Art Gallery

LAB # 8

By

Corey Henry and Geoffrey Sanchez

***“On my honor, as a Mississippi State University student, I have neither***

***given nor received unauthorized assistance on this academic work.”***

Signatures:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CSE-1284-06-201430 Intro to Computer Programming

Class Section # 6

Josh Crowson

11/5/3014

**Analysis and Conclusions**

Using and understanding how to control lists was very important in this list. We used multiple different functions to be able to call each part of the program. The most important function was defiantly the option function. This allowed the make shift database to remember what you had done and continue the additions and changes you made until you quit the program. We used a combination on adding and subtracting parts of the list with the index numbers of the list items. This way when you made a change or a search you would call the corresponding index numbers for that list of art. The code overall was very straight forward and easy, using Google to see all the different ways to edit the list was very helpful, as well as using the draw.io to make our flowchart which made it more clear to read and build our program off of. As long as the index points all matched up and the option function was always at the end of the series of code it allowed the program to run smoothly, with little to no bugs.

Source Code:

define the main function

def main():

title, genre, artist, date, value = [],[],[],[],[]

options(title, genre, artist, date, value)

#create a function that allows you to call your options.

def options(title, genre, artist, date, value):

print('\n')

print("1. Insert a artwork")

print("2. Find maximum valued art")

print("3. Find minimum valued art")

print("4. Show average value of my gallery")

print("5. Delete an artwork")

print("6. Find your art by title")

print("7. Find your arts by year")

print("8. Quit")

option = int(input("Your option: "))

# if statements that will excute your options depending on your answer

if option == 1:

InsertArtwork(title, genre, artist, date, value)

elif option == 2:

FindMaximum(title, genre, artist, date, value)

elif option == 3:

FindMinimum(title, genre, artist, date, value)

elif option == 4:

ShowAverage(title, genre, artist, date, value)

elif option == 5:

DeleteArtwork(title, genre, artist, date, value)

elif option == 6:

FindByTitle(title, genre, artist, date, value)

elif option == 7:

FindByYear(title, genre, artist, date, value)

else:

quit

#define function that will insert a new artwork

def InsertArtwork(title, genre, artist, date, value):

new\_title = input('title: ')

new\_genre = input('genre: ')

new\_artist = input('Artist name: ')

new\_date = input('Year Created: ')

new\_value = int(input('value: '))

#add items to the lists

title.append(new\_title)

genre.append(new\_genre)

artist.append(new\_artist)

date.append(new\_date)

value.append(new\_value)

print('New artwork added!')

#recall options function

options(title, genre, artist, date, value)

#define function that will find the maxium value and call corrosponding indexs

def FindMaximum(title, genre, artist, date, value):

index = value.index(max(value))

print("Maximum valued art: " + title[index])

print("Genre: " + genre[index])

print("Artist: " + artist[index])

print("Date: " + date[index])

print("Value: ", max(value))

#recall options functions

options(title, genre, artist, date, value)

#define function that will find the min value and call corrosponding indexs

def FindMinimum(title, genre, artist, date, value):

index = value.index(min(value))

print("Minimum valued art: " + title[index])

print("Genre: " + genre[index])

print("Artist: " + artist[index])

print("Date: " + date[index])

print("Value: ", min(value))

#recall options function

options(title, genre, artist, date, value)

#define the average function calling your sum devided by your len to get average

def ShowAverage(title, genre, artist, date, value):

average = sum(value)/len(value)

print("Average value of your gallery is: ", average)

#recall options function

options(title, genre, artist, date, value)

#define function that will remove artwork based on title and corrosponding indexs

def DeleteArtwork(title, genre, artist, date, value):

remove\_item = input("Give a title: ")

index = title.index(remove\_item)

title.remove(remove\_item)

genre.remove(genre[index])

artist.remove(artist[index])

date.remove(date[index])

value.remove(value[index])

#recall options

options(title, genre, artist, date, value)

#create function that will call title and corrosponding indexs

def FindByTitle(title, genre, artist, date, value):

item = input("Give a title: ")

index = title.index(item)

print("Title: " + title[index])

print("Genre: " + genre[index])

print("Artist: " + artist[index])

print("Date: " + date[index])

print("Value: ", value[index])

#recall options

options(title, genre, artist, date, value)

#define function that will call year and corrosponding indexs

def FindByYear(title, genre, artist, date, value):

year = input("Give a year: ")

index = date.index(year)

print("Title: " + title[index])

print("Genre: " + genre[index])

print("Artist: " + artist[index])

print("Date: " + date[index])

print("Value: ", value[index])

#recall options

options(title, genre, artist, date, value)

main()

Example:

