

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

## Lab session 3 - Week 3 Object Oriented Programming Paradigm

Name :- Puneet Sharma

Reg. No. :- RA1911003010331

Class :-CSE F1

### Set 2

1. Write a Python class to find the two elements that sum to zero from a set of n real numbers.

Go to the editor

Input array : [-25, -10, -7, 7, 10, -3, 8, 3]

Output : [[-10,10], [-7,7], [-3,3]]

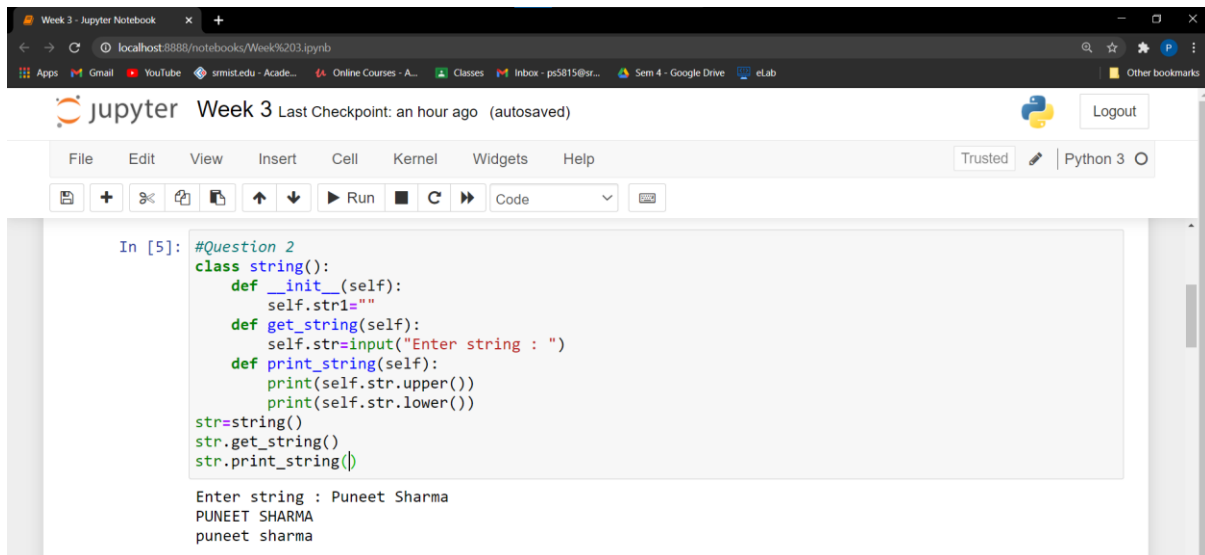
Solution:

```
In [12]: #Question 1
class pairs:
    def zerosum(self,a,n):
        result =[]
        for i in range (0,n-1):
            for j in range (i+1,n):
                if(a[i]+a[j]==0):
                    result.append([a[i],a[j]])
        return result
a=[-25,-10,-7,7,10,-3,8,3]
n=len(a)
print(sum().zerosum(a,n))

[[-10, 10], [-7, 7], [-3, 3]]
```

2. Write a Python class which has two functions `get_Str` and `print_Str`. `get_Str` accept a string from the user and `print_Str` print the string in upper case and lower case.

