**PYTHON CODE**

1. **QUESTION**:

Use variables to create a list of workers dynamically (at least 400 workers)

**Solution:**

To dynamically create a list of 400 workers, I can use unique IDs or placeholder names to represent them. For example:

**Using IDs:** Worker\_1, Worker\_2, ..., Worker\_400.

This approach ensures that each worker is identified with a unique ID number, which must conform to various scenarios.

**CODE:**

# Number of workers to create

num\_workers = 400

# Create a list of workers (dynamically)

workers = [f"Worker\_{i+1}" for i in range(num\_workers)]

# Print a few workers to verify

print(workers[:20]) # Output: ['Worker\_1', 'Worker\_2', ..., 'Worker\_20']

# Print the total number of workers

print(f"Total workers: {len(workers)}") # Output: Total workers: 400

1. **Question:**

Utilize a for loop to generate payment slips for each of the 400 workers.

**Solution:**

**Code:**

# Number of workers

num\_workers = 400

# Generate payment slips for each worker

for i in range(1, num\_workers + 1):

# Worker ID or name

worker\_name = f"Worker\_{i}"

# Payment slip details

payment\_slip = f"Payment Slip for {worker\_name}"

# Print or save the payment slip

print(payment\_slip)

**Output:**

Payment Slip for Worker\_1

Payment Slip for Worker\_2

Payment Slip for Worker\_3

Payment Slip for Worker\_4

...

Payment Slip for Worker\_400

1. **QUESTION:**

Implement the following conditional statements within the for loop:

1. If the salary is greater than $10,000 and less than $20,000, assign the Employee level as "A1."
2. If the salary is greater than $7,500 and less than $30,000 and the employee is female, set the Employee level as "A5-F."

**Solution**:

**Python Code**

import random

# Number of workers

num\_workers = 400

# Generate worker data (salary and gender)

workers\_data = [{"worker\_name": f"Worker\_{i+1}","salary": random.randint(5000,35000),"gender": random.choice(["Male", "Female"]),} for i in range(num\_workers)]

# Process each worker and assign employee level

for worker in workers\_data:

salary = worker["salary"]

gender = worker["gender"]

# Assign employee level based on conditions

if 10000 < salary < 20000:

worker["employee\_level"] = "A1"

elif 7500 < salary < 30000 and gender == "Female":

worker["employee\_level"] = "A5-F"

else:

worker["employee\_level"] = "Unspecified"

# Print worker details

print(f"Name:{worker['worker\_name']}, Salary: ${salary}, Gender: {gender}, Level: {worker['employee\_level']}")

1. **QUESTION:**

Add exception handling to your Python code to address potential errors.

**SOLUTION:**

**Code for Exception Handling**

import random

# Number of workers

num\_workers = 400

# Generate workers, salaries, and gender data

workers = [f"Worker\_{i+1}" for I in range(num\_workers)]

salaries = [random.randint(5000, 35000) for\_in range(num\_workers)]

genders = [random.choice(["Male", "Female"]) for\_in range(num\_workers)]

# Process each worker with exception handling

for i in range(num\_workers):

try:

# Retrieve worker details

worker\_name = workers[i]

salary = salaries[i]

gender = genders[i]

employee\_level = "Unspecified" #Default level

# Conditional logic for employee level assignment

if 10000 < salary < 20000:

employee\_level = "A1"

elif 7500 < salary < 30000 and gender == "Female":

employee\_level = "A5-F"

# Display worker details

print(f"{worker\_name} | Gender: {gender} | Salary: ${salary} | Level: {employee\_level}")

except IndexError as e:

print(f"Error: Index out of bounds at worker {i}: {e}")

except TypeError as e:

print(f"Error: Invalid data type encountered for worker {workers[i] if i < len(workers) else 'Unknown'}: {e}")

except Exception as e:

print(f"Unexpected error for worker {workers[i] if i < len(workers) else 'Unknown'}: {e}")