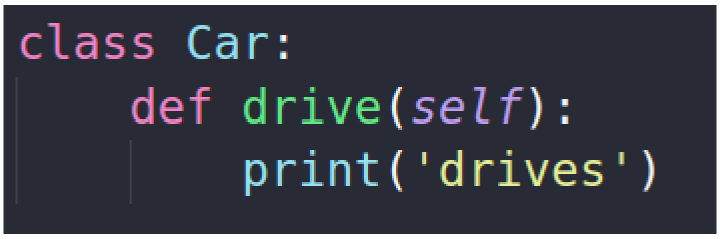
Object Oriented Programming (OOP) Exercises

1. How is Object Oriented Programming (OOP) different from traditional programming?

2. What does this program output?



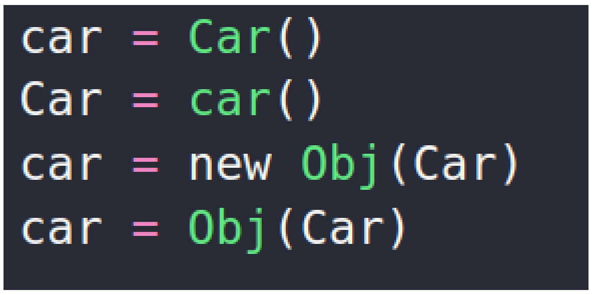
```
class Car:
    def drive(self):
        print('drives')
```

3. True or False: Objects are created from a Class

4. True or False: Objects can have methods

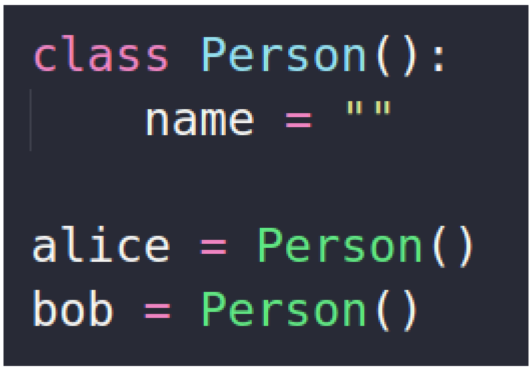
5. What is the purpose of a class?

6. Which one is correct and what does it do?



```
car = Car()
Car = car()
car = new Obj(Car)
car = Obj(Car)
```

7. Given this code, how would you set the objects variable name?



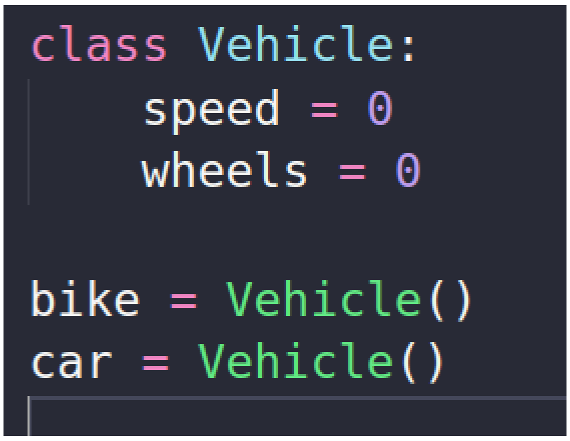
```
class Person():
    name = ""

alice = Person()
bob = Person()
```

8. Does the class Person have attributes? Does it have methods?

9. What is the maximum number of classes a Python program can have?

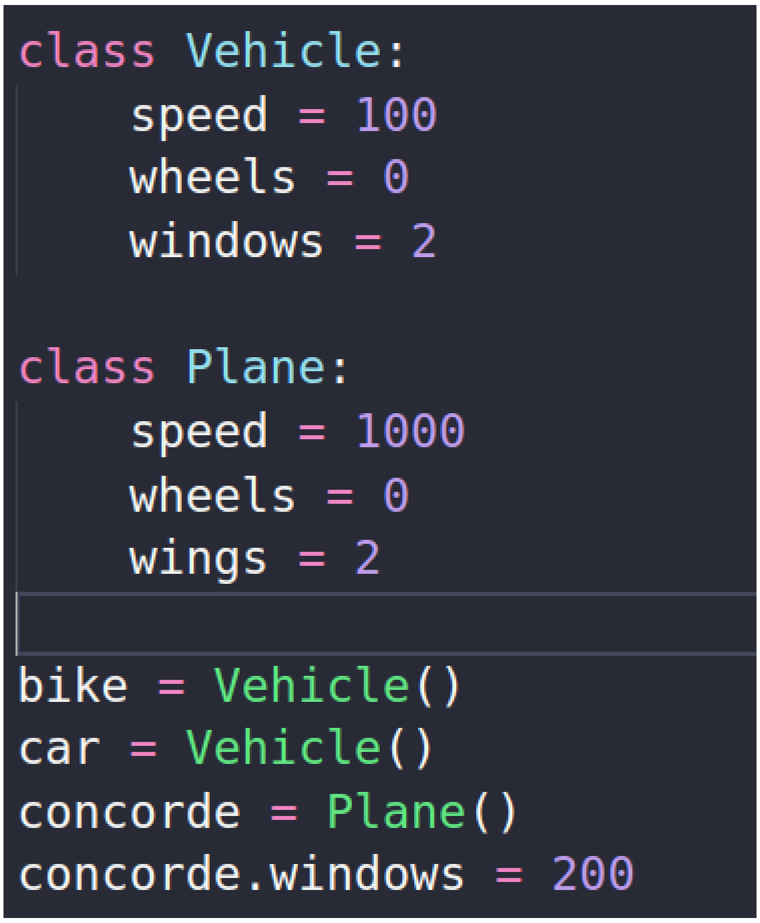
10. Given the program below, how would you set the objects variables?



```
class Vehicle:
    speed = 0
    wheels = 0

bike = Vehicle()
car = Vehicle()
```

