**HAUB SCHOOL OF BUSINESS**

**SAINT JOSEPH’S UNIVERSITY**

**DSS 615: Python Programming**

**Instructor: Michael Ghen**

**Assignment 5**

By:

Vinayak Suresh Tayshetye (10673718)

**Chapter 3 Programming Projects**

**Car Loan** Write a program to analyze a car loan. See Fig. 3.79. The user should enter the amount of the loan, the annual percentage rate of interest, and the duration of the loan in months. After each piece of data is entered, the data should be checked to make sure it is reasonable. If bad data has been supplied, the user should be so advised. Otherwise, the monthly payment and the total amount of interest paid should be displayed. The formula for the monthly payment is

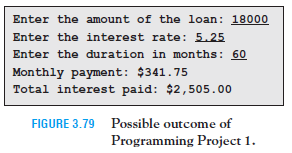
monthly payment = *[ p \* r] / [ 1 - (1 + r)-n]*

where *p* is the amount of the loan, *r* is the monthly interest rate (annual rate divided

by 12) given as a number between 0 (for 0%) and 100 (for 100%), and *n* is the duration

of the loan in months. The formula for the total interest paid is

total interest = *n \** [monthly payment] - *p*.



[Code]

# Chapter 3 Pro gramming Project #1

while True:

try:

loanAmount = int(input("Enter the amount of the loan: "))

except ValueError:

print("You have not Entered a Number Value for Loan Amount.")

print("Please Enter again with the correct Numbers.")

else:

break

while True:

try:

intrRate = float(input("Enter the interest rate : "))

except ValueError:

print("You have not Entered a Number Value for Interest Rate.")

print("Please Enter again with the correct Numbers.")

else:

break

while True:

try:

loanDuration = int(input("Enter the duration in months: "))

except ValueError:

print("You have not Entered a Number Value for Loan Duration.")

print("Please Enter again with the correct Numbers.")

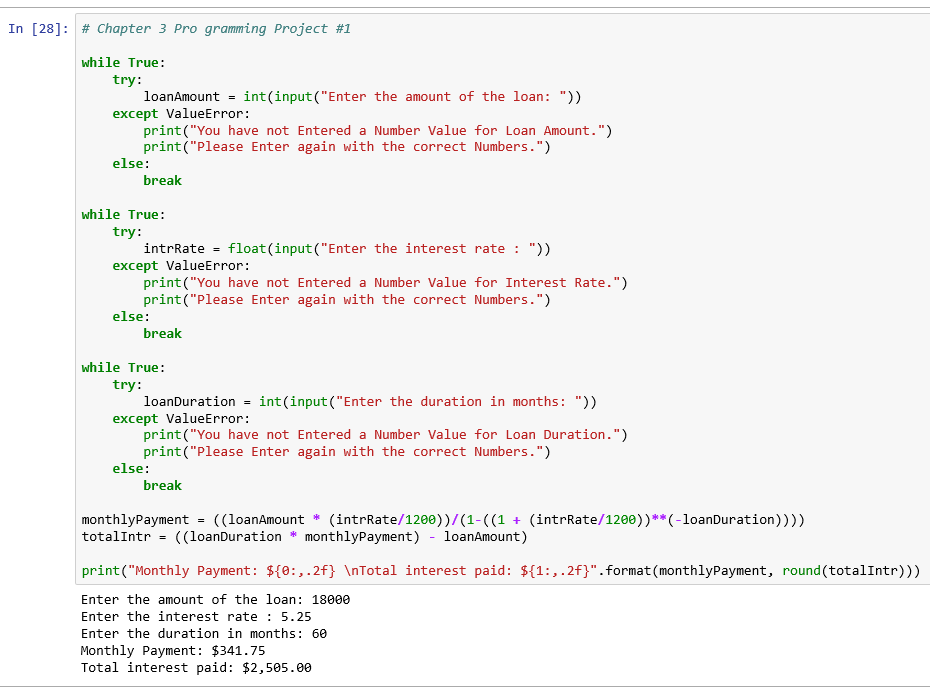
else:

break

monthlyPayment = ((loanAmount\*(intrRate/1200))/(1-((1+(intrRate/1200))\*\*(-loanDuration))))

totalIntr = ((loanDuration \* monthlyPayment) - loanAmount)

print("Monthly Payment: ${0:,.2f} \nTotal interest paid: ${1:,.2f}".format(monthlyPayment, round(totalIntr)))



