CSC 157

Name \_James Aniciete\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date \_\_5/9/2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab No. \_\_\_16\_\_\_\_

Purpose of the Lab Activity

**The purpose of this lab activity is to examine an application that simulates business intelligence.**

Source Code

# Programmer: James Aniciete

# Course No.: CSC 157

# Lab No.: 16

# Date: 5/9/2020

from datetime import datetime as DateTime

from datetime import timedelta as TimeDelta

from statistics import mean

from math import ceil

# function to add days to the start date with formatting

def addDays(days) :

start\_date = DateTime.today()

# print (start\_date.strftime("%m-%d-%Y"))

end\_date = (start\_date + TimeDelta(days)).strftime("%m-%d-%Y")

return end\_date

# List for location/placement

placement = [

"Front Store Entrance (Carboard Display)",

"CD Racks (Usual Music Collection Aisles)",

"Bargain CD Music Bins (Placed at the Front of the Store)",

"Extended Edition Released (Cardboard Display)"

]

# List for start dates of each period

dates = [DateTime.today().strftime("%m-%d-%Y"),

addDays(46), addDays(91), addDays(121)]

# Anticipated Units Sold List

# [ [CD, vinyl], ...]

# fourth period uncertain ==> only initialized

anticipated = [ [25,32], [15,19], [37,46], [0,0] ]

# CD/Vinyl Prices List

prices = [ [17,32], [15,36], [9,21], [39,53] ]

# function to ensure valid choice selection in a specified range

def getIntInRange(message, rangeBottom, rangeTop):

msg = message

while True:

try:

x = int(input(msg))

if x >= rangeBottom and x <= rangeTop:

break

else:

# error message for integer out of range + redisplay menu choices

# f"" is like str.format()

msg = f"Please enter a value between {rangeBottom} and {rangeTop}\n{message}"

# error message for wrong data type (non-integer)

except ValueError:

msg = f"Please enter an integer value \n{message}"

return x

# function to display info on anticipated CD/vinyl sales

# row = period - 1 b/c lists have zero-based indices

def printInfo(row):

antTotalCD = 0; antTotalVin = 0

# calculate total anticipated sales for CD and Vinyl

for x in range(row+1):

antTotalCD += anticipated[x][0]

antTotalVin += anticipated[x][1]

# anticipated values for the given period

#antCD = anticipated[row][0]

#antVin = anticipated[row][1]

# for Period 4, tell user data are only forecasted values

if row == 3:

if anticipated[row][0] == 0 and anticipated[row][1] == 0:

anticipated[row][0] = "Number is dependent on previous sales"

anticipated[row][1] = "Number is dependent on previous sales"

# \ for line continuation

# not indented b/c it would display as indented

print(f"\n\

Period: {row +1}\n\

Start Date: {dates[row]}\n\

Location and Placement: {placement[row]}\n\

Anticipated CDs Sold for this Period: {anticipated[row][0]}\n\

Anticipated Vinyl Sold for this Period: {anticipated[row][1]}\n\

Total Anticipated CDs Sold for this Period: {antTotalCD}\n\

Total Anticipated Vinyls Sold for this Period: {antTotalVin}\n\

CD Price: ${prices[row][0]}\n\

Vinyl Price: ${prices[row][1]}\n")

# function to calculate Period 4 sales based on {ceil(1.1 \* mean of previous periods' sales)}

# allows entry of different sales amounts for Periods 1-3 in the range [0, 1,000,000]

def calcPeriodFour():

soldCDOne = getIntInRange(f"Please input the number of CDs sold in period 1 ({dates[0]}): ", 0, 1000000)

soldCDTwo = getIntInRange(f"Please input the number of CDs sold in period 2 ({dates[1]}): ", 0, 1000000)

soldCDThree = getIntInRange(f"Please input the number of CDs sold in period 3 ({dates[2]}): ", 0, 1000000)

calcCD = ceil(mean([soldCDOne, soldCDTwo, soldCDThree])\*1.1)

soldVinOne = getIntInRange(f"Please input the number of vinyl records sold in period 1 ({dates[0]}): ", 0, 1000000)

soldVinTwo = getIntInRange(f"Please input the number of vinyl records sold in period 2 ({dates[1]}): ", 0, 1000000)

soldVinThree = getIntInRange(f"Please input the number of vinyl records sold in period 3 ({dates[2]}): ", 0, 1000000)

calcVin = ceil(mean([soldVinOne, soldVinTwo, soldVinThree])\*1.1)

print(f"\nAnticipated CDs Sold for Period 4: {calcCD}")

print(f"\nAnticipated Vinyls Sold for Period 4: {calcVin}")

print("\nNOTE: Anticipated Sales for Perid 4 is 1.1 times the average of sales from Periods 1-3\n")

anticipated[3] = [calcCD, calcVin]

# menu choices to be used with getIntInRange()

menuChoices = "\

Please select an option from the following \n\

1. View Entire Timeline for the Norwood Notes New CD Placement\n\

2. See Information for a Specific Period\n\

3. Calculate Anticipated Units Sold for Period 4\n\

4. Exit Program\n\

Your Choice: "

