JOY OF COMPUTING USING PYTHON

Experiment-1

Date:10/09/2018

**Objective**:Write a program to implement merge sort algorithm using recursion.

**Code**:

import random

def merge\_sort(any\_list):

length = len(any\_list)

if length == 1:

return any\_list

list\_1 = any\_list[:length//2]

list\_2 = any\_list[length//2:]

list\_1 = merge\_sort(list\_1)

list\_2 = merge\_sort(list\_2)

return merge(list\_1, list\_2)

def merge(list\_1, list\_2):

temp = []

while len(list\_1) != 0 and len(list\_2) != 0:

if list\_1[0]<list\_2[0]:

temp.append(list\_1[0])

list\_1.pop(0)

else:

temp.append(list\_2[0])

list\_2.pop(0)

if len(list\_1) ==0:

flag = 0

else:

flag = 1

if flag ==0:

temp = add\_remaining(temp, list\_2)

else:

temp = add\_remaining(temp, list\_1)

return temp

def add\_remaining(temp, any\_list):

for i in any\_list:

temp.append(i)

return temp

def generate\_list(n):

return [random.randint(1,9) for i in range(n)]

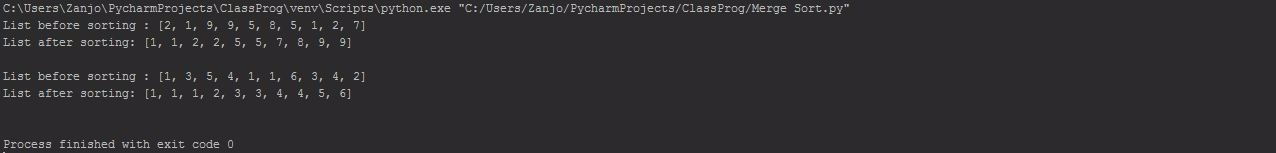
def solve(any\_list):

print("List before sorting :", any\_list, "\nList after sorting:", merge\_sort(any\_list),end="\n\n")

solve(generate\_list(10))

solve(generate\_list(10))

Output :



Experiment-2

**Objective**:Create a class and insert the details of the students such as name , subject and marks and find the average marks of the students in all the subjects .

**Code**:

class Student:

def \_\_init\_\_(self, name,\*\*kwargs):

self.subjects = {}

for subject, marks in kwargs.items():

self.subjects[subject] = marks

self.name = name

def add\_subjects(self,subject, marks):

if subject in self.subjects:

print(subject,"has already been added to the subjects list of the student!")

return

self.subjects[subject] = marks

def find\_average(self):

temp = 0

for marks in self.subjects.values():

temp+=marks

return "Total Marks Scored: "+str(temp)+"\nAverage marks scored: "+str(temp/len(self.subjects))

def \_\_str\_\_(self):

s = "-----Student Details-----" +"\nName: "+self.name+"\nMarks secured in each subject(s):"

for subject, marks in self.subjects.items():

s+="\n"+subject+" = "+ str(marks)

return s

Clay = Student("Clay Jensen",Geography=96, English=92, History=88 )

print(Clay)

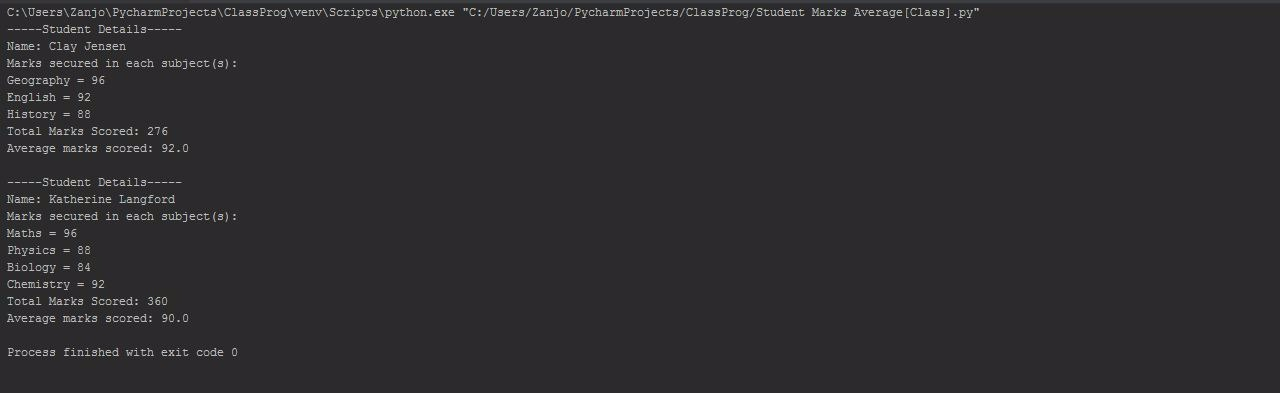
print(Clay.find\_average(),"\n")

Katherine = Student("Katherine Langford",Maths=96, Physics=88, Biology=84)

Katherine.add\_subjects("Chemistry",92)

print(Katherine)

print(Katherine.find\_average())

OUTPUT:

Experiment-3

**Objective**:Create a class and enter the details of three sport events.Take input from user.Use 3 classes for three sport events.

**Code**:

class Sports:

def \_\_init\_\_(self, number\_of\_players, venue, registration\_open):

self.number\_of\_players = number\_of\_players

self.venue = venue

self.registration\_open = registration\_open

def details(self):

print("Number of registered players:", self.number\_of\_players)

print("Venue of the event:", self.venue)

print("Registration Open:", self.registration\_open)

class TableTennis(Sports):

def \_\_init\_\_(self, number\_of\_players, venue, registration\_open):

Sports.\_\_init\_\_(self, number\_of\_players, venue, registration\_open)

def details(self):

print("\*\*\*\*\*\*\*GAME DETAILS\*\*\*\*\*\*\*\nGame: Table Tennis")

Sports.details(self)

class Badminton(Sports):

def \_\_init\_\_(self, number\_of\_players, venue, registration\_open):

Sports.\_\_init\_\_(self, number\_of\_players, venue, registration\_open)

def details(self):

print("Game: Badminton")

Sports.details(self)

class Squash(Sports):

def \_\_init\_\_(self, number\_of\_players, venue, registration\_open):

Sports.\_\_init\_\_(self, number\_of\_players, venue, registration\_open)

def details(self):

print("Game: Squash")

Sports.details(self)

def get\_values(game=""):

temp = list()

print("Enter the game details for "+game+":")

temp.append(int(input("Enter the number of registered players: ")))

temp.append(input("Venue of the event: "))

flag = int(input("Is the registration still open(0/1): "))

if flag == 0:

temp.append(False)

else:

temp.append(True)

return temp

data = get\_values("Table Tennis")

game\_1 = TableTennis(data[0], data[1], data[2])

print()

data = get\_values("Badminton")

game\_2 = Badminton(data[0], data[1], data[2])

print()

data = get\_values('Squash')

game\_3 = Squash(data[0], data[1], data[2])

print()

game\_1.details()

print()

game\_2.details()

print()

game\_3.details()

OUTPUT:

