ITI1120D – Assignment 4
(100 points)

Purpose:

To allow you to exercise with a logical thinking process to formulate algorithms, and to implement the algorithms using the Python Programming Language. The logic will include inputs and outputs, lists, files, and functions, dictionaries.

Questions:

Write a separate Python program for each of the following:

a. Create a function oneRun(range) that repetitively generates random integers in [0-range[and stops once two consecutive values are found to be equal. It returns the number of times random numbers were generated until two consecutively generated numbers were found to be equal. For example, function oneRun would return the value 12 if the following random numbers were generated: 2, 1, 5, 4, 2, 1, 7, 9, 6, 11, **5, 5** (it took 12 random number generations to find two consecutive equal numbers). (15 points)

b. Create a function Experimenter (nTimes, range) that calls the oneRun(range) function nTimes. Experimenter will print the maximum, minimum, and average that the oneRun function returned back. (10 points)

   For example, Experimenter(5, 10) will call oneRun(10) five times. If oneRun returned back values 1, 7, 4, 2, 9, the values that will be printed out are:

   Minimum: 1   Maximum: 9   Average: 4.6

c. You are given a text file that contains the ratings of multiple spectators for certain kinds of movies they watched. The following class definition defines the ratings that a specific person did for multiple kinds of movies.
   - Complete the code of the missing class below.
   - You are then to create a list of all the user ratings you found in the text file.
   - You are to process this list to print out the average rating value for each movie done by the various spectators.

```
class MovieRatings:
        def __init__(self, user_name):
        '''user_name: a string representing the name of the person
        these movie ratings belong to
        '''
        self.name = user_name
        self.scores = {}
```

```
    def rate(self, movie_name, rating):
    '''movie_name: a string representing a movie
    rating: a float out of ten representing this user's rating of the movie
    '''
    self.scores[movie_name] = rating


def get_highest_rating(self):
    '''Returns a float representing the highest rating of all
    the movies this user has rated. Returns 0 if the user
    has not yet rated any movies.
    '''
    # Your code here


def main():
     marys_ratings = MovieRatings("Mary Jones")
     # Your code here:
```