Problem Set – More on Functions

1. Prompt the user to repeatedly to do the program( input (Yes or No)). If they respond Yes, go into the loop and prompt them for last name, month and sales. Write a function to compute next month’s forecast. Pass to the function month and sales. Determine the forecast percent (see below) and compute next month’s sales to be sales x (1+forecast percent). Return next month’s sales and display the value.

Month Forecast Percent

Jan, Feb, Mar 0.10

Apr, May, Jun 0.15

Jul, Aug, Sep 0.20

Oct, Nov, Dec 0.25

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| lname | if month in ['Jan', 'Feb', 'Mar']:  return 0.10  elif month in ['Apr', 'May', 'Jun']:  return 0.15  elif month in ['Jul', 'Aug', 'Sep']:  return 0.20  elif month in ['Oct', 'Nov', 'Dec']:  return 0.25  else:  return 0 | next\_month\_sales = compute\_forecast(month, sales)  print(f"{last\_name}, your forecasted sales for next month ({month}) are: ${next\_month\_sales:.2f}")  elif user\_input == 'no':  print("Thank you for using the sales forecast program!")  break  else:  print("Invalid input. Please enter 'Yes' or 'No'.") |
| month | next\_month\_sales = sales \* (1 + forecast\_percent) |  |
| sales | ask to repeat loop |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Prompt the user to repeatedly to do the program( input (Yes or No)). If they response Yes go into the loop and prompt the user for length, width and height of a room. Write a function to compute the square footage of the room. The function should receive the length, width and height of the room and return square footage (2 x length x width (floor and ceiling) + 2 x length x height (2 of the walls) + 2 x width x height (the other 2 walls). A gallon of paint covers 50 square feet. Compute the number of gallons needed to paint the room (square footage of the room / 50). Display the number of gallons needed.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| length | floor = 2 \* L \* W \* H | gallons\_needed |
| width | wall = 2 \* L \* W \* H |  |
| height | tsquarefoot = floor + wall |  |
|  | gallons\_needed = square\_footage / 50 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Prompt the user to repeatedly to do the program (input (Yes or No)). If they response Yes go into the loop and prompt the user for make, model, electric vehicle code (Y or N) and MSRP (sticker price) of an automobile. Write a function to compute the out the door price. Pass to the function the MSRP, make, model and electric vehicle code. Determine the percent off the MSRP then compute the new MSRP and finally add 7% sales tax to the total. Return and display the total. Also sum all MSRP’s and sum of all sales price of the cars (MSRP – discount + tax).

To determine percent off MSRP Percent off MSRP

Honda Accord 0.10

Toyota Rav4 0.15

All electric vehicles 0.30

All other vehicles 0.05

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| make | get\_discount\_percent(make, model, ev\_code):  if make == "Honda" and model == "Accord":  return 0.10  elif make == "Toyota" and model == "Rav4":  return 0.15  elif ev\_code.upper() == "Y":  return 0.30 # All electric vehicles  else:  return 0.05 | print total\_msrp  print total\_salep |
| model | def compute\_out\_the\_door\_price(msrp, make, model, ev\_code):  discount\_percent = get\_discount\_percent(make, model, ev\_code)  discount\_amount = msrp \* discount\_percent  new\_msrp = msrp - discount\_amount  sales\_tax = new\_msrp \* 0.07  total\_price = new\_msrp + sales\_tax  return total\_price |  |
| ev code | ask user if they want data and to repeat loop |  |
|  | total\_msrp += msrp  total\_sales\_price += out\_the\_door\_price |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Prompt the user to repeatedly to do the program( input (Yes or No)). If they response Yes go into the loop and prompt the user for last name and miles from downtown Chicago. Write a function to compute the train ticket price. Pass to the function the miles from down town Chicago and determine the ticket price. Return the ticket price. Sum price of all tickets.

Miles from Down Town Chicago Ticket Price

30 or more $12

20 to 29 $10

10 to 19 $8

All others $5

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| lname | def compute\_ticket\_price(miles):  if miles >= 30:  return 12  elif 20 <= miles < 30:  return 10  elif 10 <= miles < 20:  return 8  else:  return 5 | print lname, ticket\_price, |
| miles from downtown | ticket\_price = compute\_ticket\_price(miles)  total\_ticket\_price += ticket\_price |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Prompt the user to repeatedly to do the program( input (Yes or No)). If they response Yes go into the loop and prompt the user for county and market value of a home. Write a function to compute the assessed value. Pass to the function the county and market value. The function will determine the assessed value percent then compute and return the assessed value. (Multiple the market value by assessed value percent. Sum and display all market values and assessed values.

County Assessed Value Percent

Cook 0.90

DuPage 0.80

McHenry 0.75

Kane 0.60

All others 0.70

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| county, market value | def compute\_assessed\_value(county, market\_value):  assessed\_value\_percent = {  "Cook": 0.90,  "DuPage": 0.80,  "McHenry": 0.75,  "Kane": 0.60  }.get(county, 0.70) | print county, value |
|  | assessed\_value = compute\_assessed\_value(county, market\_value)  total\_market\_value += market\_value  total\_assessed\_value += assessed\_value |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |