

## 全国高等学校计算机水平（广东考区）考试 II 级

### 《Python 程序设计》考试大纲

#### 一、考试目标与要求

考试目标：测试考生系统掌握 Python 基本概念及其编程思想，了解 Python 计算生态，利用 Python 进行数据处理及解决问题的能力。

考试要求：

1. 熟练掌握 Python 语言基础语法，掌握模块化程序设计思想，了解面向对象基本概念。
2. 熟练掌握 Python 常用标准库及第三方库的使用。
3. 能够阅读和分析 Python 程序，利用 Python 进行简单的数据处理及数据可视化。

#### 二、考试内容

##### (一) 基础知识

###### 【考试要求】

掌握 Python 中的常量、变量、数据类型、运算符、表达式、内置函数和基本输入输出。

###### 【主要考点】

1. 常量与变量
2. 基本数据类型
3. 运算符与表达式
4. 数据类型转换
5. 字符串类型及相关函数/方法
6. 基本输入输出
7. Python 常用内置函数
8. 注释

##### (二) 流程控制

###### 【考试要求】

掌握 Python 程序格式框架及书写风格，掌握顺序、选择、循环结构在 Python 中的实现方法，能灵活运用不同的控制结构解决实际问题。

###### 【主要考点】

1. 顺序结构
2. 选择结构：单分支、双分支、多分支选择结构，选择结构的嵌套
3. 循环结构：for 循环、while 循环，带有 else 子句的循环，break 与 continue 语句的作用
4. 程序的异常处理：try ---except

##### (三) 高级数据类型

###### 【考试要求】

掌握 Python 中列表、元组、字典、集合的使用方法以及应用场合；了解不同数据类型的



优缺点和适用领域；学会综合应用多种数据类型解决实际问题。

**【主要考点】**

1. 列表：创建、删除、修改、切片、访问、排序；多维列表
2. 元组：创建、删除、切片、访问
3. 字典：创建、删除、添加、修改及访问，有序字典的使用
4. 集合：创建、删除、访问，集合的并、交、差以及对称差等运算
5. 列表、元组、集合的区别及相互转换

#### **(四) 模块化程序设计**

**【考试要求】**

了解模块化程序设计思想，掌握分而治之的结构化程序设计方法；掌握函数定义、参数传递及高级函数的使用；了解变量作用域的含义；掌握匿名函数的定义及使用。

**【主要考点】**

1. 函数定义与使用
2. 函数参数：默认参数、可变参数、关键字参数
3. 变量作用域：局部变量和全局变量
4. 匿名函数的定义及使用
5. 高级函数（map、reduce、filter）的使用

#### **(五) 面向对象程序设计**

**【考试要求】**

了解Python中面向对象程序设计的基本方法，包括类的定义与使用、类的属性、类中的特殊方法。

**【主要考点】**

1. 类的定义与使用
2. 类成员访问控制
3. 继承与多态
4. 特殊方法与运算符重载

#### **(六) 文件操作**

**【考试要求】**

要求考生掌握文件的基本操作，掌握CSV、Excel文件的读取。

**【主要考点】**

1. 文件的使用：打开、读写、定位和关闭
2. 一二维数据组织：存储、处理及可视化
3. 读写CSV格式数据文件
4. 读取Excel格式数据文件（pandas库）

#### **(七) 模块与库**

**【考试要求】**

了解Python生态系统，掌握Python常用标准库，运用第三方库解决实际问题，包括但不限于网络爬虫、数据分析、文本处理、数据可视化、机器学习、Web应用开发、网络编程等领域。

**【主要考点】**



1. Python常用标准库：random、math、string、OS、time、datetime、urllib.request。
2. 第三方库：Matplotlib、pandas、jieba、wordcloud、requests、BeautifulSoup4、XPath、NumPy、Scrapy。

