

Exercises:-

Flow Control

1. **Customer Eligibility Checker**
Check if a user is eligible for a loan based on age, income, and credit score using conditional statements.
2. **Order Discount System**
Calculate the discount for an e-commerce order based on amount:
 - a. ₹5000 → 20%
 - b. ₹1000 → 10%
 - c. Otherwise → 0%
3. **Inventory Reorder Loop**
Loop through a list of product quantities and print those that are below the reorder level (say, 10).

Lists and Tuples

4. **Product Catalogue Management**
Manage a list of tuples where each tuple represents a product: `(product_name, price, stock_quantity)`. Add, update, and remove products.
5. **Top Sales Report**
Given a list of daily sales, sort and extract the top 5 sales days.
6. **Flight Information Search**
Given a list of flights as tuples: `(flight_no, origin, destination, price)`, write a program to search for flights between two cities under a given price.

Functions

7. **Currency Converter**
Write a function that converts currency using exchange rates. Example:
`convert(amount, from_currency, to_currency)`
8. **Email Validator Function**
Create a function that takes a string and validates if it's a proper email using basic string rules.
9. **Order Processing Function**
Create a function that takes a list of item prices and returns the total bill after adding 18% GST.

Dictionaries and Sets

10. **Employee Database**
Store employee data (name, age, department, salary) in a dictionary. Implement a simple search function to find employees by department.

11. Duplicate Data Detection

From a list of emails, use a set to remove duplicates and print only unique entries.

12. Word Frequency Counter

From a paragraph, count the occurrence of each word using a dictionary.

13. Inventory Stock Merger

Merge two dictionaries representing inventory from two warehouses and update total quantities.

OOP Concepts

14. Customer Class

Create a `Customer` class with attributes: `name`, `email`, `loyalty_points`. Add methods to add points and check status.

15. Online Order System

Create a `Product` class and an `Order` class. Allow adding products to orders and calculating total cost.

16. Student Grading System

Create a `Student` class with methods to add grades and calculate GPA.

17. Banking System with Inheritance

Create a base class `Account` and derive `SavingsAccount` and `CurrentAccount` with different withdrawal rules.

18. Library Management System

Create classes for `Book`, `Member`, and `Library`. Allow borrowing and returning books, and update book availability.

19. Vehicle Rental System (OOP - Class & Methods)

Create a class `Vehicle` with attributes like `vehicle_type`, `brand`, `rental_price_per_day`.

Then create a method `calculate_rent(days)` that returns the total rental cost.

Extend it with a child class `Car` that adds `seating_capacity`, and another class `Bike` that adds `engine_capacity`.

Use case: Ride-sharing or vehicle rental platforms like Zoomcar, Bounce.

20. Employee Attendance Tracker (OOP + Class Variables)

Create a class `Employee` with attributes: `emp_id`, `name`, `department`.

Use a class variable `total_present` to keep track of how many employees marked present today.

Implement a method `mark_attendance()` that updates this count.

Use case: HR systems or attendance management tools.

