**Lesson Number One**

Welcome to this "Challenge Course". Until now we've been leading you by the hand and working on some short and relatively easy projects. This is a **challenge** so be ready. We have faith in you!

We’re going to switch it up a bit and allow you to be the teacher of your own class. Make a gradebook for all of your students.

animal = { "name": "Mr. Fluffles", "sounds": ["meow", "purr"] } print animal["name"]

The example above is just to remind you how to create a dictionary and then to access the item stored by the "name" key.

Create three dictionaries: lloyd, alice, and tyler.

Give each dictionary the keys "name", "homework", "quizzes", and "tests".

Have the "name" key be the name of the student (that is, lloyd's name should be "Lloyd") and the other keys should be an empty list (We'll fill in these lists soon!)

|  |
| --- |
| lloyd = {  "name":"Lloyd",  "homework":[],  "quizzes":[],  "tests":[]  }  alice = {  "name":"Alice",  "homework":[],  "quizzes":[],  "tests":[]  }  tyler = {  "name":"Tyler",  "homework":[],  "quizzes":[],  "tests":[]  } |

# What's the Score?

Great work!

Now fill out your lloyd dictionary with the appropriate scores. To save you some time, we've filled out the rest for you.

Homework: 90.0, 97.0, 75.0, 92.0

Quizzes: 88.0, 40.0, 94.0

Test Scores: 75.0, 90.0

**Make sure to include the decimal points so your grades are stored as floats!** This will be important later.

|  |
| --- |
| lloyd = {  "name": "Lloyd",  "homework": [90.0, 97.0, 75.0, 92.0],  "quizzes": [88.0, 40.0, 94.0],  "tests": [75.0, 90.0]  }  alice = {  "name": "Alice",  "homework": [100.0, 92.0, 98.0, 100.0],  "quizzes": [82.0, 83.0, 91.0],  "tests": [89.0, 97.0]  }  tyler = {  "name": "Tyler",  "homework": [0.0, 87.0, 75.0, 22.0],  "quizzes": [0.0, 75.0, 78.0],  "tests": [100.0, 100.0]  } |

**Put It Together**

Now lets put the three dictionaries in a list together.

my\_list = [1, 2, 3]

The above example is just a reminder on how to create a list. Afterwards, my\_listcontains 1, 2, and 3.

Below your code, create a list called students that contains lloyd, alice, and tyler.

# For the Record

Excellent. Now you need a hard copy document with all of your students' grades.

animal\_sounds = { "cat": ["meow", "purr"], "dog": ["woof", "bark"], "fox": [], } print animal\_sounds["cat"]

The example above is just to remind you how to create a dictionary and then to access the item stored by the "cat" key.

for each student in your students list, print out that student's data, as follows:

* print the student's name
* print the student's homework
* print the student's quizzes
* print the student's tests

|  |
| --- |
| * lloyd = { * "name": "Lloyd", * "homework": [90.0, 97.0, 75.0, 92.0], * "quizzes": [88.0, 40.0, 94.0], * "tests": [75.0, 90.0] * } * alice = { * "name": "Alice", * "homework": [100.0, 92.0, 98.0, 100.0], * "quizzes": [82.0, 83.0, 91.0], * "tests": [89.0, 97.0] * } * tyler = { * "name": "Tyler", * "homework": [0.0, 87.0, 75.0, 22.0], * "quizzes": [0.0, 75.0, 78.0], * "tests": [100.0, 100.0] * } * students = [lloyd, alice, tyler] * print lloyd["name"] * print lloyd["homework"] * print lloyd["quizzes"] * print lloyd["tests"] * print alice["name"] * print alice["homework"] * print alice["quizzes"] * print alice["tests"] * print tyler["name"] * print tyler["homework"] * print tyler["quizzes"] * print tyler["tests"] |

**It's Okay to be Average**

When teaching a class, it's important to take the students' averages in order to assign grades.

5 / 2 # 2 5.0 / 2 # 2.5 float(5) / 2 # 2.5

The above example is a reminder of how division works in Python.

* When you divide an integer by another integer, the result is always an integer (rounded down, if needed).
* When you divide a float by an integer, the result is always a float.
* To divide two integers and end up with a float, you must first use float() to convert one of the integers to a float.

Write a function average that takes a list of numbers and returns the average.

