In [13]:
```
a = [1, 2, 3, 4,6]
b = [11, 22, 33, 44]
c = [-1, -3, -5]
```

In [5]:
```
m=list(map(lambda x,y,z: x+y+z ,a,b,c))
```

In [6]:
```
m
```

Out[6]:
```
[11, 21, 31]
```

In [14]:
```
n=list(zip(a,b))
print(n)
```
```
[(1, 11), (2, 22), (3, 33), (4, 44)]
```

In [21]:
```
x=[1,2,3]
def modify_content(x):
    x[0]=2
    return x
print(x)
print(id(x))
print("-"*50)

y=modify_content(x)
print(y)
print(id(y))
print("-"*50)

print(modify_content(x))
print(id(modify_content(x)))
print("-"*50)

print(x)
print(id(x))
```
```
[1, 2, 3]
140457922924800
--------------------------------------------------
[2, 2, 3]
140457922924800
--------------------------------------------------
[2, 2, 3]
140457922924800
--------------------------------------------------
[2, 2, 3]
140457922924800
```

In [23]:
```
x=[1,2,3]
def replace_content(x):
    x=[4,5,6]
    return x
print(x)
print(id(x))
print("-"*50)

s=replace_content(x)
print(y)
print(id(y))
print("-"*50)

print(replace_content(x))
print(id(replace_content(x)))
print("-"*50)

print(x)
print(id(x))
```
```
[1, 2, 3]
140457922924800
--------------------------------------------------
[4, 5, 6]
140457922912384
--------------------------------------------------
[4, 5, 6]
140457922899136
--------------------------------------------------
[1, 2, 3]
140457922924800
```

In [24]:
```
s=[4,5,6]
print(s)
print(id(s))
```
```
[4, 5, 6]
140457922897344
```

In [25]:
```
s=[9,8,7]
print(s)
print(id(s))
```
```
[9, 8, 7]
140457922835136
```

In [26]:
```
s[0]=2
print(s)
print(id(s))
```
```
[2, 8, 7]
140457922835136
```

In [27]:
```
s1=[4,5,6]
print(id(s1))
s1.append(9)
print(id(s1))
```
```
140457922897536
140457922897536
```

In [28]:
```
s2=[4,5,6]
print(id(s2))
s2[0]=4
print(id(s2))
```
```
140457922907392
140457922907392
```

In [ ]:

In [ ]:

In [29]:
```
class Student:
    pass
```

In [30]:
```
shafiq=Student()
```

In [31]:
```
shafiq
```

Out[31]:
```
<__main__.Student at 0x7fbee06fabe0>
```

In [32]:
```
type(shafiq)
```

Out[32]:
```
__main__.Student
```

In [33]:
```
shafiq.rollnumber=1
```

In [34]:
```
print(shafiq.rollnumber)
```
```
1
```

In [35]:
```
s=Student()
```

In [36]:
```
s
```

Out[36]:
```
<__main__.Student at 0x7fbee06fa880>
```

In [37]:
```
s.name="Rahul"
```

In [38]:
```
print(s.name)
```
```
Rahul
```

In [42]:
```
class Student:
    def __init__(self):
        print(self)
        self.name="shreyansh"
```

In [43]:
```
m=Student()
print(m.name)
```
```
<__main__.Student object at 0x7fbee06fa5b0>
shreyansh
```

In [41]:
```
n=Student()
print(n.name)
```
```
shreyansh
```

In [63]:
```
class Student:
    def __init__(self,new_name):
        self.name=new_name
    def __str__(self):
        return f"student name is {self.name} and his is a briliant student "
```

In [64]:
```
# m=Student('rahul')
# print(m.name)
```

In [66]:
```
s1=Student("mudit")
s2=Student("nikhil")
s3=Student("mohit")
print(s1)
print(s2)
print(s3)
```
```
student name is mudit and his is a briliant student
student name is nikhil and his is a briliant student
student name is mohit and his is a briliant student
```

In [ ]:

In [68]:
```
class Student:
    pass
```

In [69]:
```
s1=Student()
s2=Student()
s3=Student()
s1.name="mudit"
s2.name="mudit"
s3.name="mudit"
print(s1)
print(s2)
print(s3)
```
```
<__main__.Student object at 0x7fbee8995640>
<__main__.Student object at 0x7fbee8995160>
<__main__.Student object at 0x7fbee8995070>
```

In [ ]:

In [82]:
```
class Student:
    def __init__(self,new_name,rollNo=1):
        self.name=new_name
        self.rollno=rollNo
    def __str__(self):
        return f"student name is {self.name} and his rollno is {self.rollno} "
```

