Philip Brendel
Professor Kevin
CS 110
12/13/20

Final Report:

The majority of my program centers around this class called Data. This class uses the variables defined below as lists to store the data I take from txt files as parts of the Data object. The majority of my program, excluding my main function falls under this class.

```python
class Data:

   def __init__(self, month, file, wait, dayTime, data, Time, Day, count):
       self.month = month
       self.file = file
       ###file2 = '
       ## This data was sourced from thrill-data.com
       self.wait = wait
       self.dayTime = dayTime
       self.data = data
       self.Time = Time
       self.Day = Day
       self.count = 0
       for i in data:

           filler = i.split(',')
           dayTime.append(filler[1].split())
           # print(dayTime)
           Time.append(dayTime[count][1].split(':'))

           Day.append(dayTime[count][0].split('-'))
           # creates a separate list that holds the day and the time
           wait.append(int(filler[2]))
           count += 1
```

```python
def main():
   print("This program will provide you with monthly and daily data for the wait time of the Haunted Mansion a
   month = input("What month would you like to see the data for? (1-12) ")
   file = ['/Users/pjbrendel/Desktop/CS 110/DisneyDataJan.txt','/Users/pjbrendel/Desktop/CS 110/DisneyDataFeb.
   file = file[int(month) - 1]
   data = open(file, 'r')
   D = Data(month, file, [], [], data, [], [], 0)
   print("The average wait for this month was:", sum(D.wait) / len(D.wait))
   inp = str(input("Input what day of the month you would like to find wait time data for: "))
   D.findWait([], inp)
   print("The shortest wait time on", (D.month + '/' + inp), "was", D.shortest[0], "minutes long at the time",
         D.timeConvert(D.shortest[1]) + '.', "The average wait time for", (D.month + '/' + inp), "was",
         round(D.mean(D.waitofday)), "minutes", "and the longest wait was", max(D.waitofday), "minutes at",
         D.timeConvert(D.longesttime) + '.', end=' ')

main()
```

The main() function is what the uses interfaces with the class where the rest of the program resides and takes the user's input and sends it into the data class. Here the user will input the Month they would like to use as well as the date, which the main function will process and give the data object a monthly txt file, as well as a specific day from that month, which will provide the data object with enough information to return the average wait from the month, the longest and shortest wait time with the specific time of day that they were at that point, as well as the average wait time of that day.

```python
def mean(self, lst):
    avg = sum(lst) / len(lst)
    return avg
```

This mean function, is simply an easy way for me to have a callable way to calculate the average wait time of the month and the day.

```python
def timeConvert(self, time):
    time = str(time).split(':')
    AMPM = ' '
    if int(time[0]) == 24:
        AMPM = 'AM'
        changedtime = str(12) + ':' + time[1] + ' ' + AMPM
    elif int(time[0]) > 12:
        AMPM = 'PM'
        changedtime = str(int(time[0]) - 12) + ':' + time[1] + ' ' + AMPM
    else:
        AMPM = 'AM'
        changedtime = str(int(time[0])) + ':' + time[1] + ' ' + AMPM


    return changedtime
```

TimeConvert() is a function that takes time data, given in military time and unformatted and formats it nicely as well as giving it and AM/PM value to go with the shift to regular time.

```python
def findWait(self, waitofday, inp):
    self.waitofday = waitofday
    self.shortesttime = 0
    self.longesttime= 0
    for j in range(len(self.dayTime)):

        if int(inp) == int((self.dayTime[int(j)][0]).split('-')[2]):

            self.waitofday.append(int(self.wait[j]))

            if min(waitofday) == int(self.wait[j]):
                self.shortesttime = self.dayTime[int(j)][1]

            if max(waitofday) == int(self.wait[j]):
                self.longesttime = self.dayTime[int(j)][1]

    self.shortest = [min(self.waitofday), self.shortesttime]
```

The findWait() function takes in the wait time data for a particular day and parses through it in order to find the shortest wait, plus when it happened, and the longest wait, plus when it happened.

All these functions, besides the main function, fall within the class data in order to provide the object that is called and referenced in the main function. The main function takes the user's input, provides the class with a txt file and an integer that represents the particular day. The class then parses through this data, storing data from that particular day in a list, which is broken into more lists that hold the data for wait time values and the time at which those wait times were taken (every five minutes). With these lists, the class finds the average wait time of the day, the shortest wait time, and the longest wait time (with their respective time stamps). This is all passed to the main function who outputs this to the user and provides the data within a short paragraph.