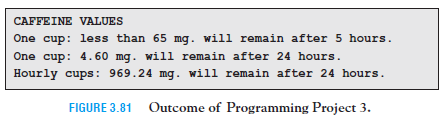
**Caffeine Absorption** After caffeine is absorbed into the body, 13% is eliminated from the body each hour. Assume a person drinks an 8-oz cup of brewed coffee containing 130 mg of caffeine, and that the caffeine is absorbed immediately into the body.

Write a program to calculate the following values. See Fig. 3.81.

**(a)** The number of hours required until less than 65 mg (one-half the original amount) remain in the body.

**(b)** The amount of caffeine in the body 24 hours after the person drinks the coffee.

**(c)** Suppose the person drinks a cup of coffee at 7 a.m. and then drinks a cup of coffee at the end of each hour until 7 a.m. the next day. How much caffeine will be in the body at the end of the 24 hours?



[code]

# Chapter 3 Pro gramming Project #3

# (a)

caffeine = 130

hours = 0

while caffeine >= 65:

caffeine -= caffeine\*0.13 #reducing hourly caffeine content by 13%

hours += 1

print("One cup: less than 65 mg. will remain after {0} hours.".format(hours))

# (b)

caffeine = 130

hours = 1

while hours <= 24:

caffeine -= caffeine\*0.13 #reducing hourly caffeine content by 13%

hours += 1

print("One cup: {0:.2f} mg. will remain after 24 hours.".format(caffeine))

# (c)

caffeine = 130

hours = 0

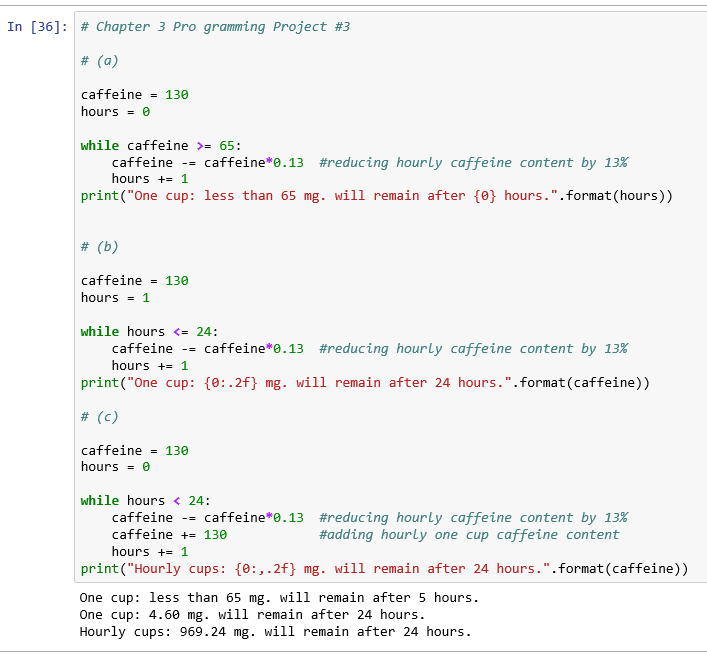
while hours < 24:

caffeine -= caffeine\*0.13 #reducing hourly caffeine content by 13%

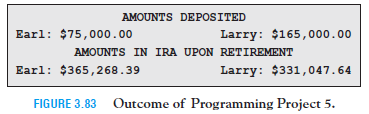
caffeine += 130 #adding hourly one cup caffeine content

hours += 1

print("Hourly cups: {0:,.2f} mg. will remain after 24 hours.".format(caffeine))



**Individual Retirement Accounts** Money earned in an ordinary savings account is subject to federal, state, and local income taxes. However, a special type of retirement savings account, called a **traditional individual retirement account** (traditional IRA), allows these taxes to be deferred until after retirement. IRAs are highly touted by financial planners. The purpose of this programming project is to show the value of starting an IRA early. Earl and Larry each begin full-time jobs in January 2015 and plan to retire in January 2063 after working for 48 years. Assume that any money they deposit into IRAs earns 4% interest compounded annually. Earl opens a traditional IRA account immediately and deposits $5,000 into his account at the end of each year for fifteen years. After that he plans to make no further deposits and just let the money earn interest. Larry plans to wait fifteen years before opening his traditional IRA and then deposit $5,000 into the account at the end of each year until he retires. Write a program that calculates the amount of money each person has deposited into his account and the amount of money in each account upon retirement. See Fig. 3.83.



