Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw4

2-15-19

Question #1: Personal Web Page Generator

'''

Deborah Barndt

2-15-19

PersonalWebPageGenerator.py

hw4: Question 1 Personal Web Page Generator

This program will ask the user for his or her name, then ask the user to enter

a sentence that describes himself or herself. Once the user has entered the

requested input, the program should create an HTML file, containing the input,

for a simple web page.

Written by Deborah Barndt.

'''

# Ask the user for his or her name.

username = input('Enter your name: ')

# Ask the user to enter a sentence that describes them.

userdesc = input('Describe yourself: ')

# Open the html file to write the html content

content = open('profile.html', 'w')

# Create the html file for the user profile.

html = '<html>\n' + \

'<head>\n' + \

'</head>\n' + \

'<body>\n' + \

'\t<center>\n' + \

'\t\t<h1>' + username + '</h1>\n' + \

'\t<center>\n' + \

'\t<hr />\n' + \

'\t' + userdesc + '\n' + \

'\t<hr />\n' + \

'</body>\n' + \

'</html>\n'

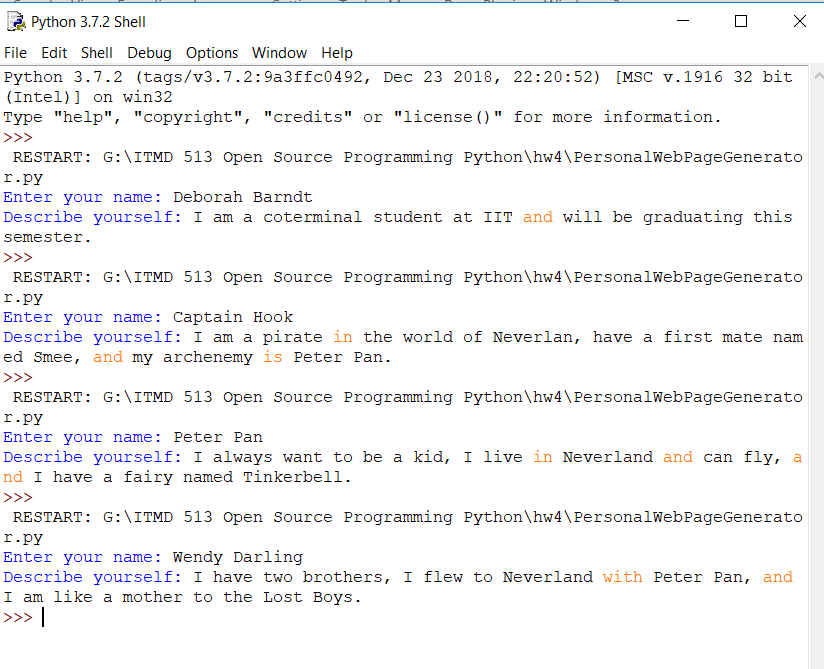
# Write the html into the file for the user profile.

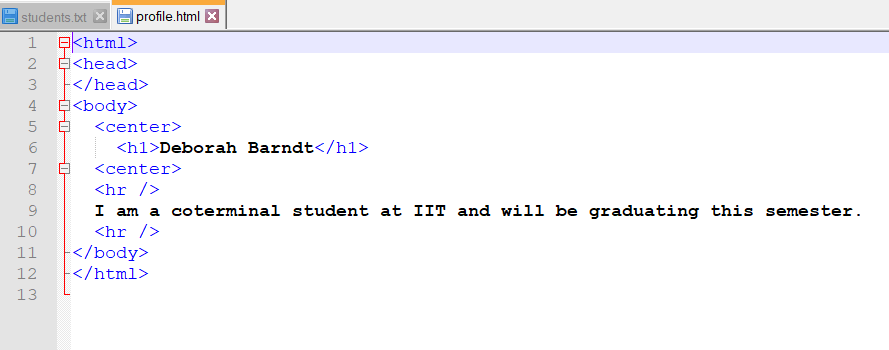
content.write(html)