# period choices for specific period selection

if menuChoices != 4:

periodChoices = f"\n\

Please select a period:\n\

Period 1: {dates[0]}\n\

Period 2: {dates[1]}\n\

Period 3: {dates[2]}\n\

Period 4: {dates[3]}\n\

Your Choice: "

# while loop to execute program until choice 4 (Exit Program) is selected

while True:

choice = getIntInRange(menuChoices, 1, 4)

if choice == 1:

for i in range(4):

printInfo(i)

elif choice == 2:

period = getIntInRange(periodChoices, 1, 4)

printInfo(period-1)

elif choice == 3:

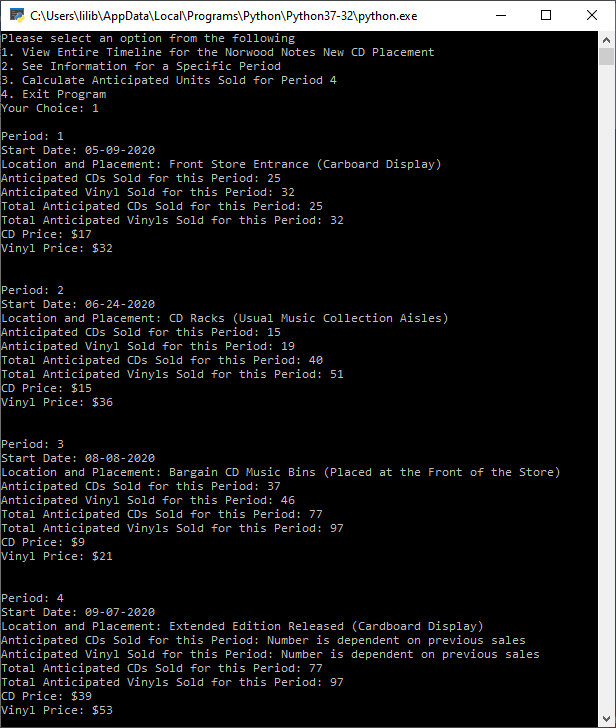
calcPeriodFour()

elif choice == 4:

break;

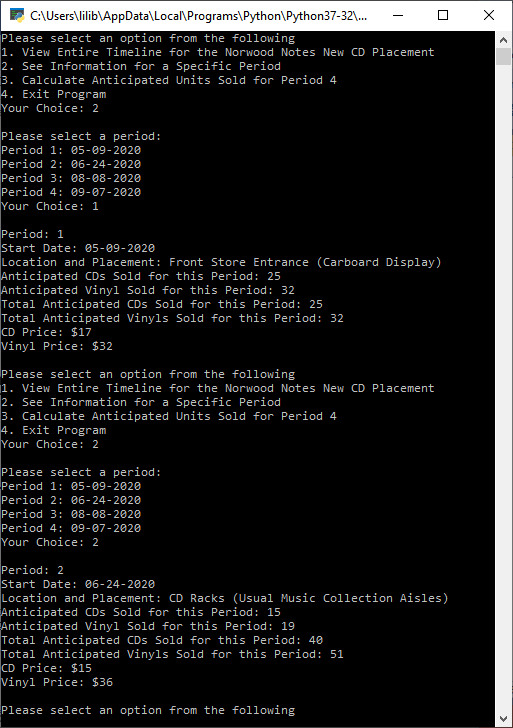
Snippet(s) of Output(s)

