):
4. return "Person's get\_info is called"
5. def print\_person(self):
6. print(self.\_\_get\_info(), end = ' ')
7. class Student(Person):
8. def \_\_get\_info(self):
9. return "Student's get\_info is called"
10. def main():
11. Person().print\_person()
12. Student().print\_person()

main()

A. 先输出Person's get\_info is called，接着输出Person's get\_info is called  
B. 先输出Student's get\_info is called，接着输出Person's get\_info is called  
C. 先输出Student's get\_info is called，接着输出Student's get\_info is called  
D. 先输出Person's get\_info is called，接着输出Student's get\_info is called

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. class Person:
3. def get\_info(self):
4. return "Person's get\_info is called"
5. def print\_person(self):
6. print(self.get\_info(), end = ' ')
7. class Student(Person):
8. def get\_info(self):
9. return "Student's get\_info is called"
10. def main():
11. Person().print\_person()
12. Student().print\_person()

main()

A. 先输出Person's get\_info is called，接着输出Person's get\_info is called  
B. 先输出Student's get\_info is called，接着输出Person's get\_info is called  
C. 先输出Student's get\_info is called，接着输出Student's get\_info is called  
D. 先输出Person's get\_info is called，接着输出Student's get\_info is called

**参考答案：**

D

1. 对于如下程序：
2. class A:
3. def \_\_init\_\_(self, i = 0):
4. self.i = i
5. class B(A):
6. def \_\_init\_\_(self, j = 0):
7. self.j = j
8. def main():
9. b = B()
10. print(b.i)
11. print(b.j)

main()

（         ）是正确的。

A. 类B是类A的子类，但类A的数据域i未被类B继承  
B. 类B是类A的子类，自动继承了类A的所有数据域  
C. 当创建一个类B的对象时，必须传递一个整数，例如B(5)  
D. 对象b不能访问数据域j

**参考答案：**

A

1. 下列程序的输出结果为（         ） 。
2. class A:
3. def \_\_new\_\_(cls):
4. A.\_\_init\_\_(cls)
5. print("A's \_\_new\_\_() invoked")
6. def \_\_init\_\_(self):
7. print("A's \_\_init\_\_() invoked")
8. class B(A):
9. def \_\_new\_\_(cls):
10. B.\_\_init\_\_(cls)
11. print("B's \_\_new\_\_() invoked")
12. def \_\_init\_\_(self):
13. print("B's \_\_init\_\_() invoked")
14. def main():
15. b = B()
16. a = A()

main()

A. 先输出B's \_\_init\_\_() invoked，接着输出A's \_\_init\_\_() invoked  
B. 先输出B's \_\_new\_\_() invoked，接着输出A's \_\_new\_\_() invoked  
C. 先输出B's \_\_new\_\_() invoked，接着输出B's\_\_init\_\_() invoked，然后输出A's \_\_new\_\_() invoked，最后输出A's \_\_init\_\_() invoked  
D. 先输出B's \_\_init\_\_() invoked，接着输出B's \_\_new\_\_() invoked，然后输出A's\_\_init\_\_() invoked，最后输出A's \_\_new\_\_() invoked

**参考答案：**

D

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_str\_\_(self):
4. return "A"
5. class B(A):
6. def \_\_str\_\_(self):
7. return "B"
8. class C(B):
9. def \_\_str\_\_(self):
10. return "C"
11. def main():
12. b = B()
13. a = A()
14. c = C()
15. print(a, b, c)

main()

A. C C C  
B. A B C  
C. A A A  
D. B B B

**参考答案：**

B

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_str\_\_(self):
4. return "A"
5. class B(A):
6. def \_\_init\_\_(self):
7. super().\_\_init\_\_()
8. class C(B):
9. def \_\_init\_\_(self):
10. super().\_\_init\_\_()
11. def main():
12. b = B()
13. a = A()
14. c = C()
15. print(a, b, c)

main()

A. C C C  
B. A B C  
C. A A A  
D. B B B

**参考答案：**

C

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_init\_\_(self):
4. self.i = 1
5. def m(self):
6. self.i = 10
7. class B(A):
8. def m(self):
9. self.i += 1
10. return self.i
11. def main():
12. b = B()
13. print(b.m())

main()