Solution :-



The screenshot shows a Jupyter Notebook window titled "Week 3 - Jupyter Notebook". The browser address bar shows "localhost:8888/notebooks/Week%203.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 code, and output. The code cell contains the following Python code:

```
In [5]: #Question 2
class string():
    def __init__(self):
        self.str1=""
    def get_string(self):
        self.str=input("Enter string : ")
    def print_string(self):
        print(self.str.upper())
        print(self.str.lower())

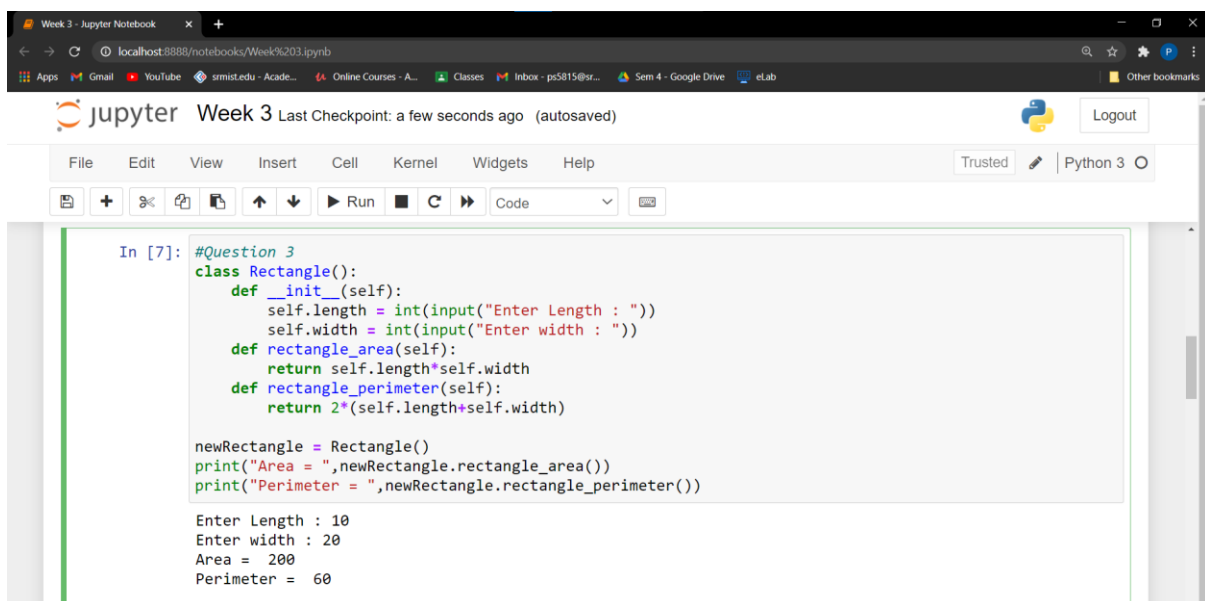
str=string()
str.get_string()
str.print_string()
```

The output of the code is:

```
Enter string : Puneet Sharma
PUNEET SHARMA
puneet sharma
```

3. Write a Python class named `rectangle` constructed by a length, breadth, and two methods which will compute the area and the perimeter of a rectangle.

Solution:-



The screenshot shows a Jupyter Notebook window titled "Week 3 - Jupyter Notebook". The browser address bar shows "localhost:8888/notebooks/Week%203.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 code, and output. The code cell contains the following Python code:

```
In [7]: #Question 3
class Rectangle():
    def __init__(self):
        self.length = int(input("Enter Length : "))
        self.width = int(input("Enter width : "))
    def rectangle_area(self):
        return self.length*self.width
    def rectangle_perimeter(self):
        return 2*(self.length+self.width)

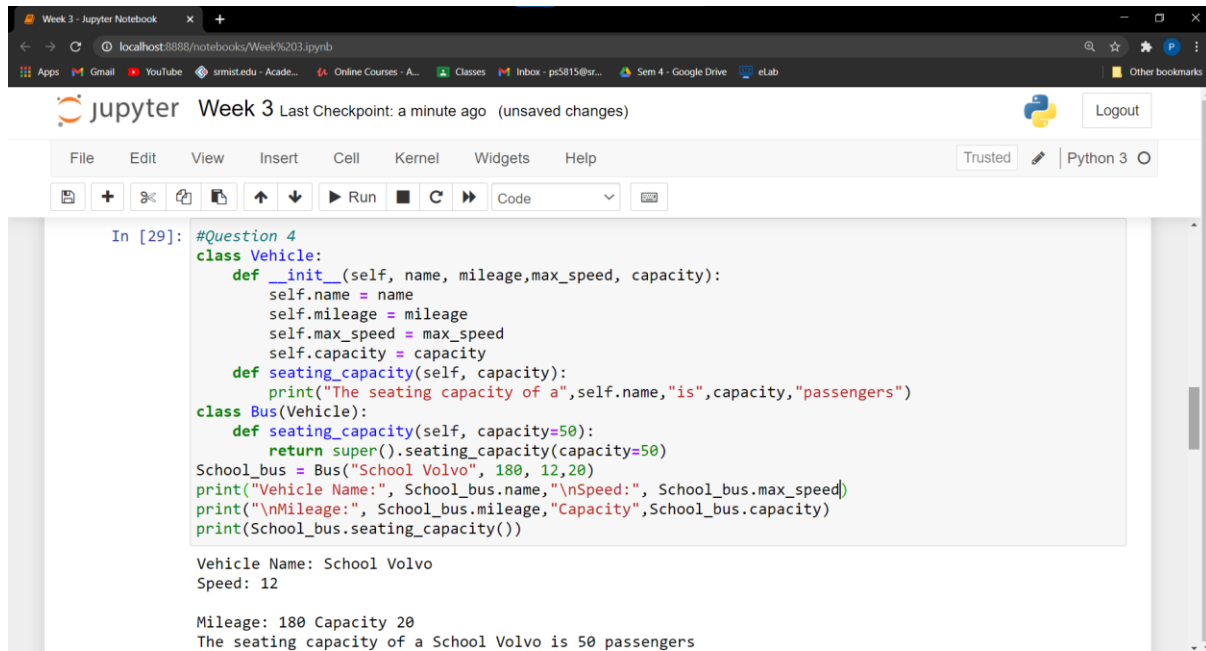
newRectangle = Rectangle()
print("Area = ",newRectangle.rectangle_area())
print("Perimeter = ",newRectangle.rectangle_perimeter())
```