### 三、考试方式

机试。考试时间为 105 分钟，满分 100 分。

### 四、考试题型题量

- 单选题（20 题，40 分）
- 基础操作题（2 题，25 分）
- 简单应用题（1 题，15 分）
- 综合应用题（1 题，20 分）

### 五、考试环境要求

Python 3.7或以上版本解释器+安装jieba库+可选装PyScripter/Pycharm等集成开发环境（IDE）

### 六、考试样题

#### （一）单选题（20 题，40 分）

- 1、下列选项中不符合 Python 语言变量命名规则的是\_\_\_\_\_。
  - A) OHOH5
  - B) \_ OHOH\_
  - C) OHOH
  - D) 5OHOH
- 2、在 print() 语句中使用 format() 函数进行格式化输出，能够将一个整数以八进制方式输出的是\_\_\_\_\_。
  - A) {:x}
  - B) {:o}
  - C) {:c}
  - D){:b}
- 3、经过表达式 x=' 2' 的赋值，x 的数据类型是\_\_\_\_\_。
  - A) int
  - B) str
  - C) float
  - D) list
- 4、在 Python 函数中，用于获取用户输入的是\_\_\_\_\_。
  - A) get()



- B) eval()
- C) input()
- D) print()

5、下面代码的输出结果是\_\_\_\_\_。

```
for a in "Hello world":  
    print(a, end="")  
    if a == "l":  
        break
```

- A) Hel
- B) Hell
- C) He
- D) Hello

6、以下选项关于分支结构的描述，正确的是\_\_\_\_\_。

- A) 单分支结构指只有 if 语句，没有 else 语句的分支结构
- B) 多分支结构是 if-elif 保留字构成的分支结构
- C) 二分支结构的简介表达方式是：<条件> if <表达式 1> else <表达式 2>
- D) 分支结构中的判断条件只能是产生 True 或 False 的表达式或函数

7、以下关于字典类型的描述，错误的是\_\_\_\_\_。

- A) 字典中的元素以值信息为索引访问
- B) 使用 {} 可以创建一个空的字典
- C) 字典中一个键信息只对应一个值信息
- D) 字典长度是可变的

8、下列代码的输出结果是\_\_\_\_\_。

```
a = 0  
b = 1  
if (a > 0) or (b / a > 2):  
    print("yes")  
else:  
    print("no")
```

- A) yes
- B) no
- C) 报错：ZeroDivisionError: division by zero
- D) 没有显示结果或错误提示

9、关于 python 的内置函数库 turtle 库的描述，以下正确的是\_\_\_\_\_。

- A) 用 from turtle import \* 语句之后，可以用 circle() 函数画圆
- B) 用 import turtle as t 语句之后，可以用 turtle.circle() 函数画图
- C) 用 import turtle 语句之后，用 circle() 函数画图
- D) turtle.pensize() 函数的别名是 turtle.size()



- 10、random.randrange(a, b)的作用是 \_\_\_\_\_。
- A) 生成一个[a, b]之间的随机整数
  - B) 生成一个[a, b]之间的随机小数
  - C) 生成一个[a, b)之间的随机小数
  - D) 生成一个[a, b)之间的随机整数
- 11、下列代码的输出结果是 \_\_\_\_\_。
- ```
>>>f= lambda x,y:y+x  
>>>f(10,10)
```
- A) 10
  - B) 20
  - C) 10, 10
  - D) 100
- 12、下面代码的输出结果是 \_\_\_\_\_。
- ```
ls = ["猫", "狗", "鸡", "兔", "狗"]  
x = "狗"  
print(ls.index(x,0))
```
- A) 0
  - B) 1
  - C) -4
  - D) -3
- 13、关于函数形参和实参的描述，以下选项正确的是\_\_\_\_\_。
- A) 函数定义中参数列表里面的参数是实际参数，简称实参
  - B) 调用函数时，参数列表中给出要传入函数内部的参数，称为形式参数，简称形参
  - C) 程序调用函数时，将形参复制给函数的实参
  - D) 程序调用函数时，将实参复制给函数的形参
- 14、已有集合 s = {1, 2, 3}和 t = {2, 3, 5}，则 s.difference(t)的输出结果是\_\_\_\_\_。
- A) {1}
  - B) {5}
  - C) {2, 3}
  - D) {1, 2, 3, 5}
- 15、下列选项中哪项不是 Python 语言的保留字\_\_\_\_\_。
- A) For
  - B) del
  - C) from
  - D) while
- 16、下列代码的输出结果是\_\_\_\_\_。



```
n = 2
def multiply(x, y = 5):
    global n
    return x*y*n
s = multiply(9)
print(s)
```

