**1. Office Supplies (Pens & Pencils)**

* A **stationery store** sells different types of **pens and pencils**. Each **pen** has a **brand, color, and ink type**, while each **pencil** has a **hardness level (HB, 2B, etc.) and an eraser (Yes/No)**.
* **What would be the class, objects, properties, and methods?**

Ans. **Class: Pen**

**Objects:**

Examples of objects that can be instantiated:

1. Pen ("Parker", "Blue", "Gel")
2. Pen ("Reynolds", "Black", "Ballpoint")

Properties:

. brand (e.g., Parker, Reynolds)

.color (e.g., Blue, Black, Red)

. ink\_type (e.g., Gel, Ballpoint, Fountain)

Methods:

.write(): Simulates writing with the pen.

Class: Pencil

**Objects:**

Examples of objects that can be instantiated:

1. Pencil("2B")
2. Pencil("HB")

Properties:

. hardness\_level (e.g., HB, 2B, 4B)

. has\_eraser (Boolean)

Methods:

.erase(): Simulates erasing if the pencil has an eraser.

. sharpen(): Simulates sharpening the pencil.

**2. Library System**

* A **library** has thousands of **books**. Each **book** has a **title, author, and genre**. The library allows users to **borrow and return books**.
* **Define the class, objects, properties, and methods for this scenario.**

Ans. Class: Book

Class:user

**Objects:**

1. book1, book2, ... (Instances of Book)
2. user1, user2, ... (Instances of User)

Properties:**Book and user**

* title (String)
* author (String)
* genre (String)
* is\_borrowed (Boolean)
* name (String)
* borrowed\_books (List of books the user has borrowed)

Methods: book and user

get\_info() → Returns book details (title, author, genre).

* borrow\_book(book, library) → Requests to borrow a book from the library.
* return\_book(book, library) → Requests to return a book to the library.
* view\_borrowed\_books() → Displays books currently borrowed by the user.

**3. Car Showroom**

* A **car showroom** has different **brands of cars**. Each **car** has a **model name, color, fuel type, and price**. Customers can **purchase a car**.
* **What would be the class, objects, properties, and methods?**

Ans. Class: Car and Customer

**Objects:**

1. car1, car2, ... (Instances of Car)
2. customer1, customer2, ... (Instances of Customer)

Properties:

**Car (Class)**

* brand (String)
* model (String)
* color (String)
* fuel\_type (String)
* price (Float)
* is\_sold (Boolean)

**Customer (Class)**

* name (String)
* purchased\_cars (List of cars the customer has bought)
* budget (Float)

Methods:

**Car Methods**

* get\_info() → Returns car details (brand, model, color, fuel type, price).

**Customer Methods**

* buy\_car(car, showroom) → Attempts to buy a car from the showroom if within budget.
* view\_purchased\_cars() → Displays cars purchased by the customer.

**4. Online Food Ordering System**

* A **restaurant** has an online menu with **different food items**. Each **food item** has a **name, price, and category (veg/non-veg)**. Users can **place an order**.
* **How would you represent this using classes and objects?**

**Ans . Classes:**

FoodItem and User

**Objects:**

1. food1, food2, ... (Instances of FoodItem)
2. user1, user2, ... (Instances of User)

Properties:

**FoodItem (Class)**

* name (String)
* price (Float)
* category (String: "Veg" or "Non-Veg")

**User (Class)**

* name (String)
* order\_history (List of Order objects)

Methods:

**FoodItem Methods**

* get\_info() → Returns food details (name, price, category).

**User Methods**

* view\_menu(restaurant) → Displays the restaurant’s menu.
* place\_order(restaurant, food\_items) → Places an order at the restaurant.
* view\_order\_history() → Displays the user’s past orders.

