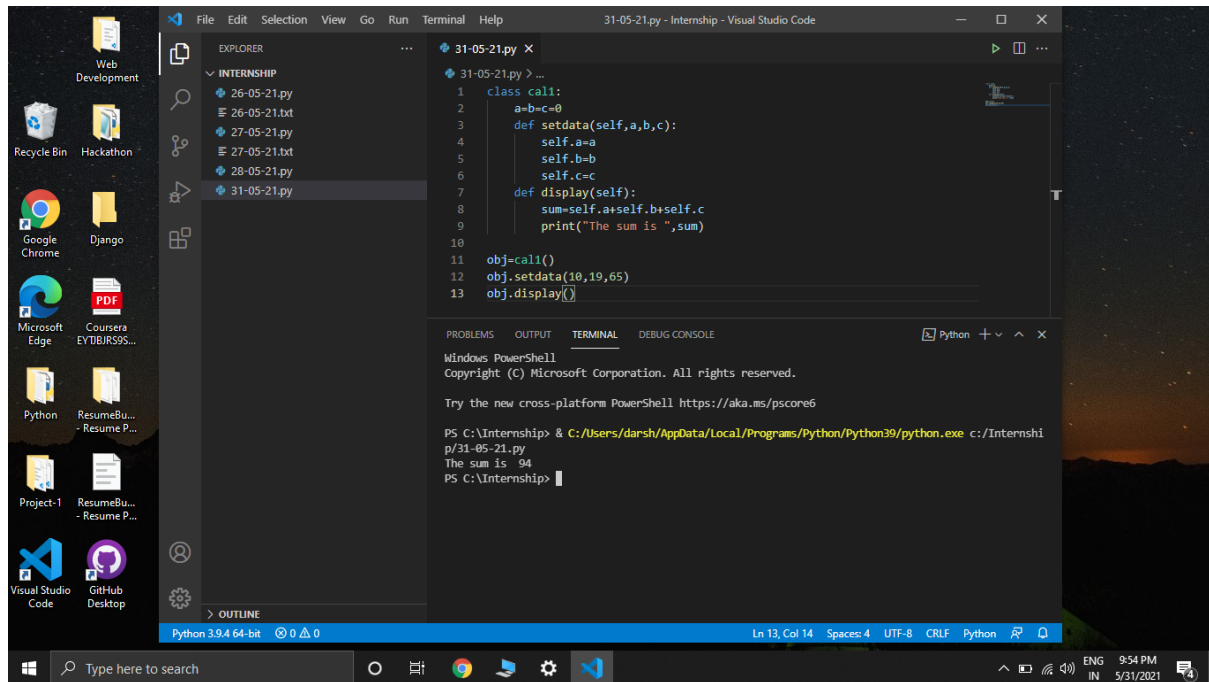
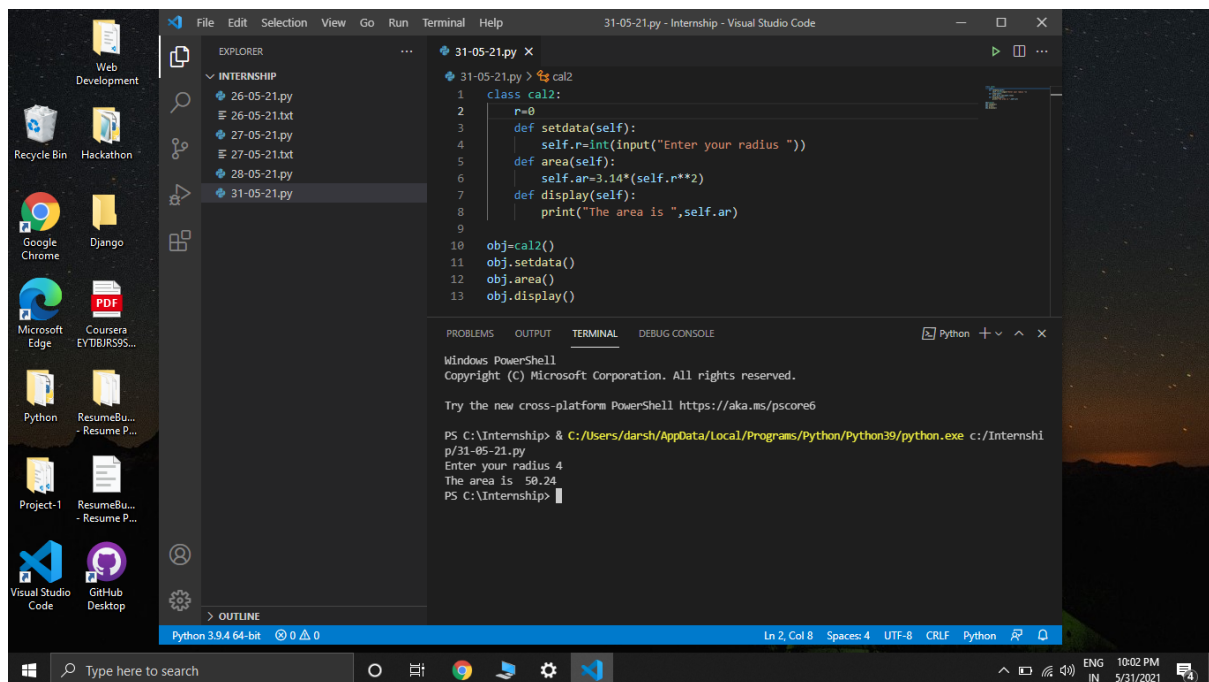


