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 | Process | output |
| Player's last name  Number of hits  Number of at bats  Ctrl+Z | Use a function to compute the batting average using the formula: batting average = hits / at bats. Hits and at bats are passed to the function. The function returns the batting average. Store each player's last name and batting average. Maintain a counter to track the number of players entered. | Display each player's last name and batting average. Display the total number of players entered. |

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 | Process | Output |
| Destination city  Miles traveled Gallons used | Function calculates miles per gallon (MPG) using the formula: MPG = miles traveled / gallons used. Miles traveled and gallons used are passed to the function. The function returns MPG. Store each destination city, miles traveled, and MPG. Maintain a counter to track the number of trips entered. | Display each trip’s destination city, miles traveled, and MPG. At the end, display the total number of trips entered. |

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.

|  |  |  |
| --- | --- | --- |
| input | process | output |
| |  | | --- | |  |  |  | | --- | | Last name   Job code  Hours worked  Ctrl+Z | | Use a function to determine the pay rate based on the job code: Job code L: $25/hr. Job code A: $30/hr. Job code J: $50/hr.  Pass the job code to the function and return the corresponding pay rate. Calculate gross pay based on hours worked and the pay rate. If hours worked > 40, calculate overtime at time and a half (1.5x the pay rate). Store each employee's last name, gross pay, and hours worked. Maintain a counter for the total number of employees and sum of gross pay. | Display each employee's last name and gross pay.  Display the total gross pay for all employees. |

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** | **ouput** |
| Student's last name  Credit hours  District code  Ctrl+ z | Use a function to compute tuition owed based on the district code and credit hours:  In-district (I): $250 per credit hour. Out-of-district (O): $550 per credit hour.  The function should receive credit hours and district code and return the tuition owed. Store each student's last name and tuition owed. Maintain a counter for the number of students and a sum for the total tuition owed. | Display each student's last name and tuition owed. Display the total tuition owed for all students. |

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.