

```

1 """
2 population_increases.py
3 @author: ITP 150
4 Date Created: Oct. 5, 2024
5 https://unicode.org/emoji/charts/full-emoji-list.html
6 """
7
8
9 import csv
10 import os
11
12
13 def main():
14     DISPLAY_LIST = 1
15     STATS = 2
16     SAVE_STATS = 3
17     QUIT = 99
18     pop_list = []
19     pop_stats = {}
20     choice = 0
21     if not os.path.isfile('us_population.csv'):
22         print('Sorry we can\'t run the program at this time.')
23     else:
24         pop_list = read_the_file(pop_list)
25     if pop_list:
26         # print(pop_list) # a sanity check to see if we read the data
27         while choice != QUIT:
28             menu_string = 'Please choose from the following menu: \
29 \nEnter 1 to print the pop list.\
30 \nEnter 2 to analyze pop statistics.\
31 \nEnter 3 to save the statistics.\
32 \nEnter 99 to Quit.'
33             valid_choices = [1, 2, 3, 99]
34             choice = input_menu_choice(menu_string, valid_choices)
35             if choice == DISPLAY_LIST:
36                 display_list(pop_list)
37             elif choice == STATS:
38                 pop_stats = calc_descriptive_stats(pop_list, pop_stats)
39             elif choice == SAVE_STATS:
40                 save_stats(pop_stats)
41             else:
42                 print('Thanks for using the US population tracker.')
43
44
45 def read_the_file(pop_list):
46     try:
47         with open('us_population.csv', newline='') as pop_file:
48             pop_reader = csv.reader(pop_file, delimiter=',') # \t tab
49             pop_list = [row for row in pop_reader]
50             for row in range(1, len(pop_list)):
51                 pop_list[row][0] = int(pop_list[row][0])
52                 pop_list[row][1] = int(pop_list[row][1])
53
54             return pop_list
55     except IOError:
56         print('An error occurred trying to read from the file.')
57     except IndexError:
58         print('An index error occurred.')
59     except Exception:
60         print('An error occurred.')
61
62
63 def input_menu_choice(menu_string, valid_choices):

```

```

64     while True:
65         try:
66             print('-'*50)
67             print(menu_string)
68             print('-'*50)
69             # Get the user's choice
70             choice = int(input())
71             if choice in valid_choices:
72                 return choice
73             else:
74                 raise Exception
75         except ValueError: # throws on invalid data type
76             print('Invalid choice. Please try again: \U0001F600')
77         except Exception: # throws on anything else
78             print('Valid datatype but invalid value. Please try again.')
79
80
81 def display_list(pop_list):
82     print('='*35)
83     print(f'{"As Of Date":12s}{"pop":>20s}')
84     try:
85         for row in range(1, len(pop_list)):
86             print(f'{pop_list[row][0]:12}{pop_list[row][1]:>20,d}')
87         print('='*35)
88     except IndexError:
89         print('Index is out of range.')
90     except Exception:
91         print('An error occurred.')
92
93
94 def calc_descriptive_stats(pop_list, pop_stats):
95     average_pop = calc_average(pop_list)
96     lowest_pop, lowest_year = find_lowest(pop_list)
97     highest_pop, highest_year = find_highest(pop_list)
98     try:
99         labels = ['Average Population', 'Lowest Population', 'Lowest Year',
100                  'Highest Population', 'Highest Year']
101         stats = [average_pop, lowest_pop, lowest_year, highest_pop,
102                  highest_year]
103         stats_zipped = zip(labels, stats)
104         pop_stats = dict(stats_zipped)
105         for key, val in pop_stats.items():
106             print(f'{key:30s}{val:>20,.0f}')
107         return pop_stats
108     except KeyError:
109         print("A key error occurred.")
110     except Exception:
111         print('An error occurred.')
112
113
114 def calc_average(pop_list):
115     sum_pop = 0
116     try:
117         for row in range(1, len(pop_list)):
118             sum_pop = sum_pop + pop_list[row][1]
119         average_pop = sum_pop / (len(pop_list) - 1)
120         # print(average_pop)
121         return average_pop
122     except IndexError:
123         print('Index is out of range')
124     except ZeroDivisionError:
125         print('A Zero Division Error occurred.')
126     except Exception:

```

```

127         print('An error occurred.')
128
129
130 def find_lowest(pop_list):
131     try:
132         # print(pop_list)
133         lowest_pop = pop_list[1][1]
134         lowest_year = pop_list[1][0]
135         for row in range(1, len(pop_list)):
136             if pop_list[row][1] < lowest_pop:
137                 lowest_pop = pop_list[row][1]
138                 lowest_year = pop_list[row][0]
139         return lowest_pop, lowest_year
140     except IndexError:
141         print('An Index Error Occurred.')
142     except Exception:
143         print('An error occurred.')
144
145
146 def find_highest(pop_list):
147     try:
148         highest_pop = pop_list[1][1]
149         highest_year = pop_list[1][0]
150         for row in range(1, len(pop_list)):
151             if pop_list[row][1] > highest_pop:
152                 highest_pop = pop_list[row][1]
153                 highest_year = pop_list[row][0]
154         return highest_pop, highest_year
155     except IndexError:
156         print('An Index Error Occurred.')
157     except Exception:
158         print('An error occurred.')
159
160
161 def save_stats(pop_stats):
162     try:
163         pop_stats_file = csv.writer(open('pop_stats.csv', 'w'))
164         for key, val in pop_stats.items():
165             pop_stats_file.writerow([key, val])
166         print("The pop_stats_file.csv has been updated.")
167     except IOError:
168         print("An error occurred trying to write to the file.")
169     except KeyError:
170         print("A key error occurred.")
171     except Exception:
172         print('An error occurred.')
173
174
175 if __name__ == '__main__':
176     main()
177

```