Records can be implemented using various data structures in Python, such as **dictionaries, namedtuples, classes, or data classes.** Each of these approaches has its own advantages and use cases.

**Using Dictionaries:**

1. Contact Book:

• Create a contact book using dictionaries, where each contact has fields for name, phone number, and email.

• Allow users to add, update, and delete contacts.

2. Student Database:

• Build a student database using dictionaries, with fields for name, roll number, and marks in different subjects.

• Implement functions to calculate average marks and find the top scorer.

3. Library Catalog:

• Design a library catalog using dictionaries, where each book record includes fields for title, author, and availability.

• Implement borrowing and returning books with status updates.

**Using Namedtuples:**

4. Employee Records:

• Create namedtuples to represent employee records with fields for name, employee ID, and department.

• Allow users to search for employees by ID and list employees in a specific department.

5. Product Inventory:

• Use namedtuples to build a simple product inventory, where each product has fields for name, price, and quantity.

• Implement functions to add products, update prices, and check stock levels.

**Using Classes:**

6. Bank Account Management:

• Define a class for bank accounts with attributes like account number, balance, and account holder name.

• Implement methods for deposit, withdrawal, and balance inquiry.

7. Student Management System:

• Design a class for students with attributes like name, roll number, and marks.

• Implement methods for calculating grade, adding marks, and displaying student details.

8. Employee Payroll System:

• Create a class for employees with attributes like name, salary, and designation.

• Develop methods for calculating bonuses, updating salaries, and displaying employee information.

**Using Data Classes:**

9. Inventory Management:

• Use data classes to create an inventory management system with fields for product name, price, and quantity.

• Implement methods to add, remove, and update products.

10. Flight Reservation System:

• Design data classes to represent flights with fields for flight number, departure, and destination.

• Develop methods to book seats, check availability, and display flight information.