# Internship task of Day-7(31-05-21)

1)



2)



3)

The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying a folder named 'INTERNSHIP' containing several Python files. The main editor window shows a file named '31-05-21.py' with the following Python code:

```

1 class cal3:
2     p=r=n=0
3     def __init__(self,p,r,n):
4         self.p,self.r,self.n=p,r,n
5     def callinterest(self):
6         self.interest=(self.p*self.r*self.n)/100
7     def display(self):
8         print("The interest is ",self.interest)
9
10 obj=cal3(14000,5,24)
11 obj.callinterest()
12 obj.display()

```

The terminal at the bottom shows the command to run the script and its output:

```

PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The interest is  16800.0
PS C:\Internship>

```

4)

The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying a folder named 'INTERNSHIP' containing several Python files. The main editor window shows a file named '31-05-21.py' with the following Python code:

```

1 class cal4:
2     a=0
3     def setdata(self,a):
4         self.a=a
5     def display(self):
6         return self.a**2
7
8 obj=cal4()
9 obj.setdata(8)
10 square=obj.display()
11 print("The square is ",square)
12

```

The terminal at the bottom shows the command to run the script and its output:

```

PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The square is  64
PS C:\Internship>

```

5)

The screenshot shows the Visual Studio Code editor with a file named `31-05-21.py`. The code defines a class `employee` with a class attribute `name-designation=""`. A subclass `employeesub` inherits from `employee` and overrides the `display` method. The `display` method prints the employee's name, designation, and salary. An object `obj` is created from `employeesub` and the `display` method is called with arguments "Darsh", "Data Scientist", and "90000".

```
1 class employee:
2     name-designation=""
3
4 class employeesub(employee):
5     salary=0
6     def display(self,name,designation,salary):
7         self.name,self.designation,self.salary=name,designation,salary
8         print("The name of employee is ",self.name," His designation is ",self.designation," His salary is ",self.salary)
9
10 obj=employeesub()
11 obj.display("Darsh","Data Scientist","90000")
```

The terminal output shows the execution of the script:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The name of employee is Darsh His designation is Data Scientist His salary is 90000
PS C:\Internship>
```

6)

The screenshot shows the Visual Studio Code editor with a file named `31-05-21.py`. The code defines a class `cal5` with an `__init__` method that initializes `length` and `breadth` attributes. It also has a `calArea` method that calculates the area and a `display` method that prints the area. An object `obj` is created from `cal5` with arguments 10 and 15, and the `calArea` and `display` methods are called.

```
1 class cal5:
2     def __init__(self,length,breadth):
3         self.length,self.breadth=length,breadth
4     def calArea(self):
5         self.area=self.length*self.breadth
6     def display(self):
7         print("The area of the rectangle is ",self.area)
8
9 obj=cal5(10,15)
10 obj.calArea()
11 obj.display()
12
13
14
```

The terminal output shows the execution of the script:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The area of the rectangle is 150
PS C:\Internship>
```

7)

