1.

```
class Thing:
Pass
print(Thin)
<class '_main_.Thing'>
example = Thing()
print(example)
_main_.Thing object as 0*1006f3fd0>
```

2.

```
class Thing2:
letters = 'abc'
print(Thing2.letters)
abc
```

3.

```
class Thing3:
def__init__(self):
self.letters = 'xyz'
print(Things3.letters)
Traceback (most recent call last):
File '<stdin>",line 1, in <module>
Attributeerror: type object 'Thing3' has no attribute 'letters'
something = Thing3()
print(something.letters)
xyz
```

4.

```
class Element:
def__inint__(self,name,symbol,number):
self.name=name
self.symbol=symbol
self.number=number

hydrogen =Element('Hydrogen','H',1)
```

5.

```
el_dict = {'name':'Hydrogen','sumbol':'H','number':1}

hydrogen= Element(el_dict['name'], el_dict['symbol'], el_dict['number'])

hyrogen.name
'Hydrogen'

hyrogen= Element(**el_dict)
hyrogen.name
'Hydrogen"
```

6.

```
class Element:
def__init__(self,name,symbol,number):
self.name=name
self.symbol=symbol
```

```
self.number=number
def dump(self):
print('name=%s , symbol=%s , number = %s' %
       (self.name, self.symbol,self.number))
hyrogen = Element(**el_dict)
hyrogen.dump()
name=Hydrogen, symbol=H, number=1
```

7.

```
print(hyrogen)
<__main__.Element object at 0*1006f5310>
class Element:
def__init__(self,name,symbol,number):
self.name=name
self.symbol=symbol
self.number=number
def__str__(Self):
return ('name=%s, symbol=%s, number=%s %
           (self.name, self.symbol,self.number))
hyrogen = Element(**el_dict)
print(hyrogen)
name=Hydrogen, symbol=H, number=1
```

8.

```
class Element:
def__init__(self,name,symbol,number):
self.name=name
self.symbol=symbol
self.number=number

@property
def name(Self):
return self.__name
@property
def symbol(Self)
return self.__symbol
@property
def number(Self)
return self.__number

hydrogen = Element('Hydrogen','H',1)
hydrogen.name
'Hydrogen'
hydrogen.symbol
'H'
hyrogen.number
1
```

9.

```
class Bear:
def eats(self):
return 'berries"

class Rabbit:
def eats(self):
return 'clover'
```

```
class Octothorpe:
def eats(Self):
return 'campers'

b=Bear()
r=Rabbit()
o=Octothorpe()
print(b.eats())
berries
print(r.eats())
clover
print(o.eats())
campers
```

10.

```
class Laser:
def does(self):
return 'crush'

class SmartPhone:
def does(Self):
return 'ring'

class Robot:
def__init__(self):
self.laser=Laser()
self.claw=Claw()
self.smartphone=SmartPhone()

def does(Self):
return '''I have many attachments:
My laser, to %s.
My claw. to %s.
My smartphone, to %s. ''' % (
       self.laser.does(),
       self.claw.does(),
       self.smartphone.does())

robbie = Robot()
print(robbie.does())

I have many attachments:
My laser, to disintegrate.
My claw, to crush.
My smartphone, to ring.
```