# Chapter 3 Pro gramming Project #5

intr = 0.04

#Details for Earl's deposit and earnings

totalDepAmt1 = 0

totalAmt1 = 0

year = 2016

while year < 2063:

if year <= 2030: #Condition for first 15 years

totalDepAmt1 += 5000

totalAmt1 += 5000

totalAmt1 += intr \* totalAmt1

else:

totalAmt1 += intr \* totalAmt1 #Calculation for next 32 years

year += 1

#Details for Larry's deposit and earnings

totalDepAmt2 = 0

totalAmt2 = 0

year = 2016

while year <= 2063:

if year > 2030 and year < 2063:

totalDepAmt2 += 5000

totalAmt2 += 5000

totalAmt2 += intr \* totalAmt2

elif year == 2063: #Calculation for the last year 2063 without interest

totalDepAmt2 += 5000

totalAmt2 += 5000

else: #Calculation for next 32 years

totalAmt2 += intr \* totalAmt2

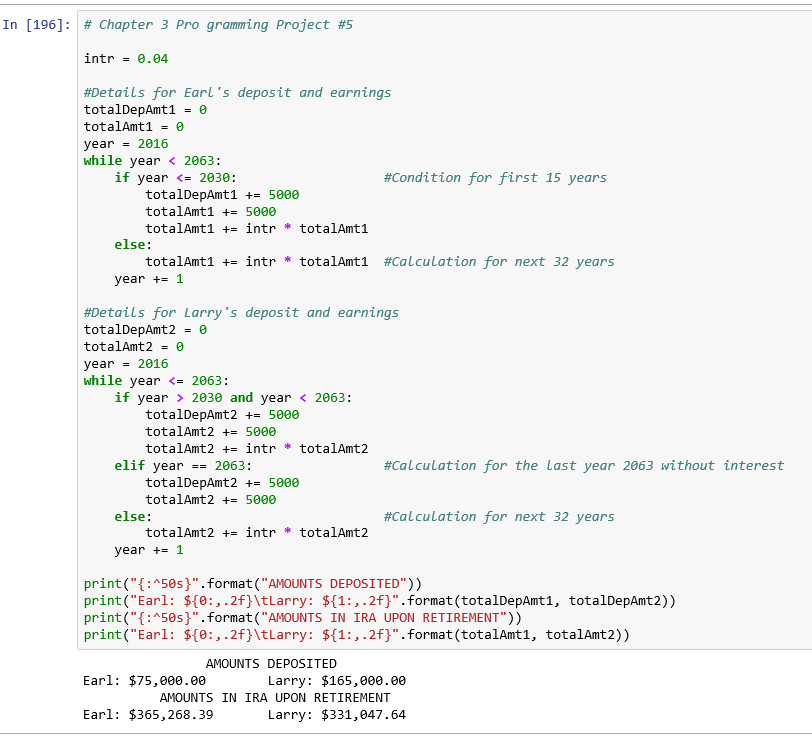
year += 1

print("{:^50s}".format("AMOUNTS DEPOSITED"))

print("Earl: ${0:,.2f}\tLarry: ${1:,.2f}".format(totalDepAmt1, totalDepAmt2))

print("{:^50s}".format("AMOUNTS IN IRA UPON RETIREMENT"))

print("Earl: ${0:,.2f}\tLarry: ${1:,.2f}".format(totalAmt1, totalAmt2))



**Palindrome** A *palindrome* is a word or phrase that reads the same forward and backward,

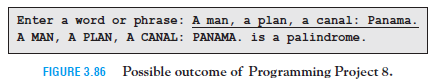
character for character, disregarding punctuation, case, and spaces. Some examples

are “racecar”, “Madam, I’m Adam”, and “Was it a cat I saw?”. Write a program

that asks the user to input a word or phrase and then determines if it is a palindrome.

