Track Two - Review of Python Solutions

Old members are going to groan at the sight of seeing this sentence again, but if you have ***any*** questions at all about these solutions, do not hesitate to reach out to Fillip or Arnav. We would be more than happy to help out!

fcutiuba@raleighcharterhs.org
asareen@raleighcharterhs.org

Here are the solutions!

```
word = input("Input a word: ")

for char in range(len(word) - 1, -1, -1):
 print(word[char], end="")
print("\n")


my_password = "FunPassw0rd!@"

def detector (password):
 val = False
 counter = 0
 alphabet =
["a","b","c","d","e","f","g","h","i","j","k","
l","m","n","o","p","q","r","s","t","u","v","w"
, "x", "y","z"]
 numbers =
["1","2","3","4","5","6","7","8","9","0"]
 characters =
```

```
["!","@","#","$","%","^","&","*"]
 for x in alphabet:
   if x in password:
     counter += 1
     print("done")
 for y in numbers:
   if y in password:
     counter += 1
     print("done")
 for z in characters:
   if z in password:
     counter += 1
     print("done")
 if len(password) > 5:
   counter += 1
   print("done")
 if counter > 3:
   val = True

 if val == True:
   print("The password is safe and valid")

 elif val != True:
   print("The password is not safe and not
valid")
```

```
detector(my_password)




fillip = { "name":"Fillip Cutiuba",
         "assignment" : [80, 50, 40, 20],
         "test" : [75, 75],
         "lab" : [78.20, 77.20]
       }


james = { "name":"James Potter",
          "assignment" : [82, 56, 44, 30],
          "test" : [80, 80],
          "lab" : [67.90, 78.72]
        }

arnav = { "name" : "Arnav Sareen",
          "assignment" : [77, 82, 23, 39],
          "test" : [78, 77],
          "lab" : [80, 80]
        }


paul = { "name" : "Paul Walker",
```

```
        "assignment" : [67, 55, 77, 21],
        "test" : [40, 50],
        "lab" : [69, 44.56]
      }


tom = { "name" : "Tom Hanks",
       "assignment" : [29, 89, 60, 56],
       "test" : [65, 56],
       "lab" : [50, 40.6]
     }
 # 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 "F"
 # 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 = [fillip, james, arnav, paul, tom]
 # 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)))
```