The output of the code is:

```
Enter Length : 10
Enter width : 20
Area = 200
Perimeter = 60
```

4. A vehicle class consists of vehicle data and standard methods to manipulate vehicle data. Write a Python child class Bus that will inherit all of the variables and methods of the Vehicle class. Child class consists of new methods that update the value of vehicle class and display in the output.

Solution:-



The screenshot shows a Jupyter Notebook interface with a browser window at the top displaying 'localhost:8888/notebooks/Week%203.ipynb'. The notebook title is 'Week 3' and it shows 'Last Checkpoint: a minute ago (unsaved changes)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations and a 'Run' button. The code cell is labeled 'In [29]:' and contains the following Python code:

```
#Question 4
class Vehicle:
    def __init__(self, name, mileage, max_speed, capacity):
        self.name = name
        self.mileage = mileage
        self.max_speed = max_speed
        self.capacity = capacity
    def seating_capacity(self, capacity):
        print("The seating capacity of a", self.name, "is", capacity, "passengers")
class Bus(Vehicle):
    def seating_capacity(self, capacity=50):
        return super().seating_capacity(capacity=50)
School_bus = Bus("School Volvo", 180, 12, 20)
print("Vehicle Name:", School_bus.name, "\nSpeed:", School_bus.max_speed)
print("\nMileage:", School_bus.mileage, "Capacity", School_bus.capacity)
print(School_bus.seating_capacity())
```

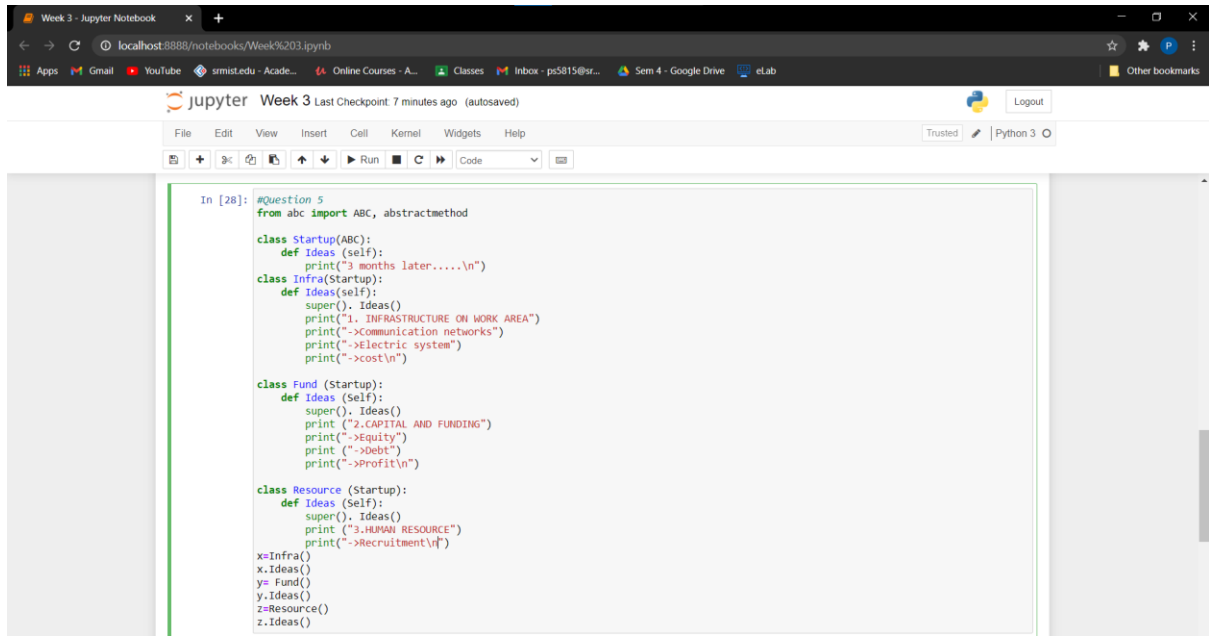
The output of the code is displayed below the cell:

```
Vehicle Name: School Volvo
Speed: 12

Mileage: 180 Capacity 20
The seating capacity of a School Volvo is 50 passengers
```