**5. Mobile Phone Shop**

* A **mobile shop** sells **smartphones**. Each **phone** has a **brand, model, RAM, and price**. Customers can **buy a phone** and **check phone details**.
* **Define the class, objects, and methods.**

Ans. **Classes:**

* Smartphone
* Customer

**Objects:**

1. **phone1, phone2, ... (Instances of Smartphone)**
2. **customer1, customer2, ... (Instances of Customer)**

**Properties:**

**Smartphone (Class)**

* **brand (String)**
* **model (String)**
* **RAM (String)**
* **price (Float)**
* **is\_sold (Boolean)**

**Customer (Class)**

* **name (String)**
* **purchased\_phones (List of smartphones bought)**
* **budget (Float)**

**Methods:**

**Smartphone Methods**

* **get\_info() → Returns smartphone details (brand, model, RAM, price).**

**Customer Methods**

* **buy\_phone(phone, shop) → Attempts to buy a smartphone from the shop if within budget.**
* **view\_purchased\_phones() → Displays phones purchased by the customer.**

**6. Movie Ticket Booking System**

* **A movie theatre allows users to book movie tickets. Each movie has a name, duration, and language. Users can book, cancel, or check available seats.**

**Ans . Classes:**

* **Movie**
* **User**
* **Booking**

**Objects:**

1. **movie1, movie2, ... (Instances of Movie)**
2. **user1, user2, ... (Instances of User)**
3. **booking1, booking2, ... (Instances of Booking)**

**Movie (Class)**

* **name (String)**
* **duration (Float, in hours)**
* **language (String)**

**User (Class)**

* **name (String)**
* **bookings (List of Booking objects)**

**Booking (Class)**

* **user (Reference to User object)**
* **movie (Reference to Movie object)**
* **num\_tickets (Integer)**
* **status (String: "Booked", "Cancelled")**

**Methods:**

**Movie Methods**

* **get\_info() → Returns movie details (name, duration, language).**

**User Methods**

* **view\_movies(theatre) → Displays available movies in the theatre.**
* **book\_ticket(theatre, movie, num\_tickets) → Requests ticket booking.**
* **cancel\_booking(booking, theatre) → Requests booking cancellation.**
* **view\_bookings() → Displays the user's past and current bookings.**

**Booking Methods**

* **confirm\_booking() → Marks a booking as confirmed.**
* **cancel\_booking() → Marks a booking as canceled and releases seats.**

**7. Hospital Management System**

* **A hospital has multiple doctors and patients. Each doctor has a name, specialization, and years of experience, while each patient has a name, age, and disease.**
* **What are the classes, objects, and methods?**

**Ans. Classes:**

* **Doctor**
* **Patient**
* **Appointment**

**Objects:**

1. **doctor1, doctor2, ... (Instances of Doctor)**
2. **patient1, patient2, ... (Instances of Patient)**
3. **appointment1, appointment2, ... (Instances of Appointment)**

**Properties:**

**Doctor (Class)**

* **name (String)**
* **specialization (String)**
* **years\_of\_experience (Integer)**

**Patient (Class)**

* **name (String)**
* **age (Integer)**
* **disease (String)**

**Appointment (Class)**

* **doctor (Reference to Doctor object)**
* **patient (Reference to Patient object)**
* **date (String or Date object)**
* **status (String: "Scheduled", "Completed", "Cancelled")**

**Methods:**

**Doctor Methods**

* **view\_patients() → Displays all patients assigned to the doctor.**
* **view\_appointments() → Shows the doctor's upcoming appointments.**

**Patient Methods**

* **view\_doctors(hospital) → Displays available doctors in the hospital.**
* **book\_appointment(hospital, doctor, date) → Requests an appointment with a doctor.**
* **cancel\_appointment(appointment, hospital) → Requests cancellation of an appointment.**
* **view\_medical\_history() → Displays past medical records.**

**Appointment Methods**

* **confirm\_appointment() → Marks an appointment as confirmed.**
* **complete\_appointment() → Marks an appointment as completed.**
* **cancel\_appointment() → Cancels the appointment.**