* Define a function called average that has one argument, numbers.
* Inside that function, call the built-in sum() function with the numbers list as a parameter. Store the result in a variable called total.
* Like the example above, use float() to convert total and store the result in total.
* Divide total by the length of the numbers list. Use the built-in len() function to calculate that.
* Return that result.

|  |
| --- |
| * def average(numbers): * total = sum(numbers) * total = float(numbers) * return total / len(number) |

**Just Weight and See**

Great! Now we need to compute a student’s average using weighted averages.

cost = { "apples": [3.5, 2.4, 2.3], "bananas": [1.2, 1.8], } return 0.9 \* average(cost["apples"]) + \ 0.1 \* average(cost["bananas"])

1. In the above example, we create a dictionary called cost that contains the costs of some fruit.
2. Then, we calculate the average cost of apples and the average cost of bananas. Since we like apples much more than we like bananas, we weight the average cost of apples by 90% and the average cost of bananas by 10%.

The \ character is a *continuation character*. The following line is considered a *continuation* of the current line.

Write a function called get\_average that takes a student dictionary (like lloyd, alice, or tyler) as input and returns his/her weighted average.

* Define a function called get\_average that takes one argument called student.
* Make a variable homework that stores the average() of student["homework"].
* Repeat the above step for "quizzes" and "tests".
* Multiply the 3 averages by their weights and return the sum of those three. Homework is 10%, quizzes are 30% and tests are 60%.

|  |
| --- |
| * def get\_average(student): * homework = average(student["homework"]) * quizzes = average(student["quizzes"]) * tests = average(student["tests"]) * total = homework\*0.1 + quizzes\*0.3 + tests\*0.6 * return total |

# Sending a Letter

Great work!

Now let's write a get\_letter\_gradefunction that takes a number score as input and returns a string with the letter grade that that student should receive.

Define a new function called get\_letter\_grade that has one argument called score. Expect score to be a number.

Inside your function, test scoreusing a chain of if: / elif: / else: statements, like so:

* If score is 90 or above: return "A"
* Else if score is 80 or above: return "B"
* Else if score is 70 or above: return "C"
* Else if score is 60 or above: return "D"
* Otherwise: return "F"

Finally, test your function!

Print the resulting letter grade with print. Call the get\_letter\_gradefunction and pass in get\_average(lloyd).

|  |
| --- |
| def get\_letter\_grade(score):  if score >= 90:  return "A"  elif score >=80 and score < 90:  return "B"  elif score >= 70 and score < 80:  return "C"  elif score >= 60 and score < 70:  return "D"  else:  return "F"  print get\_letter\_grade(get\_average(lloyd)) |

# Part of the Whole

Good! Now let's calculate the class average.

You need to get the average for each student and then calculate the average of those averages.

Define a function called get\_class\_average that has one argument class\_list. You can expect class\_list to be a list containing your three students.

First, make an empty list called results.

For each student item in the class\_list, calculate get\_average(student) and then call results.append() with that result.

Finally, return the result of calling average() with results.

|  |
| --- |
| def get\_class\_average(class\_list):  results = []  for student in class\_list:  student\_avg = get\_average(student)  results.append(student\_avg)  return results |

# How is Everybody Doing?

Awesome! You're doing great. Now let's use the functions you've created to check on the grade of the class overall.

Create a list called students and fill it with the three students, alice, lloyd, and tyler.

|  |
| --- |
| students = [alice, lloyd, tyler] |

Find the average grade of the class. Print this numerical grade to the terminal.

print out the result of calling get\_class\_average with your students list.

|  |
| --- |
| print get\_class\_average(students) |

|  |
| --- |
| [91.14999999999999, 80.55, 79.9] |

Finally, determine the letter grade for the class's average and print it to the terminal.

|  |
| --- |
| print get\_letter\_grade(get\_class\_average(students)) |

|  |
| --- |
| lloyd = {  "name": "Lloyd",  "homework": [90.0, 97.0, 75.0, 92.0],  "quizzes": [88.0, 40.0, 94.0],  "tests": [75.0, 90.0]  }  alice = {  "name": "Alice",  "homework": [100.0, 92.0, 98.0, 100.0],  "quizzes": [82.0, 83.0, 91.0],  "tests": [89.0, 97.0]  }  tyler = {  "name": "Tyler",  "homework": [0.0, 87.0, 75.0, 22.0],  "quizzes": [0.0, 75.0, 78.0],  "tests": [100.0, 100.0]  }  # Add your function below!  def average(numbers):  total = sum(numbers)  total = float(total)  return total/len(numbers)  def get\_average(student):  homework = average(student["homework"])  quizzes = average(student["quizzes"])  tests = average(student["tests"])  return 0.1 \* homework + 0.3 \* quizzes + 0.6 \* tests  def get\_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"  def get\_class\_average(class\_list):  results = []  for student in class\_list:  student\_avg = get\_average(student)  results.append(student\_avg)  return average(results)  students = [lloyd, alice, tyler]  print get\_class\_average(students)  print get\_letter\_grade(get\_class\_average(students)) |