In [83]:
```
s1=Student("mudit")
s2=Student("nikhil")
s3=Student("mohit",12)
print(s1)
print(s2)
print(s3)
```
```
student name is mudit and his rollno is 1
student name is nikhil and his rollno is 1
student name is mohit and his rollno is 12
```

In [92]:
```
class Student:
    def __init__(variable,new_name,rollNo=1):
        variable.name=new_name
        variable.rollno=rollNo
    def __str__(variable):
        return f"student name is {variable.name} and his rollno is {variable.rollno} "
```

In [93]:
```
s1=Student("mudit")
s2=Student("nikhil")
s3=Student("mohit",12)
print(s1)
print(s2)
print(s3)
```
```
student name is mudit and his rollno is 1
student name is nikhil and his rollno is 1
student name is mohit and his rollno is 12
```

In [ ]:

In [94]:
```
class Vehicle:
  def __init__(thisVehicle, name, mileage, capacity):
    thisVehicle.name = name
    thisVehicle.mileage = mileage
    thisVehicle.capacity = capacity

  def __str__(thisVehicle):
    return 'Vehicle Name={} \nMileage={} \nCapacity={}'.format(thisVehicle.name, thisVehicle.mileage, thisVehicle.capacity)


truck = Vehicle("Truck", 25, 500)
print(truck)
```
```
Vehicle Name=Truck
Mileage=25
Capacity=500
```

In [ ]:

In [131…
```
class Student:
    counter=100 # class variable
    def __init__(self,new_name):
        self.name=new_name
        Student.counter+=1
        self.rollNo=Student.counter

    def __str__(self):
        return f"student name is {self.name} and his rollno is {self.rollNo} "
```

In [132…
```
s1=Student("mudit")
s2=Student("nikhil")
s3=Student("mohit")
s4=Student("satya")
print(s1)
print(s2)
print(s3)
print(s4)
```
```
student name is mudit and his rollno is 101
student name is nikhil and his rollno is 102
student name is mohit and his rollno is 103
student name is satya and his rollno is 104
```

In [126…
```
print(Student.counter)
```
```
104
```

In [127…
```
print(s1.rollNo)
```
```
101
```

In [113…
```
Student.counter=1000
```

In [114…
```
print(Student.counter)
```
```
1000
```

In [116…
```
print(s1.counter)
```
```
1000
```

In [117…
```
s1.counter=10000
```

In [119…
```
print(s1.counter)
```
```
10000
```

In [120…
```
s5=Student("satya")
print(s5)
```
```
student name is satya and his rollno is 1001
```

In [121…
```
print(s2)
```
```
student name is nikhil and his rollno is 102
```

In [122…
```
print(s1.counter)
```
```
10000
```

In [123…
```
print(s2.counter)
```
```
1001
```

In [133…
```
class Vehicle:

    country = "India"

    def __init__(self, name, mileage):
        self.name = name
        self.mileage = mileage

    def __str__(self):
        return 'Vehicle Name={} \nMileage={}'.format(self.name, self.mileage)

v1 = Vehicle("minivan", 10)

print(v1.country)
v1.country = "USA"
print(Vehicle.country)
print(v1.country)
```
```
India
India
USA
```

In [ ]:

In [141…
```
class Student:
    counter=0 # class variable
    def __init__(self,new_name):
        self.name=new_name
        Student.counter+=1
        self.rollNo=Student.counter
    def intro(self):
        print(f" Hello my name is {self.name}")

    def __str__(self):
        return f"student name is {self.name} and his rollno is {self.rollNo} "
```

In [146…
```
s1=Student("mudit")
s2=Student("nikhil")
s3=Student("mohit")
s4=Student("satya")
print(s1)
print(s2)
print(s3)
print(s4)
s2.intro()
```
```
student name is mudit and his rollno is 17
student name is nikhil and his rollno is 18
student name is mohit and his rollno is 19
student name is satya and his rollno is 20
 Hello my name is nikhil
```

- Create a **class** called `Account`, which **refers to a bank account**

- Create **attributes** that will be unique for each instance of `Account`

1. `id` ---> this has to be **incremented** and **assigned automatically**
2. `bal` ---> this will give balance amount for each account
   - `bal` needs to be **assigned a value as soon as an account is created**
   - As soon as **account is created**, it should have some **opening balance**

- Create **2 instances** of accounts, **a1** and **a2**
  - `a1` should have `id = 1` and `bal = 100`
  - `a2` should have `id = next id` and `bal = 0`

