COMP B10 – INTRO TO PROGRAMMING METHODOLOGIES USING PYTHON
ASSIGNMENT #7 (CLASSES AND OOP)

Create the Python code for a program that will allow a user to maintain a student assignment score database.

I am providing you with a mostly completed starting program including the shells for all the functions and classes that you will need. You will complete the provided shell without adding additional functions, classes or class methods. This assignment is a test to see if you can "read" code and understand OOP.

You will complete the methods (class functions) for the following:

- Course.addStudent
- Course.deleteStudent
- Student.addScore
- Student.getID
- Student.getName
- Student.getGrades

You will complete the program functions for the following:

- addStudent
- displayStudentAverage

Additional notes:

- You MUST use the provided program template.
- Your messages and prompts should look EXACTLY like those in the sample. Duplicating all blank lines and spacing.
- I am providing the datafile.dat file just so you have some starting data, but you don't need this to test your program because the program will create the file if it does not exist.
- You do not need to document your variable for this program.
- Pay attention to spelling and all other things visual -- you will be graded on this.
- Staple multi-page outputs.
- Use appropriate white-space and line-continuations in your source code.
- Upload ONLY your Python program to Canvas in the designated area.

SAMPLE RUN:

```
************************
Grade Management System
************************

Course data loaded.


1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 1

Student Grade Report:

                    ID: 002
                    Name: Mary Jones
                    Average is 89.00%

                    ID: 001
                    Name: Mike Wilson
                    Average is 76.00%




1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 2

Course: COMP B10 - Intro to Structured Programming using Python

Student ID: 002, Student Name: Mary Jones
                    (Quiz 1 - 88.0,100.0)
                    (Midterm Exam - 90.0,100.0)
Student ID: 001, Student Name: Mike Wilson
                    (Quiz 1 - 73.0,100.0)
                    (Quiz 3 - 90.0,100.0)
                    (Midterm Exam - 65.0,100.0)
```

```
1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 3
Enter the NEW student's ID: 004
Enter the NEW student's name: Jim Brady
Student Added

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 3
Enter the NEW student's ID: 001
Student ID already in the database

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 5

Enter student ID: 0004
Student ID not found.

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 5

Enter student ID: 004
Enter the task description: Quiz 1
Enter the points received: 25
Enter the points possible: 30
```

```
Record updated

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 1

Student Grade Report:

                    ID: 002
                    Name: Mary Jones
                    Average is 89.00%

                    ID: 004
                    Name: Jim Brady
                    Average is 83.33%

                    ID: 001
                    Name: Mike Wilson
                    Average is 76.00%



1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 6

Enter student ID: 001
Mike Wilson's average = 76.00%

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 7

Enter student ID: 004
Student ID: 004
```

Student Name: Jim Brady
Quiz 1 - 83.33%

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 4

Enter student ID: 00004
Student ID not found.

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 4

Enter student ID: 004
Are you sure you want to delete this student (Y/N)? y

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 8

Enter student ID: 004
Student ID not found.

