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** |
| --- | --- | --- |
|  | def calculate\_batting\_average(number\_of\_hits, bats):  return number\_of\_hits / bats |  |
| number\_of\_hits | player\_count = 0  r = input("Do you want to start? (Yes or No): ").lower() | Yes or No |
| bats | while r == "yes":  last\_name = input("Enter Last Name: ")  hits = int(input("Enter Number of Hits: "))  at\_bats = int(input("Enter Number of At Bats: "))  bat\_avg = calculate\_batting\_average(hits, at\_bats)  print("Last Name:", last\_name)  print("Batting Average:", bat\_avg)  player\_count += 1  print("Total Players:", player\_count)  r = input("Do you want to continue? (Yes or No): ").lower() |  |
|  | Display: Last Name, Bat Avg, Total Players | Last Name, Bat Avg, Total Players |

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** |
| --- | --- | --- |
| calculate\_mpg | def calculate\_mpg(miles\_traveled, gallons):  return miles\_traveled / gallons |  |
|  | gallons = 0  miles\_traveled = 0  total\_miles\_traveled = 0 |  |
| Destination  miles\_traveled  Gallons | r = input("Do you want to start? (Yes or No): ").lower()  while r == "yes":  destination = input("Enter the destination you want to visit: ")  miles\_traveled = float(input("Enter the number of miles travled: "))  gallons = int(input("Enter the amount of gallons used: ")) |  |
| total\_number\_trips | calculate\_mpg(miles\_traveled, gallons) |  |
| Print | print(f"Destination: {destination}")  print(f"Miles traveled: {miles\_traveled}")  print(f"Miles per gallon each trip: {total\_number\_trips}") | Destination  Miles traveled  Miles per gallon each trip |
|  | total\_miles\_traveled += 1 |  |
|  | Display: Total number of trips | total\_miles\_traveled |
| r = input("Do you want to start? (Yes or No): ").lower() | Enter yes or no | Yes or no |
| Print: "Total number of trips: ", total\_miles\_traveled | Display: total\_miles\_traveled | total\_miles\_traveled |

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** |
| --- | --- | --- |
|  | def calculate\_payrate(job\_code):  pay\_rate = 0.0  if job\_code == 'L':  pay\_rate = 25  elif job\_code == 'A':  pay\_rate = 30  elif job\_code == 'J':  pay\_rate = 50  return pay\_rate  total\_gross\_pay = 0.0 |  |
| last\_name  job\_code  Hours | while r == "yes":  last\_name = input("Enter last name: ")  job\_code = input("Enter job code: ")  hours = int(input("Enter hours worked: "))  pay\_rate = calculate\_payrate(job\_code) |  |
| Hours  Gross\_pay | if hours > 40:  gross\_pay = (pay\_rate \* 40) + (1.5 \* pay\_rate) \* (hours - 40)  else:  gross\_pay = pay\_rate \* hours  total\_gross\_pay += gross\_pay |  |
|  | print(f'Last name: {last\_name}')  print(f'Gross Pay $: {gross\_pay}') | last\_name  gross\_pay |
| r = input("Do you want to continue? (Yes or No): ").lower() | Display: Yes or no | Yes or no |
| Total Gross Pay for all employees: $ | Display”: total\_gross\_pay | total\_gross\_pay |

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** |
| --- | --- | --- |
|  | def tuition\_owed(credit\_hours, district\_code):  tuition = 0.0  if district\_code == 'I':  tuition = 250 \* credit\_hours  elif district\_code == 'O':  tuition = 550 \* credit\_hours  return tuition  total\_tuition = 0.0 |  |
| r = input("Do you want to start? (Yes or No): ").lower() |  | Yes or no |
| last\_name  district\_code  credit\_hours  Tuition | while r == "yes":  last\_name = input("Enter last name: ")  district\_code = input("Enter your district code: ")  credit\_hours = int(input("Enter your credit hours: "))  tuition = tuition\_owed(credit\_hours, district\_code)  total\_tuition += tuition |  |
| print(f'Last name: {last\_name}')  print(f'Tuition owed $: {tuition}') | Display: last\_name  tuition | last\_name  tuition |
| r = input("Do you want to continue? (Yes or No): ").lower() | Display: Yes or no | Yes or no |
| "Total Tuition owed for all students: $” | Display: total\_tuition | total\_tuition |

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.