**1. View Entire Timeline**

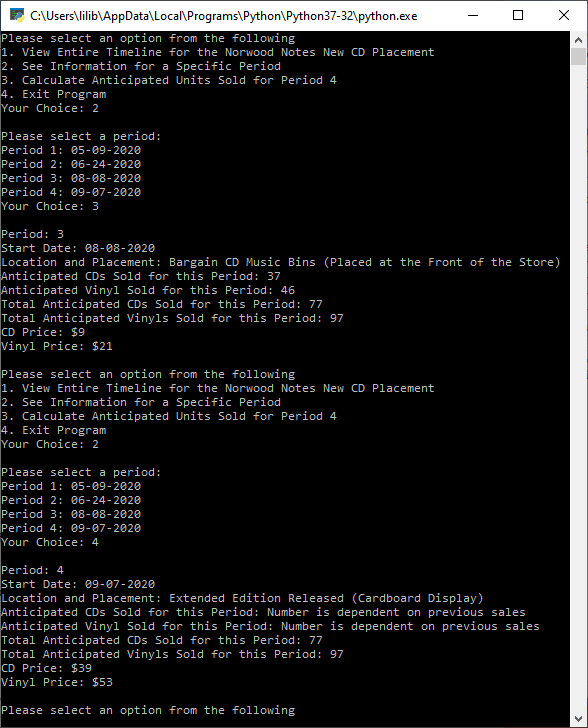


**2. See Information for a Specific Period**

**Periods 1 and 2:**

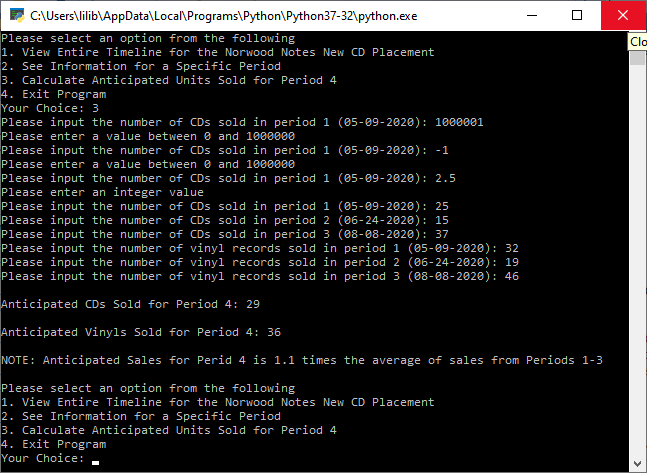


**Periods 3 and 4:**

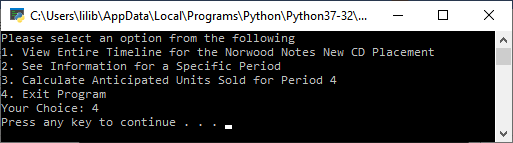


**3. Calculate Anticpated Units Sold for Period 4**

**Anticipated Units Sold = Units Sold:**



**4. Exit Program**



Modified Source Code

**n/a**

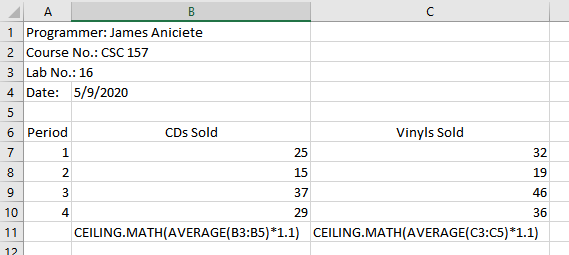
Snippet(s) of Output(s) from execution of modified Code

**n/a**

Excel Spreadsheet (when Calculations are involved)

**3. Calculate Anticpated Units Sold for Period 4**

**Anticipated Units Sold = Units Sold:**



Answers to Questions (Be sure to copy the questions themselves!)

**(1)** This particular project’s starter code uses the **datetime** library.

With a function call having your own arguments, test the following user - defined function.

**from datetime import datetime**

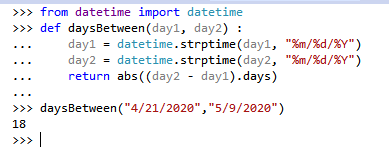
**def daysBetween(day1, day2) :**

**day1 = datetime.strptime(day1, "%m/%d/%Y")**

**day2 = datetime.strptime(day2, "%m/%d/%Y")**

**return abs((day2 - day1).days)**

**This function returns the difference between two dates with the “m/d/yyyy" format in terms of days.**



**(2)** Explain what is accomplished by the following code segment.

**import datetime**

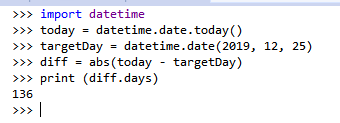
**today = datetime.date.today()**

**targetDay = datetime.date(2019, 12, 25)**

**diff = abs(today - targetDay)**

**print (diff.days)**

**This function returns how many days have passed since last Christmas.**



**(3)** Execute the following program segment, using today’s date.

**import datetime**

**print ("Today's Date:", datetime.datetime.today())**

**date\_today = datetime.date.today()**

**print (date\_today)**

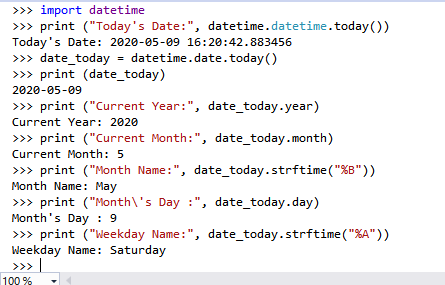
**print ("Current Year:", date\_today.year)**

**print ("Current Month:", date\_today.month)**

**print ("Month Name:", date\_today.strftime("%B"))**

**print ("Month\'s Day :", date\_today.day) print ("Weekday Name:", date\_today.strftime("%A"))**

**This program takes today’s date and prints the date in “yyyy-mm-dd” format, the year, the number for the month, the name of the month spelled out, the number for the day of the month, and the name of the weekday spelled out.**

  
 **(4)** Explain how business intelligence is used in this project.

**Business intelligence is used in this project to report anticipated sales for the four periods and forecast anticipated sales for period 4 based on the respective sales from periods 1-3.**

**(5)** What have you learned from performing and coding this lab assignment?

**I have learned about string formatting using f”” , the statistics module, and formatting dates using the datetime module.**