**CPSC 223P Assignment #1**

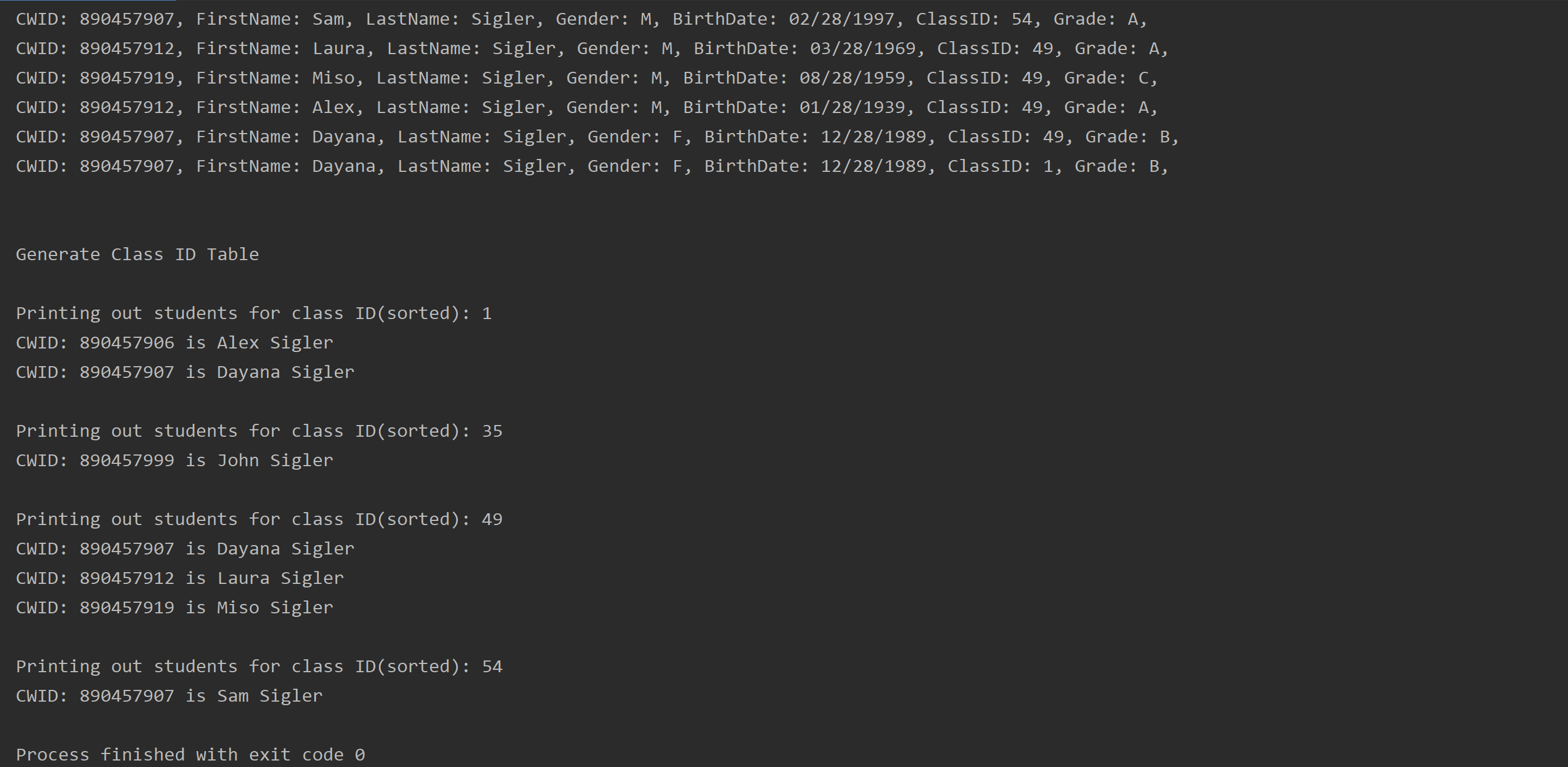
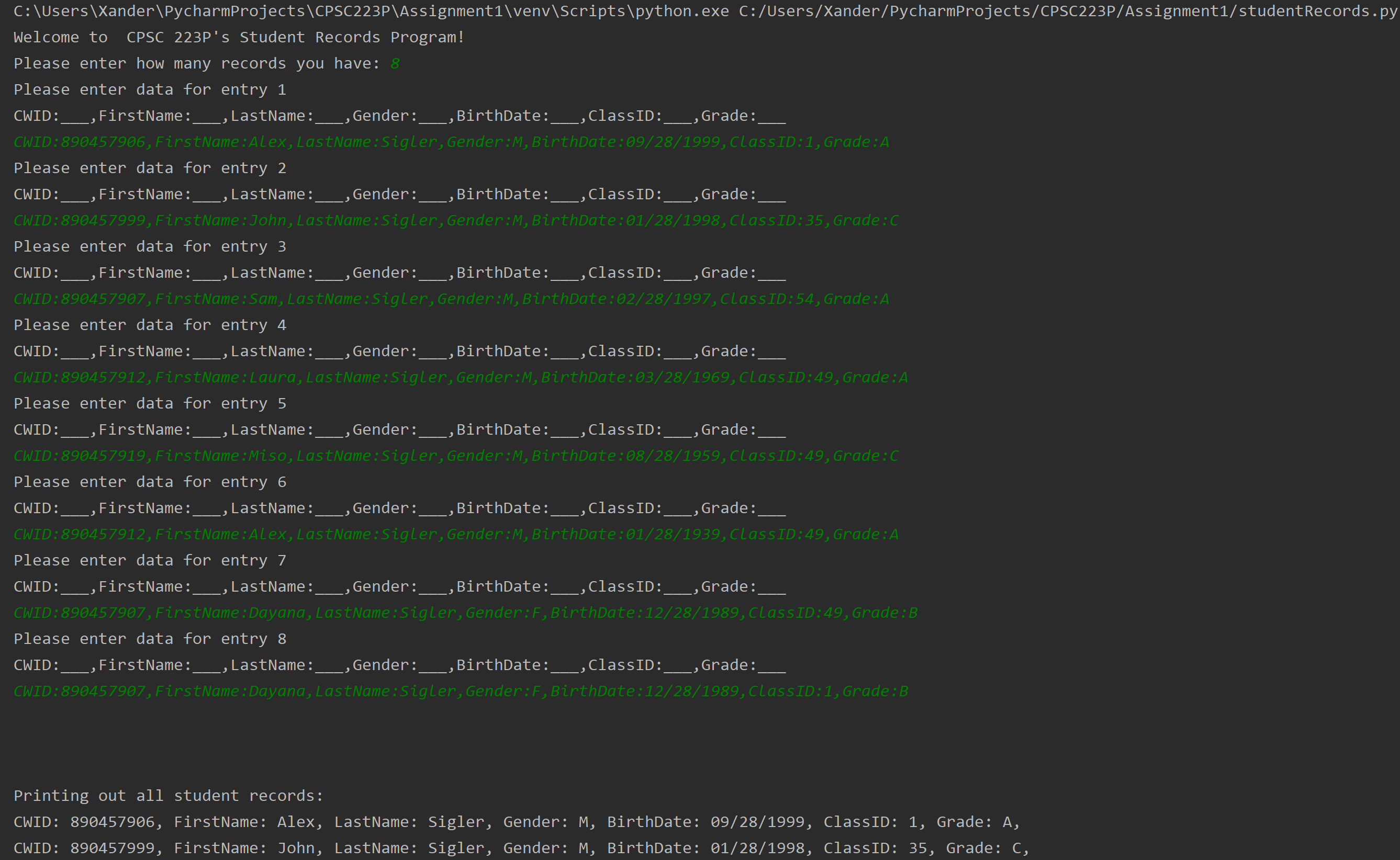
**Alexander Sigler**

