**第12.2节超类和子类**

12.1分析以下代码：

class A:

def \_\_init\_\_(self, i = 0):

self.i = i

class B(A):

def \_\_init\_\_(self, j = 0):

self.j = j

def main():

b = B()

print(b.i)

print(b.j)

main()

 A. B类继承A，但A中i中的数据字段不是继承的。

 B. B类继承A并自动继承A中的所有数据字段。

 C.创建对象B时，必须传递一个整数，如B（5）。

 D.对象b无法访问数据字段j。

A

答案分析：A

12.2以下代码将显示什么？

class A:

def \_\_init\_\_(self, i = 1):

self.i = i

class B(A):

def \_\_init\_\_(self, j = 2):

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

self.j = j

def main():

b = B()

print(b.i, b.j)

main()

A. 0 0

B. 0 1

C. 1 2

D. 0 2

E. 2 1

C

答案分析：C

12.3以下代码的输出是什么？

class ParentClass:

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

self.\_\_x = 1

self.y = 10

def print(self):

print(self.\_\_x, self.y)

class ChildClass(ParentClass):

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

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

self.\_\_x = 2

self.y = 20

c = ChildClass()

c.print()

A. 1 10

B. 1 20

C. 2 10

D. 2 20

B

答案分析：B

12.4假设A是B的子类，要从A调用B中的\_\_init\_\_方法，你要写\_\_\_\_\_\_\_\_\_。

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

B. super().\_\_init\_\_(self)

C. B.\_\_init\_\_()

D. B.\_\_init\_\_(self)

AD

答案分析：AD

12.5你可以在B类的第三行放什么代码来调用B的超类构造函数？

class A:

def \_\_init\_\_(self, i = 1):

self.i = i

class B(A):

def \_\_init\_\_(self, j = 2):

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

self.j = j

def main():

b = B()

print(b.i, b.j)

main()

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

B. super().\_\_init\_\_()

C. A.\_\_init\_\_()

D. A.\_\_init\_\_(self)

BD

答案分析：BD

**第12.3节覆盖方法**

12.6以下代码将显示什么？

class A:

def \_\_init\_\_(self, i = 0):

self.i = i

def m1(self):

self.i += 1

class B(A):

def \_\_init\_\_(self, j = 0):

A.\_\_init\_\_(self, 3)

self.j = j

def m1(self):

self.j += 1

def main():

b = B()

b.m1()

print(b.i, b.j)

main()

A. 2 0

B. 3 1

C. 4 0

D. 3 0

E. 4 1

B

答案分析：B

12.7以下哪项陈述是正确的？

 A.子类​​是超类的子集。

 B.从子类调用构造函数时，总是调用其超类的无参数构造函数。

 C.您可以覆盖超类中定义的非私有方法。

 D.您可以覆盖超类中定义的初始化程序。

 E.您可以覆盖超类中定义的私有方法。

CD

答案分析：CD

**第12.4节对象类**

12.8以下代码将显示什么？

class A:

def \_\_new\_\_(self):

self.\_\_init\_\_(self)

print("A's \_\_new\_\_() invoked")

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

print("A's \_\_init\_\_() invoked")

class B(A):

def \_\_new\_\_(self):

print("B's \_\_new\_\_() invoked")

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

print("B's \_\_init\_\_() invoked")

def main():

b = B()

a = A()

main()

 A. B的\_\_new \_\_（）被调用，后跟A的\_\_init \_\_（）被调用

 B. B调用\_\_new \_\_（），然后调用A的\_\_new \_\_（）

 C.调用\_\_new \_\_（），然后调用A的\_\_init \_\_（），然后调用A的\_\_new \_\_（）

 D. A的\_\_init \_\_（）被调用，然后调用A的\_\_new \_\_（）

C

答案分析：C

12.9以下哪项陈述是正确的？

 A.默认情况下，\_\_ new \_\_（）方法调用\_\_init\_\_方法。

 B. \_\_new \_\_（）方法在对象类中定义。

 C. \_\_init \_\_（）方法在对象类中定义。

 D. \_\_str \_\_（）方法在对象类中定义。

 E. \_\_eq \_\_（其他）方法在对象类中定义。

ABCDE

答案分析：ABCDE

**第12.5节多态性和动态绑定**

12.10以下代码将显示什么？

class A:

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

self.i = 1

def m(self):

self.i = 10

class B(A):

def m(self):

self.i += 1

return self.i

def main():

b = B()

print(b.m())

main()

