For each problem below, create the pseudocode and program code solution.

Create a new repository in Github for Session 4. Upload Pseudocode and code the repository.

Post the link to Session 4 Assignment Upload link in Blackboard and paste the link into the text area after pressing the Write Submission button.

Problems

1. Read and study the various built in functions within your programming language. Create a problem and solution in both pseudocode and program code using at least one built in function.

Sum is one of the most used functions in Python. In my case the example is about return value:

numbers = [2.5, 3, 4, -5]  
  
*# start parameter is not provided*numbers\_sum = sum(numbers)  
print(numbers\_sum)  
  
*# start = 10*numbers\_sum = sum(numbers, 10)  
print(numbers\_sum)

After I tested the code the result is:

"C:\Users\Nina Lalova-Toncheva\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:/Users/Nina Lalova-Toncheva/PycharmProjects/pythonProject/main.py"

4.5

14.5

Process finished with exit code 0

1. The problem is to allow a user to enter student last name and two exam scores and determine a total score. Use a function to do this. The first exam score is worth 40% of the total score and the second exam is worth 60%.

Your functions should receive the two exam scores by value, determine the total and return the total score.

Display the student last name, two exam scores and the total.

*# 1. Jack's dictionary*jack = {**"name"**: **"Jack Frost"**,  
 **"assignment"**: [80, 50, 40, 20],  
 **"test"**: [75, 75],  
 **"lab"**: [78.20, 77.20]  
 }  
  
*# 2. James's dictionary*james = {**"name"**: **"James Potter"**,  
 **"assignment"**: [82, 56, 44, 30],  
 **"test"**: [80, 80],  
 **"lab"**: [67.90, 78.72]  
 }  
  
  
*# Function calculates average*def get\_average(marks):  
 total\_sum = sum(marks)  
 total\_sum = float(total\_sum)  
 return total\_sum / len(marks)  
  
  
*# Function calculates total average*def calculate\_total\_average(students):  
 assignment = get\_average(students[**"assignment"**])  
 test = get\_average(students[**"test"**])  
 lab = get\_average(students[**"lab"**])  
  
 *# Return the result based  
 # on weightage supplied  
 # 10 % from assignments  
 # 70 % from test  
 # 20 % from lab-works* return (0.1 \* assignment +  
 0.7 \* test + 0.2 \* lab)  
  
  
*# Calculate letter grade of each student*def assign\_letter\_grade(score):  
 if score >= 90:  
 return **"A"** elif score >= 80:  
 return **"B"** elif score >= 70:  
 return **"C"** elif score >= 60:  
 return **"D"** else:  
 return **"E"***# Function to calculate the total  
# average marks of the whole class*def class\_average\_is(student\_list):  
 result\_list = []  
  
 for student in student\_list:  
 stud\_avg = calculate\_total\_average(student)  
 result\_list.append(stud\_avg)  
 return get\_average(result\_list)  
  
 *# Student list consisting the  
  
  
# dictionary of all students*students = [jack, james]  
  
*# Iterate through the students list  
# and calculate their respective  
# average marks and letter grade*for i in students:  
 print(i[**"name"**])  
 print(**"=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+="**)  
 print(**"Average marks of %s is : %s "** % (i[**"name"**],  
 calculate\_total\_average(i)))  
  
 print(**"Letter Grade of %s is : %s"** % (i[**"name"**],  
 assign\_letter\_grade(calculate\_total\_average(i))))  
  
 print()  
  
*# Calculate the average of whole class*class\_av = class\_average\_is(students)  
  
print(**"Class Average is %s"** % (class\_av))  
print(**"Letter Grade of the class is %s "** % (assign\_letter\_grade(class\_av)))

And after I tested the code:

Jack Frost

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

Average marks of Jack Frost is : 72.79

Letter Grade of Jack Frost is : C

James Potter

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

Average marks of James Potter is : 75.962

Letter Grade of James Potter is : C

Class Average is 72.79

Letter Grade of the class is C

1. This problem entails allowing the user to enter the number of airlines tickets and cost per seat for a purchase. (We will assume all seats are the same price). You are to use a function that receives the number of tickets and price per seat and calculates the cost, tax (7% of cost) and total (cost + tax).

Your function should “return” the cost, tax and total. Therefore, you should pass these arguments by reference. The number of tickets and price per ticket can be passed by value.

Display number of tickets, price per ticket, cost, tax and total.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
|  |  |  |
| tax | Fun\_compute(tax, cost\_per\_seat)  Cost=tix\*cost\_per\_seat  Tax= cost\*0.07  Return cost, tax, total | Tix  Cost\_per\_seat  Cost  Tax  total |
| Cost\_per\_seat |  |  |
|  | Display tix, cost per seat, tax, total |  |
|  |  |  |
|  |  |  |
|  |  |  |

def fun\_calcs (qty\_tix, tix\_price)  
 ext\_price = float(qty\_tix) \* float(tix\_price)  
 tax = ext\_price \* 0.07  
 totale = ext\_price + tax  
 return (ext\_price, tax, total)  
 qty\_tix = input(**"Enter nimber of tickets ordered:"**)  
 tix\_price = input (**"Enter price of a ticket:"**)  
  
 ext\_price, tax, total = fun\_calcs(qty\_tix,tix\_price)  
  
 print(**"Extended Price: "**, ext\_price)  
 print(**"Tax amount: "**, tax)  
 print(**"Total: "**, total)