A. 1  
B. 2  
C. 10  
D. b不能访问i

**参考答案：**

B

1. 检查一个对象o是否是类A的一个实例，使用（         ）。

A. o.isinstance(A)  
B. A.isinstance(o)  
C. isinstance(o, A)  
D. isinstance(A, o)

**参考答案：**

C

1. 下列程序的输出结果为（         ）。
2. class A:
3. def \_\_init\_\_(self, i = 0):
4. self.i = i
5. def m1(self):
6. self.i += 1
7. class B(A):
8. def \_\_init\_\_(self, j = 0):
9. A.\_\_init\_\_(self, 3)
10. self.j = j
11. def m1(self):
12. self.j += 1
13. def main():
14. b = B()
15. b.m1()
16. print(b.i, b.j)

main()

A. 2 0  
B. 3 1  
C. 4 0  
D. 3 0  
E. 4 1

**参考答案：**

B

1. 若try语句中的代码引发了异常，则会执行else语句中的代码。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. try语句可以有一个或多个finally语句。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 断言语句assert通常在程序调试阶段使用，辅助调试程序。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 通常情况下，用户自定义异常类都应该派生自Exception类。（1分）

A. 对  
B. 错

**参考答案：**

A

1. try语句可以有一个或多个except语句。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 异常是程序运行时错误。严格说来，语法错误和逻辑错误不属于异常。  
   （1分）

A. 对  
B. 错

**参考答案：**

A

1. 异常处理结构也不是万能的，处理异常的代码也有引发异常的可能。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 不论是否发生异常，finally语句中的代码总是会执行的。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 异常处理将错误检测和错误处理分隔开来，使程序复杂化。程序中异常处理结构在绝大多数情况下是没必要的。

（1分）

A. 对  
B. 错

**参考答案：**

B

1. 内置异常类的父类是Exception。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。

（1分）

A. 对  
B. 错

**参考答案：**

B

1. 若输入-10，下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. try:
3. temperature= eval(input("Enter a temperature: "))
4. if temperature > 40:
5. raise RuntimeError("It is too hot")
6. if temperature < 0:
7. raise RuntimeError("It is too cold")
8. except RuntimeError as ex:
9. print(ex)
10. except:
11. print("Other errors")
12. else:

print(temperature)

**参考答案：**

It is too cold

1. 假设下面try-except块中的statement2语句出现一个异常：
2. try:
3. statement1
4. statement2
5. statement3
6. except Exception1:
7. # Handle exception 1
8. except Exception2:
9. # Handle exception 2
10. except Exception3:
11. # Handle exception 3
12. finally:
13. statement4

statement5

如果异常类型是Exception3，那么statement4和statement5会被执行吗？\_\_\_\_\_\_\_\_ （注意：只能填写"是"或"否"）。（2分）

**参考答案：**

是

1. 若输入50，下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. try:
3. temperature= eval(input("Enter a temperature: "))
4. if temperature > 40:
5. raise RuntimeError("It is too hot")
6. if temperature < 0:
7. raise RuntimeError("It is too cold")
8. except RuntimeError as ex:
9. print(ex)
10. except:
11. print("Other errors")
12. else:

print(temperature)

**参考答案：**

It is too hot

1. 使用math.sqrt(x)数学函数求x的平方根，若x为负数，将产生\_\_\_\_\_\_\_\_异常。**参考答案：**

ValueError

1. 假设下面try-except块中的statement2语句出现一个异常：
2. try:
3. statement1
4. statement2
5. statement3
6. except Exception1:
7. # Handle exception 1
8. except Exception2:
9. # Handle exception 2
10. except Exception3:
11. # Handle exception 3
12. finally:
13. statement4

statement5

如果异常未被捕获，statement5会被执行吗？\_\_\_\_\_\_\_\_ （注意：只能填写"是"或"否"）。

**参考答案：**

否

1. 假设list1= ['a', 'b', 'c']，执行list1[3]，将抛出\_\_\_\_\_\_\_\_异常。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（2分）

