Python 程序设计基础互助课堂

期末复习 2

易翔

> 注：本章练习中的所有语法规则、代码执行结果等对 python 版本存在依赖关系的内容，均以 python==3.10.9 为准。

2023 年 6 月 5 日

一. 判断题 本大题共计 10 小题

1. Python allows mixing tabs and spaces for indentation. ______

2. Set object **does not** support item assignment. ______

3. The **suffix** name of a file (like ".csv") determines the type of file. In other word, a file with **suffix** ".txt" will never be a binary file. ______

4. **Pickle** stores any Python data structure into a binary file. ______

5. @staticmethod decorator must be declare after the method definition. ______

6. *args and **kwargs allow you to pass a variable number of arguments to the function definition. ______

7. __getattr__ and __getattribute__ have identical functions(功能). ______

8. You can use the Python keyword super to initialize the parent class in def __init__ ():... . ______

9. Similar to the **list comprehension**, python has **set comprehension** and **dictionary comprehension** ______

10. assert will raise a **AssertionException** when the expression after assert is False ______

二. 不定项选择题 本大题共计 10 小题

1. Python is ______ ....( )

   A. Machine Language   B. Assembly Language

   C. High-level Language   D. *None of them above.*

2. According to **PEP8**, which of following coding style is better? ______

```
# A
income = (gross_wages +
          taxable_interest +
          (dividends - qualified_dividends) -
          ira_deduction -
          student_loan_interest)
# B
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

3. What are the minimum methods required to implement the following code (TypeError is not allowed)? ______

```
class A():
    # some code
a = A()
print(a[1])
```

....( )

A. only `__getitem__`

B. only `__setitem__`

C. both `__getitem__` and `__setitem__`

D. *None of them above.*

4. Which of the following variable name is valid? ________ . . . . . . ( )

   A. `nonlocal` B. `s1mple` C. `_foobar_` D. `<name>`

   E. `v$50` F. `don't_change_this`

5. What is the result of the following expression? ______

   `-2 ** 5 + 3 * (7 - 4)` . . . . ( )

   A. 41 B. -768 C. 288 D. -23

6. What is the result of the following expression? ______

   `False or None or 0 or [ ]` . . . . ( )

   A. `False` B. `None` C. `0` D. `[]`

7. Which of the correspondence of result(s) is/are **true**? ____________ ( )

   A. `math.ceil(2.2) # 3.0` B. `round(5.5) # 6`

   C. `int(2.8) # 2` D. `float('inf') # inf`

8. Which of the correspondence of result(s) is/are **true**? ________

```
# input format:
# 1 2 3
in = input()
out = sum(in.split(' '))
print(out, ending="")
```

. . . . ( )

A. line 3: `SyntaxError` B. line 4: `TypeError`

C. line 5: `ValueError`

9. For operators and methods of *set*, the corresponding **correct** one(s) is/are (such that methods and operators have the same function) ________ ( )

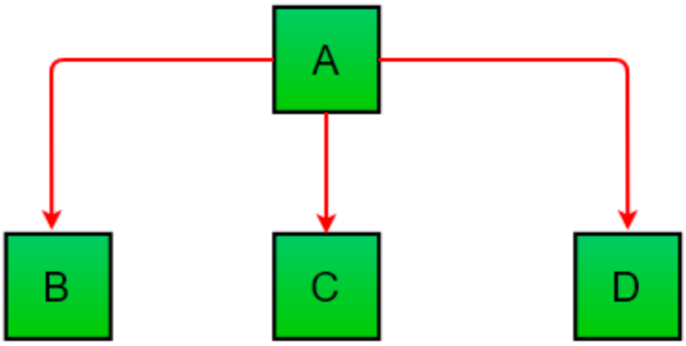
   A. `intersection()` and &

   B. `union()` and |

   C. `difference()` and -

   D. `symmetric_difference()` and ˆ

10. What is the types of inheritance of the following figure? ____



....( )

   A. Multiple Inheritance B. Multilevel Inheritance

   C. Hierarchical Inheritance D. Hybrid Inheritance

三. 简答题　本大题共 3 小题

1. 请写出下列代码的输出。

```
for a in []:
    print("Access for loop.")
else:
    print("Out for loop.")
while 0:
    print("Access while loop.")
else:
    print("Out while loop.")
```

2. 请简述集合操作中remove() 和 discard ()方法的区别。

3. 请简述 if __name__ == '__main__': ... 的作用，并描述解释器在执行**通过 import 调用到的 module** 和**顶层 module** 时，二者之间的区别。

四. 代码题　本大题共 2 小题

1. 考虑下列代码，请指出执行时代码将会遇到的问题。并且在不影响其预期功能的前提下加以修改，使得代码能够正确运行[1]。

```
class B():
    def __init__(self，data):
        self .data = data

    def __getattribute__(self，name):
        return self .data[name]

b = B({"foo": "bar"})
print(b.foo) # We want to see "bar" through instance field access.
```

2. 现有两个字典，这两个字典满足下面的条件：1. 两个字典均不为空；2. 两个字典等长；3. 两个字典的键集合是完全相同的（i.e. a.keys() == b.keys()）。然而，对于任意一个键，两个字典中**有且仅有**一个字典对应的值为有效值（y 有效值指的是不为None的值，字符串"None", ""和0均为有效值）。要求合并得到的字典所有键对应的值均为有效值。[2]

   (1) 下列代码是一串用于合并这两个字典的程序，然而在某些情况下，这个程序无法按照预期的方式完成任务。请指出程序在哪些情况下会出现预期之外的结果，并给出修改之后的程序。

```
a = {"1": None, "2": "python", "3": "CS112"}
b = {"1": "Study", "2": None, "3": None}

def merge(dict1，dict2 ):
    # Is there any difference between "if None:" and "if False:"?
    return {k: dict1 [k] or dict2 [k] for k in dict1 .keys()}
print(merge(a, b))
```

[1] 摘自 *Effective Python, 59 Specific Ways to Write Better Python*, 此书是授课老师推荐的参考书籍之一。

[2] 此题思路来源于 https://www.bilibili.com/video/BV1SN411y7P7/

(2) 下列代码是这个问题的另一种实现方式，但是在某些情况下仍然存在问题。请指出程序在哪些情况下会出现预期之外的结果，并给出修改之后的程序。

```
# This is a hint!
import numpy as npy

a = {"1": None, "2": "python", "3": "CS112"}
b = {"1": "Study", "2": None, "3": None}

def merge(dict1, dict2 ):
    for k in dict1 :
        if dict1 [k] != None:
            dict2 [k] = dict1[k]
        else:
            # value in dict2 is OK
            pass
    return dict2
print(merge(a, b))
```

(3) 拓展这个问题：在其他条件不变的情况下，如果我们需要合并的是任意数量的字典，该用什么方式来实现这个要求呢（字典数量范围为 $[2, 10^5]$）？

```
a = {"1": None, "2": "Python", "3": None, "4": "is", "5": None}
b = {"1": "Study", "2": None, "3": "CS112", "4": None, "5": "funny"}
c = {"a": 1, "b": None, "c": None}
d = {"a": None, "b": 2, "c": None}
e = {"a": None, "b": None, "c": 3}
# Your code here. Start with func define: def merge ...






# Your code end.
print(merge(a, b))
print(merge(c, d, e))
```