The screenshot shows the Visual Studio Code editor with a file named `31-05-21.py`. The code defines a class `cal6` with methods `setdata`, `area`, and `display`. The `setdata` method takes user input for the length of a square. The `area` method calculates the area as `length**2`. The `display` method prints the area. An object `obj` is created and its methods are called.

```
1 class cal6:
2     def setdata(self):
3         self.length=int(input("Enter the length of square "))
4     def area(self):
5         self.area=self.length**2
6     def display(self):
7         print("The area of the square is ",self.area)
8
9 obj=cal6()
10 obj.setdata()
11 obj.area()
12 obj.display()
13
14
```

The terminal output shows the execution of the script:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
Enter the length of square 14
The area of the square is 196
PS C:\Internship>
```

8)

The screenshot shows the Visual Studio Code editor with a file named `31-05-21.py`. The code defines two classes: `book` and `tape`, both inheriting from a base class `publisher`. The `book` class has attributes `title` and `pageno`, and methods `setdata` and `display`. The `tape` class has attributes `playingtime` and `ptime`, and methods `setdata` and `display`. Objects `obj1` and `obj2` are created and their methods are called.

```
1 class publisher:
2     title=""
3
4 class book(publisher):
5     pageno=0
6     def setdata(self,title,pno):
7         self.title=title
8         self.pageno=pno
9     def display(self):
10        print("The title of book is ",self.title)
11        print("The page number is ",self.pageno)
12
13 class tape(publisher):
14     playingtime=0
15     def setdata(self,ptime):
16         self.playingtime=ptime
17     def display(self):
18         print("The playing time is ",self.playingtime)
19
20 obj1=book()
21 obj2=tape()
22 obj1.setdata("Mystic Falls",71)
23 obj2.setdata(98)
24 obj1.display()
25 obj2.display()
```

The terminal output shows the execution of the script:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The title of book is Mystic Falls
The page number is 71
The playing time is 98
PS C:\Internship>
```

9)

The screenshot shows a Visual Studio Code editor window with a file named `31-05-21.py`. The code defines a `customer` class with attributes `scheme_id`, `scheme_name`, `outgoing_rate`, `message_charge`, `cust_id`, and `mobile_no`. It includes methods `setdata` and `display`. The `display` method prints the values of these attributes. The code is executed, and the output is shown in the terminal window at the bottom.

```
1 class scheme:
2     scheme_id=0
3     scheme_name=""
4 class customer(scheme):
5     cust_id=0
6     name=""
7     def setdata(self,scheme_id,name,rate,charge,cust_id,mno):
8         self.scheme_id=scheme_id
9         self.scheme_name=name
10        self.outgoing_rate=rate
11        self.message_charge=charge
12        self.cust_id=cust_id
13        self.mobile_no=mno
14    def display(self):
15        print("The scheme id is ",self.scheme_id)
16        print("The name of scheme is ",self.name)
17        print("The outgoing rate is ₹",self.outgoing_rate," per minute")
18        print("The message cahrges is ₹",self.message_charge)
19        print("The customer id is ",self.cust_id)
20        print("The customer mobile number is ",self.mobile_no)
21
22 obj=customer()
23 obj.setdata(194,"Cricket Season Plan","1","2","67","9409180436")
24 obj.display()
```

Terminal Output:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The scheme id is 194
The name of scheme is
The outgoing rate is ₹ 1 per minute
The message cahrges is ₹ 2
The customer id is 67
The customer mobile number is 9409180436
```

10)

The screenshot shows a Visual Studio Code editor window with a file named `31-05-21.py`. The code defines a `arith` class with attributes `answer`, `a`, and `b`. It includes methods `__init__`, `add`, `subtract`, and `multiply`. The `add`, `subtract`, and `multiply` methods print the result of the operation. The code is executed, and the output is shown in the terminal window at the bottom.

```
1 class arith:
2     answer=0
3     def __init__(self,a,b):
4         self.a=a
5         self.b=b
6     def add(self):
7         self.answer=self.a+self.b
8         print("The addition is ",self.answer)
9     def subtract(self):
10        self.answer=self.a-self.b
11        print("The subtraction is ",self.answer)
12    def multiply(self):
13        self.answer=self.a*self.b
14        print("The multiplication is ",self.answer)
15
16
17 obj=arith(15,8)
18 obj.add()
19 obj.subtract()
20 obj.multiply()
21
```

Terminal Output:

```
PS C:\Internship> & C:/Users/darsh/AppData/Local/Programs/Python/Python39/python.exe c:/Internship/31-05-21.py
The addition is 23
The subtraction is 7
The multiplication is 120
PS C:\Internship>
```