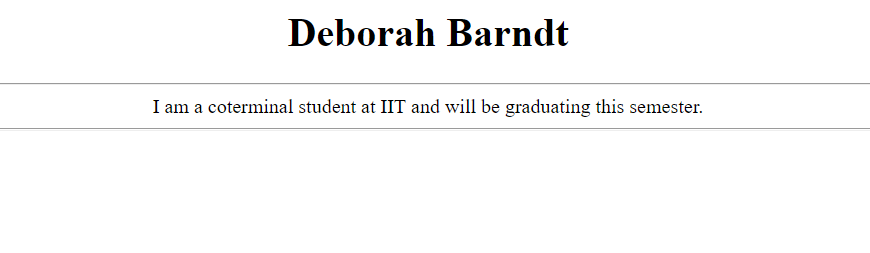
# Close the file with the html content.

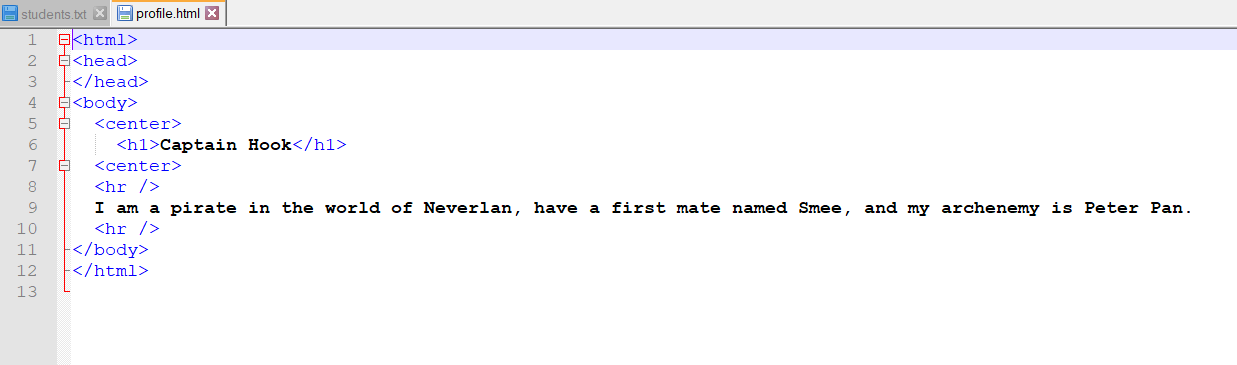
content.close()

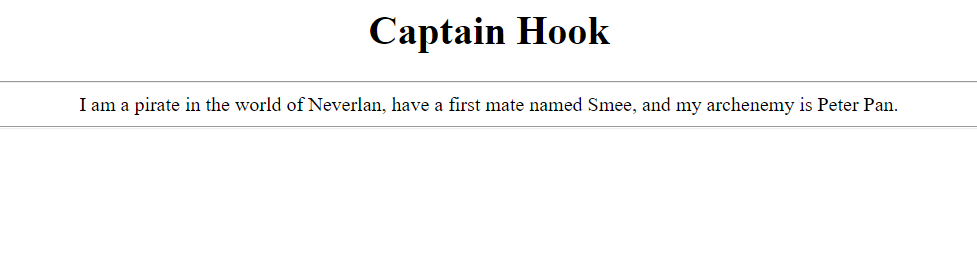
Output Results:

