

Industry Problem Statement

Employee Performance & Payroll Processing System (Python)

Business Background

An IT services company wants to build a Python-based internal tool to manage:

- Employee details
- Performance ratings
- Salary calculations
- Bonus and tax processing

You are asked to implement this system incrementally, following good coding practices used in real projects.

Task 1: Capture Employee Details (Input Validation)

Objective

Collect and store basic employee information.

Requirements

Write a program to accept:

- Employee ID
- Employee Name
- Basic Monthly Salary

Business Rules

- Salary must be greater than ₹15,000

- If invalid, display a meaningful error message

Expected Outcome

Validated employee record ready for further processing.

Task 2: Store Performance Ratings

Objective

Record employee performance over time.

Requirements

- Accept monthly performance ratings (scale 1–5)
- Store ratings in a list

Business Rules

- Rating must be between 1 and 5
- Ignore or reject invalid entries

Expected Outcome

A clean list of performance ratings.

Task 3: Calculate Average Performance Score

Objective

Evaluate overall employee performance.

Requirements

- Calculate the average of the ratings list
- Round the result to 2 decimal places

Expected Outcome

A single performance score used for appraisals and bonuses.

Task 4: Performance Classification

Objective

Convert numerical ratings into business-friendly categories.

Rules

Average Rating	Performance Level
≥ 4.5	Excellent
≥ 4.0	Good
≥ 3.0	Average
< 3.0	Needs Improvement

Expected Outcome

Readable performance classification for HR reporting.

Task 5: Gross Salary Calculation

Objective

Compute gross monthly salary from basic salary.

Salary Components

- HRA = 20% of Basic Salary
- DA = 10% of Basic Salary

Formula

Gross Salary = Basic + HRA + DA

Expected Outcome

Accurate gross salary value.

Task 6: Tax Deduction Logic

Objective

Apply income tax based on salary slabs.

Tax Rules

Gross Salary	Tax Rate
$\geq ₹80,000$	20%
$\geq ₹50,000$	10%
$< ₹50,000$	No Tax

Expected Outcome

Correct monthly tax amount.

Task 7: Net Salary Calculation

Objective

Determine take-home salary.

Formula

Net Salary = Gross Salary - Tax

Expected Outcome

Final payable salary amount.

Task 8: Bonus Eligibility Check

Objective

Decide if employee qualifies for performance bonus.

Rules

Employee is eligible if:

- **Experience \geq 5 years**
- **Performance level is Good or Excellent**

Expected Outcome

Clear Yes/No bonus eligibility decision.

Task 9: Bonus Calculation

Objective

Calculate bonus amount for eligible employees.

Rule

- **Bonus = 10% of Basic Salary**

Expected Outcome

Bonus amount (0 if not eligible).

Task 10: Consolidated Payroll Summary (Procedural)

Objective

Generate a payroll summary using variables and functions.

Summary Should Include

- Employee ID & Name
- Performance Level
- Gross Salary
- Tax
- Bonus
- Net Pay

Expected Outcome

A readable payroll statement.

Task 11: Employee Class Design (OOP)

Objective

Model employee as a real-world object.

Create a class **Employee** with:

Attributes

- emp_id
- name
- basic_salary
- experience
- ratings

Task 12: Salary-Related Methods

Objective

Encapsulate salary logic inside the class.

Methods

- `calculate_gross_salary()`
- `calculate_tax()`
- `calculate_net_salary()`

Task 13: Performance Methods

Objective

Handle performance logic within the class.

Methods

- `calculate_average_rating()`
- `get_performance_level()`

Task 14: Bonus Processing Method

Objective

Automate bonus calculation.

Method

- `calculate_bonus()`

Bonus should be added to net salary if eligible.

Task 15: Final Payroll Report Generation

Objective

Generate a professional payroll report.

Output Format (Example)

Employee ID : E102

Name : Anjali

Performance : Good

Gross Salary : ₹65,000

Tax Deducted : ₹6,500

Bonus : ₹5,000

Net Pay : ₹63,500