```
from MP7_StudentClass import Student

#
# Machine Problem 7
# CS 1300
# Bruce Miller
#
# Description:
#
# Get info from txt data and create student objects with: studentd ID, name,
testScores, and testAvg
# Put everything in a dict, with student ID as the Keys, and student objects as
linked value
# Prompt for student ID, use to create new student object or change existing
# If new student: prompt for new studentName, and loop prompt testScores for
received student ID\
# If old student: Ask for change type: name / testScores, and receive related input
# Make changes to student according to input
# Re-prompt testScores and studentID until none value returned
#

def getStudents():
    #
    # Opens the data file of student info... studentID, firstName, lastName, then
    # testScores - the number of test scores is variable, from zero up... reads
    # each line of data as a str, divides the line into the values... str, str, str,
    # int, int, int... (a variable number of int values - could be none)...
    # instantiates a new Student object, uses the Student object's instance methods
    # to add those values to the object, adds the Student object to a dictionary of
    # Student objects using studentID as the key and the Student object as the
value,
    # and returns the dictionary of Student objects.
    #
    # There are no parameters.
    #
    # Returns a dict of objects from the Student class, using student ID as the key
    # and the Student object as the value
    #

    file = open("C:\\Users\\Bruce\\Desktop\\Python Coding\\Class Projects\\1300 -
MP6 Data.txt")
    lines = file.readlines()
    file.close()
    for linePos in range(len(lines)):
        lines[linePos] = lines[linePos].split()

    roster = {}
    for studentInfo in lines:
        currentStudent = Student(studentInfo[0])
        currentStudent.setName(studentInfo[1], studentInfo[2])
        for testScore in studentInfo[3:]:
            currentStudent.addTest(int(testScore))
        roster[studentInfo[0]] = currentStudent
    return roster

def printStudents(roster):
    #
    # Prints the information for each Student object in roster... including
studentID,
    # firstName, lastName, testScores, and average for each object.
    #
    # roster A dict of objects from the Student class, each object contains a
    # studentID(str), firstName(str), lastName(str),
    # testScores(list of int value), and average (float).
    #
    # There is no return value.
    #

    print("Students in roster:")
    for student in roster.values():
        print()
        print(student)

def updateName(roster, stuID):
    #
    # Prompts the user to enter the first name and last name of a student. Then,
    # uses the setName() instance method to change the name for the Student object
    # in the dictionary roster with the key = stuID.
    #
    # roster A dictionary of objects from the Student class, using stuID as the
    # key, and a Student object as the value.
    # stuID A str that is the key for a Student object in the dictionary roster.
    #
    # There is no return value.
    #
    roster[stuID].setName(input("First name: "), input("Last Name: "))


def updateTests(roster, stuID):
    #
    # Prompts the user to enter test scores for a student using addTest() to add
    # each one to the Student object in the dictionary roster with the key = stuID.
    #
    # roster A dictionary of objects from the Student class, using stuID as the
    # key, and a Student object as the value.
    # stuID A str that is the key for a Student object in the dictionary roster.
    #
    # There is no return value.
    #
    currentStudent = roster[stuID]
    score = input("Enter a test Score (<enter> to stop): ")
    while score:
        currentStudent.addTest(int(score))
        score = input("Enter a test Score (<enter> to stop): ")



roster = getStudents()
printStudents(roster)

print()
print()
print("You may update any student info, or add a student.")

print()
currentID = input("Enter Student ID (<enter> to stop): ")
while currentID:
    if currentID in roster:
        print(roster[currentID])

        print()
        print("(1) Change the Name")
        print("(2) Add a Test")
        print("(3) Do Nothing")

        print()
        choice = input("What would you like to do? ")
        if choice == "1":
            updateName(roster, currentID)
        if choice == "2":
            updateTests(roster, currentID)

        print()
        print(roster[currentID])
    else:
        print("This student does not exist. You may enter the info now.")
        roster[currentID] = Student(currentID)
        updateName(roster, currentID)
        updateTests(roster, currentID)

        print()
        print("New Student Added:")
        print(roster[currentID])

    print()
    currentID = input("Enter Student ID (<enter> to stop): ")
```