A) 系统提示错误  
B) 45  
C) 54  
D) 90

17、一个已经打开的文件 exam.txt 保存在变量 file 中，下列哪项从文件中读入所有行，并保存在一个列表中\_\_\_\_\_。

- A) file.readline()
- B) file.readlines()
- C) file.readall()
- D) file.read()

18、下列代码的输出结果是\_\_\_\_\_。

```
class Point():
    def __init__(self, x=0, y=0):
        self.x = x+1
        self.y = y+1
```

```
p1 = Point(3)
print(p1.x, p1.y)
```

- A) 0 0
- B) 4 1
- C) None None
- D) x, y

19、关于面向对象方法中继承的叙述，以下选项中错误的是\_\_\_\_\_。

- A) 继承仅仅允许单重继承，即不允许一个子类有多个父类
- B) 继承是父类和子类之间共享数据和方法的机制
- C) 继承定义了一种类与类之间的关系
- D) 继承关系中的子类将拥有父类的全部属性和方法

20、以下选项中，不是 Python 文件处理.seek()方法的参数是\_\_\_\_\_。

- A) 1
- B) 2
- C) 0
- D) -1



## (二) 基本操作题 (2 题, 25 分)

1、(10 分) 考生文件夹下存在一个文件 PY101.py, 请写代码替换横线, 不修改其他代码, 实现以下功能:

遍历字符串 "789Python2", 要求其输出结果为 "-P--y--t--h--o--n-!"。其中字符的输出要求在同一行中连续输出, 每个字符的输出宽度为 3 个字符, 居中放置, 以减号字符 "-" 填充, 例如: -p-。注意: 程序中所有字符串均使用双引号""。

请把编号(1)~(5)和对应下划线删除, 填空完成程序中的语句, 不能修改已有的代码。

```
____(1)____ "789Python2": #1 遍历字符串
    if i.isnumeric():
        ____ (2) ____ #2 程序最后仅输出 python!
        #3 每个字符输出宽度为 3, 居中放置, 以 "-" 作为填充,
        #4 将整个单词在同一行连续输出
        print(" {:_:(3)_} ".format(i), ____ (4) ____ )
____ (5) ____: # 5 输出 Python 后面的惊叹号
    print(" !")
```

2、(15 分) 考生文件夹下存在一个文件 PY102.py, 请写代码替换横线, 不修改其他代码, 实现以下功能:

编程计算下列式子的值:

$$1!+3!+5!+\dots+n!$$

从键盘上输入一个整数  $n$ , 把上面公式的值打印输出到屏幕。如果输入的不是整数是小数、字母或其他符号, 则程序就在屏幕上输出“异常信息: 你输入的数据必须是阿拉伯数字!”字样。

如键盘上输入数字 7, 则输出:

输入 7 时, 得到式子的值为: 5167

如键盘上输入数字 8, 则输出:

输入 8 时, 得到式子的值为: 5167

如键盘上输入 5.3 则输出:

异常信息: 你输入的数据必须是阿拉伯数字!

如键盘上输入 a 则输出:

异常信息: 你输入的数据必须是阿拉伯数字!

