Guo, Xilun

```
def get_initial_input():
	reset = 1;
	while (reset == 1):
		num_tests = int(input("Enter the number of tests:  "));
		num_assignments = int(input("Enter the number of assignments:  "));
		num_quizzes = int(input("Enter the number of quizzes:  "));
		num_labs = int(input("Enter the number of labs:  "));
		if(num_tests < 0 or num_assignments < 0 or num_quizzes < 0 or
num_labs < 0):
			reset = int(input("You probably don't want negative number
above, so type 1 to try again: "));
		else:
			return num_tests, num_assignments, num_quizzes, num_labs
def get_weight():
	reset = 1;
	while (reset == 1):
		weight_final = float(input("Enter the weight of final in decimals,if you
don't have final in the course type 0:  "));
		weight_test = float(input("Enter the weight of test in decimals:  "));
		weight_assignment = float(input("Enter the weight of assignment in
decimals:  "));
		weight_quizze = float(input("Enter the weight of quizze in decimals:
"));
		weight_lab = float(input("Enter the weight of lab in decimals:  "));
		if(weight_final + weight_test + weight_assignment +weight_quizze +
weight_lab != 1):
			reset = int(input("weights must equal to 1. Try again? (1-yes,
0-no): "));
		else:
			return weight_final, weight_test, weight_assignment,
weight_quizze, weight_lab;

def get_scores(scores):
	for i in range(len(scores)):
		scores[i] = float(input("Enter scores "));

def calculate_class_grade(test_weighted_avg, assignment_weighted_avg,
quizze_weighted_avg, lab_weighted_avg, final_weighted_avg):
	print((test_weighted_avg + assignment_weighted_avg + quizze_weighted_avg
+ lab_weighted_avg + final_weighted_avg), "%");

def main():
	num_tests, num_assignments, num_quizzes, num_labs = get_initial_input();
```

```
        weight_final, weight_test, weight_assignment, weight_quizze, weight_lab =
get_weight();

        print("Enter the test scores");
        scores = [0] * num_tests;
        get_scores(scores);
        test_weighted_avg = (sum(scores)/num_tests) * weight_test;

        print("Enter the assignment scores");
        scores = [0] * num_assignments;
        get_scores(scores);
        assignment_weighted_avg = (sum(scores)/num_assignments) *
weight_assignment;

        print("Enter the quizze scores");
        scores = [0] * num_quizzes;
        get_scores(scores);
        quizze_weighted_avg = (sum(scores)/num_quizzes) * weight_quizze;

        print("Enter the lab scores");
        scores = [0] * num_labs;
        get_scores(scores);
        lab_weighted_avg = (sum(scores)) * weight_lab;

        print("Enter the final score, if you don't have final type 0: ");
        final_score = float(input("Enter your final score "));
        final_weighted_avg = final_score * weight_final;

        calculate_class_grade(test_weighted_avg, assignment_weighted_avg,
quizze_weighted_avg, lab_weighted_avg, final_weighted_avg);

        redo = int(input( "Do you want to use again to calcuate another course's
grade? 1-yes, 0-no: "));
        if (redo == 1):
                main();
        else:
                exit();
main();
```

1. I change to don't ask the user if they have final in their course or not. I believe it is necessary for me due to it is too much things to worry about if I write yes final or no final. So I think it is much easier for me to ask the user if they don't have final just type 0, if yes just type the weight of their final.
2. In the pervious code, I calculate the final grade by sum all scores and then divide by the max scores and then times the weight. That should be a good way to do. But Jennifer told I don't need to worry about the full scores isn't 100. Hence I change to the easy way.