BBM 200: Introduction to Programming

CAT 1: Python Programming Questions for Business Cases

John Karuitha, PhD

NB: As much as possible, write your programs in the form of reusable functions and anticipate errors that you can resolve using the try …. except blocks.

Question ONE: Sales Tax Calculator

Write a program that calculates the total price of an item after applying a sales tax. The program should:
- Take the item's price and tax rate as inputs.
- Calculate and display the total price.

Example Input:
Price: 100, Tax Rate: 15%
Expected Output:
Total Price: 115

NB: Write this program as a function (the def keyword)

```
def total_price(price, tax_rate):
    """
    Enter price and tax rate. The tax rate should be in the form 0.15 for 15%
    """
    return (price + (price * tax_rate))

total_price(100, 0.20)
```

```
120.0
```

Question TWO: Employee Salary Bonus

Create a program that calculates the annual salary of an employee after including a performance bonus.

- The user enters the employee's monthly salary and the bonus percentage.

- The program calculates the total annual salary, including the bonus.

  Example Input:
  Monthly Salary: 20,000, Bonus: 10%
  Expected Output:
  Annual Salary (with bonus): 264,000

```
def total_pay(month_pay, bonus_rate):
    """
    Enter monthly pay and a bonus like 0.15 for 15%
    """
    annual_pay = month_pay * 12
    bonus = annual_pay * bonus_rate
    return(annual_pay + bonus)

total_pay(200000, 0.15)
```

```
2760000.0
```

Question THREE: Inventory Reorder Alert

Develop a program that checks whether an item in stock needs to be reordered.
- The program should take the current stock and reorder threshold as inputs.
- If the stock is below the threshold, display "Reorder Needed"; otherwise, display "Stock is Sufficient."

Example Input:
Current Stock: 15, Reorder Threshold: 20
Expected Output:
Reorder Needed

```python
def reorder(current_stock, threshold):
    """
    Enter the current stock and the threshold stock
    that will trigger a reorder
    """
    if current_stock < threshold:
        return("Reorder needed.")
    else:
        return("Stock is OK")

print(reorder(1000, 2000))
```

```
Reorder needed.
```

Question FOUR: Discounted Price Calculation

Write a program for a shop owner to calculate the discounted price of an item.
- The user inputs the original price and discount percentage.
- The program calculates and displays the discounted price and the amount saved.

Example Input:
Original Price: 200, Discount: 25%
Expected Output:
Discounted Price: 150, You Saved: 50

```python
def discounted_price(price, discount):
    """
    Enter price and discount as a fraction like 0.15 for 15%
    """
    payment = price * (1 - discount)
    return(payment)

discounted_price(1000, 0.20)
```

```
800.0
```

Question FIVE: Loan Payment Estimator

Create a program that calculates the monthly payment for a loan.
- Inputs: Loan amount, annual interest rate (as a percentage), and the loan term (in years).
- The program should calculate the monthly payment using the formula:

$$M = \frac{P \cdot r \cdot (1+r)^n}{(1+r)^n - 1}$$

Where (P) is the loan amount, (r) is the monthly interest rate (annual rate divided by 12), and (n) is the total number of payments (years (×) 12).

Example Input:
Loan Amount: 500,000, Annual Interest Rate: 5%, Loan Term: 15 years
Expected Output:
Monthly Payment: 3950.33

```
def loan_repayment(amount, period, interest_rate):
    """
    Enter loan amount.
    Enter Loan period in years
    Enter interest rate as 0.15 for 15%.
    """
    interest_rate = interest_rate / 12
    period = period * 12
    num = amount * interest_rate * ((1 + interest_rate) ** period)
    den = ((1 + interest_rate) ** period) - 1
    return(round(num / den))

loan_repayment(800000, 5, 0.12)
```

```
17796
```

Question SIX: Customer Feedback Analysis

Write a program to analyze customer feedback by counting the occurrences of positive and negative words.
- Input: A string containing customer feedback, and lists of positive and negative keywords.
- Output: Count the number of positive and negative words in the feedback.

Example Input:
Feedback: “The service was excellent, but the delivery was slow.”
Positive Words: [“excellent”, “good”, “amazing”]
Negative Words: [“bad”, “slow”, “poor”]
Expected Output:
Positive Words: 1, Negative Words: 1

NB: This is a bit hard but give it a try anyway.

```
def sentiment(customer_feedback):
    positive_count = 0
    negative_count = 0
    positive_words = ["excellent", "good", "amazing"]
    negative_words = ["bad", "slow", "poor"]
    words = customer_feedback.replace(",", "").replace(".", "").lower().split()
    for word in words:
        if word in positive_words:
            positive_count += 1
        elif word in negative_words:
            negative_count += 1
    return(f"Positive words: {positive_count}; Negative words: {negative_count}")

sentiment("The service was bad, but the delivery was slow.")
```

```
'Positive words: 0; Negative words: 2'
```