**📘 Problem Statement: *VAS Telecom Billing System Using Java 8 Features***

**🧩 Context:**

A telecom company offers **Voice**, **SMS**, and **Data** services to its users as part of Value-Added Services (VAS). Customers are billed monthly based on usage. The system must:

* Process daily usage
* Apply correct billing
* Generate detailed bill
* Use **Java 8 functional style** wherever possible

**✅ Requirements:**

**1. Customer and Service Modeling**

* A Customer has:
  + customerId, name, phoneNumber
  + a list of VASUsageRecord (one per service per day)
* A VASUsageRecord includes:
  + usageType (VOICE, SMS, DATA)
  + date, unitsUsed (mins, count, MB)

**2. VAS Interface with Default Billing**

Use a **Java 8 interface** for VASService with default billing logic:

Implement 3 concrete classes: VoiceService, SMSService, DataService.

**3. Billing Engine**

* Take a list of VASUsageRecord for a customer
* Use **Java 8 Stream API** to:
  + Group by usageType
  + Calculate total charge for each service
  + Use **method references** where possible

**4. Report Generator**

* Use **StringJoiner** to prepare a bill summary:

Monthly Bill for Avinash (9999988888)

-------------------------------------

VOICE : ₹25.0

SMS : ₹10.0

DATA : ₹150.0

-------------------------------------

Total : ₹185.0

**5. Optional Handling**

* If a customer has **no usage** for the month, use Optional to handle and print:

No usage records found for customer Avinash.

**6. Manual Test Requirements**

* At least 3 customers with varying usage patterns
* Use Java 8 forEach and lambdas to loop through and generate bills

**🎯 Java 8 Features to Be Used**

| **Feature** | **Where to Use** |
| --- | --- |
| **Streams** | Billing summary, grouping by service |
| **Lambda** | Custom operations during billing |
| **Method Reference** | e.g., System.out::println or billing methods |
| **Optional** | Checking presence of usage data |
| **forEach** | Iterating customer list and usage |
| **StringJoiner** | Formatting bill output |
| **Default Methods** | Interface billing logic |

**🧪 Example Scenario:**

Customer c1 = new Customer("1001", "Avinash", "9999988888");

c1.addUsage(new VASUsageRecord(VOICE, "2025-07-01", 10));

c1.addUsage(new VASUsageRecord(SMS, "2025-07-01", 5));

c1.addUsage(new VASUsageRecord(DATA, "2025-07-01", 300)); // in MB

Expected Billing Output:

Monthly Bill for Avinash (9999988888)

-------------------------------------

VOICE : ₹20.0

SMS : ₹5.0

DATA : ₹150.0

-------------------------------------

Total : ₹175.0

**🏁 Deliverables**

1. **Customer**, **VASService**, **VASUsageRecord** classes
2. Concrete services: VoiceService, SMSService, DataService
3. BillingEngine using Java 8 Streams
4. Main.java or test class that adds data and prints bills

**🧪 Manual Test Case Sheet**

| **Test Case ID** | **Customer Name** | **VOICE (min)** | **SMS (count)** | **DATA (MB)** | **Expected Output Summary** |
| --- | --- | --- | --- | --- | --- |
| TC\_001 | Avinash | 10 | 5 | 300 | VOICE: ₹20, SMS: ₹5, DATA: ₹150, Total: ₹175 |
| TC\_002 | Priya | 2 | 0 | 100 | VOICE: ₹4, DATA: ₹50, Total: ₹54 |
| TC\_003 | Ravi | 0 | 0 | 0 | No usage records found |