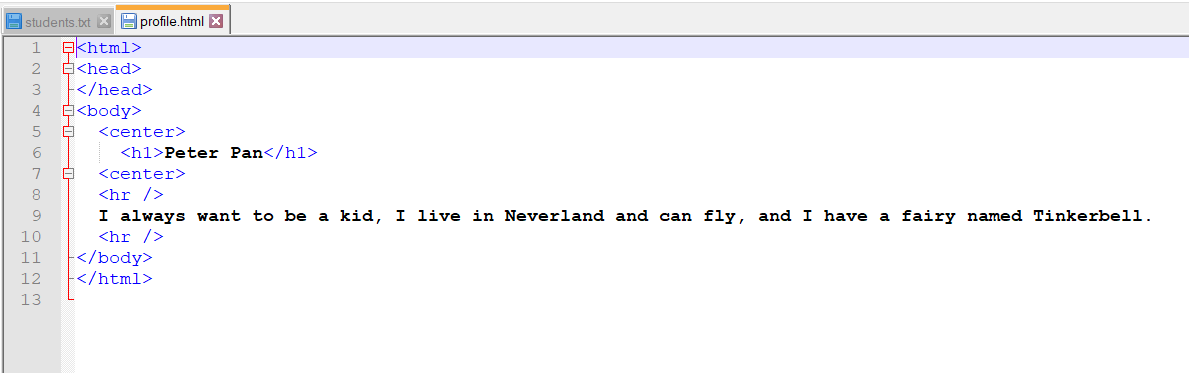


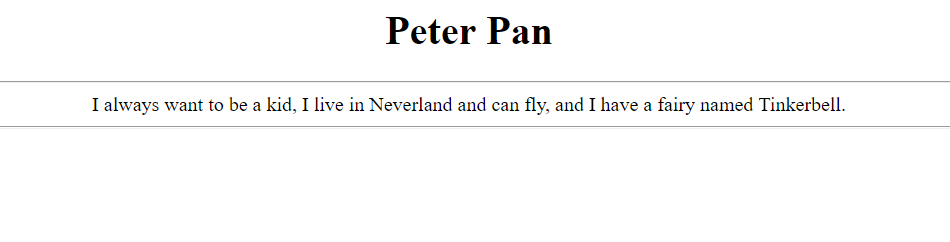


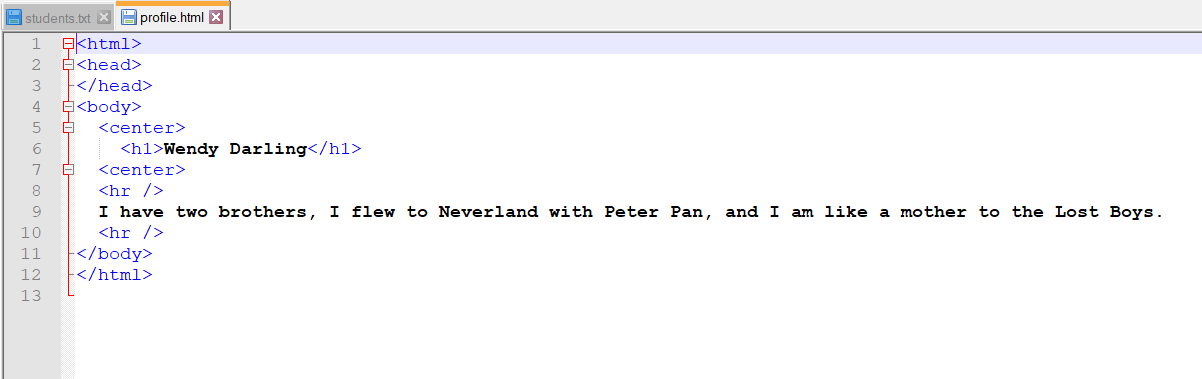


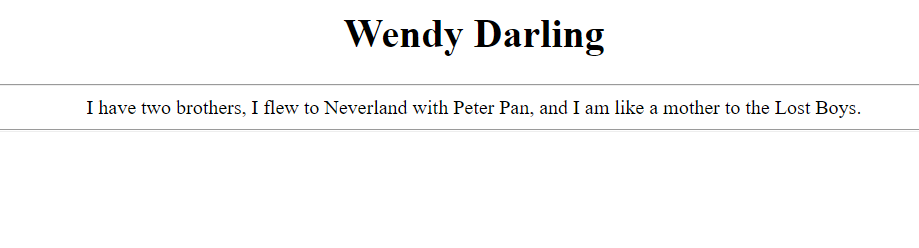












Question #2

'''

Deborah Barndt

2-15-19

StudentGradeReports.py

hw4: Question 2 Student Grade Reports

This program will create student grade reports, and save the report in a file

separate filename. The grade report for each student will contain the student's

name and ID number, a list of courses taken, the credits and grade for each

course, and a semester grade point average (GPA).

