# Source/Pseudo Code

print("###########################################################################")

print("# #")

print("# Part 1: #")

print("# Write a program that calculates the total amount of a meal purchased at #")

print("# a restaurant. The program should ask the user to enter the charge for #")

print("# the food and then calculate the amounts with an 18 percent tip and 7 #")

print("# percent sales tax. Display each of these amounts and the total price. #")

print("# #")

print("###########################################################################")

def get\_cost():

cost = input("\nPlease enter the charge for the food in whole dollars or dollars and cents and press enter (Ex. 10.23): $")

# guard for invalid entries

try:

if "." in cost and len(cost.split(".")[-1]) < 2:

print("Please enter a valid amount in dollars and cents.")

return get\_cost()

cost = float(cost)

except:

print("Please enter a valid amount in dollars and cents.")

return get\_cost()

return cost

# get meal total from user

meal\_cost\_dollars = get\_cost()

print("\nYou entered: $%0.2f" % meal\_cost\_dollars)

# solution per written instruction "then calculate the amounts with an 18 percent tip and 7 percent sales tax."

print("\nSolution per written instruction 'then calculate the amounts with an 18 percent tip and 7 percent sales tax.\nNote: Here tip is applied pretax, and tax is applied to both cost and tip.'")

print(" 18%% tip: $%0.2f" % round(meal\_cost\_dollars \* 0.18, 3))

print(" 7%% tax: $%0.2f" % round(round(meal\_cost\_dollars \* 1.18, 3) \* 0.07, 3))

print("Total with tax and tip: $%0.2f" % (round(round(meal\_cost\_dollars \* 1.18, 3) \* 1.07, 3)))

# some states sales tax is not required to be applied to tips, add sales tax first then tip

print("\nAlternate solution, some states sales tax is not required to be applied to tips, add sales tax first then tip.\nNote: Tip is applied to pretax cost and tax is only applied to the original cost.")

print(" 18%% tip: $%0.2f" % round(meal\_cost\_dollars \* 0.18, 3))

print(" 7%% tax: $%0.2f" % round(meal\_cost\_dollars \* 0.07, 3))

print("Total with tax and tip: $%0.2f" % round(meal\_cost\_dollars \* (1 + 0.07 + 0.18), 3))

print("\n\n")

print("###########################################################################")

print("# #")

print("# Part 2: #")

print("# Many people keep time using a 24-hour clock (11 is 11am and 23 is 11pm, #")

print("# 0 is midnight). If it is currently 13 and you set your alarm to go off #")

print("# in 50 hours, it will be 15 (3pm). Write a Python program to solve the #")

print("# general version of the above problem. Ask the user for the time now (in #")

print("# hours) and then ask for the number of hours to wait for the alarm. Your #")

print("# program should output what the time will be on a 24-hour clock when the #")

print("# alarm goes off. #")

print("# #")

print("###########################################################################")

def get\_time():

# although prompt asks for hours

time = input("\nPlease enter the time in military time (just hours 0-23)\nand press enter (Ex. if its 1735, enter 17): ")

# guard for invalid entries

try:

time = int(time)

if not (0 <= time and time <= 23):

print("\nPlease enter a valid time in military time.")

return get\_time()

except:

print("\nPlease enter a valid time in military time.")

return get\_time()

return time

def get\_hours\_to\_wait():

# although prompt asks for hours

hours\_to\_wait = input("\nPlease enter hours to wait before alarm and press enter: ")

# guard for invalid entries

try:

if not hours\_to\_wait.isnumeric():

print("\nPlease enter an integer.")

return get\_hours\_to\_wait()

hours\_to\_wait = int(hours\_to\_wait)

except:

print("\nPlease enter an integer.")

return get\_hours\_to\_wait()

return hours\_to\_wait

# get user input for current time

time\_now = get\_time()

# get user input for hours to wait before alarm

hours\_to\_wait = get\_hours\_to\_wait()

# calculate alarm time

time\_alarm = (time\_now + hours\_to\_wait) % 24

# format time

def format\_time(time):

if time < 10:

return "0" + str(time)

else:

return str(time)

time\_now\_display = format\_time(time\_now)

time\_alarm\_display = format\_time(time\_alarm)

# output current time and alarm time

print("\nThe current time is %s00 hours." % time\_now\_display)

print("Alarm will go off in %s hour(s)." % hours\_to\_wait)

print("Alarm will go off at %s00 hours." % time\_alarm\_display)

# Application Screenshot

A screenshot of a computer program

Description automatically generated

# Github link

<https://github.com/wheyluhai/csuglobal/tree/main/CSC500-1/Module_3>