A. 1

B. 2

C. 10

D. i is not accessible from b.

B

答案分析：B

12.11以下代码将显示什么？

class A:

def \_\_str\_\_(self):

return "A"

class B(A):

def \_\_str\_\_(self):

return "B"

class C(B):

def \_\_str\_\_(self):

return "C"

def main():

b = B()

a = A()

c = C()

print(a, b, c)

main()

A. C C C

B. A B C

C. A A A

D. B B B

B

答案分析：B

12.12以下代码将显示什么？

class A:

def \_\_str\_\_(self):

return "A"

class B(A):

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

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

class C(B):

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

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

def main():

b = B()

a = A()

c = C()

print(a, b, c)

main()

A. C C C

B. A B C

C. A A A

D. B B B

C

答案分析：C

12.13以下代码将显示什么？

class A:

def \_\_init\_\_(self, i = 2, j = 3):

self.i = i

self.j = j

def \_\_str\_\_(self):

return "A"

def \_\_eq\_\_(self, other):

return self.i \* self.j == other.i \* other.j

def main():

x = A(1, 2)

y = A(2, 1)

print(x == y)

main()

A. True

B. False

C. 2

D. 1

A

答案分析：A

12.14以下代码将显示什么？

class Person:

def getInfo(self):

return "Person's getInfo is called"

def printPerson(self):

print(self.getInfo(), end = ' ')

class Student(Person):

def getInfo(self):

return "Student's getInfo is called"

def main():

Person().printPerson()

Student().printPerson()

main()

A. Person's getInfo is called Person's getInfo is called

B. Person's getInfo is called Student's getInfo is called

C. Student's getInfo is called Person's getInfo is called

D. Student's getInfo is called Student's getInfo is called

B

答案分析：B

12.15以下代码将显示什么？

class Person:

def \_\_getInfo(self):

return "Person's getInfo is called"

def printPerson(self):

print(self.\_\_getInfo(), end = ' ')

class Student(Person):

def \_\_getInfo(self):

return "Student's getInfo is called"

def main():

Person().printPerson()

Student().printPerson()

main()

A. Person's getInfo is called Person's getInfo is called

B. Person's getInfo is called Student's getInfo is called

C. Student's getInfo is called Person's getInfo is called

D. Student's getInfo is called Student's getInfo is called

A

答案分析：A

12.16分析以下代码：

class A:

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

self.setI(20)

print("i from A is", self.i)

def setI(self, i):

self.i = 2 \* i;

class B(A):

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

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

def setI(self, i):

self.i = 3 \* i;

b = B()

 A.不调用A类的\_\_init\_\_方法。

 B.调用A类的\_\_init\_\_方法，它显示“i from A is 0”。

 C.调用A类的\_\_init\_\_方法，它显示“i from A is 40”。

 D.调用A类的\_\_init\_\_方法，它显示“i from A is 60”。

D

答案分析：D

12.17分析以下代码：

class A:

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

self.setI(20)

def setI(self, i):

self.i = 2 \* i;

class B(A):

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

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

print("i from B is", self.i)

def setI(self, i):

self.i = 3 \* i;

b = B()

 A.不调用A类的\_\_init\_\_方法。

 B.调用A类的\_\_init\_\_方法，它显示“i from B is 0”。

 C.调用A类的\_\_init\_\_方法，它显示“i from B is 40”。

 D.调用A类的\_\_init\_\_方法，它显示“i from B is 60”。

D

答案分析：D

**第12.6节isinstance函数**

12.18要检查对象o是否是A类的实例，请使用\_\_\_\_\_\_\_\_\_。

 A. o.isinstance（A）

 B. A.isinstance（o）

 C. isinstance（o，A）

 D. isinstance（A，o）

C

答案分析：C

**第12.8节类关系**

12.19什么样的关系适合公司和员工？

 A.关联

 B.组成

 C.继承

B

答案分析：B

12.20什么样的关系适合课程和教师？

 A.关联

 B.组成

 C.继承

A

答案分析：A

12.21什么样的关系适合学生和个人？

 A.关联

 B.组成

 C.继承

C

答案分析：C

12.22什么样的关系适合House和Window？

 A关联

 B.组成

 C.继承

B

答案分析：B

12.23账户和储蓄账户有什么关系？

 A.关联

 B.组成

 C.继承

C

答案分析：C