DSA Lab-2
Hasan Amin
374866

OOP:

```
class Flights:
    def __init__(self):
        self.__flightNo=0
        self.__destination=""
        self.__fuel=0.0
        self.__distance=0.0
    def calfuel(self):
        if self.__distance<=1000:
            self.__fuel=500
        elif self.__distance>1000 and self.__distance<=2000:
            self.__fuel=1100
        else:
            self.__fuel=2200
    def feedinfo(self,flightNo,destination,distance):
        self.__flightNo=flightNo
        self.__destination=destination
        self.__distance=distance

    def show_info(self):
        print("Flight Number:", self.__flightNo)
        print("Destination:", self.__destination)
        print("Distance:", self.__distance, "miles")
        print("Fuel Required:", self.__fuel, "gallons")


class Batsman:
    def __init__(self):
        self.__bcode=0
        self.__bname=""
        self.__innings=0
        self.__notout=0
        self.__runs=0
        self.__batavg=0.0
    def calcavg(self):
        try:
            self.__batavg =self.__runs/(self.__innings-self.__notout)
        except ZeroDivisionError:
            pass
    def readdata(self,bcode:int,bname:str,innings:int,notout:int,runs:int):
        self.__bcode=bcode
        self.__bname=bname
        self.__innings=innings
        self.__notout=notout
        self.__runs=runs
        self.calcavg()
    def __repr__(self):
        return f'Code:{self.__bcode}--{hex(id(self.__bcode))} \n Name:{self.__bname}--
{hex(id(self.__bname))} \n Innings: {self.__innings}--{hex(id(self.__innings))} \n Notouts:
{self.__notout}--{hex(id(self.__notout))} \n Runs: {self.__runs}--{hex(id(self.__runs))}'
```

```
class Person:
    def __init__(self, name):
        self.name = name
        self.last_call=None
        self.last_stuff=None
```

```
    def say(self, stuff):
        self.last_stuff=stuff
        self.last_call=self.say
        return stuff
```

```
    def ask(self, stuff):
        self.last_stuff=stuff
        self.last_call=self.ask
        return self.say("Would you please " + stuff)
```

```
    def greet(self):
        self.last_call=self.greet
        return self.say("Hello, my name is " + self.name)
```

```
    def repeat(self):
        if self.last_call is not None:
            if self.last_stuff is not None:
                return self.last_call(self.last_stuff)
            else:
                return self.last_call()
        else:
            return "No method called yet"
```

```
def main():
    obj=Flights()
    obj.feedinfo(20,"Karachi",2000)
    obj.calfuel()
    obj.show_info()
    print("----------------------")
    obj2=Batsman()
    obj2.readdata(2000,"John",10,5,500)
    print(obj2)
    print("-------------------")
    obj3=Person("Hasan")
    print(obj3.say("Hello Hasan!"))
    print(obj3.repeat())
    print(obj3.greet())
    print(obj3.repeat())
    print(obj3.ask("move aside"))
    print(obj3.repeat())
main()
```

```
week-2/OOPs.py
Flight Number: 20
Destination: Karachi
Distance: 2000 miles
Fuel Required: 1100 gallons
----------------------
Code:2000--0x1b0d575bc30
 Name:John--0x1b0d59a2f70
 Innings: 10--0x7ffc371ee448
 Notouts: 5--0x7ffc371ee3a8
 Runs: 500--0x1b0d575bc70
--------------------
Hello Hasan!
Hello Hasan!
Hello, my name is Hasan
Hello, my name is Hasan
Would you please move aside
Would you please move aside
```

2-D Lists:

```python
list1 = [[1, 2, 3], [2, 3, 3], [1, 3, 3]]

def find_max(mylist):
    max_sum=0
    for i in mylist:
        if sum(i)>max_sum:
            max_sum=sum(i)
            index=list1.index(i)
    return index
```

```python
print(f"The row with the highest sum is at index:{find_max(list1)}")
```

```
PS C:\Users\Hasan\Desktop\University Resources\DSA\La
/University Resources/DSA/Labs/week-2/2d-lists.py"
The row with the highest sum is at index:1
```