

Chap.-8 || Practise Book Que.

P.b.- 557

```
In [1]: 1 class Animal:
2         count = 0 # class variable
3         def __init__(self, a, l):
4             self.arms = a
5             self.legs = l
6             Animal.count += 1
7         def limbs(self):
8             return (self.arms + self.legs)
9
10 spider = Animal(4,4)
11 spidlimbs = spider.limbs()
12 print('Limbs :',spidlimbs) # Limbs : 8
13
14 #----- Extra :-----
15 monkey = Animal(2,2)
16 print(monkey.arms) # 2
17
18 octopus = Animal(8,0)
19 print(octopus.legs) # 0
20
21 print(Animal.count) # 3
```

Limbs : 8

2

0

3

P.b.- 568

```
In [35]: 1 # ['apple', 'banana', 'find', 'dictionary', 'set', 'tuple', 'list', 'malayalam', 'nayan', 'grind', 'apricot']
2
3 class wordPlay():
4     def __init__(self,l):
5         self.l = l
6
7     def words_with_length(self,s):
8         nl = []
9         for i in self.l:
10             if(len(i) == s):
11                 nl.append(i)
12             print(f"Length {s} : {nl}") # ['apple', 'tuple', 'nayan', 'grind']
13
14     def starts_with(self,m):
15         nl = []
16         for i in self.l:
17             if(i[0] == m):
18                 nl.append(i)
19             print(f"Starts with {m} : {nl}")
20
21     def ends_with(self,e):
22         nl = []
23         for i in self.l:
24             if(i[-1] == e):
25                 nl.append(i)
26             print(f"Ends with {e} : {nl}")
27
28     def palindromes(self):
29         nl = []
30         for i in self.l:
31             if(i == i[::-1]):
32                 nl.append(i)
33             print(f"Palindrome : {nl}")
34
35     def only(self,m):
36         nl = []
37         s1 = set(m)
38         for word in self.l:
39             s2 = set(word)
40             if s1 == s2:
41                 nl.append(word)
42
43             print(f"only '{m}' : {nl}")
44
45     def avoids(self, a):
46         nl = []
47         for i in self.l:
48             f = True
49             for c in a:
50                 if(c in i):
51                     f = False
52                     break
53             if f:
54                 nl.append(i)
55             print(f"avoids ('{a}') : {nl}")
56
57 f = open('pb_568.txt')
58 l = f.read().split()
59 print(l)
60
61 ob1 = wordPlay(l)
62 ob1.words_with_length(5)
63 ob1.starts_with('a')
64 ob1.ends_with('d')
65 ob1.palindromes()
66 ob1.only('bna')
67 ob1.avoids('amkd')
68
69 # Output :
70 # ['apple', 'banana', 'find', 'dictionary', 'set', 'tuple', 'list', 'malayalam', 'nayan', 'grind', 'apricot']
71 # Length 5 : ['apple', 'tuple', 'nayan', 'grind']
72 # Starts with a : ['apple', 'apricot']
73 # Ends with d : ['find', 'grind']
74 # Palindrome : ['malayalam', 'nayan']
75 # only 'bna' : ['banana']
76 # avoids ('amkd') : ['set', 'tuple', 'list']
```

```
['apple', 'banana', 'find', 'dictionary', 'set', 'tuple', 'list', 'malayalam', 'nayan', 'grind', 'apricot']
Length 5 : ['apple', 'tuple', 'nayan', 'grind']
Starts with a : ['apple', 'apricot']
Ends with d : ['find', 'grind']
Palindrome : ['malayalam', 'nayan']
only 'bna' : ['banana']
avoids ('amkd') : ['set', 'tuple', 'list']
```

```
In [37]: 1 # List Comprehension
        2
        3 x for x in self.l if len(x) == ln
```

P.b.- 572

```
In [ ]: 1 class SQ:
        2     def __init__(self, initial_list=None):
        3         if initial_list is None:
        4             self.data = []
        5         else:
        6             self.data = initial_list
        7
        8     def shift(self):
        9         if not self.data:
       10             raise IndexError("shift from empty list")
       11         return self.data.pop(0)
       12
       13     def unshift(self, element):
       14         self.data.insert(0, element)
       15
       16     def push(self, element):
       17         self.data.append(element)
       18
       19     def pop(self):
       20         if not self.data:
       21             raise IndexError("pop from empty list")
       22         return self.data.pop()
       23
       24     def remove(self):
       25         if not self.data:
       26             raise IndexError("remove from empty list")
       27         max_element = max(self.data)
       28         self.data.remove(max_element)
       29         return max_element
       30
       31 # Example usage:
       32 sq = SQ([1, 2, 3, 4, 5])
       33 print(sq.shift()) # Output: 1
       34 sq.unshift(0)
       35 print(sq.data)    # Output: [0, 2, 3, 4, 5]
       36 sq.push(6)
       37 print(sq.data)    # Output: [0, 2, 3, 4, 5, 6]
       38 print(sq.pop())   # Output: 6
       39 print(sq.data)    # Output: [0, 2, 3, 4, 5]
       40 print(sq.remove()) # Output: 5
       41 print(sq.data)    # Output: [0, 2, 3, 4]
```