- Create a **string representation** for each account
  - When we print an account, like `print(a1)`
  - It should print out:

  `Account {id} has Rs. {balance}.`

In [149…
```
class Account:
    counter=0
    def __init__ (self,openingbal=0):
        Account.counter+=1
        self.id=Account.counter
        self.bal=openingbal
    def __str__(self):
        return f"Account {self.id} has Rs. {self.bal}."
```

In [150…
```
a1=Account(100)
a2=Account()
print(a1)
print(a2)
```
```
Account 1 has Rs. 100.
Account 2 has Rs. 0.
```

In [151…
```
class Account:
    counter=0
    def __init__ (self,openingbal=0):
        Account.counter+=1
        self.id=Account.counter
        self.bal=openingbal

    def deposit(self,amount):
        self.bal+=amount

    def __str__(self):
        return f"Account {self.id} has Rs. {self.bal}."
```

In [152…
```
a1=Account(100)
print(a1)
a1.deposit(100)
print(a1)
```
```
Account 1 has Rs. 100.
Account 1 has Rs. 200.
```

In [157…
```
class Account:
    counter=0
    def __init__ (self,openingbal=0):
        Account.counter+=1
        self.id=Account.counter
        self.bal=openingbal

    def deposit(self,amount):
        if amount>0: # condition for adding amount
            self.bal+=amount

    def withdraw(self,amount):
        if amount>0 and self.bal>=amount:
            self.bal -= amount

    def __str__(self):
        return f"Account {self.id} has Rs. {self.bal}."
```

In [158…
```
a2=Account(100)
print(a2)
a2.deposit(100)
print(a2)
a2.withdraw(50)
print(a2)
```
```
Account 1 has Rs. 100.
Account 1 has Rs. 200.
Account 1 has Rs. 150.
```

In [155…
```
a2=Account(0)
print(a2)

a2.withdraw(50)
print(a2)
```
```
Account 2 has Rs. 0.
Account 2 has Rs. -50.
```

In [164…
```
a2=Account(100)
print(a2)
a2.withdraw(150)
print(a2)
```
```
Account 7 has Rs. 100.
Account 7 has Rs. 100.
```

In [179…
```
class Account:
    counter=0
    def __init__ (self,openingbal=0):
        Account.counter+=1
        self.id=Account.counter
        self.bal=openingbal

    def deposit(self,amount):
        if amount>0: # condition for adding amount
            self.bal+=amount

    def withdraw(self,amount):
        if amount>0 and self.bal>=amount:
            self.bal -= amount

    def __str__(self):
        return f"Account {self.id} has Rs. {self.bal}."
```

In [180…
```
a1 = Account(100)
a2 = Account()
a1.deposit(50)
print(a1)
a1.withdraw(10)
print(a1)

print(a1)
print(a2)
print(a2)
```
```
Account 1 has Rs. 150.
Account 1 has Rs. 140.
Account 1 has Rs. 140.
Account 2 has Rs. 0.
Account 2 has Rs. 0.
```

In [ ]:

In [ ]:

In [200…
```
x=[1,2,3,4,5]
a=x
b=list(x)
c=x.copy()
d=x[:]
e=copy.deepcopy(x)
print(x)
print(a)
print(b)
print(c)
print(d)
print(e)
```
```
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
```

In [201…
```
print(id(x))
print(id(a))
print(id(b))
print(id(c))
print(id(d))
print(id(e))
```
```
140457785792192
140457785792192
140457922801984
140457108078976
140457922802816
140457376197504
```

In [202…
```
x[0]=100
print(x)
print(a)
print(b)
print(c)
print(d)
print(e)
```
```
[100, 2, 3, 4, 5]
[100, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
```

In [194…
```
import copy
y=[[1,2,3],[4,5,6],[7,8,9]]
a=y
b=list(y)
c=y.copy()
d=copy.deepcopy(y)

print(y)
print(a)
print(b)
print(c)
print(d)
```
```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [195…
```
print(id(y))
print(id(a))
print(id(b))
print(id(c))
print(id(d))
```
```
140457107747968
140457107747968
140457922801792
140457108079168
140457107949888
```

In [196…
```
y[0][0]=100
print(y)
print(a)
print(b)
print(c)
print(d)
```
```
[[100, 2, 3], [4, 5, 6], [7, 8, 9]]
[[100, 2, 3], [4, 5, 6], [7, 8, 9]]
[[100, 2, 3], [4, 5, 6], [7, 8, 9]]
[[100, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [ ]: