

18CSC207J-Advance Programming Practice - Structured Programming – Lab Programs

Lab session 4 - Revision of structure, procedure and object oriented Programming

Name :- Puneet Sharma

Reg. No. :- RA1911003010331

Class :-CSE F1

Set:- 9

WEEK-1

1. Friend of three are going for the weight loss challenge. The winner will get exciting prizes from other two.

input:

Challenge day:

friend1-63

friend2-47

friend3-39

one month later:

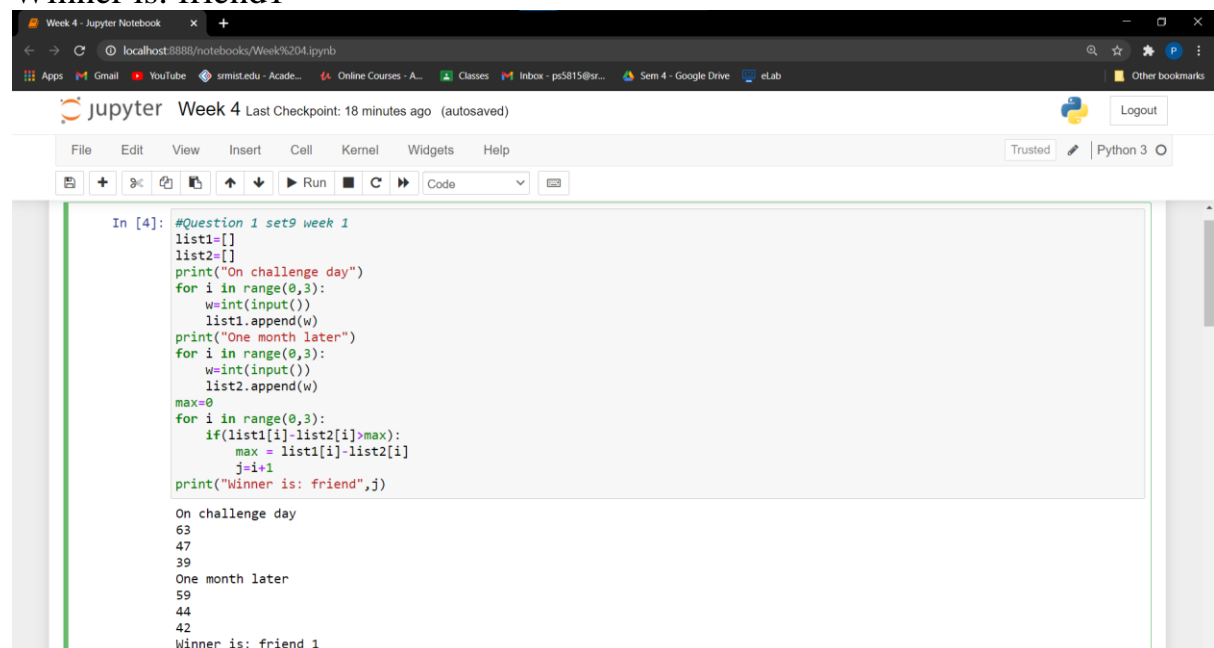
friend1-59

friend2-44

friend3-42

output:

Winner is: friend1



```
In [4]: #Question 1 set9 week 1
list1=[]
list2=[]
print("On challenge day")
for i in range(0,3):
    w=int(input())
    list1.append(w)
print("One month later")
for i in range(0,3):
    w=int(input())
    list2.append(w)
max=0
for i in range(0,3):
    if(list1[i]-list2[i]>max):
        max = list1[i]-list2[i]
        j=i+1
print("Winner is: friend",j)

On challenge day
63
47
39
One month later
59
44
42
Winner is: friend 1
```

2. Using python programming, Find the electrical bill to be paid from the previous and

current reading. Rule slab is given below.

Units 0-100 - Rs.0

Units 101-200 - Rs.1.00

Units 201-300 - Rs. 2.00

Units 301 - more - Rs. 3.00

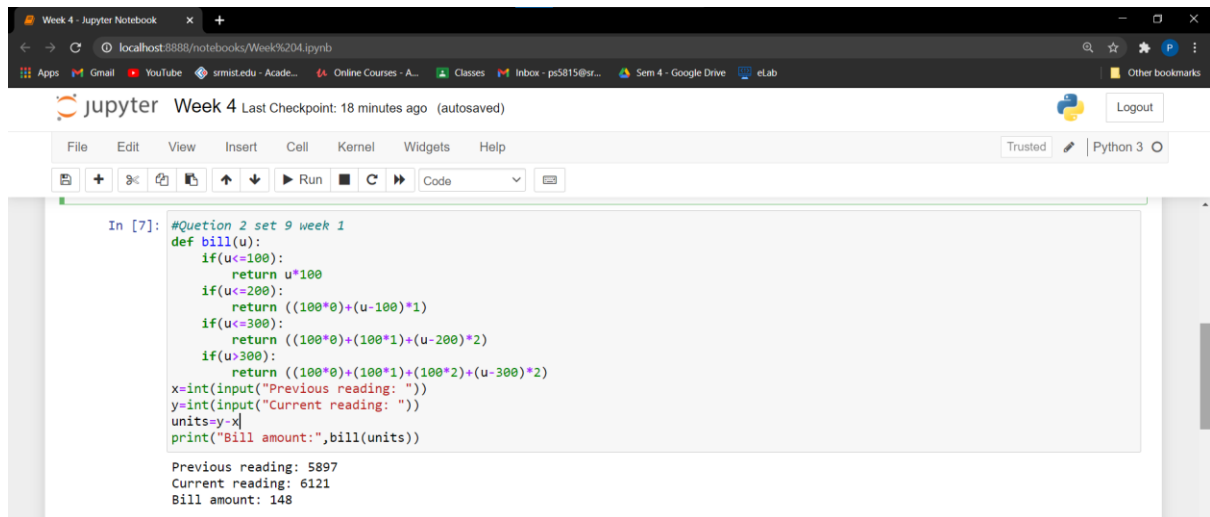
Input:

Previous reading: 5897

Current reading: 6121

Output:

Bill amount: 148



The screenshot shows a Jupyter Notebook window titled "Week 4 - Jupyter Notebook". The browser address bar shows "localhost:8888/notebooks/Week%204.ipynb". The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cell contains the following Python code:

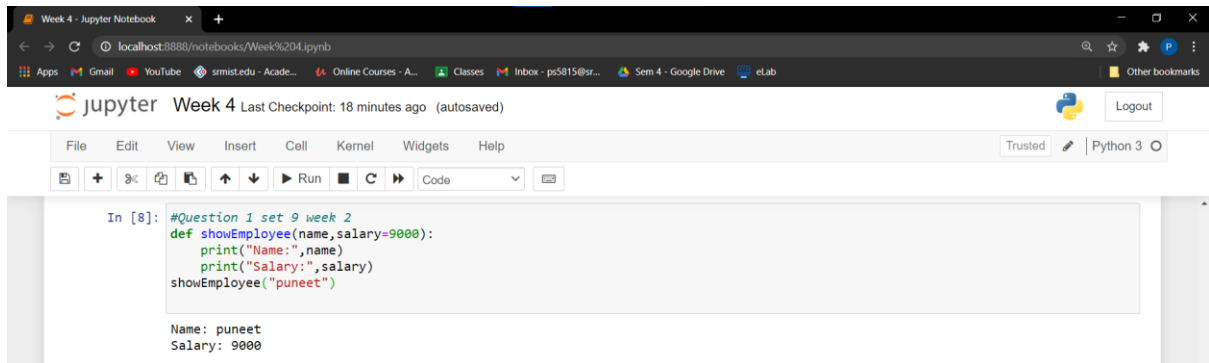
```
In [7]: #Question 2 set 9 week 1
def bill(u):
    if(u<=100):
        return u*100
    if(u<=200):
        return ((100*0)+(u-100)*1)
    if(u<=300):
        return ((100*0)+(100*1)+(u-200)*2)
    if(u>300):
        return ((100*0)+(100*1)+(100*2)+(u-300)*3)
x=int(input("Previous reading: "))
y=int(input("Current reading: "))
units=y-x
print("Bill amount:",bill(units))
```

The output of the code is displayed below the cell:

```
Previous reading: 5897
Current reading: 6121
Bill amount: 148
```

WEEK-2

1. Create a function showEmployee() in such a way that it should accept employee name, and it's salary and display both, and if the salary is missing in function call it should show it as 9000



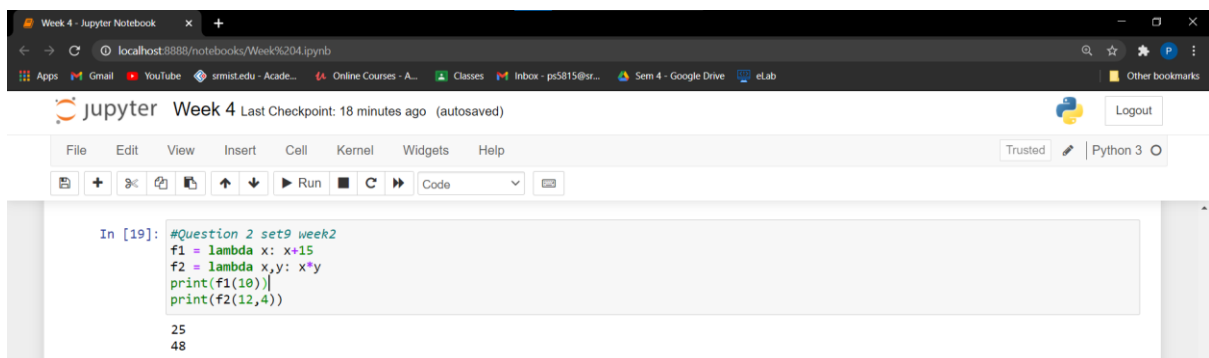
A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Week%204.ipynb'. The notebook title is 'Week 4'. The last checkpoint was 18 minutes ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The code cell contains the following Python code:

```
In [8]: #Question 1 set 9 week 2
def showEmployee(name,salary=9000):
    print("Name:",name)
    print("Salary:",salary)
showEmployee("puneet")
```

The output of the code is displayed below the cell:

```
Name: puneet
Salary: 9000
```

2. Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and print the result.



A screenshot of a Jupyter Notebook interface, similar to the one above. The code cell contains the following Python code:

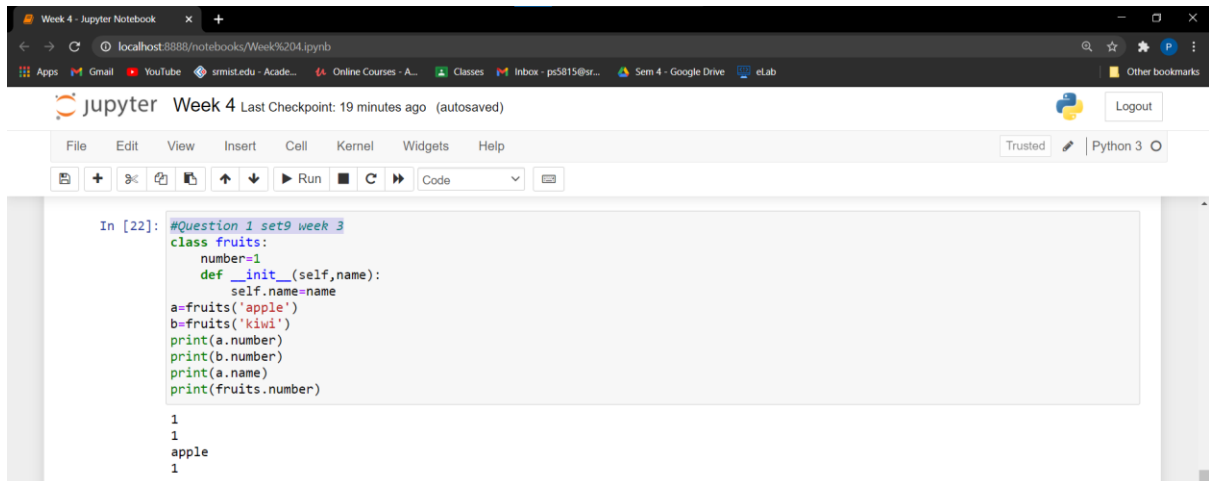
```
In [19]: #Question 2 set9 week2
f1 = lambda x: x+15
f2 = lambda x,y: x*y
print(f1(10))
print(f2(12,4))
```

The output of the code is displayed below the cell:

```
25
48
```

WEEK-3

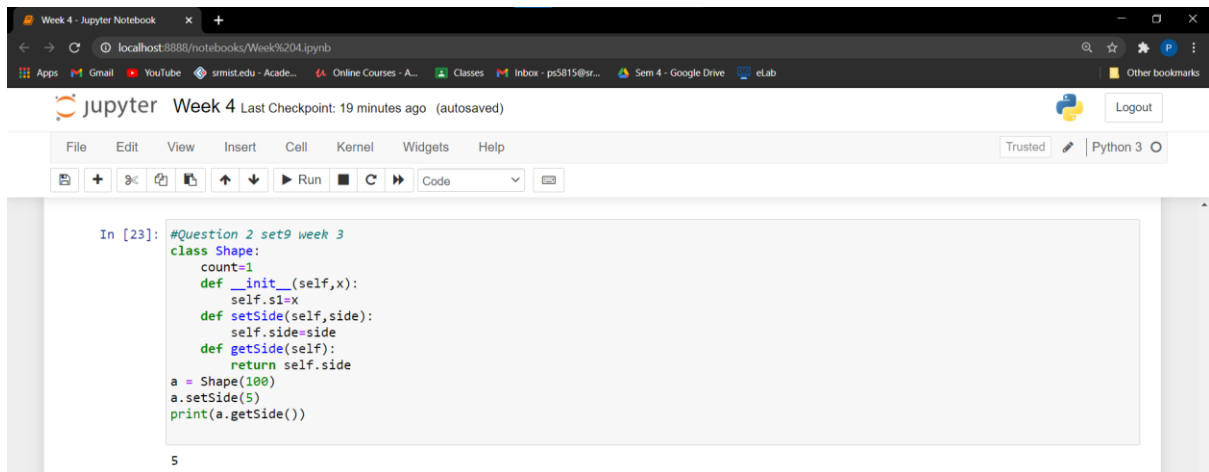
1. Write a Python program to show that the variables with a value assigned in class declaration, are class variables and variables inside methods and constructors are instance variables.



```
In [22]: #Question 1 set9 week 3
class fruits:
    number=1
    def __init__(self,name):
        self.name=name
a=fruits('apple')
b=fruits('kiwi')
print(a.number)
print(b.number)
print(a.name)
print(fruits.number)

1
1
apple
1
```

2. Write a Python program to show that we can create instance variables inside methods



```
In [23]: #Question 2 set9 week 3
class Shape:
    count=1
    def __init__(self,x):
        self.s1=x
    def setSide(self,side):
        self.side=side
    def getSide(self):
        return self.side
a = Shape(100)
a.setSide(5)
print(a.getSide())

5
```