11. What's wrong with this code?



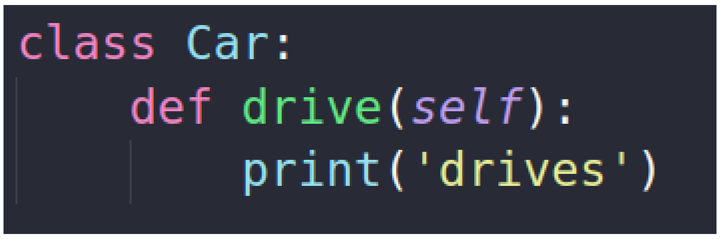
```
class Vehicle:
    speed = 100
    wheels = 0
    windows = 2

class Plane:
    speed = 1000
    wheels = 0
    wings = 2

bike = Vehicle()
car = Vehicle()
concorde = Plane()
concorde.windows = 200
```

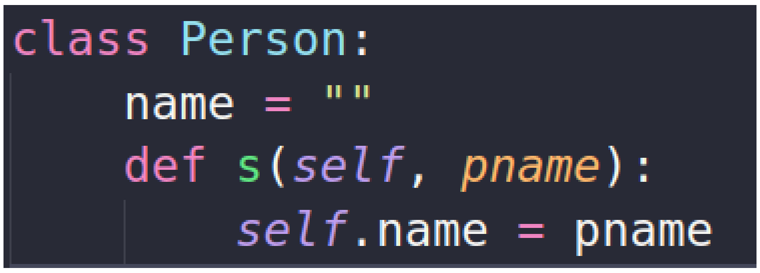
12. How many types of objects are there in the previous code?

13. What is self in the program:



```python
class Car:
    def drive(self):
        print('drives')
```

14. What does this method do?



```python
class Person:
    name = ""
    def s(self, pname):
        self.name = pname
```

15. What is a constructor?

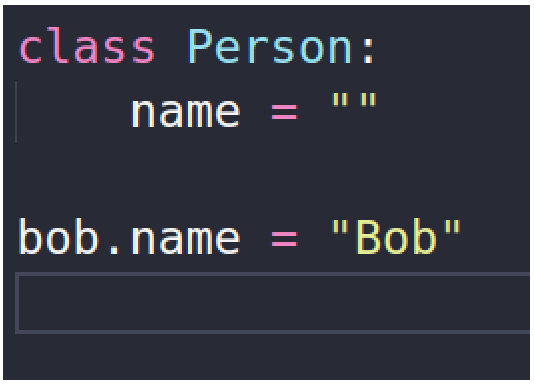
16. Do you have to explicitly create a constructor and destructor?

17. What is a private variable?

18. What types of variables exist in OOP?

19. True or False: Objects can create new variables

20. What's wrong with this program?



```python
class Person:
    name = ""

bob.name = "Bob"
```

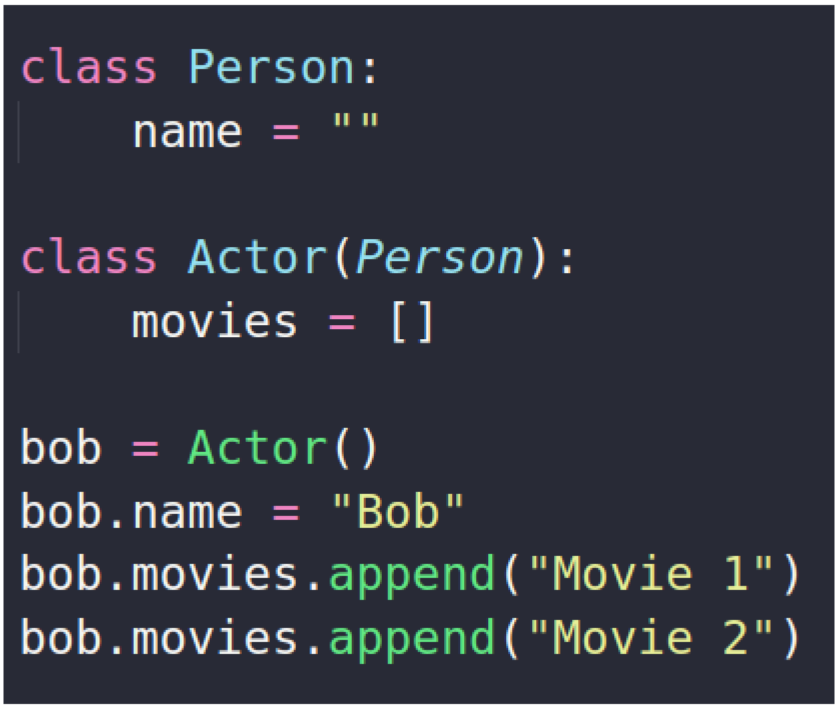
21. Which other programming languages support Object Orientated Programming?

22. Who can inherit: Classes or Objects?

23. What does it mean when we say “inherits” ?

24. Is there a maximum number of inheritances?

25. Find the error in this code:

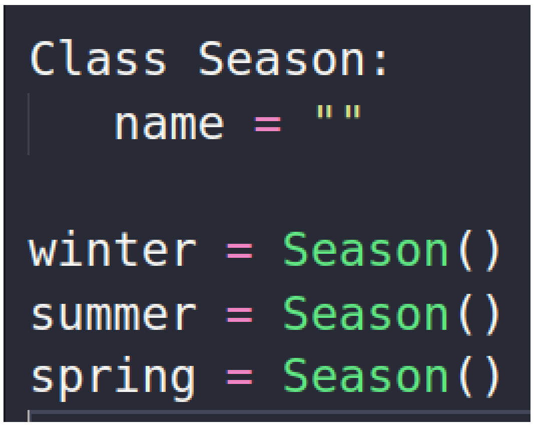


```
class Person:
    name = ""

class Actor(Person):
    movies = []

bob = Actor()
bob.name = "Bob"
bob.movies.append("Movie 1")
bob.movies.append("Movie 2")
```

26. Find the error in this program



```
Class Season:
    name = ""

winter = Season()
summer = Season()
spring = Season()
```

27. What is method over loading?

28. What is an interface?

29. What’s the difference between an interface and inheritance?

30. Create a class Cat that inherits from the class Animal. The class Animal should have the string species and the class Cat should have the string name

31. Whats the purpose of a Factory method?
32. When do you use a Factory Method?

33. What is object serialization?

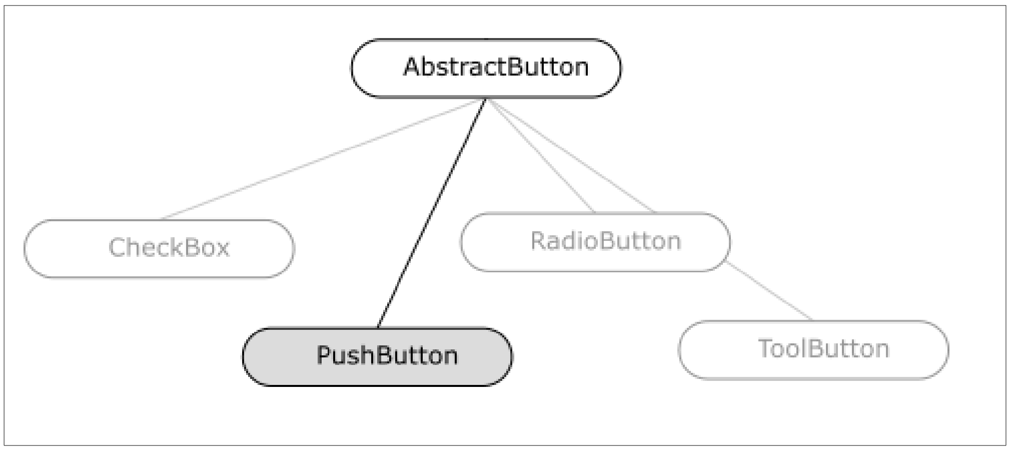
34. Is it possible to transfer objects in Python to another language?

