Problem Set – Basic if-else logic. Each of these problems will contain a basic if statement. In addition, they should contain at least one function.

1. Develop a Flowgorithm and code for this problem.

The input to this program should be quantity of items ordered and unit price. Use a function that receives the quantity and unit price and returns the extended price (quantity x unit price). Compute tax of 7% of the extended price.

Next, determine the shipping charge. For orders of $100 or more, then there is free shipping (shipping charge is 0). For orders under $100, shipping is 10% of the order.

Display quantity ordered, unit price, shipping charge and total order (extended price + tax + shipping).

1. Develop pseudocode and code for this problem.

The program asks the user for a widget and quantity. Determine the price of the widgets based on the table below using a function. You can assume that the user will enter either A or B as the Widget (in other words if the Widget is not A then it must be B even though they may enter any character – B will be the default. You may also assume that capital letters are entered, otherwise B or the default price will be used).

Compute the extended price to be quantity times the price per widget.

Display the widget, price per widget, and extended price.

Widget Price Per Widget

A $10.00

B $20.00

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| widget | Fun get price (widget)  If widget = “A”  Pruce per widget = 10.00  Else  Price per widget = 20.00  Return price per widget | Price per widget  Ext\_price |
|  |  |  |
|  |  |  |
|  |  |  |
| qty | Ext price = qty\* price per widget |  |
|  |  |  |
|  | Display price per widget, ext price |  |

def fun\_get\_up(widget):  
 if widget = **"A"** up=10.00  
 else:  
  
  
 return(up)  
  
*#input phase*widget = input (**"Enter Widget (A or B)"**)  
qty = input (**"Enter quantity ordered:"**)  
  
*#process phase*up = fun\_get\_up(widget)  
  
ext\_price = float(qty) \* float(up)  
  
*#output phase*print(**"Unit Price: $"**, up)  
print(**"Extended Price: $"** , ext\_price)

1. Develop pseudocode and code for this problem.

Create a program to prompt the user for employee last name, hours and rate per hour and then compute gross pay (hours \* rate) in a function. Include a calculation to give 1.5 times the hourly rate for any overtime (hours worked above 40 hours (overtime hours = hours - 40). For example, 50 hours worked at $10 per hour with overtime is $550.

Display employee last name, hours, rate and gross pay.

|  |
| --- |
| hrs = input("Enter Hours:") |
|  | h = float(hrs) |
|  | xx = input("Enter the Rate:") |
|  | x = float(xx) |
|  | if h <= 40: |
|  | print( h \* x) |
|  | elif h > 40: |
|  | print(40\* x + (h-40)\*1.5\*x) |

1. Extra Credit. Modify Problem 3 above to display regular hours, regular pay, overtime hours and overtime pay as well as employee last name, hours, rate and gross pay.

Hint: For hours of 40 or less, overtime hours and overtime pay is 0.