1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

```
Select menu option to perform: 8

Enter student ID: 001
Student ID: 001, Student Name: Mike Wilson
                    (Quiz 1 - 73.0,100.0)
                    (Quiz 3 - 90.0,100.0)
                    (Midterm Exam - 65.0,100.0)


1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: 2

Course: COMP B10 - Intro to Structured Programming using Python

Student ID: 002, Student Name: Mary Jones
                    (Quiz 1 - 88.0,100.0)
                    (Midterm Exam - 90.0,100.0)
Student ID: 001, Student Name: Mike Wilson
                    (Quiz 1 - 73.0,100.0)
                    (Quiz 3 - 90.0,100.0)
                    (Midterm Exam - 65.0,100.0)


1. Display course roster and individual averages
2. Dump all course records
3. Add a student
4. Delete a student
5. Add a score to student's record
6. Display a student's overall average
7. Display a student's grades
8. Dump a student's record
Q. Quit program

Select menu option to perform: q

Saving course data.


Run complete. Press the Enter key to exit.
```

```
#-----------------------------------------------------------------------
# Program name: Course Database
# Author:
# Date:
# Purpose: Manage student grades for a course
#-----------------------------------------------------------------------

# FUNCTION DEFINITIONS

import pickle

class Course:
    # The Course class consists of 3 data attributes (string: course ID, string: Course Description,
    # dictionary: of student objects where the KEY is the string: student ID and the VALUE is a
    # student object)

    def __init__(self, cID, cDesc):
        self.__ID = cID
        self.__Desc = cDesc
        self.__students = {}

    def addStudent(self, stuToAdd):
        # if the student ID from the passed student object is in the dictionary
        #   return False (unsuccessful add)
        # else
        #   insert the passed student object into the dictionary AND return True (successful)

    def deleteStudent(self, stuID):
        # if student ID found in dictionary, deletes a student entry from the dictionary
        # otherwise prints "Student ID is not in the database"

    def getStudent(self, stuToGet):
        # if student ID found in dictionary, returns just the student object (dictionary value)
        # from the dictionary
        # otherwise returns None
        if stuToGet in self.__students:
            stuObj = self.__students[stuToGet]
            return stuObj
        else:
            return None

    def getCourseID(self):
        # returns the course id
        return self.__cID
```

```
    def getCourseDescription(self):
        # returns the course description
        return self.__cDesc

    def getStudents(self):
        # returns the dictionary containing the student objects
        return self.__students

    def __str__(self):
        # returns a string containing the course object data attribute dump
        self.__tmp = ""
        for stuKey in self.__students:
            self.__tmp += str(self.__students[stuKey])
        return "\nCourse: " + self.__ID + " - " + self.__Desc + "\n\n" + self.__tmp

class Student:
    # The Student class consists of 3 data attributes (string: student ID, string: student name,
    # list: where each element in a list is 3 values - string: a task name, float: points received
    # for that task, float: points possible for that task

    def  __init__(self, stuID, stuName):
        self.__ID = stuID
        self.__name = stuName
        self.__grades = []

    def addScore(self, taskName, pointsRec, pointsPoss):
        # appends a student task element list to a student task list

    def displayAverage(self):
        # returns the overall average percentage for a student
        totReceived = 0.0
        totPossible = 0.0
        for item, rec, poss in self.__grades:
            totReceived += rec
            totPossible += poss
        if totPossible == 0.0:
            return "No scores to average"
        else:
            return "{:.2%}".format(totReceived/totPossible)

    def getID(self):
        # Returns the student ID
```

```
    def getName(self):
        # Returns the student name

    def getGrades(self):
        # Returns the list containing student grades

    def __str__(self):
        # returns a string containing the student object data attribute dump
        self.__tmp = ""
        for item1, item2, item3 in self.__grades:
            self.__tmp += "\t(" + item1 + " - " + str(item2) + "," + str(item3) + ")\n"
        return "Student ID: " + self.__ID + ", Student Name: " + self.__name + "\n" + self.__tmp

def displayGradeReport():
    # Displays students and their current course average
    print("\nStudent Grade Report:\n")
    for student in myClass.getStudents().values():
        print("\tID: " + student.getID() + "\n\tName: " + student.getName() + "\n\tAverage is " + \
              student.displayAverage() + "\n")
    print("\n")

def getStudent():
    # Returns a student object if found
    stuID = input("\nEnter student ID: ")
    stuObject = myClass.getStudent(stuID)
    if not stuObject:
        print("Student ID not found.")
        return None
    else:
        return stuObject

def addStudent():
    # Prompts user for student info and adds a student to the database
    # Print an error message and prevents addition if student ID already in database

def deleteStudent():
    # Prompts user for a student ID and then deletes the student record (object) from the database
    stuObj = getStudent()
    if stuObj:
        answer = input("Are you sure you want to delete this student (Y/N)? ").upper()
        if answer == "Y":
            myClass.deleteStudent(stuObj.getID())
```

```
def addScore():
    # calls getStudent to get a specific student object and then
    # adds a new scored item to a specific student record
    stuObj = getStudent()
    if stuObj:
        task = input("Enter the task description: ")
        received = float(input("Enter the points received: "))
        possible = float(input("Enter the points possible: "))
        stuObj.addScore(task, received, possible)
        print("Record updated")

def displayStudentAverage():
    # calls getStudent to get a specific student object and then
    # displays a specific student's class average

def displayStudentGrades():
    # calls getStudent to get a specific student object and then
    # displays a specific student's tasks and percentage grades
    stuObj = getStudent()
    if stuObj:
        tmp = ""
        for task, received, possible in stuObj.getGrades():
            tmp += "\t" + task + " - " + "{:.2%}".format(received/possible) + "\n"
        print("Student ID: " + stuObj.getID() + "\nStudent Name: " + stuObj.getName() + "\n" + tmp)

def startUp():
    # Loads student database if file exists. Creates a new one otherwise
    global myClass

    print ("\n" + "*" * 25 + "\nGrade Management System\n" + "*" * 25)

    try:
        dataFile = open("datafile.dat", "rb")
        myClass = pickle.load(dataFile)
        print("\nCourse data loaded.\n")
        dataFile.close()
    except IOError:
        print("\nData file does not exist. Creating new course.\n")
        myClass = Course("COMP B10", "Intro to Structured Programming using Python")

def saveDataAndExit():
    # Saves the student database
    print("\nSaving course data.\n")
    dataFile = open("datafile.dat", "wb")
```

```
    pickle.dump(myClass, dataFile)
    dataFile.close()

def getMenuOption():
    # Displays menu and gets menu selection
    menuStr = "\n" + \
              "1. Display course roster and individual averages\n" + \
              "2. Dump all course records\n" + \
              "3. Add a student\n" + \
              "4. Delete a student\n" + \
              "5. Add a score to student's record\n" + \
              "6. Display a student's overall average\n" + \
              "7. Display a student's grades\n" + \
              "8. Dump a student's record\n" + \
              "Q. Quit program\n\n"

    return input(menuStr + "Select menu option to perform: ").upper()

def main():
    # Controlling routine
    startUp()
    option = getMenuOption()
    while option != "Q":

        if option == "1":
            displayGradeReport()
        elif option == "2":
            print(myClass)
        elif option == "3":
            addStudent()
        elif option == "4":
            deleteStudent()
        elif option == "5":
            addScore()
        elif option == "6":
            displayStudentAverage()
        elif option == "7":
            displayStudentGrades()
        elif option == "8":
            stu = getStudent()
            if stu:
                print(stu)

        option = getMenuOption()
```

```
    saveDataAndExit()

    input("\nRun complete. Press the Enter key to exit.")

#---------------------------------------------------------------------
# PROGRAM'S MAIN LOGIC

main()
```