```
# CS021 Module 10
# Final Project
# Harrison Davis and Bella Barbera
# November 29, 2018
# Write a program that takes, organizes and outputs homework assignments in order of priority

# Main Function
# Bella B
def main():
    # Title and introduction to program
    print('Welcome to your Homework Priority Manager, or HPM for short, you may call me BOB!')
    print('Please enter your homework assignments below!')

    # Ask user for how many classes they are taking along with error handling to make sure
    # user enters a number not a string
    try:
        classes = int(input('How many classes are you taking? '))
    except ValueError:
        classes = int(input('Please enter a valid input (1,2,3...): '))


    # Creates dictionary course_load which is the main dictionary
    course_load = {}
    # Creates dictionary assignmenting which is the nested dictionary used for ordering the assignments
    assignmenting = {}
    # setting dummy variable
    i = 0
    # Loop that iterates through each course that the user entered with a range of
    # how many courses the user is taking (entered above)
    for i in range(classes):
        # Asks user for the name of the first course
        course = input("Enter the name of a course: ")
        # calling function assignment and passing the number of classes to it, the dummy variable and the
        # nested ordeing dictionary
        assign = assignments(classes, i, assignmenting)
        # assigning the key course and assign as the value to the dictionary course_load
        course_load[course] = assign

    # printing the title for the homework assignment checklist
```

```python
    print()
    print('Here Is Your Homework Checklist:')
    print('------------------------')

    # calling the function cp_calc which sorts the courses and assignments into
    # the order of top priority
    # The dictionary courseload and the dictioary assinmenting are passed to the function cp_calc
    cp_calc(course_load, assignmenting)

# function cp_calc which takes the dictionaries course_load and assignmenting
# sorts the dictionaries based on their values in ascending order and stores the result
# into a list
# Harrison D
def cp_calc(course_load, assignmenting):
    import operator
    # Sorts the course_load dictionary by adding all of the values per class and putting them into
    # a list in ascending order
    dict1 = dict((i,sum(course_load[i].values())) for i in course_load)
    sorted_dict1 = sorted(dict1.items(), key=operator.itemgetter(0), reverse=True)

    # creates new list alpha which contains only the keys in ascending order from sorted_dict1
    alpha = [item[0] for item in sorted_dict1]

    # printing title for Course Order Preference
    print()
    print("Courses in Order of Preference:")
    # Printing the course list in order of priority (highest at top)
    for y in range(len(alpha)):
        print(' - ' + alpha[y])

    # printing title for assignment order preference
    print()
    print("Assignments in Order of Preference:")

    # Sorts the assignmenting dictionary by comparing all of the values per assignment and putting them into
    # a list in ascending order
    sorted_dict2 = sorted(assignmenting.items(), key=operator.itemgetter(1), reverse=True)
    beta = [item[0] for item in sorted_dict2]

    # Printing the assignment list in order of priority (highest at top)
    for b in range(len(beta)):
        print(' - ' + beta[b])


# assignments function which gets all the information from the user regarding assignments for each course and assigns
# the assignments to each course to a dictionary with a nested dictionary within it which has the assignments
# and their associated worth (calcualted priority) which is p
# Harrison D
def assignments(classes, i, assignmenting):
    # creates dictionary assign of assignments
    assign = {}
    again = 'yes'
    # assuming user entered yes or to start the loop
    while again == 'yes':
        # asks use for assignment name
        try:
            inp = input('Enter an assignment: ')
        except ValueError:
            print("Please enter a valid input")
            inp = input('Enter an assignment: ')

        try:
            # asks user for the percentage of the assignment in relation to final grade
            per = int(input('What percent of your final grade is this worth? (to the nearest whole number) '))
            # making sure input in 0-100
            while per < 0 or per > 100:
                per = int(input('Percentage must be between 0-100, try again '))
        except:
            print("Please enter a valid input.")
            per = int(input('What percent of your final grade is this worth? (to the nearest whole number) '))
            while per < 0 or per > 100:
                per = int(input('Percentage must be between 0-100, try again '))

        try:
            # Asking the number of days till the assignment is due
            dd = int(input("How many days till it's due? (If number is greater than 14, just enter 14) "))
            # Making sure user input fits bounds
            while dd < 0 or dd > 14:
                dd = int(input('Value must be greater than zero and less than 15, try again' ))
        except:
            print("Please enter a valid input.")
            dd = int(input("How many days till it's due? (If number is greater than 14, just enter 14) "))
            while dd < 0 or dd > 14:
                dd = int(input('Value must be greater than zero and less than 15, try again'))
        try:
            # Asking how hard the user finds the class
            print('On a scale from 1 -', classes)
            hard = int(input('How hard do you find this class? '))
            # making sure that user enters number between bounds
            while hard > classes or hard < 0:
                hard = int(input('Answer must be between 1 and the number of classes being taken. Try again. '))
        except ValueError:
            print("Please enter valid input.")
            print('On a scale from 1 -', classes)
            hard = int(input('How hard do you find this class? '))
            while hard > classes or hard < 0:
                hard = int(input('Answer must be between 1 and the number of classes being taken. Try again. '))

            # calls priority function which determines the worth of the assignment or how urgent it is
        p = priority(per, dd, hard, classes)

        # asks user if they have another assignment for that course
        again = input('Would you like to add another assignment? (yes/no) ')
        again = again.lower()

        # Checking to make sure user said yes or no
        # Bella B
        while again != 'yes' and again != 'no':
            print('You must answer either "yes" or "no". Try again"')
            again = input('Would you like to add another assignment? (yes/no) ')

        # if no transfer user to enter next course
        if again == 'no' and i != classes-1:
            print('Awesome! Next Course:')
```

```python
        # assigning users input of assigment and its worth to dictionary assignmenting
        assignmenting[inp] = p
        # assigning users input of assigment and its worth to dictionary assign
        assign[inp] = p

    # returning assign value to assignments function
    return assign


# function priority which uses the users input regarding the courses to calculate how urgent
# the class/assignment is which is nessecary for them to be put into order
# Bella B
def priority(p, d, h, c):
    # Convert all values to a scale of 1-100 and calculate a general value
    SCALE1 = 100/c
    hrd = h * SCALE1
    SCALE2 = 100/14
    day = 100 - (d * SCALE2)
    value = int(p + hrd + day)
    # return that value
    return value

# Call main function
main()

```