CSC500

Module 3 Critical Thinking Assignment

GIT Repository: <https://github.com/focomapper/CSC500>

Introduction:

The module 3 Critical Thinking Assignment involves two parts that both obtain user input, perform a calculation, and print formatted results. Part 1 gets user input for a meal price, calculates tax, tip and total cost, and prints out the resulting amounts formatted into dollar and cents. Part 2 gets user input for current time and a time to wait for an alarm to go off, both in hours on a 24-hour clock, calculates based on the user input times when the alarm will go off, and prints that time. Both parts present opportunities to validate user input and to format the results.

Part 1, Python code calculating total cost of meal with tax and tip included:

Pseudo code-

1. Initialize list for meal price, tip and tax
2. Get user input for price of meal and validate it is a number (float), update list
3. Make user input is greater than 0
4. Calculate total cost (meal price + tax + tip) from list
5. Format and print values for tip, tax and total cost

Source code-

# Initialize meal list  
meal = [0, 0.18, 0.07]  
  
# Get price of meal from user and validate that it's a number  
try:  
 meal[0] = (float(input("Enter the price of meal: ")))  
except ValueError:  
 print("Your input was not a number, run program again and enter valid number")  
 exit()  
  
# Check that cost is greater than 0, no free meals at this establishment  
if not meal[0] > 0:  
 print("Your input must be greater than 0, run program again and enter valid number")  
 exit()  
  
# Calculate total meal cost  
total\_cost = meal[0] + (meal[0] \* meal[1]) + (meal[0] \* meal[2])  
  
# Output results with formatting  
print('Total tip is: ' + '${:,.2f}'.format(meal[0] \* meal[1]))  
print('Total tax is: ' + '${:,.2f}'.format(meal[0] \* meal[2]))  
print('Total meal cost is: ' + '${:,.2f}'.format(total\_cost))

Screenshot of code executing-

A screenshot of a computer program

Description automatically generated

Part 2, Python code calculating time alarm will go off given current time and hours to wait for alarm:

Pseudo code-

1. Get user input for current time a validate it is a number, exit if not
2. Validate user input for time from 0 to 23, exit if not
3. Get user input for time to wait for alarm and validate it is a number, exit if not
4. Calculate time alarm will go off
5. Print alarm time

Source code-

# Initialize list  
alarm\_specs = [0, 0]  
  
# Get time from user and validate that it's a whole number  
try:  
 alarm\_specs[0] = (int(input("Enter time in hours: ")))  
except ValueError:  
 print("Your input was not a whole number (hours), run program again and enter valid number")  
 exit()  
  
# Make sure time entered is 23 or less  
if not (23 > alarm\_specs[0] >= 0):  
 print("Your input must be from 0 to 23, run program again and enter valid time")  
 exit()  
# Get time to wait for alarm from user and validate that it's a number  
try:  
 alarm\_specs[1] = (int(input("Enter hours to wait for alarm: ")))  
except ValueError:  
 print("Your input was not a number (hours), run program again and enter valid number")  
 exit()  
  
# Make sure alarm time is a positive number  
if not (23 > alarm\_specs[1] >= 0):  
 print("Your input must be from greater than 0, run program again and enter valid time")  
 exit()  
  
# Calculate time for alarm  
alarm\_time = (alarm\_specs[0] + alarm\_specs[1]) % 24  
  
# Format and print output  
print('Alarm will go off at ' + str(int(alarm\_time)) + ' hundred hours')

Screenshot of code executing-

A screenshot of a computer program

Description automatically generated

References:

Vahid, F., Lysecky, R., & Miler, B. (2022). *CSC500: Principles of programming*. zyBooks. www.zybooks.com. ISBN: 979-8-203-05060-1