Mayar Magdy

Question 1:

How many issues in this code? From a correctness perspective.

1. class Employee:
2. def \_\_init\_\_(self, salary):
3. self.salary = salary
5. def get\_salary(self):
6. return self.salary
8. def set\_salary(self, value):
9. self.salary = value if value > 0 else 0 # ternary operator
11. salary = property(get\_salary, set\_salary)
13. if \_\_name\_\_ == '\_\_main\_\_':
14. em = Employee(100)
15. print(em.salary)

0

1

2

3

4+

1) infinite calls as salary attribute is same as property name. 2) creating object with -ve salary, will put it with -ve NOT 0. We need to call set\_salary from init

Top of Form

Question 2:

Property Class can help us to change a variable name while not break the old code that depends on this attribute

True

False

Top of Form

Question 3:

In python, we don't provide getters and setters as a default coding style

True

False

Top of Form

Question 4:

We can access a static variable with both the class and the object.

True

False

Top of Form

Question 5:

Static variables can help us maintain statistics about the number of calls a specific function received from different objects.

True

False

Top of Form

Question 6:

Static methods are used when you want a global function, but find it natural to let the function belong to a class and be prefixed with the class name.

True

False

Top of Form

Question 7:

Static variables are typically created inside a class but outside any method

True

False

Top of Form

Question 8:

Static variables are shared amongst objects of the class

True

False

Top of Form

Question 9:

Guess the output of this program:

1. class Employee:
2. total\_money\_paid = 0
4. def \_\_init\_\_(self, salary):
5. self.salary = salary
7. def pay\_employee(self):
8. Employee.total\_money\_paid += self.salary
10. if \_\_name\_\_ == '\_\_main\_\_':
11. Employee(10).pay\_employee()
12. Employee(20).pay\_employee()
13. Employee(30).pay\_employee()
14. print(Employee(40).total\_money\_paid)

60

100

Error

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

QQBottom of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Question 10:

Guess the output of this program:

1. class Employee:
2. total\_money\_paid = 0
4. def \_\_init\_\_(self, salary):
5. self.salary = salary
7. def pay\_employee(self):
8. Employee.total\_money\_paid += self.salary
10. @staticmethod
11. def print\_overall():
12. print('Total paid: ' + str(Employee.total\_money\_paid))
13. Employee.total\_money\_paid = 0
15. if \_\_name\_\_ == '\_\_main\_\_':
16. Employee(10).pay\_employee()
17. Employee(20).pay\_employee()
18. Employee(30).pay\_employee()
19. Employee(40).print\_overall()

Total paid: 60

Total paid: 100

Error

Top of Form

Question 11:

Guess the output of this program:

1. class Em:
2. def \_\_init\_\_(self):
3. self.x = [1, 2, 3]
4. e1 = Em()
5. e2 = e1
6. Em.x = [5, 6]
7. del e1.x
8. print(e2.x)

[1, 2, 3]

[5, 6]

Error

Top of Form

Question 12:

This code works well?

1. class Em:
2. def \_\_init\_\_(self):
3. self.x = [1, 2, 3]
4. e1 = Em()
5. e2 = e1
6. Em.x = [5, 6]
7. del e1.x
8. del Em.x
9. print(e2.x)

Yes

No

Top of Form

Question 13:

Guess the output of this program:

1. class A:
2. name = 'I am a class'
4. def \_\_init\_\_(self):
5. self.name = 'I am attribute'
7. def print(myobj):
8. print(myobj.name)
10. if \_\_name\_\_ == '\_\_main\_\_':
11. A.print(A)

I am a class

I am attribute

Error

Top of Form

Question 14:

Guess the output of this program:

1. class A:
2. def our\_print(self):
3. print('Method')
5. def our\_print(dummy):
6. print('Function')
8. if \_\_name\_\_ == '\_\_main\_\_':
9. A.our\_print = our\_print
10. A().our\_print()

Function

Method

Error

Bottom of Form

Question 15:

Does this code work?

1. class Person:
2. def \_\_init\_\_(self, salary):
3. self.salary = salary
5. @property
6. def salary(self):
7. return self.\_\_salary
9. @salary.setter
10. def salary(self, value):
11. if value < 0:
12. value = 0
13. self.\_\_salary = value


17. person = Person(100)
18. del person.salary

Yes

No

Bottom of Form

Method

Error

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Top of Form

Question 15:

Does this code work?

1. class Person:
2. def \_\_init\_\_(self, salary):
3. self.salary = salary
5. @property
6. def salary(self):
7. return self.\_\_salary
9. @salary.setter
10. def salary(self, value):
11. if value < 0:
12. value = 0
13. self.\_\_salary = value


17. person = Person(100)
18. del person.salary

Yes

No

Bottom of Form

Top of Form

Question 16:

Guess the output?

1. class Employee:
2. def \_\_init\_\_(self):
3. self.age = 1
5. @property
6. def salary(self):
7. return 0
9. @salary.setter
10. def salary(self, value):
11. pass


15. emp = Employee()
16. emp.salary = 10
17. print(list(emp.\_\_dict\_\_.keys()))

['age']

['age', 'salary']

properties are not added to the instance dict

Top of Form

Question 17:

Guess the output of this program:

1. class Hello:
2. def print(self, name):
3. print(name \* 2)

6. if \_\_name\_\_ == '\_\_main\_\_':
7. Hello.print(None, 'Most')

Error

MostMost

Top of Form

Question 18:

Guess the output of this program:

1. class Hello:
2. def \_\_init\_\_(self):
3. self.counter = 2
5. def print(self, name):
6. print(name \* self.counter)

9. class Hack:
10. counter = 2
12. if \_\_name\_\_ == '\_\_main\_\_':
13. Hello.print(Hack, 'Most')

Error

MostMost

Top of Form

Question 19:

Guess the output of this program:

1. class Hello:
2. def \_\_init\_\_(self):
3. self.counter = 1234

6. class Hack:
7. pass

10. if \_\_name\_\_ == '\_\_main\_\_':
11. h = Hack()
12. Hello.\_\_init\_\_(h)
13. print(h.counter)

Error

1234Bottom of Form

Bottom of Form

Bottom of Form

Top of Form

Bottom of Form