

```

import tkinter as tk
from tkinter import simpledialog
from tkinter import messagebox
import math

win = tk.Tk()
my_canvas = tk.Canvas(win, width=800, height=400, bg="lightsteelblue")
my_canvas.pack()
win.title("Grade Calculator")
my_label = tk.Label(win, text="Grade Calculator", font="Helvetica 16
bold", bg="lightsteelblue", fg="dimgray")
my_label.place(x=310, y=30)

enter_categories = simpledialog.askstring(title="Create categories",
prompt="How many grade categories would you like?")
enter_desired = simpledialog.askstring(title="Enter your desired grade",
prompt="What is your desired grade?")

# class for if all grades are known
class totalGrade():
    def __init__(self, weight_list, grade_list, desired_grade):
        self.weight = weight_list
        self.grade = grade_list
        self.desired = desired_grade
    def calc(self):
        run_avg = 0
        for item in range(len(self.weight)):
            weights = float(self.weight[item]) * 0.01
            grades = float(self.grade[item])
            points = weights * grades
            run_avg += points
        return run_avg
    def compare(self):
        if self.calc() >= float(self.desired):
            return True
        else:
            return False
    def __str__(self):
        if self.compare() == True:
            return "Your current average is " +
str('{0:.2f}'.format(self.calc())) + ", which meets your desired grade of
" + str('{0:.2f}'.format(int(self.desired))) + "."
        else:
            return "Your current average is " +
str('{0:.2f}'.format(self.calc())) + ", which does not meet your desired
grade of " + str('{0:.2f}'.format(int(self.desired))) + "."

```

```

# class for if one grade is unknown
class ugradeCalc():
    def __init__(self, weights_list, grades_list, desire_grade):
        self.weights = weights_list
        self.grades = grades_list
        self.desire = desire_grade
    def ucalc(self):
        running_avg = 0
        for item in range(len(self.weights)-1):
            weight1 = float(self.weights[item]) * 0.01
            gradel = float(self.grades[item])
            point = weight1 * gradel
            running_avg += point
        needed = (float(self.desire) - running_avg) /
(float(self.weights[-1]) * 0.01)
        return needed
    def __str__(self):
        return "In order to get your desired grade of " +
str('{0:.2f}'.format(int(self.desire))) + ", you need to get a grade of "
+ str('{0:.2f}'.format(self.ucalc())) + " in the selected category."

# class to create boxes and add contents
class addBox():
    def __init__(self, h, categories, canvas):
        self.h = h
        self.categories = categories
        self.canvas = canvas
        self.all_weights = []
        self.all_grades = []
    def makeBoxes(self):
        for i in range(int(self.categories)):
            weight_label = tk.Label(win, text="Weight",
bg="lightsteelblue")
            weight_label.place(x=90, y=self.h-10)
            entry = tk.Entry(win)
            self.canvas.create_window(200, self.h, window=entry)
            self.all_weights.append(entry)
            grade_label = tk.Label(win, text="Grade", bg="lightsteelblue")
            grade_label.place(x=490, y=self.h-10)
            entry1 = tk.Entry(win)
            self.canvas.create_window(600, self.h, window=entry1)
            self.all_grades.append(entry1)
            self.h = self.h + 50
    def getWeights(self):
        return self.all_weights
    def getGrades(self):
        return self.all_grades

```

```

hei = 100
create_boxes = addBox(hei, enter_categories, my_canvas)
create_boxes.makeBoxes()

def inputs(w_list, g_list, d_grade):
    for i in g_list:
        if i == "U":
            uaverage = ugradeCalc(w_list, g_list, d_grade)
            return messagebox.showinfo("Results", uaverage)
    average = totalGrade(w_list, g_list, d_grade)
    return messagebox.showinfo("Results", average)

# initiates calculation and retrieves list
def final():
    global enter_desired, create_boxes
    new_w = []
    new_g = []
    final_weights = create_boxes.getWeights()
    final_grades = create_boxes.getGrades()
    for i in range(len(final_weights)):
        new_w += [final_weights[i].get()]
    for i in range(len(final_grades)):
        new_g += [final_grades[i].get()]
    try:
        inputs(new_w, new_g, enter_desired)
    except:
        messagebox.showerror("Error", "Please enter valid inputs for
grades and weights. Make sure that you enter 'U' last if you have an
unknown category and that your weights are percentages, not decimals.")

# calculate button, should add up what's in boxes
calculate_button = tk.Button(win, text="Calculate", font="Helvetica 12",
bg="firebrick", pady=4, padx=4)
calculate_button.pack()
calculate_button['command'] = final

# quit button
quit_button = tk.Button(win, text="Quit", font="Helvetica 12",
bg="firebrick", pady=4, padx=4)
quit_button.pack()
quit_button['command'] = win.destroy

win.mainloop()

```