Written by Deborah Barndt.

'''

# Function that returns the number value of the letter grade.

def studentGrade(courseGrade):

# Get the number value from the course grade.

if (courseGrade == 'A'):

grade = 4

elif (courseGrade == 'B'):

grade = 3

elif (courseGrade == 'C'):

grade = 2

elif (courseGrade == 'D'):

grade = 1

else:

grade = 0

return grade

# Function the writes the total number of semester course credits and the semester gpa.

def studentGPA(gpa, totalcredits, outputData):

outputData.write('\n')

outputData.write('Total Semester Course Credits Completed: ' + str(totalcredits) + '\n')

outputData.write('Semester GPA: ' + "%.2f" % gpa + '\n\n\n')

# Function to write the student reports from the data.

def main():

studentid = 0

previd = 0

grade = 0

totalcredits = 0

gpa = 0

studentname = ' '

courseGrade = ' '

credit = 0

courseCredit = ' '

code = ' '

numstudent = 0

# Open the data file students.txt as an input file.

inputData = open('students.txt', 'r')

# Create the report as gradereport.txt for the output file.

outputData = open('gradereports.txt', 'w')

# Read lines from students.txt file.

for line in inputData:

# Split the data elements using : as a separator.

line = line.split(':')

# Read studentID from students.txt file.

studentid = int(line[0])

if (studentid != previd):

# Count the number of students in the students.txt file.

numstudent += 1

previd = studentid

studentname = line[1]

# Check if the student is the same, if not calculate the gpa for student.

if (numstudent > 1):

# Calculate the gpa to gradereports.txt file.

gpa = gpa / totalcredits

# Write the student gpa and semester credits at end of student report.

studentGPA(gpa, totalcredits, outputData)

# Reset the variables for the gpa and the total number of credits.

gpa = 0

totalcredits = 0

# Get the course credits from the students.txt and count the total credits.

courseCredit = int(line[3])

totalcredits += courseCredit

# Get the course grade from students.txt and calculate the grades.

courseGrade = line[4].strip()

grade = studentGrade(courseGrade)

# Calculate the student gpa for the semester.

gpa += grade \* courseCredit

code = line[2]

# Write the name and ID number for each student and data headers to reports.

outputData.write('Student Name: ' + studentname + '\n')

outputData.write('Student ID Number: ' + str(studentid) + '\n\n')

outputData.write('Course Code\t\t\tCourse Credits\t\t\tCourse Grade\n')

outputData.write('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n')

# Write the course code, course credits, and course grade for each student in the report.

outputData.write(code + '\t\t\t\t\t\t\t\t\t' + str(courseCredit) + '\t\t\t\t\t\t\t\t\t\t' + courseGrade + '\n')

elif (studentid == previd):

# Get the course credits from the students.txt and count the total credits.

courseCredit = int(line[3])

totalcredits += courseCredit

# Get the course grade from students.txt and calculate the grades.

courseGrade = line[4].strip()

grade = studentGrade(courseGrade)

# Calculate the student gpa for the semester.

gpa += grade \* courseCredit

code = line[2]

# Write the course code, course credits, and course grade for each student in the report.

outputData.write(code + '\t\t\t\t\t\t\t\t\t' + str(courseCredit) + '\t\t\t\t\t\t\t\t\t\t' + courseGrade + '\n')

# Calculate and write the student gpa for the last student in the students.txt file.

gpa = gpa / totalcredits

# Write the total number of semester credits and student gpa at the end of the report.

studentGPA(gpa, totalcredits, outputData)

# Close the student.txt and gradereports.txt file.

inputData.close()

outputData.close()

print('Student reports written to gradereports.txt file.')

main()

Output Results:

