

```

#
# Aravind Kumar Kaspe
# Banner ID : 001291145
#
# Description : This program manages student information using a class called
#               Student. So, first it imports Student class and then, the main
#               program reads data from a file, processes it, and stores each
#               student's information in a list of Student objects. The
#               getStudents() function opens a data file, reads student information
#               (including studentID, firstName, lastName, and testScores),
#               creates a new Student object for each student, populates it with
#               the read data, and adds the object to a list. The printOneStudent()
#               function prints the details of a Student object, including
#               studentID, name, testScores, and average. The program also includes
#               a findStudent() function, which conducts a sequential search in the
#               list of Student objects based on studentID or lastName, returning
#               the matching object or None if no match is found. Overall, this
#               program allows for reading, processing, and searching student
#               information, offering functionality for adding new students and
#               updating existing student data.
#
#
from Student import *

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
    # list of objects,and returns the list of Student objects.
    #
    # There are no parameters.
    #
    # Returns a list of objects from the Student class
    #
    with open('5930 - MP5 Data.txt') as data:
        sourceData = data.readlines()
        details = [line.rstrip() for line in sourceData]
        objList = []
        for data in details:
            stuList =data.split()
            student = Student(stuList.pop(0))
            student.setName(stuList.pop(0),stuList.pop(0))
            for test in stuList:
                student.addTest(int(test))
            objList.append(student)
    return objList

```

```

def printOneStudent(student):
    #
    # Prints the information for one Student object including studentID, firstName,
    # lastName, testScores, and average for each object.
    #
    # student An object from the Student class, the object contains a
    # studentID(str), name (list of firstName(str), lastName(str)),
    # testScores(list of int values), and average(float).
    #
    # There is no return value.
    #
    print("Student:", student.getID(), student.getName()[0], student.getName()[1])
    scores = student.getTestScores()
    if len(scores)>0:
        print("Test Scores:",end=" ")
        for score in scores:
            print(score,end=" ")
        print()
        print(f"Test Average: {student.getAvg():1.2f}")
        print()
    else:
        print("Test Scores: ")
        print(f"Test Average: {student.getAvg():1.2f}")
        print()

def findStudent(objlist,studentInfo):
    #
    # Executes a Sequential Search of roster, looking for an object of the Student
    # class with either a studentID == studentInfo, or lastName == studentInfo. If
    # the search is unsuccessful, return None... otherwise, return a pointer to
    # the student object that was found.
    #
    # roster A list of objects from the Student class, each element in the list
    # contains a studentID(str), Name(list of firstName(str), lastName(str)),
    # testScores(list of int values), and average(float).
    # studentInfo A str that contains either a lastName or studentID
    #
    # Returns a reference to the Student object that was found, or None if the
    # search is unsuccessful
    #
    for student in objlist:
        if(student.getID() == studentInfo or student.getName()[1] == studentInfo):
            return student
    return None

#
# *****Main Program*****

```

```

#
studentList = getStudents()
for student in studentList:
    printOneStudent(student)
print()
print("You may update any student info, or add a student.")
print()
temp = True
while temp:
    studentInfo = input("Enter Student ID or Last Name (<enter> to stop): ")
    if studentInfo:
        foundStudent = findStudent(studentList,studentInfo)
        if foundStudent is None:
            print('This student does not exist. You may enter the info now.')
            studentID = input("Student ID: ")
            studentFN = input("First Name: ")
            studentLN = input("Last Name: ")
            student = Student(studentID)
            student.setName(studentFN, studentLN)
            flag = True
            while flag:
                score = input("Enter a Test Score (<enter> to stop): ")
                if(score != ''):
                    student.addTest(int(score))
                else:
                    flag = False
            studentList.append(student)
            print()
            print("New Student Added:")
            printOneStudent(student)
        else:
            printOneStudent(foundStudent)
            print("(1) Change the Name")
            print("(2) Add a Test")
            print("(3) Do Nothing")
            print()
            value = input("What would you like to do? ")
            valid_input = True
            while valid_input:
                if value == '1':
                    studentFN = input("First Name: ")
                    studentLN = input("Last Name: ")
                    print()
                    foundStudent.setName(studentFN,studentLN)
                    printOneStudent(foundStudent)
                    valid_input = False
                elif value == '2':
                    flag = True
                    while(flag):
                        score = input("Enter a Test Score (<enter> to stop): ")

```

```
        if(score != ''):
            foundStudent.addTest(int(score))
        else:
            flag = False
        print()
        printOneStudent(foundStudent)
        valid_input = False
    elif value == '3':
        print()
        valid_input = False
    else:
        print("Enter a valid number")
        print()
        value = input("What would you like to do? ")

else:
    temp = False
```