**1. Java Fundamentals**

**Problem**: Create a CLI-based application that captures and displays **Missed Call Alert** details.  
**Features**:

* Accept user input for caller number, time of call.
* Store up to the last 10 missed calls.
* Display missed calls when user requests.

**2. OOPS / Inheritance**

**Problem**: Implement a VAS service hierarchy for **Caller Tunes** and **Voice Mail Services**.  
**Features**:

* Create a base class VASService with properties like serviceName, activationDate.
* Derive CallerTuneService and VoiceMailService classes.
* Override a method activateService() for each service.

**3. Abstraction / Packages / Exception Handling**

**Problem**: Design an application for **SMS Subscription Services** using interfaces and proper exception handling.  
**Features**:

* Use an interface Subscribable with methods subscribe() and unsubscribe().
* Implement it in SMSPack, DataPack classes.
* Throw custom exceptions like InvalidPackException.

**4. Java 8 Features**

**Problem**: Analyze usage logs of different VAS services using **Streams and Lambdas**.  
**Features**:

* Create a list of service usage entries (service name, user ID, usage time).
* Use streams to filter logs of a particular service or sort by usage time.
* Use Optional to handle missing data.

**5. JUnit / Code Coverage**

**Problem**: Write JUnit test cases for a **VAS Billing System** that calculates charges based on service type and usage.  
**Features**:

* Create BillCalculator class with methods like calculateCharge(serviceType, usageTime).
* Write unit tests for normal and edge cases.
* Use code coverage tools like **JaCoCo** to verify coverage.

**6. Collections**

**Problem**: Build a **VAS Usage Tracker** that stores usage details per user using Maps and Lists.  
**Features**:

* Use Map<String, List<ServiceUsage>> where key = userID.
* Track daily usage of different services like MissedCallAlert, CallerTune.
* Support features like total usage summary per user.

**7. RDBMS / SQL / JDBC**

**Problem**: Build a JDBC application to store and retrieve **VAS Subscription Details** in a relational database.  
**Features**:

* Table: subscriptions (id, userId, serviceName, status, activationDate)
* Implement CRUD operations using JDBC.
* Use parameterized queries to prevent SQL injection.

**8. Combined (Capstone Style - Advanced)**

**Problem**: Create a **VAS Management Dashboard** with:

* **Java Fundamentals** for CLI
* **OOPs** for services hierarchy
* **Abstraction/Exception Handling**
* **Java 8** streams for reporting
* **Collections** to manage in-memory cache
* **JDBC** for persistent storage
* **JUnit** for test coverage

**Scenario**: Admin can:

* Add/Remove services
* View active subscriptions
* Generate usage reports
* Backup to DB