Quiz: OOP II (Practice Problems)

1 Inheritance

Note 1. Whenever an attribute is not found in a subclass, Python will check the superclass for the attribute before raising the `AttributeError` exception.

Problem 1. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    pass
a = Foo()
Foo.message = 'hello world'
b = Bar()
Bar.message = 'hola mundo'
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 2. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    pass
a = Foo()
Foo.message = 'hello world'
b = Bar()
a.message = 'hola mundo'
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 3. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    pass
a = Foo()
Foo.message = 'hello world'
b = Bar()
a.message = 'hola mundo'
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 4. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    pass
a = Foo()
a.message = 'hello world'
b = Bar()
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 5. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    pass
a = Foo()
Foo.message = 'hello world'
b = Bar()
Bar.message = 'hola mundo'
try:
    print('a.message=', a.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

1.1 With Constructors

Note 2. The constructor of a superclass will only be called if it is explicitly called in the constructor of the subclass. You should use the `super` function to get the superclass.

Problem 6. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self):
        self.message = 'hello world'
class Bar(Foo):
    def __init__(self):
        self.message = 'hola mundo'
a = Foo()
b = Bar()
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 7. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self):
        self.message = 'hello world'
class Bar(Foo):
    def __init__(self):
        self.message = 'hola mundo'
        super().__init__()
a = Foo()
b = Bar()
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 8. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self):
        self.message = 'hello world'
class Bar(Foo):
    def __init__(self):
        super().__init__()
        self.message = 'hola mundo'
a = Foo()
b = Bar()
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 9. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
class Bar(Foo):
    def __init__(self, message=None):
        self.message = message
        super().__init__(message)
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 10. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        Foo.message = message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 11. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        Foo.message = message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
b = Bar('hola mundo')
a = Foo('hello world')
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 12. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    pass
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__()
        Foo.message = message
a = Foo()
b = Bar('hola mundo')
c = Bar()
try:
    print('b.message=', b.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

1.2 With built-in Classes

Note 3. Recall that when you subclasss an existing class, you get all the superclass's functionality "for free". It is very common in python to subclass the built-in python classes in order to extend their functionality. Again, these problems focus on the "what/how" instead of the "why".

Problem 13. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo(list):
    def __init__(self, xs=[]):
        super().__init__(xs + [0])
xs = Foo([1])
xs = Foo([1, 2])
xs = Foo([1, 2, 3])
try:
    print('xs=', xs)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 14. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo(list):
    def __init__(self, xs=[]):
        xs.append(len(xs))
        super().__init__(xs)
xs = Foo([1, 2, 3])
xs = Foo(xs)
xs = Foo(xs)
try:
    print('xs=', xs)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 15. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo(list):
    def __init__(self, xs=[]):
        super().__init__(xs)
        xs.append(len(xs))
xs = Foo()
xs = Foo()
xs = Foo()
try:
    print('xs=', xs)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 16. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo(list):
    def __init__(self, xs=[]):
        xs.append(len(xs))
        super().__init__(xs)
xs = Foo()
xs = Foo([1, 2, 3])
xs = Foo()
try:
    print('xs=', xs)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

2 Static Methods

2.1 Without Inheritance

Problem 17. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
    def foo(self):
        return self.message
a = Foo('hello world')
try:
    print('a.foo()=', a.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 18. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    message = 'salve munde'
    def __init__(self, message=None):
        self.message = message
    @staticmethod
    def foo():
        return Foo.message
a = Foo('hello world')
try:
    print('a.foo()=', a.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 19. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    message = 'salve munde'
    def __init__(self, message=None):
        self.message = message
    @staticmethod
    def foo():
        return Foo.message
a = Foo('hello world')
try:
    print('Foo.foo()=', Foo.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 20. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    message = 'salve munde'
    def __init__(self, message=None):
        Foo.message = message
    @staticmethod
    def foo():
        return Foo.message
a = Foo('hello world')
try:
    print('Foo.foo()=', Foo.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

2.2 With Inheritance

Problem 21. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
    def foo(self):
        return self.message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
    def bar(self):
        return self.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('b.foo()=', b.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 22. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
    def foo(self):
        return self.message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
    def bar(self):
        return self.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('a.bar()=', a.bar())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 23. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
    def foo(self):
        return self.message
class Bar(Foo):
    message = 'salve munde'
    def __init__(self, message=None):
        super().__init__(message)
    @staticmethod
    def bar():
        return Bar.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('b.bar()=', b.bar())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 24. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
    def foo(self):
        return self.message
class Bar(Foo):
    message = 'salve munde'
    def __init__(self, message=None):
        super().__init__(message)
    @staticmethod
    def bar():
        return Bar.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('Bar.bar()=', Bar.bar())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 25. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    message = 'salve munde'
    def __init__(self, message=None):
        Bar.message = message
    @staticmethod
    def foo():
        return Bar.message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
    @staticmethod
    def bar():
        return Foo.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('Bar.foo()=', Bar.foo())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 26. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    message = 'salve munde'
    def __init__(self, message=None):
        Bar.message = message
    @staticmethod
    def foo():
        return Bar.message
class Bar(Foo):
    def __init__(self, message=None):
        super().__init__(message)
    @staticmethod
    def bar():
        return Foo.message
a = Foo('hello world')
b = Bar('hola mundo')
try:
    print('Bar.bar()=', Bar.bar())
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

3 Recursive Classes

Problem 27. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
foo = Foo('hello world')
foo.child = Foo('hola mundo')
foo.child.child = Foo('salve munde')
try:
    print(' foo.child.message=', foo.child.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 28. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
foo = Foo('hello world')
foo.child = Foo('hola mundo')
foo.child.child = Foo('salve munde')
try:
    print(' foo.child.child.child.message=', foo.child.child.child.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 29. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
foo = Foo('hello world')
foo.child = foo
try:
    print(' foo.child.child.child.message=', foo.child.child.child.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 30. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
foo = Foo('hello world')
foo.child = Foo('hola mundo')
foo.child.child = foo
try:
    print(' foo.child.child.child.message=', foo.child.child.child.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

Problem 31. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo:
    def __init__(self, message=None):
        self.message = message
foo = Foo('hello world')
foo.child = Foo('hola mundo')
foo.child.child = foo.child
try:
    print(' foo.child.child.child.message=', foo.child.child.child.message)
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```

3.1 With built-in classes

Problem 32. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
xs = [1, 2, 3]
xs.append(xs)
xs[0] = xs
try:
    print('xs[-1][0][2]=',xs[-1][0][2])
except TypeError:
    print('TypeError')
EOF
$ python3 foo.py
```

Problem 33. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
xs = [1, 2, 3]
xs.append(xs)
xs[0] = xs
try:
    print('xs[1][0][2]=',xs[1][0][2])
except TypeError:
    print('TypeError')
EOF
$ python3 foo.py
```

Problem 34. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
class Foo(list):
    def __init__(self, xs=[]):
        xs.append(len(xs))
        super().__init__(xs)
xs = Foo()
xs.append(Foo())
xs[0] = 1
xs.append(Foo())
try:
    print('xs[0]=', xs[0])
except AttributeError:
    print('AttributeError')
EOF
$ python3 foo.py
```