5. Friend of two are jointly planning for the startups, during month one their discussion was about the, “1. Infrastructure of the work area” & “2. Capital and funding” and “3. Human Resources” Three months later they started implementing. Represent this concept by using Abstract class.

Solution:-



```
In [28]: #Question 5
from abc import ABC, abstractmethod

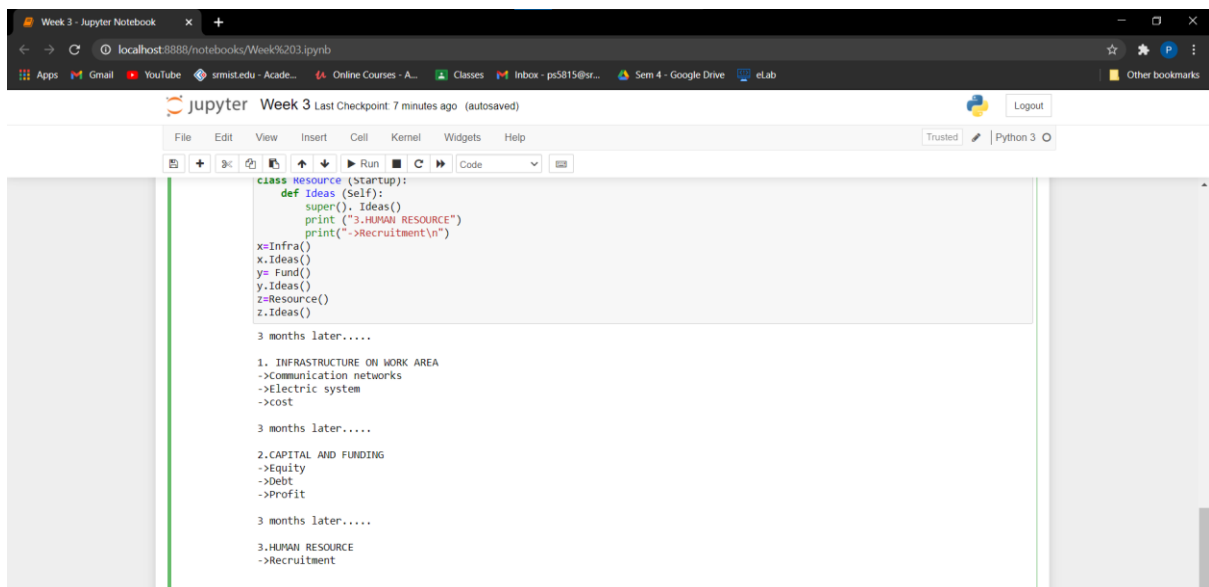
class Startup(ABC):
    def Ideas(self):
        print("3 months later.....\n")

class Infra(Startup):
    def Ideas(self):
        super().Ideas()
        print("1. INFRASTRUCTURE ON WORK AREA")
        print("->Communication networks")
        print("->Electric system")
        print("->cost\n")

class Fund(Startup):
    def Ideas(self):
        super().Ideas()
        print("2.CAPITAL AND FUNDING")
        print("->Equity")
        print("->Debt")
        print("->Profit\n")

class Resource(Startup):
    def Ideas(self):
        super().Ideas()
        print("3.HUMAN RESOURCE")
        print("->Recruitment\n")

x=Infra()
x.Ideas()
y=Fund()
y.Ideas()
z=Resource()
z.Ideas()
```



```
class Resource(Startup):
    def Ideas(self):
        super().Ideas()
        print("3.HUMAN RESOURCE")
        print("->Recruitment\n")

x=Infra()
x.Ideas()
y=Fund()
y.Ideas()
z=Resource()
z.Ideas()

3 months later.....

1. INFRASTRUCTURE ON WORK AREA
->Communication networks
->Electric system
->cost

3 months later.....

2.CAPITAL AND FUNDING
->Equity
->Debt
->Profit

3 months later.....

3.HUMAN RESOURCE
->Recruitment
```