CIS 106
Problems – Using Nested if and Compound Relational Conditions

For each problem, develop the IPO and Code.

1. The input to the problem is quantity of widgets. Your program should determine the price to charge based on the schedule below. Calculate the extended price (quantity x price). Calculate tax at 7%. Display the extended price, tax amount and total.

Quantity	Price
>10000	$10
5000 to 10000	$20
Below 5000	$30

- Input quantity of widgets	- If quantity > 10,000, then price is 10 Elif quantity >= 5,000, then price is 20 Else price is 30 - Extended price = quantity x price - Tax = .07 x extended price - Total = extended piece + tax	- Print extended price - Tax amount - total

```
#input phase
quantity = float(input("enter quantity of widgets"))

#process phase
if quantity > 10000:
  price = 10
elif quantity >= 5000:
  price = 20
else:
  price = 30

extended_price = quantity * price
tax = extended_price * 0.07
total = extended_price + tax

#output phase
print("Extended price:$"), extended_price
print("Tax amount:$", tax)
print("total:$", total)
```

2. Enter a part number and quantity Determine the cost per unit using the table below. Then calculate the total cost (quantity x unit cost). Display the part number, cost per unit and total cost. Note: Part number can be an integer but it can also be a string because you are not doing arithmetic on it. However in your code if statement be sure to compare using consistency, that is, if item == "10" when item is a string and if item == 10 when item is an integer.

Part	Unit Cost
10 **or** 55	1.00
99	2.00
80 **or** 70	3.00
All others	5.00

- Input part number - Quantity	- If part == "10" or "55", then cost is 1 Elif part == "99", then cost is 2 Elif part == "80" or "70", then cost 3 Else, cost is 5 - Total cost = quantity x cost	- Print part number - Cost - Total cost

```
#input phase
part = float(input("Enter part number:"))
qty = float(input("Enter quantity:"))

#process phase
if part == 10 or part == 55:
  cost = 1.00
elif part == 99:
  cost = 2.00
elif part == 80 or part == 70:
  cost = 3.00
else:
  cost = 5.00

total = cost * qty

#output phase
print("part number:", part)
print("unit cost:$", cost)
print("total cost", total)
```

3. Enter a principle amount of a CD and year to maturity of CD. Determine the interest rate based on the amount of the principle **and** maturity (see below). Calculate first year interest (principle x interest rate). Display principle, interest rate and the interest amount for first year.

Principle	Years to Maturity	Interest Rate
>$100,000	5	6%
$50,000 to $100,000	10	5%
$50,000 to $100,000	5	4%
Any other principle and years		2%

- Input principle of CD - Years to maturity	- If principle is >100,000 AND years == 5, then rate is 6% Elif principle is >= 50,000 AND years == 10, then rate is 5% Elif principle >= 50,000 AND years == 5, then rate is 4% Else, rate is 2% - Interest = principle x rate - Total = principle + interest	- Print principle - Interest rate - Total interest amount for first year

```
#input phase
principle = float(input("Enter principle of CD:"))
years = float(input("Enter years to maturity:"))

#process phase
if principle > 100000 and years == 5:
  rate = .06
elif principle >= 50000 and years == 10:
  rate = .05
elif principle >= 50000 and years == 5:
  rate = .04
else:
  rate = .02

interest = principle * rate
total = principle + interest

#output phase
print("principle:", principle)
print("interest rate:", rate)
print("total interest for first year", total)
```

4. Allow the user to enter number of concert tickets. The price per ticket depends on the volume (see below). Display the number of tickets, price per ticket and the total cost (number of tickets x Price Per Ticket).

Quantity	Price Per Ticket
>=25	$50
10 to 24	$60
5 to 9	$70
Less 5	$75

- Input qty of concert tickets	- If qty >= 25, then price is 50 Elif qty >=10, then price is 60 Elif qty >=5, then price is 70 Else price is 75 - Total = qty x price	- Print # of tickets - Price per ticket - Total cost

```
#input phase
qty = float(input("enter quantity of tickets:"))

#process phase
if qty >= 25:
  price = 50
elif qty >= 10:
  price = 60
elif qty >= 5:
  price = 70
else:
  price = 75

total = qty * price

#output phase
print("quantity of tickets:", qty)
print("price per ticket:$", price)
print("total cost:$", total)
```

5. The user will enter employee last name, salary and job level (as noted below). Use the job level to determine the bonus rate. Then compute bonus to be salary times bonus rate. Display employee last name and bonus.

Job Level	Bonus Rate
10 and above	25%

5 to 9 20%

All others 10%

- Input employee last name - Salary - Job level	- If level >= 10, then rate 25% Elif level >= 5, then rate 20% Else rate 10% - Bonus = salary x rate	- Print employee last name - bonus

```
#input phase
name = input("Enter last name:")
salary = float(input("Enter salary:"))
level = float(input("Enter job level:"))

#process phase
if level >= 10:
  rate = .25
elif level >= 5:
  rate = .20
else:
  rate = .10

bonus = salary * rate

#output phase
print("Last name:", name)
print("bonus amount:$", bonus)
```