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, sales |  |  |
| Month | Def CompForecast(month)  If month = jan, feb, mar  Percent = .1  If month = apr, may, jun  Percent = .15  If month = jul, aug, sep  Percent = .2  If month = oct, nov, dec  Percent = .25  Else  Percent = 0  (return) |  |
|  | R = input(“do you want to run this program?)  Input lname, month, sales  Nextsales = sales \* CompForecast(month) + 1  Print CompForecast, nextsales  R = input(“do you want to run the program?”) | CompForecast(month)  nextsales |
|  |  |  |
|  |  |  |

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, width, height | CompSqFt(length, width, height):  sqft = 2 \* (length \* width) + 2 \* (length \* height) + 2 \* (width \* height)  return sqft |  |
|  | r = input("Do you want to run the program? ")  while r == "yes":  input length, width, height  sqft = CompSqFt(length, width, height)  print("Square footage of the room: ", sqft)  gal = sqft / 50  print("Gallons of paint needed: ", gal)  r = input("Do you want to run the program? ") | Sqft  gal |
|  |  |  |
|  |  |  |
|  |  |  |

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, model, ecode, msrp | def CompPercent(make, model, ecode):  if make + model == "Honda Accord":  percent = 0.10  return percent  if make + model == "Toyota Rav4":  percent = 0.15  return percent  if ecode == "Y":  percent = 0.30  return percent  if ecode == "N":  percent = 0.05  return percent |  |
|  | make = input("Enter car make: ")  model = input("Enter car model: ")  ecode = input("Enter electric vehicle code (Y or N): ")  msrp = float(input("Enter car MSRP: $"))  percent = CompPercent(make, model, ecode)  off = msrp \* percent  newmsrp = msrp - off  tax = newmsrp \* .07  total = newmsrp + tax  sum = sum + total | Total, sum |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Miles | Ticketprice(miles)  if miles >= 30:  ticket = 12  return ticket  if miles >= 20 and miles <= 29:  ticket = 10  return ticket  if miles >= 10 and miles <= 19:  ticket = 8  return ticket |  |
|  | r = input("Do you want to run this program? ")  while r == "yes":  lname = input("Enter your last name: ")  miles = int(input("How many miles from downtown Chicago? "))  ticket = TicketPrice(miles)  print("Your train ticket price is: $" + str(TicketPrice(miles)))  r = input("Do you want to run this program again? ")  sum = sum + ticket  print("Sum: $", sum) | Output sum, ticket |
|  |  |  |
|  |  |  |

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, mvalue |  |  |
|  | AssessedValue(county):  if county == "Cook":  percent = 0.90  return percent  if county == "DuPage":  percent = 0.80  return percent  if county == "McHenry":  percent = 0.75  return percent  if county == "Kane":  percent = 0.60  return percent  else:  percent = 0.7  return percent |  |
|  | r = input("Do you want to run this program? ")  while r == "yes":  mvalue = float(input("Enter market value: "))  county = input("House county: ")  percent = (AssessedValue(county))  avalue = percent \* mvalue  output mvalue, avalue  r = input("Do you want to run this program again? ")  sum avalue, mvalue | Mvalue, avalue, sums |
|  |  |  |