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 |
| Quantity | Def total(quantity, price)  Comput total = quantity \* price  If comput total > 10,000  Discount = compute total \* 10%  else  discount = 0  grand total = compute total – 0  return grand total | The quantity is {quantity} |
| Price | Sum extended price = 0 | The price is {price} |
| Prompt = Want to run the program? | While Y  Grand total = total(quantity, price)  Sum extended price = sum extended price + grand total | The total is {grand total} |
| Prompt = Want to run the program again? |  | The sum of all the totals is {sum extended price} |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Prompt | Number of players | Player last name, their batting average is batting average |
| Player last name | While prompt  Player last name  Number of hits  At bats  At bats = average(number of hits, at bats) | The number of players entered |
| Number of hits | Number of players + 1 |  |
| Prompt | Def average(number of hits, at bats  Batting average = number of hits / at bats  Return batting average |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Number of trips = 0 | Def compute(miles travelled, gallons used)  Miles per gallon = miles travelled / gallons used  Return miles per gallon | Your end point is destination city |
| Prompt = Run the program? | While prompt = Y  Miles per gallon = compute(miles travelled, gallons used) | You travelled miles travelled |
| Destination City = What is the destination? | Number of trips = number of trips +1 | Your MPG is miles per gallon |
| Miles travelled = How many miles? |  | The number of trips entered is number of trips |
| Gallons used = How many gallons? |  |  |
| Prompt = Run the program again? |  |  |
|  |  |  |

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 |
| Total gross pay = 0 | Def compute(job code)  If job code = L  Pay rate = 25  Elif job code = A  Pay rate = 30  Else  Pay rate = 50  Return pay rate | {last name} your gross pay is $ {gross pay} |
| Prompt = Run the program Y/N? | While prompt = Y  Pay rate = compute(job code)  If hours worked >= 40  Gross pay = (pay rate \*40) + ((hours worked – 40) \* (pay rate \* 1.5))  Else  Gross pay = hours worked \* pay rate  Total gross pay = total gross pay + gross pay | This is the total gross pay for all employees: $ {total gross pay} |
| Last name = What is your last name? |  |  |
| Job code = What is your job code? (L/A/J) |  |  |
| Hours worked = int(What is your hours worked?) |  |  |
| Prompt = Run the program again? |  |  |
|  |  |  |

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 |
| Prompt = Run the program? (Y/N) | Def compute(credit hours, district code)  If district code == I  Credit charge = 250  Else  Credit charge = 500  Tuition = credit charge \* credit hours  Return tuition | {Last name} your tuition owed is $ |
| Last name = What is your last name? | Total all tuition = 0 | This is the total tuition cost for all students: ${total alltuition} |
| Credit hours = int(How many credit hours are you taking?) | While prompt = Y  Tuition = compute(credit hours, district code  Total all tuition = total all tuition + tuition |  |
| District code = What is your district code? (I/O) |  |  |
| Prompt = Run the program again? |  |  |
|  |  |  |
|  |  |  |

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.