Assignment #6 – ‘The Savoy Ticket Seating’

Andrew McDonald

W0426368

**The Math:**

**Constants:**

**Questions:**

NONE!

**Pseudocode:**

(I found a flow chart to be far more useful this time around. Although it is not created in Visio, I have included it at the end of my Pseudocode as it is what I referenced the most.)

1. Ask if purchase or refund
2. If purchase:
3. Display all available and sold seats
4. Ask for input of Row, Seat Number, and Name
5. If seat is empty:

Put name in seat

1. If seat is full:

Display error message Seat already full

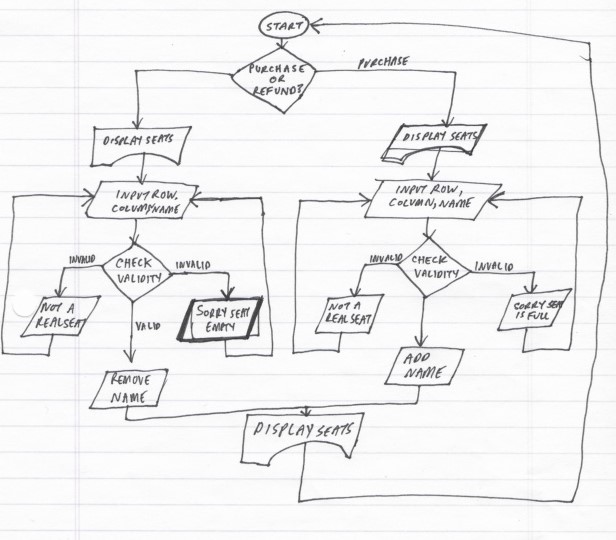
1. Otherwise display invalid entry error message
2. Loop back to 1
3. If refund:
4. Display all available and sold seats
5. Ask for input of Row, Seat Number, and Name
6. If seat is full:

Remove name from seat

1. If seat is empty:

Display error message Seat already empty

1. Otherwise display invalid entry error message
2. Loop back to 1

**Flowchart:**

Refund

**Code:**

#Program: Assignment #6

#Written By: Andrew McDonald (W0426368)

#Date written: Started - 4 Apr 19

#Purpose: Create a program for the Savoy Theatre seating system

#Seating Chart

def Display\_seats(seat\_chart):

row = 0

col = 0

print()

print('\t\tSavoy Theatre Seating Plan')

print()

printline = format ('', '5') + '\t'

for col in range(7):

printline += format('Seat ' + str(col), '5') + '\t'

print(printline)

for row in range(11):

printline = 'Row ' + str(row) + '\t'

for col in range (7):

printline += format(seat\_chart [row] [col], '5') + '\t'

print(printline)

return seat\_chart

#Inputs for purchase

def Take\_inputsAdd(seat\_chart):

print()

print()

#Zeroize input data

Seat\_row = [0]

Seat\_col = [0]

Cust\_name = ['']

#Gather data

Seat\_row = int(input('Enter Row '))

Seat\_col = int(input('Enter Seat '))

Cust\_name = input('Customer Name? ')

#Check if seat exists and is empty

try:

if seat\_chart[Seat\_row][Seat\_col] == 'empty':

#change item in the list

seat\_chart[Seat\_row][Seat\_col] = Cust\_name

elif seat\_chart[Seat\_row][Seat\_col] != 'empty':

print('\nSorry that seat is already sold')

except:

print('Sorry, invalid seat entered')

#Inputs for Refund

def Take\_inputsRem(seat\_chart):

print()

print()

#Zeroize input data

Seat\_row = [0]

Seat\_col = [0]

Cust\_name = ['']

#Gather data

Seat\_row = int(input('Enter Row '))

Seat\_col = int(input('Enter Seat '))

Cust\_name = input('Customer Name? ')

#Check if seat exists and is empty

try:

if seat\_chart[Seat\_row][Seat\_col] == Cust\_name:

#change item in the list

seat\_chart[Seat\_row][Seat\_col] = 'empty'

elif seat\_chart[Seat\_row][Seat\_col] == 'empty':

print('\nSorry that seat cannot be refunded')

except:

print('Sorry, invalid seat entered')

#Main

def main():

#Begin

print('\t\t Welcome to the Savoy Theatre')

print()

print()

#initalize array with 'empty' as value in all seats

Seat\_Chart = [['empty' for \_ in range(7)] for \_ in range (11)]

#Start Loop

#Have user choose Purchase or Refund

Purpose = input('Enter \'p\' to purchase a seat or enter \'r\' to refund a seat: ')

CustEntry = 1000

while CustEntry >= 0:

#If processing a Purchase

if Purpose == 'p':

#Call seat chart

seating = Display\_seats(Seat\_Chart)

#Gather our inputs

Take\_inputsAdd(seating)

#Display NEW seating chart

seating = Display\_seats(Seat\_Chart)

print()

print()

#If processing a Refund

elif Purpose == 'r':

#Call seat chart

seating = Display\_seats(Seat\_Chart)

#Gather our inputs

Take\_inputsRem(seating)

#Display NEW seating chart

seating = Display\_seats(Seat\_Chart)

print()

print()

CustEntry -=CustEntry

#End Loop

Purpose = input('Enter \'p\' to purchase a seat or enter \'r\' to refund a seat: ')

main()