Python Exercises

**Topics Covered:**

* Classes and Objects
* Constructors (\_\_init\_\_)
* Instance Methods
* Lists of Objects
* Basic Input/Output
* Filtering, Searching, Updating in lists

**Exercise 1: Create an Employee Class**

**Problem Statement:**

Create a class named Employee with the following attributes:

* emp\_id (integer)
* name (string)
* department (string)
* salary (float)

Add a constructor that initializes these fields. Write a method display\_details() that prints the employee details.

**Exercise 2: Create a Project Class**

**Problem Statement:**

Define a class named Project with the following attributes:

* project\_id (string)
* project\_name (string)
* client\_name (string)

Add a constructor to initialize these attributes. Add a method show\_project() to display project information.

**Exercise 3: List of Employees**

**Problem Statement:**

Create a list of 5 Employee objects using constructor input.  
Write a method to print details of all employees in the list.

**Exercise 4: Assign Projects to Employees**

**Problem Statement:**

Extend the Employee class to include a list of assigned projects (as Project objects).  
Write a method assign\_project(self, project) to add a project to an employee.  
Write another method list\_projects() to print all assigned projects for the employee.

**Exercise 5: Find Employee by Department**

**Problem Statement:**

Given a list of employees, write a function that takes a department name as input and prints the details of all employees in that department.

**Exercise 6: Calculate Total Salary Expense**

**Problem Statement:**

Write a function that takes a list of employees and calculates the total salary expense of the company.

**Exercise 7: Search Employee by ID**

**Problem Statement:**

Ask the user to input an emp\_id.  
Write a function that searches for an employee with that ID in the employee list and displays their details.  
If the employee is not found, print an appropriate message.

**Exercise 8: Filter High-Earning Employees**

**Problem Statement:**

Write a function that takes the employee list and a salary threshold as input.  
Return all employees who earn more than the given threshold.

**Exercise 9: Remove Project from Employee**

**Problem Statement:**

Add a method in the Employee class to remove a project from the employee’s project list by project ID.

**Exercise 10: Display All Employees with Their Projects**

**Problem Statement:**

Write a function that prints all employee details along with the names of the projects assigned to each employee.

**Exercise 11: Department-wise Employee Count**

**Problem Statement:**

Write a function that groups employees by department and prints the count of employees in each department.

**Exercise 12: Update Employee Salary**

**Problem Statement:**

Write a method in the Employee class to update the salary.  
Then write a function that takes an employee ID and a new salary, and updates the salary of that employee.

**Exercise 13: Employee-Project Relationship Mapping**

**Problem Statement:**

Assume a single project can be assigned to multiple employees.  
Write code to:

* Create a list of projects.
* Assign projects to multiple employees.
* Print the list of employees assigned to each project.

**Exercise 14: Sort Employees by Salary**

**Problem Statement:**

Write a function that sorts the employee list based on salary in descending order and displays them.

**Exercise 15: Generate Company Report**

**Problem Statement:**

Generate a report that includes:

* Total number of employees
* Average salary
* Department-wise count
* Number of projects assigned to each employee

Use a combination of list traversals, conditions, and print formatting.