**890457906**

**03/07/2020**

Code (GIT: <https://github.com/97WaterPolo/CPSC223P/blob/master/HW1_studentRecords.py> )

print("Welcome to CPSC 223P's Student Records Program!")  
numOfRecords = input("Please enter how many records you have: ")  
columns = ["CWID", "FirstName", "LastName", "Gender", "BirthDate", "ClassID", "Grade"]  
  
# Variable (lists and dictionary) delceration  
classIds = {} # Empty dictionary for class Ids  
sortedClassIds = {} # Empty dictionary for after class Ids are sorted  
records = [] # A list of all the student records  
for x in range(int(numOfRecords)):  
 print("Please enter data for entry " + str(x+1))  
 studentInput = input(":\_\_\_,".join(columns)+":\_\_\_\n") # Print out the format of how input should be  
 studentInput = studentInput.split(",") # After getting input, split into n objects based on the "," comma  
  
 while len(studentInput) != len(columns): # Data verification, make sure they entired the right amt of items  
 print("Incorrect format, try again!")  
 studentInput = input(":\_\_\_,".join(columns) + ":\_\_\_\n")  
 studentInput = studentInput.split(",")  
  
 tmpDict = {} # Temp variable that holds the key/value pair before added to records  
 for i in range(len(studentInput)):  
 attribute = studentInput[i].split(":") # Split the individual attribute to put into key/value  
 tmpDict[attribute[0]] = attribute[1] # Add the key/value attribute to the dictionary  
  
 # This section is to sort the CWID of the students into the proper class ID  
 classId = tmpDict[columns[5]] # Gets the class ID value from dictionary  
 cwid = tmpDict[columns[0]] # Gets the CWID value from the dictionary  
 if classId in classIds.keys(): # If the Class ID IS in dictionary, add it to the set  
 classIds[classId].add(cwid) # Since set, any duplicates won't be added.  
 else:  
 classIds[classId] = {cwid} # Create a new set of single CWID, and add it to the class ID records  
  
 records.append(tmpDict) # Append the dictionary record to our list of student records  
  
for x in sorted(classIds.keys()): # Loop through all SORTED class IDs (sort the ids for the classes)  
 sortedClassIds[x] = sorted(classIds[x]) # Sort the individual set based on CWID (Ascending) and put in new dict  
  
print("\n\n\nPrinting out all student records: ")  
for r in records: # Loop through all the student records entered  
 for c in r.keys(): # Loop through all the individual keys of the record  
 print(c + ": " + r[c] + ", ", end="") # Print combined records in 1 line, no newline char at end  
 print() # Generate new line entry  
  
print("\n\nGenerate Class ID Table")  
for classId in sortedClassIds.keys(): # Loop through all the keys in our sorted class Ids  
 print("\nPrinting out students for class ID(sorted): " + classId)  
 for cwid in sortedClassIds[classId]: # Loop through all the CWIDs for each class ID  
 for r in records: # Loop through all of the student records  
 if r[columns[0]] == cwid and r[columns[5]] == classId: # make sure CWID & class ID match up  
 print("CWID: " + r[columns[0]] + " is " + r[columns[1]] + " " + r[columns[2]]) # Print out CWID,F/LName  
 break

Screenshots of Output Code