P.b.- 573

```
In [18]: 1 class Atm:
2         def __init__(self):
3             self.pin = None
4             self.bal = 0
5             self.flag = False
6             self.menu()
7
8         def create_pin(self):
9             self.pin = input("Enter a new PIN: ")
10            self.bal = float(input("Initialize your account balance: "))
11            print("PIN created and balance initialized.")
12            self.flag = True
13
14        def change_pin(self):
15            op = input("Enter your current PIN: ")
16            if(op == self.pin):
17                self.pin = input("Enter a new PIN: ")
18                print("PIN changed successfully.")
19            else:
20                print("Enter correct PIN.")
21
22        def check_balance(self):
23            ep = input("Enter your PIN: ")
24            if ep == self.pin:
25                print(f"Your current balance is: {self.bal}")
26            else:
27                print("Enter correct PIN.")
28
29        def withdraw(self):
30            ep = input("Enter your PIN: ")
31            if ep == self.pin:
32                amt = float(input("Enter amount to withdraw: "))
33                if amt > self.bal:
34                    print("Insufficient funds.")
35                else:
36                    self.bal -= amt
37                    print(f"Withdrawal successful. New balance: {self.bal}")
38            else:
39                print("Enter correct PIN.")
40
41        def deposit(self):
42            ep = input("Enter your PIN: ")
43            if ep == self.pin:
44                amt = float(input("Enter amount to deposit: "))
45                self.bal += amt
46                print(f"Deposit successful. New balance: {self.bal}")
47            else:
48                print("Enter correct PIN.")
49
50        def menu(self):
51            while True:
52                print("\nMenu:")
53                print("1. Create PIN")
54                print("2. Change PIN")
55                print("3. Check Balance")
56                print("4. Withdraw Money")
57                print("5. Deposit Money")
58                print("6. Exit")
59                ch = int(input("Enter your choice: "))
60                if ch == 1:
61                    self.create_pin()
62                elif(ch == 2 and self.flag):
63                    self.change_pin()
64                elif(ch == 3 and self.flag):
65                    self.check_balance()
66                elif(ch == 4 and self.flag):
67                    self.withdraw()
68                elif(ch == 5 and self.flag):
69                    self.deposit()
70                elif((ch==1 or ch==2 or ch==3 or ch==4 or ch==5)and(self.flag==False)):
71                    print("Please Create a Pin & and Initialize the bank balalnce!!")
72                    self.create_pin()
73                elif ch == 6 :
74                    print("Exited the System :)")
75                    break
76                else:
77                    print("Invalid choice. Please try again.")
78
79        atm = Atm()
```

```
Menu:
1. Create PIN
2. Change PIN
3. Check Balance
4. Withdraw Money
5. Deposit Money
6. Exit
Enter your choice: 5
Please Create a Pin & and Initialize the bank balalnce!!
Enter a new PIN: 1234
Initialize your account balance: 5000
PIN created and balance initialized.
```

```
Menu:
1. Create PIN
2. Change PIN
3. Check Balance
4. Withdraw Money
5. Deposit Money
6. Exit
Enter your choice: 3
Enter your PIN: 1234
Your current balance is: 5000.0
```

```
Menu:
1. Create PIN
2. Change PIN
3. Check Balance
4. Withdraw Money
5. Deposit Money
6. Exit
Enter your choice: 5
Enter your PIN: 1
Enter correct PIN.
```

```
Menu:
1. Create PIN
2. Change PIN
3. Check Balance
4. Withdraw Money
5. Deposit Money
6. Exit
Enter your choice: 5
Enter your PIN: 1234
Enter amount to deposit: 50
Deposit successful. New balance: 5050.0
```

```
Menu:
1. Create PIN
2. Change PIN
3. Check Balance
4. Withdraw Money
5. Deposit Money
6. Exit
Enter your choice: 6
Exited the System :)
```

P.b.- 566

In []:

1