**参考答案：**

IndexError

1. 若输入t，下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. try:
3. temperature= eval(input("Enter a temperature: "))
4. if temperature > 40:
5. raise RuntimeError("It is too hot")
6. if temperature < 0:
7. raise RuntimeError("It is too cold")
8. except RuntimeError as ex:
9. print(ex)
10. except:
11. print("Other errors")
12. else:

print(temperature)

**参考答案：**

Other errors

1. Python内置异常类的父（基）类是\_\_\_\_\_\_\_\_（注：填写英文）。

**参考答案：**

BaseException

1. 对于下面try-except块：
2. try:
3. # some code here
4. except ArithmeticError:
5. print("ArithmeticError")
6. except ZeroDivisionError:

print("ZeroDivisionError")

这样的写法是否正确？\_\_\_\_\_\_\_\_ （注意：只能填写"是"或"否"）。

**参考答案：**

否

1. 执行123 + "abc"，将抛出\_\_\_\_\_\_\_\_异常。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。

**参考答案：**

TypeError

1. 假设dict1= {0: 'female', 1: 'male'}，执行dict1[2]，将抛出\_\_\_\_\_\_\_\_异常。  
   【来源】  
   《Python程序设计基础》第9章思考与练习。（2分）

**参考答案：**

KeyError

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. def main():
3. try:
4. value = 30
5. if value < 40:
6. raise Exception("value is too small")
7. except Exception as ex:
8. print(ex)

main()

**参考答案：**

value is too small

1. 假设下面try-except块中的statement2语句出现一个异常：
2. try:
3. statement1
4. statement2
5. statement3
6. except Exception1:
7. # Handle exception 1
8. except Exception2:
9. # Handle exception 2

statement4

statement3会被执行吗？\_\_\_\_\_\_\_\_ （注意：只能填写"是"或"否"）。

**参考答案：**

否

1. 若输入10，下面程序的输出结果为\_\_\_\_\_\_\_\_。
2. try:
3. temperature= eval(input("Enter a temperature: "))
4. if temperature > 40:
5. raise RuntimeError("It is too hot")
6. if temperature < 0:
7. raise RuntimeError("It is too cold")
8. except RuntimeError as ex:
9. print(ex)
10. except:
11. print("Other errors")
12. else:

print(temperature)

**参考答案：**

10

1. 假设下面try-except块中的statement2语句出现一个异常：
2. try:
3. statement1
4. statement2
5. statement3
6. except Exception1:
7. # Handle exception 1
8. except Exception2:
9. # Handle exception 2

statement4

如果异常被捕获，statement4会被执行吗？\_\_\_\_\_\_\_\_ （注意：只能填写"是"或"否"）。

**参考答案：**

是

1. 下列程序执行时，将发生（         ）异常。
2. a = 12
3. b= 12 - 24

my\_result= a / (a + b)

（2分）

A. ValueError  
B. NameError  
C. SyntaxError  
D. ZeroDivisionError  
E. OverflowError

**参考答案：**

D

1. 下列程序的输出结果为（         ）。
2. def main():
3. try:
4. f()
5. print("After the functioncall")
6. except ZeroDivisionError:
7. print("Divided by zero!")
8. except:
9. print("Exception")
10. def f():
11. print(1 / 0)

main()

A. 先输出"After the function call"，接着输出"Divided by zero!"  
B. "After the function call"  
C. "Divided by zero!"  
D. 先输出"Divided by zero!"，接着输出"Exception"

**参考答案：**

C

1. 下列程序的输出结果为（         ）。
2. try:
3. list = 10 \* [0]
4. x = list[9]
5. print("Done")
6. except IndexError:
7. print("Index out of bound")
8. else:
9. print("Nothing is wrong")
10. finally:

print("Finally we are here")

A. 先输出"Done"，接着输出"Nothing is wrong"  
B. 先输出"Done"，接着输出"Nothing is wrong"，最后输出"Finally we are here"  
C. 先输出"Index out of bound"，接着输出"Nothing is wrong"，最后输出"Finally we are here"  
D. 先输出"Nothing is wrong"，接着输出"Finally we are here"

**参考答案：**

B

1. 下列程序执行时，将发生（         ）异常。

int("3.4")

A. ValueError  
B. NameError  
C. SyntaxError  
D. ZeroDivisionError  
E. OverflowError

**参考答案：**

A

1. 下列程序的输出结果为（         ）。
2. try:
3. list = 10 \* [0]
4. x = list[10]
5. print("Done")
6. except IndexError:
7. print("Index out of bound")
8. else:
9. print("Nothing is wrong")
10. finally:

print("Finally we are here")

A. 先输出"Done"，接着输出"Nothing is wrong"  
B. 先输出"Done"，接着输出"Nothing is wrong"，最后输出"Finally we are here"  
C. 先输出"Index out of bound"，接着输出"Nothing is wrong"，最后输出"Finally we are here"  
D. 先输出"Nothing is wrong"，接着输出"Finally we are here"  
E. 先输出"Index out of bound"，接着输出"Finally we are here"

**参考答案：**

E

1. 下列程序执行时，将发生（         ）异常。
2. a = 30
3. b = 74353634.89

my\_result= a \*\* b

A. ValueError  
B. NameError  
C. SyntaxError  
D. ZeroDivisionError  
E. OverflowError

**参考答案：**

E

1. 下列程序的输出结果为（         ）。
2. try:
3. list = 5 \* [0]
4. x = list[5]
5. print("Done")
6. except IndexError:

print("Index out of bound")

A. 先输出"Done"，接着输出"Index out of bound"  
B. "Index out of bound"  
C. "Done"  
D. 没有输出

**参考答案：**

B

1. 下列程序执行时，将发生（         ）异常。
2. a = 12
3. b = 13

my\_result = (a + b / 2a)

A. ValueError  
B. NameError  
C. SyntaxError  
D. ZeroDivisionError  
E. OverflowError

**参考答案：**

C

1. 下列程序的输出结果为（         ）。
2. def main():
3. try:
4. f()
5. print("After the function call")
6. except IndexError:
7. print("Index out of bound")
8. except:
9. print("Exception in main")
10. def f():
11. try:
12. s = "abc"
13. print(s[3])
14. except ZeroDivisionError:
15. print("Divided by zero!")

main()

A. 先输出"Divided by zero!"，接着输出"After the function call"  
B. "After the functioncall"  
C. "Index out of bound"  
D. 先输出"Divided by zero!"，接着输出"Exception in main"

**参考答案：**

C

1. 下列程序执行时，将发生（         ）异常。

my\_result = a \* 8

A. ValueError  
B. NameError  
C. SyntaxError  
D. ZeroDivisionError  
E. OverflowError

**参考答案：**

B

1. 文件对象的seek()方法用来返回文件位置指示器（文件指针）的当前位置。（1分）

A. 对  
B. 错

**参考答案：**

B

1. Python第三方扩展库openpyxl支持Excel文件的读写操作。

（1分）

A. 对  
B. 错

**参考答案：**

A

1. 序列化指的是从字节流中提取出对象的过程。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 打开文件的含义是指将文件对象与外存上的文件建立联系。成功打开文件后，对该文件的操作都将通过文件对象来完成。

A. 对  
B. 错

**参考答案：**

A

1. 内存文件不是存放在外存上的真正文件，而是存放在内存中的虚拟文件。（1分）

A. 对  
B. 错

**参考答案：**

A

1. CSV文件是纯文本文件。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 好的习惯，文件使用完毕后必须显式关闭文件。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 文件默认以二进制文件模式打开。打开文本文件要显式指定文本文件模式。

A. 对  
B. 错

**参考答案：**

B

1. 为了防止文件中已存在的数据被意外清除，在打开文件前可以检测该文件是否存在。使用os.path模块中的isdir方法判断一个文件是否存在。（1分）

A. 对  
B. 错

**参考答案：**

B

1. 以写模式打开的文件无法进行读操作。（1分）

A. 对  
B. 错

**参考答案：**

A

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. f = open("temp.txt","w+")
3. f.write("Lux et Veritas")
4. f.seek(7)
5. s = f.read(3)
6. f.close()

print(s)

**参考答案：**

Ver

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. def main():
3. lst = ["Hello", "Aloha\n"]
4. outfile = open("temp.txt", "w")
5. outfile.writelines(lst)
6. outfile.close()
7. infile = open("temp.txt", "r")
8. text = infile.read().rstrip()
9. infile.close()
10. print(text)

main()

