

Python Class Exercise Set 5

Exercise

- Let's say you have two lists:
 lstA = [1, 2, 3, 4, 5]
 lstB = [10, 20, 30, 40, 50]
- Write a single for loop to add corresponding elements of the two lists, and put the result in another list called lstC
- Then execute the following code:
 print (lstC)

Answer: [11, 22, 33, 44, 55]

Answer

```
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lstC = []
for a, b in zip(lstA, lstB):
    lstC.append(a+b)
print (lstC)
```

Exercise

- Write a function called addLists that takes two lists as parameters, uses a single for loop to add corresponding elements of the two lists, and returns the resulting list
- Then execute the following code:
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]
lstC = addLists (lstA, lstB)
print (lstC)

Answer: [11, 22, 33, 44, 55]

Answer

```
def addLists (lst1, lst2):  
    lst = []  
    for l1, l2 in zip(lst1, lst2):  
        lst.append(l1+l2)  
    return (lst)  
  
lstA = [1, 2, 3, 4, 5]  
lstB = [10, 20, 30, 40, 50]  
  
lstC = addLists (lstA, lstB)  
print (lstC)
```

```
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lstC = []
for a, b in zip(lstA, lstB):
    lstC.append(a+b)
print (lstC)
```

```
def addLists (lst1, lst2):
    lst = []
    for l1, l2 in zip(lst1, lst2):
        lst.append(l1+l2)
    return (lst)

lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lstC = addLists (lstA, lstB)
print (lstC)
```

Exercise

- Write a class called listComposites that can be instantiated with two lists. Then write a class method called addLists that uses a single for loop to add corresponding elements of the two lists, and returns the resulting list
- Then execute the following code:
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]
lc = listComposites(lstA, lstB)
lstC = lc.addLists()
print (lstC)

Answer: [11, 22, 33, 44, 55]

Answer

```
class listComposites:

    def __init__(self, lst1, lst2):
        self.lst1 = lst1
        self.lst2 = lst2

    def addLists(self):
        lst = []
        for l1, l2 in zip(self.lst1, self.lst2):
            lst.append(l1+l2)
        return (lst)

lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lc = listComposites(lstA, lstB)

lstC = lc.addLists()
print (lstC)
```



```
def addLists (lst1, lst2):  
    lst = []  
    for l1, l2 in zip(lst1, lst2):  
        lst.append(l1+l2)  
    return (lst)
```

```
lstA = [1, 2, 3, 4, 5]  
lstB = [10, 20, 30, 40, 50]
```

```
lstC = addLists (lstA, lstB)  
print (lstC)
```

```
class listComposites:
```

```
    def __init__(self, lst1, lst2):  
        self.lst1 = lst1  
        self.lst2 = lst2
```

```
    def addLists(self):  
        lst = []  
        for l1, l2 in zip(self.lst1, self.lst2):  
            lst.append(l1+l2)  
        return (lst)
```

```
lstA = [1, 2, 3, 4, 5]  
lstB = [10, 20, 30, 40, 50]
```

```
lc = listComposites(lstA, lstB)
```

```
lstC = lc.addLists()  
print (lstC)
```

Exercise

- Include another class method called subLists that uses a single for loop to subtract corresponding elements of the two lists, and returns the resulting list
- Then execute the following code:

```
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]
lc = listComposites(lstA, lstB)
lstC = lc.addLists()
print (lstC)
lstC = lc.subLists()
print (lstC)
```

Answer:

[-9, -18, -27, -36, -45]

Answer

```
class listComposites:

    def __init__(self, lst1, lst2):
        self.lst1 = lst1
        self.lst2 = lst2

    def addLists(self):
        lst = []
        for l1, l2 in zip(self.lst1, self.lst2):
            lst.append(l1+l2)
        return (lst)

    def subLists(self):
        lst = []
        for l1, l2 in zip(self.lst1, self.lst2):
            lst.append(l1-l2)
        return (lst)

lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lc = listComposites(lstA, lstB)

lstC = lc.addLists()
print (lstC)

lstC = lc.subLists()
print (lstC)
```

Exercise

- Put the class definition in a module called listModule.
- Then execute the following code:

```
lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]
lc = listModule.listComposites(lstA, lstB)
lstC = lc.addLists()
print (lstC)
lstC = lc.subLists()
print (lstC)
```

Answer:

[11, 22, 33, 44, 55]

[-9, -18, -27, -36, -45]

Answer

listExercise.py

```
import listModule

lstA = [1, 2, 3, 4, 5]
lstB = [10, 20, 30, 40, 50]

lc = listModule.listComposites(lstA, lstB)

lstC = lc.addLists()
print (lstC)

lstC = lc.subLists()
print (lstC)
```

listModule.py

```
class listComposites:

    def __init__(self, lst1, lst2):
        self.lst1 = lst1
        self.lst2 = lst2

    def addLists(self):
        lst = []
        for l1, l2 in zip(self.lst1, self.lst2):
            lst.append(l1+l2)
        return (lst)

    def subLists(self):
        lst = []
        for l1, l2 in zip(self.lst1, self.lst2):
            lst.append(l1-l2)
        return (lst)
```