

Python Comprehension - Real-Time Scenario-Based Assignments

1. Employee Email Extractor (List Comprehension)

Extract all email IDs from a list of employee records.

Sample Input:

```
employee_data = [  
    'John Doe - john.doe@company.com',  
    'Jane Smith - jane.smith@company.com',  
    'Michael Lee - michael.lee@company.com'  
]
```

Expected Output:

```
['john.doe@company.com', 'jane.smith@company.com', 'michael.lee@company.com']
```

2. Active Project Count per Department (Nested List Comprehension)

From nested list of department-wise project statuses, count active projects.

Sample Input:

```
project_status = [  
    ['completed', 'active', 'active'],  
    ['active', 'completed'],  
    ['completed', 'active', 'active', 'active']  
]
```

Expected Output:

```
[2, 1, 3]
```

3. HR - Salary Filter (Dict Comprehension)

From a dictionary of employee salaries, keep only those with salary > 50,000.

Sample Input:

```
salaries = {  
    'John': 48000,  
    'Jane': 55000,  
    'Mark': 51000,  
    'Lucy': 45000  
}
```

Expected Output:

```
{'Jane': 55000, 'Mark': 51000}
```

4. Department-wise Average Ratings (Nested Dict Comprehension)

Extract department-wise average ratings.

Sample Input:

```
employee_ratings = {  
    'IT': {'John': 3.5, 'Alice': 4.0},  
    'HR': {'Bob': 4.2, 'Emma': 3.8}  
}
```

Expected Output:

```
{'IT': 3.75, 'HR': 4.0}
```

5. Client Feedback Flags (List of Dicts Comprehension)

Extract client feedbacks where rating is below 3.

Sample Input:

```
feedback = [  
    {'client': 'A', 'rating': 4},  
    {'client': 'B', 'rating': 2},  
    {'client': 'C', 'rating': 5},  
    {'client': 'D', 'rating': 1}  
]
```

Expected Output:

```
[{'client': 'B', 'rating': 2}, {'client': 'D', 'rating': 1}]
```

6. Finance - Tax Filing Eligibility Filter (Nested Dict Comprehension)

Filter employees with income > 5L and investment < 1.5L.

Sample Input:

```
employee_finance = {  
    'Sales': {  
        'E001': {'income': 650000, 'investment': 140000},  
        'E002': {'income': 450000, 'investment': 100000}  
    },  
    'IT': {  
        'E003': {'income': 800000, 'investment': 200000},  
        'E004': {'income': 700000, 'investment': 130000}  
    }  
}
```

```
}
```

Expected Output:

```
{'Sales': {'E001': ...}, 'IT': {'E004': ...}}
```

7. Healthcare - Critical Patient Alerts (Nested List Comprehension)

Extract patient IDs with BP > 140 and Sugar > 200.

Sample Input:

```
patient_data = [  
    {'id': 'P101', 'bp': 130, 'sugar': 210},  
    {'id': 'P102', 'bp': 145, 'sugar': 190},  
    {'id': 'P103', 'bp': 150, 'sugar': 220}  
]
```

Expected Output:

```
['P103']
```

8. Logistics - Package Status Summary (Dict of List Comprehension)

Generate count of delivered packages per city.

Sample Input:

```
package_status = {  
    'Mumbai': ['delivered', 'pending', 'delivered'],  
    'Chennai': ['pending', 'pending'],  
    'Delhi': ['delivered', 'delivered', 'delivered']  
}
```

Expected Output:

```
{'Mumbai': 2, 'Chennai': 0, 'Delhi': 3}
```

9. HR - Skillset Filter (Nested Dict Comprehension)

Filter employees having both 'Python' and 'SQL'.

Sample Input:

```
skills = {  
    'Data': {  
        'Emp01': ['Python', 'SQL', 'ML'],  
        'Emp02': ['Excel', 'SQL']  
    },  
    'Analytics': {  
        'Emp03': ['Python', 'SQL'],
```

```
    'Emp04': ['Python']  
}  
}
```

Expected Output:

```
{'Data': {'Emp01': [...]}, 'Analytics': {'Emp03': [...]}}
```

10. Retail - Salesperson Monthly Incentive Calculation (Dict Comprehension)

Calculate 10% incentive if sales > 50000.

Sample Input:

```
monthly_sales = {  
    'S001': 72000,  
    'S002': 49000,  
    'S003': 51000,  
    'S004': 30000  
}
```

Expected Output:

```
{'S001': 7200.0, 'S003': 5100.0}
```