**参考答案：**

HelloAloha

1. 文件使用完毕后必须关闭文件。\_\_\_\_\_\_\_\_（注：填写英文）方法用来显式关闭文件。（2分）

**参考答案：**

close()、close

1. 若有以下程序：
2. 1 f = open("temp.txt","w+")
3. 2 f.write("Lux et Veritas")

3 f.seek(4,1)

程序中存在的错误在第\_\_\_\_\_\_\_\_行（注：填写阿拉伯数字）。（2分）

**参考答案：**

3

1. Python内置函数\_\_\_\_\_\_\_\_（注：填写英文）用来打开或创建文件并返回文件对象。（2分）

**参考答案：**

open()、open

1. 下面程序执行后，文件temp.txt中内容是\_\_\_\_\_\_\_\_。
2. def main():
3. fun("temp.txt", "amazing")
4. fun("temp.txt", "awesome")
5. def fun(filename,s):
6. outfile = open(filename, "w")
7. outfile.write(s)
8. outfile.close()

main()

**参考答案：**

awesome

1. 通常有两种类型的文件：文本文件和\_\_\_\_\_\_\_\_（注：填写中文）。（2分）

**参考答案：**

二进制文件

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. def main():
3. x = 20
4. y = 30
5. outfile = open("temp.txt", "w")
6. outfile.write(str(x) + " " +str(y))
7. outfile.close()
8. infile = open("temp.txt","r")
9. s = infile.read()
10. numbers = [eval(value) for value in s.split()]
11. for number in numbers:
12. print(number, end='#')
13. infile.close()

main()

**参考答案：**

20#30#

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. def main():
3. outfile = open("temp.txt", "w")
4. outfile.write("Hello\n")
5. outfile.write("Aloha\n")
6. outfile.close()
7. infile = open("temp.txt", "r")
8. text = ""
9. for line in infile:
10. text += line.rstrip()
11. infile.close()
12. print(text)

main()

**参考答案：**

HelloAloha

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ （注意：不要有任何多余的空格）。
2. f = open("temp.txt", "w")
3. f.write("Lux et Veritas")
4. f.close
5. f = open("temp.txt", "rb")
6. f.read(3)
7. f.seek(4,1)

print(f.tell())

**参考答案：**

7

1. 使用上下文管理\_\_\_\_\_\_\_\_（注：填写英文单词）语句可以自动管理文件对象，不论何种原因，都能保证文件被正确关闭。（2分）

**参考答案：**

with

1. Python提供了一个专门处理CSV文件的\_\_\_\_\_\_\_\_（注：填写英文）模块。

**参考答案：**

csv

1. 下面程序的输出结果为\_\_\_\_\_\_\_\_ 。
2. def main():
3. iofile = open("temp.txt", "w+")
4. for i in range(1, 10):
5. iofile.write(str(i))
6. iofile.seek(7)
7. print(iofile.read(1))
8. iofile.close()

main()

**参考答案：**

8

1. 对文件进行写入操作之后，\_\_\_\_\_\_\_\_（注：填写英文）方法用来在不关闭文件对象的情况下强制将缓冲区内容写入文件。（2分）

**参考答案：**

flush()、flush

1. 解决数据永久性保存的有效方式是使用\_\_\_\_\_\_\_\_（注：填写中文）。

**参考答案：**

文件

1. readlines()方法返回（         ）。

（2分）

A. 一个字符串  
B. 一个字符串（行）列表  
C. 一个浮点数列表  
D. 一个整数列表

**参考答案：**

B

1. 打开文件c:\scores.dat进行二进制读操作，使用（         ）。

A. infile = open("c:\\scores.dat", "rb")  
B. infile = open("c:\\scores.dat", "r")  
C. infile = open("c:\scores.dat", "wrb")  
D. infile =open("c:\\scores.dat", "r")