35. Name some languages that are not a Object Oriented Programming language?

36. What are the advantages of Object Oriented programming?

37. Consider a Desktop App. Are the buttons objects?

38. What is happening here?



39. What is the downside of using to much inheritance?

40. How many children can inherit from a parent class?

OOP Answers

1. Traditional programming uses functions to organize code. Object-oriented programming (as the name implies) uses objects to organize code.
2. Nothing, no object is created and no method is called
3. True
4. True. An object can have any number of methods. (A method is a function that’s used inside a class.)
5. To define metadata (sometimes seen as blue print). This metadata is then used to create objects.
6. car = Car() it creates a new object with the name car, using the class Car.
7. By alice.name = “Alice” and bob.name = “Bob”.
8. It has one attribute (name), but no methods.
9. There is no maximum
10. By bike.wheels = 2, bike.speed = 120, car.wheels = 4, car.speed = 100
11. The object concorde is of the class Plane, the class plane does not have windows
12. There are 2 types of objects: Vehicle objects and Plane objects
13. A reference to the objects properties. In English a similar word is “my”. You could have “my salary”, “my name”. These would set your properties. Another persons “my salary” and “my name” would be different.
14. Set the objects name using the parameter pname. The method can be called like Alice.s(“Alice”).
15. A method that is called when you create a new object
16. No, that is optional
17. A variable whose value you cannot change by referencing the object.
18. Public variables (accessible by anyone), private variables (accessible only in the class) and protected variables
19. False: all variables must be defined in the class
20. The object is never created, bob = Person() is missing
21. Most modern programming languages including C++, Java, PHP, Ruby, C# and others
22. Classes can inherit

23. The Class will get the methods and attributes of the parent class.

24. No, there is no limit to the number of parent classes a class can inherit from. Unlike the real world, a class can have 5 parent classes

25. There is no error

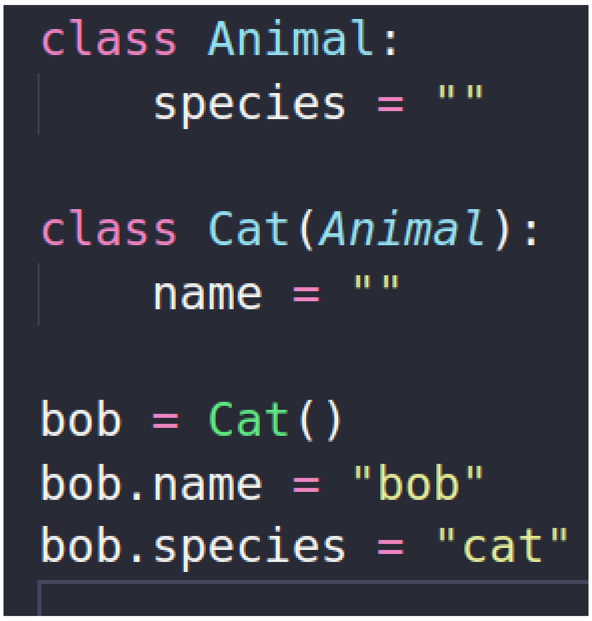
26. Class definition must start with lower case class

27. If the method has a variable number of parameters, we call that method overloading

28. A class definition without implementation of the methods. A class can then inherit this class. This is great for extensibility of your program.

29. If a class inherits, it gets all of the parents class methods. If a class inherits an interface, it gets all of the parents class methods. The difference is that with normal inheritance, the class has method implementation.

30. The solution is this:



```
class Animal:
    species = ""

class Cat(Animal):
    name = ""

bob = Cat()
bob.name = "bob"
bob.species = "cat"
```

31. Create objects based on the methods parameters. Like a real world factory, it creates objects

32. If the number of objects to be created is not known before starting the program. Think an office program, you don't know how many documents the user will open. Think tetris, it's not sure how many blocks (objects) need to be created before starting.

33. The object is converted to data. That data can then be transferred outside of the program

34. Yes, by using object serialization and serialization. For instance, an object can be created in Python, serialized and then deserialized in JavaScript.

35. The languages C, Haskell, Elixir and assembly

36. Reusability (use objects in other programs), Extensibility (using interfaces or inheritance) and Readability (easier than a list of 50.000 instructions)

37. Yes. The buttons are objects. Possibly from the class Button (depending on the implementation of the code)
38. The interface AbstractButton creates empty methods. Possibily the method onClick and onMouseOver. Because there is no implementation of the methods, the child classes CheckBox, PushButton, RadioButton and ToolButton can have their own implementation
39. It creates tight coupling. If you want to change the parent class (add a new method), all child classes need this new method.
40. There is no limit to the number of children a parent can have