**Problem 1: Fitness Tracking App**

**Problem 1 Statement:**

Decompose the development of a fitness tracking app that can track users' workouts, monitor their diet, and provide health insights.

**Instructions:**

1. Define the main goal of the application.
2. Use the top-down approach to break down the app into its major features.
3. Further decompose each feature into smaller, manageable tasks.
4. Identify opportunities for modularization.

**Main Goal of the Application**

To build a fitness tracking app that helps users:

* Log and monitor their **workouts**
* Track their **diet and nutrition**
* Receive **personalized health insights**

**Top-Down Approach – Step 1: Identify Major Features**

Break the app into its main functional areas:

1. **Workout Tracking**
2. **Diet Monitoring**
3. **Health Insights**
4. **User Management**

**Top-Down Approach – Step 2: Decompose Each Feature**

**1. Workout Tracking**

* Add a new workout (type, duration, calories)
* View workout history
* Plan future workouts (calendar, goals)
* Display workout progress (charts, summaries)

**2. Diet Monitoring**

* Log daily meals and snacks
* Search food database (name, brand, macros)
* Track calories and macronutrients (carbs, proteins, fats)
* Monitor water intake

**3. Health Insights**

* Generate weekly/monthly reports
* Suggest personalized recommendations (e.g., increase protein intake, reduce sugar)
* Set and track health goals (weight loss, muscle gain, maintenance)
* Integrate with fitness wearables (e.g., Fitbit, Apple Watch)

**4. User Management**

* User registration and login
* Store personal data (age, height, weight, activity level)
* Set personal preferences and goals
* Manage app settings and notifications

**Opportunities for Modularization**

Each core feature can be implemented as a **separate module**, enabling independent development and testing:

| **Module** | **Responsibility** |
| --- | --- |
| WorkoutModule | Add, view, plan, and analyze workouts |
| NutritionModule | Log meals, calculate calories/macros, manage food database |
| HealthInsightsModule | Generate reports, offer health recommendations |
| UserModule | Handle login, profile, preferences |
| DataStorageModule | Store and retrieve workout/diet/user data |
| AnalyticsModule | Produce graphs, summaries, and track progress |
| IntegrationModule | Optional: Sync with external APIs and wearables |

**Step-by-Step Summary**

1. **Define the broad goal** – support users in managing fitness and nutrition.
2. **Use top-down design** – break the app into main functional areas.
3. **Decompose each function** – list all sub-tasks and features under each area.
4. **Modularize** – implement each part as an independent module to ensure clean, maintainable, and scalable code.

**Problem 2: Create Online Learning Platform**

**Problem 2 Statement:**

Decompose the creation of an online learning platform that supports course creation, user enrollment, content delivery, and progress tracking.

**Instructions:**

1. Define the main goal of the platform.
2. Use the top-down approach to break down the platform into its major features.
3. Further decompose each feature into smaller, manageable tasks.
4. Identify opportunities for modularization.

**Main Goal of the Platform**

To build an online learning platform that allows:

* Instructors to **create and manage courses**
* Students to **enroll in courses and access content**
* The system to **deliver content**, **track progress**, and **support learning**

**Top-Down Approach – Step 1: Identify Major Features**

Divide the platform into key functional components:

1. **Course Management**
2. **User Enrollment & Authentication**
3. **Content Delivery**
4. **Progress Tracking**
5. **Communication & Interaction (optional)**
6. **Admin Panel (optional)**

**Top-Down Approach – Step 2: Decompose Each Feature**

**1. Course Management (For Instructors)**

* Create/edit/delete courses
* Add modules, lessons, and quizzes
* Upload files (PDFs, videos, slides)
* Set course requirements and schedules

**2. User Enrollment & Authentication**

* User registration (students & instructors)
* Login/logout and role-based access
* Enroll or unenroll from courses
* View enrolled courses dashboard

**3. Content Delivery**

* Stream video lessons
* Display text materials and resources
* Provide download links for files
* Structure content by modules or sections

**4. Progress Tracking**

* Track completed lessons/modules
* Display course progress bar
* Show quiz results or scores
* Provide certificates (optional)

**5. Communication & Interaction (Optional)**

* Discussion forums per course
* Live Q&A or chat functionality
* Instructor feedback on assignments

**6. Admin Panel (Optional)**

* Manage users and permissions
* Monitor course statistics and activity
* Moderate content and discussions

**Opportunities for Modularization**

Each major feature can be built as a **standalone module**, allowing developers to work independently and reuse components:

| **Module** | **Responsibility** |
| --- | --- |
| CourseModule | Handles course creation, structure, content upload |
| UserModule | Registration, authentication, and user roles |
| EnrollmentModule | Course enrollment management |
| ContentDeliveryModule | Streaming, displaying, and organizing course materials |
| ProgressModule | Tracks student progress and quiz performance |
| CommunicationModule | Forums, messaging, and feedback systems (optional) |
| AdminModule | Platform-wide settings, user/course moderation (optional) |
| StorageModule | Handles file uploads and downloads |

**Step-by-Step Summary**

1. **Define the main goal** – support online teaching and learning with course management, content access, and tracking.
2. **Top-down design** – split the platform into broad feature areas.
3. **Decompose features** – list out detailed tasks for each area (e.g., lessons under courses, registration under user management).
4. **Modularize** – turn each logical area into a module for cleaner architecture, scalability, and parallel development.