Daniel Hanna
Assignment 1 (REVISED)

Assignment 1: tcnjParkingSpaces (REVISED)

VM Path: `/home/student1/tcnjParkingSpaces`

Directory Contents:

`Assignment 1.pdf`	`tcnjParkingSpaces.rb`
`historical_parking.csv`	`parking_lot_contraints.csv`
`readme.pdf`	`revisions.pdf`

Revisions:

1. A significant increase/improvement in modularity has been implemented.
    a. The old program had a very large method titled “possibleParkingLots”, in which a list of legal lots for the user to park in is generated. However, this method included checking for the day of the week as well as the specific time constraints for each day.
    b. The new program utilizes the possibleParkingLots method to obtain the user’s day of week, time, and day of month. It will then utilize this information to transfer over to one of three new methods: weekDayCalculation, fridayCalculation, or weekEndCalculation.
        i. These three methods will calculate the legal lots based off of the user’s time input. In general, all of these methods are much smaller in size, which helps towards increasing the program’s general logic and overall modularity.
    c. As a result, the program runs much more reliably, and any errors that occur will be easier to fix, since the error will be much more localized.

2. Error handling has been improved
    a. The previous program generally handled errors well, but the program failed to ensure that the user’s input for a day of the month was a valid date
        i. This was improved by utilizing an Argument Error rescue, which would then allow the user to re-enter all of their information
        ii. If the user did not enter a proper day of the month, they will still be able to view all the legal lots that they can park in, since they entered a proper day of the week, time, and member title.

3. Case sensitive inputs have been addressed
    a. The previous program required the user's input to be entered as specified in order to proceed with the rest of the program
    b. The new program does not require *any* input to be case sensitive. This includes the user's membership title, time of day, day of week, day of month, and the CSV file locations as well. The user's time of day input may include a space between the number and "AM" or "PM", but it is not required (ie. the user can input <08:00AM> or <08:00 AM> or <8:00AM> or <8:00 AM>, or any combination of these, which includes all capitalized or uncapitalized)
        i. This was also made possible by converting cell contents in the CSV files. No noticeable delay or "hit in performance" is seen by this.

4. Program restarting has been implemented
    a. The previous program did not allow the user to restart the program if desired, and would instead exit after completing the algorithm
    b. The program now allows the user to restart the program as many times as they would like, and even allows them to re-enter the CSV file locations (in case there are different parking lot constraints and historical parking data that they would like to use)

5. CSV Header and Time constants have been implemented
    a. The previous program "hard-wired" the CSV headers and the time constraints into the program. If any of these headers or times were modified in the CSV files, the program would most likely break
    b. The new program implements the use of constants, which "house" all of this header and time data. If new time constraints or CSV column headers are implemented, they can easily be reflected in the program by modifying the constants
        i. These constants are NOT manipulated during run time, but are referenced in order to allow for successful completion of the algorithm.