请把编号(1)~(7)和对应下划线删除, 填空完成程序中的语句, 不能修改已有的代码。

```
try:
    n=int(input("请输入一个整数: "))
    sum = 0
    i = 0
    while ____ (1) ____: #1
        i = i + 1
        if i > n:
            ____ (2) ____ #2
        if ____ (3) ____: #3
            continue
    else:
```

```
#求第 i 项的阶乘,product 为第 i 项的阶乘的积
product=j=1
while j<=i:
    ___(4)___=j          #4 计算阶乘
    j+=1
    ___(5)___= product    #5 计算阶乘的和
print("输入%d 时, 得到式子的值为: %d" ___(6)___) #6 把结果输出
except ___(7)___:        #7
    print("异常信息: 你输入的数据必须是阿拉伯数字!")
```

### (三) 简单应用题 (1 题, 15 分)

1、(15 分) 考生文件夹下存在一个文件 PY103.py, 请写代码替换横线, 不修改其他代码, 实现以下功能:

学习语文时有中文重组句子, 英语里也有重组句, 定义一个函数 `comb(sentence)`, 其功能是把一英语句子的单词打乱, 并把打乱的单词用一个空格连接起来, 然后输出。如果句子的单词小于等于 2 个就输出: 原句的单词小于等于 2 个。

(注意程序中的字符串全部使用双引号"表示)

请把编号(1)~(7)和对应下划线删除, 填空完成程序中的语句, 不能修改已有的代码。

```
import random
def comb(sentence):          #定义一个函数, 参数为 sentence
    words=___(1)___()        #1 把单词分割出来
    print("-"*60)
    if len(words)==1 ___(2)___ len(words)==2: #2 如果句子中的单词小于 2 个, 则不进行打乱, 输出"
原句的单词小于等于 2 个"
        print("原句的单词小于等于 2 个")
    else:
        jumble=[]
        while ___(3)___:      #3 对句子的单词进行打乱
            site = ___(4)___(len(words)) #4
            jumble.___(5)___      #5
            words=___(6)___      #6 切片时位置如果是表达式要用括号()括起来
            s=___(7)___(jumble)   #7 把单词用一个空格连接起来
        print("句子打乱顺序后的单词组合为:\n",s)

if __name__=="__main__":
    txt="The Beijing Organising Committee for the 2022 Olympic and Paralympic Winter Games is a pu
blic institution with legal person status"
    print("原句为:\n", txt)
    comb(txt)
```

### (四) 综合应用题 (1 题, 20 分)

1、(20 分) 考生文件夹下存在一个文件 PY104.py, 请写代码替换横线, 不修改其他代码, 实现以下功能:





汉语中结构助词主要表示附加成分和中心语之间的结构关系，在书面语里结构助词习惯写成三个字：“的”、“地”、“得”。这样可以使书面语里的结构关系更清楚。请统计 sefile104.txt 文件中的结构助词的种类，并把每种结构助词按个数从少到多排列，然后把个数输出到屏幕，格式要求：宽度为 5 个字符，减号字符-填充，右对齐。

注意 sefile104.txt 文件的编码为 UTF-8（UTF-8 使用大写），程序中的字符串全部使用双引号 "" 表示。

请把编号(1)~(10)和对应下划线删除，填空完成程序中的语句，不能修改已有的代码。

```
import jieba
list_names=["的","地","得"]
list1=[]
count={} #用来计数
with open("sefile104.txt","r",__(1)__) as f: #1
    txt=f.__(2)___ #2
    words=__ (3)__(txt) #3 使用精确模式进行分词
    for word in words:
        if word __ (4)___ list_names: #4 如果单词不是结构助词，那就不记录这个分词了
            __ (5)___ #5
        list1.append(word)
        count[word]= __ (6)___ +1 #6
    kind=__ (7)___(list1) #7 统计文件中结构助词出现的种类
    print("sefile104.txt 文件中结构助词的种类有： %d"%kind)
    items=__ (8)___(count.items()) #8 将其返回为列表类型
    items.sort(key=lambda x:x[1],__(9)___) #9 按个数升序排序
    for i in range(kind):
        var,number=items[i]
        print('文件中" {0}"字出现次数为{ __(10)___ }'.format(var,number)) #10
```