**Warm-Up: Thinking in Classes**

Today we’ll practice breaking down real-world problems into pieces of software.  
Don’t worry about being “right.” The goal is to *practice thinking like a programmer*.

**How to do this activity**

1. Read each short scenario and its requirements. Your group picks ONE.
2. Brainstorm: *What classes (nouns) might we need?*
   * Example: A library system might need a Book class and a Patron class.
3. EXTRA: Do you think any of the classes might be related to one another? How?
4. EXTRA: Pick **one** class you listed and design it.
   * Give it some properties (with Java data types).
   * List a few methods it might have (just names are fine for now).

**Scenarios**

**1) Public Library Checkout**

* A patron can check out up to 10 books.
* Books have authors, copies, and due dates.
* Late returns cost money.

**2) Fitness Class Sign-Ups**

* Members can sign up for classes (like yoga, spin, pilates).
* Each class has a max number of spots.
* Instructors teach classes at set times.

**3) Food Delivery App**

* Customers order food from restaurants.
* Drivers deliver orders.
* Orders include multiple food items.

**4) School Grading System**

* Students enroll in courses.
* Teachers assign grades.
* Courses have assignments and exams.

**5) Online Store**

* Shoppers add items to a cart.
* Each item has a price and description.
* Orders can be paid with a credit card.

**6) Movie Theater Tickets**

* People can buy tickets for a movie showing.
* Each showing has a date, time, and theater room.
* Tickets have seat numbers.

**7) Pet Adoption Center**

* Animals are available for adoption.
* People can apply to adopt.
* Each animal has a name, species, and age.

**8) Ride-Sharing App**

* Riders request rides.
* Drivers accept rides.
* Each ride has a start and end location.

**9) Hotel Booking**

* Guests can reserve rooms.
* Rooms have different types (single, double, suite).
* Bookings have check-in and check-out dates.

**10) Music Playlist App**

* Users create playlists.
* A playlist can contain many songs.
* Songs have titles, artists, and lengths.

**Extra Credit Example**

Pick one class you listed and “design” it:  
**Example: Book**

* Properties:
  + String title
  + String author
  + boolean isCheckedOut
  + LocalDate dueDate
* Methods:
  + checkOut()
  + returnBook()
  + renew()

👉 Your turn: Pick a class from one of the scenarios and sketch out its properties and methods!