

Week 5 - Classes

Exercise 1 - Classes in Python

Question 1.

```
In [21]: class Item():
        def __init__(self):
            print('This is an item')

Apple = Item()

This is an item
```

Question 2 and 3.

```
In [22]: class Item():
        def __init__(self, Description, Number, UnitPrice):
            self.Description = Description
            self.Number = Number
            self.UnitPrice = UnitPrice
            print('Created a new item: ', self.Description)
```

Question 4.

```
In [23]: Apple = Item('Apple', 1, 0.5)
        print(Apple.Description, Apple.Number, Apple.UnitPrice)

Created a new item:  Apple
Apple 1 0.5
```

Question 5.

```
In [24]: class Item():
        def __init__(self, Description, Number, UnitPrice):
            self.Description = Description
            self.Number = Number
            self.UnitPrice = UnitPrice
            print('Created a new item: ', self.Description)

        def PrintItemInfo(self):
            print('Item description:', self.Description)
            print('Item number:', self.Number)
            print('Item unit price:', self.UnitPrice)

Apple = Item('Apple', 1, 0.5)
Apple.PrintItemInfo()

Created a new item:  Apple
Item description: Apple
Item number: 1
Item unit price: 0.5
```

Question 6.

```
In [26]: class Item():
        def __init__(self, Description, Number, UnitPrice):
            self.Description = Description
            self.Number = Number
            self.UnitPrice = UnitPrice
            print('Created a new item:', self.Description)

        def PrintItemInfo(self):
            print('Item description:', self.Description)
            print('Item number:', self.Number)
            print('Item unit price:', self.UnitPrice)

        def __str__(self):
            return self.Description

# now to create the shopping list:
Apple = Item('Apple', 1, 0.5)
Orange = Item('Orange', 4, 1)
Coffee = Item('Coffee', 1, 5)

ShoppingList = [Apple, Orange, Coffee]

for l in ShoppingList:
    print(l)

Created a new item: Apple
Created a new item: Orange
Created a new item: Coffee
Apple
Orange
Coffee
```

Question

```
In [27]: total_price = 0
        for l in ShoppingList:
            l.PrintItemInfo()
            total_price = l.Number * l.UnitPrice

        print('The total price is:', total_price)

Item description: Apple
Item number: 1
Item unit price: 0.5
Item description: Orange
Item number: 4
Item unit price: 1
Item description: Coffee
Item number: 1
Item unit price: 5
The total price is: 5
```

Exercise 2 - Inheritance

Question 1.

```
In [31]: class SpecialItem(Item):
        def __init__(self, Description, Number, UnitPrice, SpecialInfo):
            super().__init__(Description, Number, UnitPrice)
            self.SpecialInfo = SpecialInfo
```

Question 2 and 3.

```
In [44]: class SpecialItem(Item):
        def __init__(self, Description, Number, UnitPrice, SpecialInfo):
            super().__init__(Description, Number, UnitPrice)
            self.SpecialInfo = SpecialInfo

        def __str__(self):
            return self.Description + ': ' + self.SpecialInfo

        def PrintItemInfo(self):
            print('Item description:', self.Description)
            print('Item number:', self.Number)
            print('Item unit price:', self.UnitPrice)
            print('Special info:', self.SpecialInfo)
```

Question 4.

```
In [48]: Paracetamol = SpecialItem('Paracetamol', 1, 0.5, 'Take two tablets every six hours')

        print(Paracetamol)
        print('-----')

        Paracetamol.PrintItemInfo()
        print('-----')

        Apple.PrintItemInfo()

Created a new item: Paracetamol
Paracetamol: Take two tablets every six hours
-----
Item description: Paracetamol
Item number: 1
Item unit price: 0.5
Special info: Take two tablets every six hours
-----
Item description: Apple
Item number: 1
Item unit price: 0.5
```

Exercise 3 - Vectors

Question

```
In [50]: from math import *

        sqrt(100)

Out[50]: 10.0
```

Question

```
In [97]: class Vector():
        def __init__(self, x, y, z):
            self.x = x
            self.y = y
            self.z = z

        def __str__(self):
            return '<' + str(self.x) + ', ' + str(self.y) + ', ' + str(self.z) + '>'

        def __add__(self, other):
            return Vector(self.x + other.x, self.y + other.y, self.z + other.z)

        def __mul__(self, other):
            if type(other) == float:
                return Vector(self.x * other, self.y * other, self.z * other)
            elif type(other) == Vector:
                return self.x * other.x + self.y * other.y + self.z * other.z

        def __rmul__(self, other):
            if type(other) == float:
                return self * other

v = Vector(1, 3, 2)
print(v)

w = Vector(5, 0, 1)

print(v + w)

print(v * 2.0)

<1,3,2>
<6,3,3>
<2.0,6.0,4.0>
```

Question

```
In [ ]:
```

Question

```
In [ ]:
```

Question

```
In [ ]:
```

Question

```
In [ ]:
```

Question

```
In [ ]:
```