```python
#
# Machine Problem 6
#
# Drei Luna
#
# Description: This script uses class to print out the Name(with student ID),
# Test scores, and average test score when given a file with that information.
# Now allows for users to change information in the list of students. For example
# changing a test score or name.
#

from student import Student

def makeStudent(studentInfo):
    numTests = len(studentInfo) - 3
    localStudent = Student(studentInfo[0],[studentInfo[1],studentInfo[2]])
    for i in range(numTests):
        localStudent.addTest(studentInfo[i+3])
    return localStudent


def getStudent():
    #
    # 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 list of
objects,
    # and returns the list of Student objects.
    #
    # There are no parameters.
    #
    # Returns a list of objects from the Student class
    #
    roster = {}
    with open('1300 - MP6 Data.txt') as file:
        lines = file.readlines()
        for i in range(len(lines)):
            current_student = lines[i].split()
            roster[current_student[0]] = makeStudent(current_student)
    return roster


def printStudents(roster):
    #
    # Prints the information for each Student object in roster... including
studentID,
    # firstName, lastName, testScores, and average for each object.
```

```python
    #
    #\ roster A list 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.
    #
    for key in roster:
        print(roster[key])

def updateName(roster, stuID):
    if(stuID not in roster):
        print('This student does not exist. You may enter the info now.')
    first = input('First name: ')
    last = input('Last name: ')
    if(stuID not in roster):
            current_student = [stuID, first, last]

            roster[stuID] = makeStudent(current_student)
    else:
        roster[stuID].setName(first, last)

def updateTests(roster, stuID):
    test_score =' '
    score_list = []
    while test_score != '':
        test_score = input('Enter a Test Score (<enter> to stop):')
        score_list.append(test_score)
    for i in range(len(score_list)-1):
        roster[stuID].addTest(score_list[i])

print('Students in Roster:\n')
classList = getStudent()
printStudents(classList)

print('You may update any student info, or add a student.\n')
user_input = input('Enter Student ID (<enter> to stop): ')
while(user_input != ''):

    if(user_input not in classList):
        updateName(classList, user_input)
        updateTests(classList, user_input)
        print('\nNew Student Added:')
    else:
        print(classList[user_input])
        print('(1) Change the Name')
        print('(2) Add a Test')
        print('(3) Do Nothing')
        print()
        action = input('What would you like to do? ')
```

```python
    if(action == '1'):
        updateName(classList, user_input)
    elif(action =='2'):
        updateTests(classList, user_input)
    print()
print(classList[user_input])
user_input = input('Enter Student ID (<enter> to stop): ')
```