DSA Lab-2
Hasan Amin
374866

OOP:

```python
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} \n Name:{self._bname} \n Innings: {self._innings} \n Notouts: {self._notout} \n Runs: {self._runs} \n stored at {hex(id(self))}'




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)
main()

```

```
Flight Number: 20
Destination: Karachi
Distance: 2000 miles
Fuel Required: 1100 gallons
----------------------
Code:2000
 Name:John
 Innings: 10
 Notouts: 5
 Runs: 500
 stored at 0x1394e580810
```

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

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
```