Problem Set - Functions Pass By Value

1. Allow the user to enter a quantity and price, use ctl+z to stop. Use a function to compute the total (quantity times price). The function should be passed the quantity and price and then return the total. In the function, provide a 10% discount if the total is over $10,0000.00. Display quantity, price and total. Sum and display the extended price.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
|  | CompExtPrice(qty, unitprice)  Extprice = qty\*unitprice  If extprice > 10000  Discamt = extprice \* 0.10  Else  Discamt = 0  newExtPrice = extPrice – discamt  return newExtPrice |  |
| Qty |  | Extprice |
| price | Main  totalExtPrice = 0  Do you want to do this program (Yes or No)  While (Yes)  Input qty, price  Extprice = CompExtPrice(qty,price)  Display qty, price, Extprice  totalExtPrice = totalExtPrice + extprice  Do you want to continue with this program? |  |
|  |  |  |
|  | Display totalExtPrice | totalExtPrice |
|  |  |  |
|  |  |  |

1. Enter players last name, number of hits and at bats at the keyboard, use ctl+z to stop. Use a function to compute batting average. Pass the hits and at bats to the function. The function should return batting average. Display last name and batting average. Give a count of the number of players entered.

|  |  |  |
| --- | --- | --- |
| Input | Proccess | Output |
| Student’s last name | Prompt for students ln | |  | | --- | | show student's ln |  |  | | --- | |  | |
| Credit hours | Prompt for credit and validate unput | Show credit hours |
| Discrict code | Prompt for district code and validate input | Show district code |
|  | Use function ‘compute\_tuition(credit\_hours, district\_code) to calculate tuition | Compute and display tuition owed |
|  | Check if district code is 'I' or 'O' and apply respective rate per credit hour |  |
|  | |  | | --- | | Compute total tuition by summing up individual tuition amounts |  |  | | --- | |  | | Compute and display tuition owed |
|  | |  | | --- | | Handle Ctrl+Z (EOFError) to stop input loop |  |  | | --- | |  | |  |
|  | |  | | --- | | Accumulate total tuition owed for all students |  |  | | --- | |  | |  |
|  | |  | | --- | | Handle invalid input (e.g., non-numeric credit hours) and prompt user again |  |  | | --- | |  | | Display error message if invalid |

1. Enter the destination city, miles travelled and gallons used for a trip, use ctl+z to stop. Use a function to compute miles per gallon. Pass miles travelled and gallons used to the function. The function should return miles per gallon. Count the number of entries made (number of trips) Display destination city, miles and mpg. At end display the number of entries made.

|  |  |  |
| --- | --- | --- |
| Input | proccess | output |
| Destination city | Prompt for destination city | Show destination city |
| Miles traveled | Prompt for miles traveled and validate input | Show miles traveled |
| Gallons used | Prompt for gallons used and validate input | Show gallons used |
|  | Use function compute\_mpg(miles\_traveled, gallons\_used) to calculate miles per gallon | Compute and show miles per gallons (mpg) |
|  | |  | | --- | | Calculate miles per gallon using the formula: mpg = miles traveled / gallons used |  |  | | --- | |  | |  |
|  | |  | | --- | | Count the number of trips (entries made) |  |  | | --- | |  | | Display # of entries made |
|  | |  | | --- | | Handle Ctrl+Z (EOFError) to stop input loop |  |  | | --- | |  | |  |
|  | |  | | --- | | Handle invalid input (e.g., non-numeric miles or gallons) and prompt user again |  |  | | --- | |  | | Display error message if invalid |
|  |  |  |

1. Allow the employee to enter last name, job code and hours worked, use ctl+z to stop. Use a function to determine the pay rate. Pass to this function the job code and it should return rate of pay. Use Job code L is $25/hr, A is $30/hr and J is $50/hr for respective pay rates. Compute gross pay. Give time and a half for overtime. Display last name and gross pay. Sum and display total of all gross pay.

|  |  |  |
| --- | --- | --- |
| Employee’s ln | Prompt for employee’s ln | Show employee’s ln |
|  | |  | | --- | | Prompt for job code and validate input (L, A, J) |  |  | | --- | |  | | Show job code |
|  | Prompt for hours worked and validate input | Display hours worked |
|  | Use function get\_pay\_rate(job\_code) to determine pay rate based on job code | Determine and display pay rate |
|  | Calculate gross pay: |  |
|  | If hours worked <= 40: gross pay = hours worked \* pay rate If hours worked > 40: gross pay = (40 \* pay rate) + ((hours worked - 40) \* (pay rate \* 1.5)) |  |
|  | Accumulate total gross pay for all employees | Display gross pay |
|  | Handle Ctrl+Z (EOFError) to stop input loop | Show total gross pay |
|  | Handle invalid input (e.g., non-numeric hours worked) and prompt user again |  |
|  |  | Show error message if invalid |

1. Allow the user to enter student last name, credit hours and district code, use ctl+z to stop. Use a function to compute tuition owed. Charge In district (code of I) $250 per credit hour. Out of district (code of O) is $550 per credit hour. The function should receive credit hours and district code and return tuition owed. Display student name and tuition owed. Sum and display total of all tuition owed.

|  |  |  |
| --- | --- | --- |
| input | process | Output |
| Student ln | Prompt for student’s ln | Show student’s ln |
| Credit hours | Prompt for credit hours and validate input | Show credit hours |
| |  | | --- | |  | | |  | | --- | |  |   Prompt for district code and validate input | Show district code |
|  | |  | | --- | |  |   Use function compute\_tuition(credit\_hours, district\_code) to calculate tuition owed | Compute and display tuition owed |
|  | |  | | --- | |  |   Check if district code is 'I' or 'O' and apply respective rate per credit hour |  |
|  | |  | | --- | |  |   Compute total tuition by summing up individual tuition amounts | Display total tution owed |
|  | |  | | --- | |  |   Handle Ctrl+Z (EOFError) to stop input loop |  |
|  | Accumulate total tuition owed for all students |  |
|  | |  | | --- | | Handle invalid input (e.g., non-numeric credit hours) and prompt user again |  |  | | --- | |  | | Display error message if invalid |

Examples

1. Enter the number of Points and redemption code. For redemption code C then compute value as 2 x rewards points. Redemption code X then they get 3 x rewards points. All other codes get 1.5 x rewards points. Write a function that receives points and redemption code and computes rewards points. Display points, redemption code and rewards points.
2. Enter two numbers and operation code (A, S, M, D). Write a function that receives the two numbers and uses the operation code to perform an operation on the two numbers (A=addition, S=Subtraction, M=Multiplication, D=Division). Check for dividing by 0. If the second number is 0 then set result to -999. Display two number, operation code, result and message if attempt to divide by zero.
3. Allow the user to enter a string. The string can be entered with any case (all upper, all lower of mixed). Write a function that accepts the string and returns all lower case when the original string is all upper or mixed. If the original string is all lower then make the string all upper case. The function should return the new string. Display both the original and new string.