**Section 12.2 Superclasses and Subclasses**

12.1 Analyze the following code:

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. Class B inherits A, but the data field in i in A is not inherited.

B. Class B inherits A and automatically inherits all data fields in A.

C. When you create an object B, you have to pass an integer such as B(5).

D. The data field j cannot be accessed by object b.

A

Answer analysis:A

12.2 What will be displayed by the following code?

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

Answer analysis:C

12.3 What is the output of the following code?

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

Answer analysis:B

12.4 Suppose A is a subclass of B, to invoke the \_\_init\_\_ method in B from A, you write \_\_\_\_\_\_\_\_\_.

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

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

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

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

AD

Answer analysis:AD

12.5 What code can you put in the third line in class B to invoke B's superclass's constructor?

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

Answer analysis:BD

**Section 12.3 Overriding Methods**

12.6 What will be displayed by the following code?

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

Answer analysis:B

12.7 Which of the following statements is true?

A. A subclass is a subset of a superclass.

B. When invoking a constructor from a subclass, its superclass's no-arg constructor is always invoked.

C. You can override a non-private method defined in a superclass.

D. You can override the initializer defined in a superclass.

E. You can override a private method defined in a superclass.

CD

Answer analysis:CD

**Section 12.4 The object Class**

12.8 What will be displayed by the following code?

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's \_\_new\_\_() invoked and followed by A's \_\_init\_\_() invoked

B. B's \_\_new\_\_() invoked followed by A's \_\_new\_\_() invoked

C. B's \_\_new\_\_() invoked, followed by A's \_\_init\_\_() invoked, and followed by A's \_\_new\_\_() invoked

D. A's \_\_init\_\_() invoked and followed by A's \_\_new\_\_() invoked

C

Answer analysis:C

12.9 Which of the following statements is true?

A. By default, the \_\_new\_\_() method invokes the \_\_init\_\_ method.

B. The \_\_new\_\_() method is defined in the object class.

C. The \_\_init\_\_() method is defined in the object class.

D. The \_\_str\_\_() method is defined in the object class.

E. The \_\_eq\_\_(other) method is defined in the object class.

ABCDE

Answer analysis:ABCDE

**Section 12.5 Polymorphism and Dynamic Binding**

12.10 What will be displayed by the following code?

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

Answer analysis:B

12.11 What will be displayed by the following code?

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

Answer analysis:B

12.12 What will be displayed by the following code?

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

Answer analysis:C

12.13 What will be displayed by the following code?

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

Answer analysis:A

12.14 What will be displayed by the following code?

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

Answer analysis:B

12.15 What will be displayed by the following code?

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

Answer analysis:A

12.16 Analyze the following code:

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. The \_\_init\_\_ method of class A is not called.

B. The \_\_init\_\_ method of class A is called and it displays "i from A is 0".

C. The \_\_init\_\_ method of class A is called and it displays "i from A is 40".

D. The \_\_init\_\_ method of class A is called and it displays "i from A is 60".

D

Answer analysis:D

12.17 Analyze the following code:

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. The \_\_init\_\_ method of class A is not called.

B. The \_\_init\_\_ method of class A is called and it displays "i from B is 0".

C. The \_\_init\_\_ method of class A is called and it displays "i from B is 40".

D. The \_\_init\_\_ method of class A is called and it displays "i from B is 60".

D

Answer analysis:D

**Section 12.6 The isinstance Function**

12.18 To check whether an object o is an instance of class A, use \_\_\_\_\_\_\_\_\_.

A. o.isinstance(A)

B. A.isinstance(o)

C. isinstance(o, A)

D. isinstance(A, o)

C

Answer analysis:C

**Section 12.8 Class Relationships**

12.19 What relationship is appropriate for Company and Employee?

A. association

B. composition

C. inheritance

B

Answer analysis:B

12.20 What relationship is appropriate for Course and Faculty?

A. association

B. composition

C. inheritance

A

Answer analysis:A

12.21 What relationship is appropriate for Student and Person?

A. association

B. composition

C. inheritance

C

Answer analysis:C

12.22 What relationship is appropriate for House and Window?

A. association

B. composition

C. inheritance

B

Answer analysis:B

12.23 What relationship is appropriate for Account and Savings Account?

A. association

B. composition

C. inheritance

C

Answer analysis:C