See Fig. 3.86. ***Note:*** Remove all spaces and punctuation before analyzing the word or

phrase.



# Chapter 3 Pro gramming Project #8

word = input("Enter a word or phrase: ")

wordSpaceReplace = word.replace(" ", "") #Removing empty spaces to make a continuous string.

characters = [":", ",", ".", "?", "'"]

continuous = list(wordSpaceReplace.lower())

for ch in continuous: #Removing characters other than alphabets.

if ch in characters:

continuous.remove(ch)

reverse = []

for i in range(0, len(continuous), 1): #Creating a reverse string.

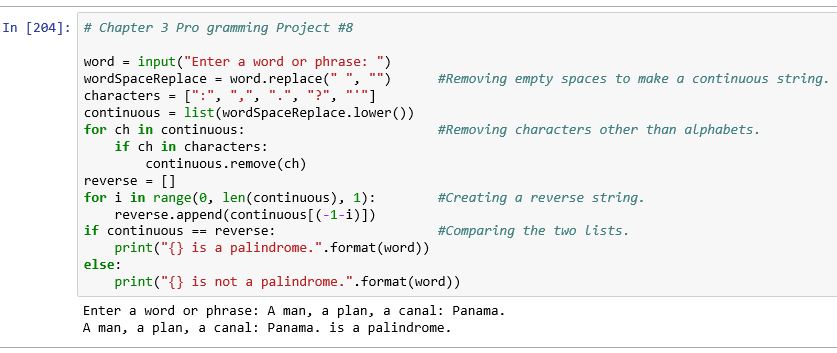
reverse.append(continuous[(-1-i)])

if continuous == reverse: #Comparing the two lists.

print("{} is a palindrome.".format(word))

else:

print("{} is not a palindrome.".format(word))



Instructor Assignments

Fix a busted CSV file

import re

file1 = open("G:/SJU/10 - Python/Assignments/Data Files/digestive\_illnesses.csv", 'r')

digIllness = []

rowNumber = 0

for row in file1:

row = " ".join(re.split("\s+", row, flags=re.UNICODE)) #removing spaces in the rows

row = ",".join(re.split(",+", row, flags=re.UNICODE)) #removing commas in the rows

if rowNumber == 0: #Keeping header of each column unchanged

digIllness.append(row)

rowNumber += 1

elif rowNumber % 2 == 1: #Condition to save odd rows

newRow= row

rowNumber += 1

else:

newRow += row

newRow=newRow.replace(", ,",",")

dataIllness.append(newRow) #Condition to add even rows to odd row

rowNumber += 1

#Creating a new file digestive\_illnesses\_New\_Sorted.csv to write the output

with open("G:/SJU/10 - Python/Assignments/Data Files/digestive\_illnesses\_New\_Sorted.csv", 'w') as file2:

for i in range(len(dataIllness)-1):

file2.write(dataIllness[i])

file2.write("\n")

file2.close()

file1.close()



Build a better data set

import pandas as pd

import re

data1 = pd.read\_csv('G:/SJU/10 - Python/Assignments/Data Files/health\_codes.csv')

characters = ["SNOMED-CT", "HCPCS"]

i=0

j=[]

for i in range(0, len(data1)-1):

j = data1.find(",") # will find index of last comma

newRow = data1[j:] + data1[0:j] #will bring ahead description column

for ch in characters:

if ch in data1:

data1.remove(ch) # will remove "SNOMED-CT", "HCPCS" from the data

else:

break

data1.insert(0,("Description" "SNOMED-CT", "HCPCS", "ICD-10",)) #adding header

print(data1)

#data1.head()

Alternate---------------------

col = ['remove1', 'SNOMED-C','remove2', 'HCPCS', 'remove3', 'ICD-10','Description']

data = pd.read\_csv('G:/SJU/10 - Python/Assignments/Data Files/health\_codes.csv', usecols=(6,1,3,5))

data.head