**参考答案：**

A

1. write()方法将（        ）写入文件。

A. 一个字符串  
B. 一个布尔值  
C. 一个浮点数  
D. 一个整数

**参考答案：**

A

1. 从infile文件对象读取文件的一行数据并作为一个字符串返回，使用（        ）。

A. infile.read(2)  
B. infile.read()  
C. infile.readline()  
D. infile.readlines()

**参考答案：**

C

1. 从infile文件对象读取2个字符，使用（        ）。

A. infile.read(2)  
B. infile.read()  
C. infile.readline()  
D. infile.readlines()

**参考答案：**

A

1. 打开文件C:\scores.txt追加数据，使用（         ）。

A. outfile = open("C:\\scores.txt", "a")  
B. outfile = open("C:\\scores.txt", "rw")  
C. outfile = open(file = "C:\scores.txt", "w")  
D. outfile = open(file = "C:\\scores.txt", "w")

**参考答案：**

A

1. 从infile文件对象读取文件的所有数据并作为一个字符串返回，使用（        ）。

A. infile.read(2)  
B. infile.read()  
C. infile.readline()  
D. infile.readlines()

**参考答案：**

B

1. 打开文件C:\scores.txt进行读操作，使用（         ）。

A. infile = open("C:\scores.txt", "r")  
B. infile = open("C:\\scores.txt", "r")  
C. infile = open(file = "C:\scores.txt", "r")  
D. infile = open(file = "C:\\scores.txt", "r")

**参考答案：**

B

1. 下列（        ）说法是正确的。

A. 当打开一个文件进行读操作，如果文件不存在，出现一个错误  
B. 当打开一个文件进行写操作，如果文件不存在，出现一个错误  
C. 当打开一个文件进行读操作，如果文件不存在，程序将打开一个空文件

**参考答案：**

A

1. writelines()方法将（         ）写入文件。

A. 一个字符串  
B. 一个字符串列表  
C. 一个浮点数列表  
D. 一个整数列表

**参考答案：**

B

1. 调用（        ）方法将字符串转换成原始字节数据。

A. encode()  
B. decode()  
C. convert()  
D. toString()

**参考答案：**

A

1. 打开文件C:\scores.txt进行写操作，使用（         ）。

A. outfile = open("C:\scores.txt", "w")  
B. outfile = open("C:\\scores.txt", "w")  
C. outfile = open(file = "C:\scores.txt", "w")  
D. outfile = open(file = "C:\\scores.txt", "w")

**参考答案：**

B

1. 从infile文件对象读取文件的所有行数据，使用（        ）。

A. infile.read(2)  
B. infile.read()  
C. infile.readline()  
D. infile.readlines()

**参考答案：**

D

1. 可以使用（        ）方法进行二进制读操作。

A. read  
B. input  
C. load  
D. receive

**参考答案：**

C

1. 打开文件c:\scores.dat进行二进制写操作，使用（         ）。

A. outfile = open("c:\\scores.dat", "wb")  
B. outfile = open("c:\\scores.dat", "w")  
C. outfile = open("c:\scores.dat", "a")  
D. outfile = open("c:\\scores.dat", "w")

**参考答案：**

A

1. 下列（        ）说法是错误的。

A. 当打开一个文件进行写操作，如果文件不存在，创建一个新文件  
B. 当打开一个文件进行写操作，如果文件已经存在，这个存在的文件被新文件覆盖  
C. 当打开一个文件进行写操作，如果文件已经存在，出现一个错误

**参考答案：**

C

1. 调用（        ）方法将原始字节数据转换成字符串。（2分）

A. encode()  
B. decode()  
C. convert()  
D. toString()

**参考答案：**

B

1. （        ）函数可以用来检查文件f是否存在。

A. os.path.isFile(f)  
B. os.path.exists(f)  
C. os.path.isfile(f)  
D. os.isFile(f)

**参考答案：**

C

1. 打开文件c:\scores.txt进行读写操作，使用（         ）。

A. iofile = open("c:\\scores.txt","r")  
B. iofile = open("c:\\scores.txt", "w")  
C. iofile = open("c:\\scores.txt", "ab")  
D. iofile = open("c:\\scores.txt", "r+")

**参考答案：**

D

1. 可以使用（        ）方法进行二进制写操作。

A. write  
B. output  
C. dump  
D. send

**参考答案：**

C