**a. Explanation of Scenario and Designed System Using a Diagram**

**Scenario: In this system, we are creating a virtual cafe where customers can order either coffee or juice. The system is designed using OSGi (Open Service Gateway Initiative), where different functionalities are separated into distinct bundles that interact through service interfaces.**

**The main components of the system are:**

1. **Coffee Service – A service that brews coffee.**
2. **Juice Service – A service that prepares juice.**
3. **Waiter – The mediator that takes customer orders and forwards the information to the appropriate services (coffee or juice).**
4. **Order Screens – Separate screens to display coffee and juice orders.**

**Explanation:**

* **Producer Bundles:** These are CoffeeBrewer and JuiceMaker, responsible for preparing coffee and juice respectively.
* **Consumer Bundles:** These are Waiter, which interacts with customers, and CoffeeOrderScreen and JuiceOrderScreen that handle the display of orders.
* The **Waiter** interacts with **CoffeeOrderScreen** or **JuiceOrderScreen** to display or announce the order based on the customer's choice.

**b. Manifest Implementation (Exported Services / Imported Services)**

For each OSGi bundle, we need to define the **Exported Services** (services that a bundle offers to others) and **Imported Services** (services that a bundle uses from other bundles).

1. **CoffeeBrewer (Producer) Manifest:**
   * Exported Services: com.cafe.api.CoffeeService (service that provides coffee making functionality).
   * Imported Services: com.cafe.api.CoffeeOrderListener (from Waiter bundle, to notify the order completion).

**Manifest (CoffeeBrewer):**

plaintext

CopyEdit

Bundle-Name: CoffeeBrewer

Bundle-SymbolicName: com.cafe.coffeebrewer

Bundle-Version: 1.0.0

Export-Package: com.cafe.coffeebrewer;version="1.0.0"

Import-Package: com.cafe.api;version="1.0.0"

1. **JuiceMaker (Producer) Manifest:**
   * Exported Services: com.cafe.api.JuiceService (service that provides juice making functionality).
   * Imported Services: com.cafe.api.JuiceOrderListener (from Waiter bundle, to notify the order completion).

**Manifest (JuiceMaker):**

plaintext

CopyEdit

Bundle-Name: JuiceMaker

Bundle-SymbolicName: com.cafe.juicemaker

Bundle-Version: 1.0.0

Export-Package: com.cafe.juicemaker;version="1.0.0"

Import-Package: com.cafe.api;version="1.0.0"

1. **Waiter (Consumer) Manifest:**
   * Imported Services: com.cafe.api.CoffeeService (to call the coffee making service).
   * Imported Services: com.cafe.api.JuiceService (to call the juice making service).
   * Exported Services: com.cafe.api.CoffeeOrderListener and com.cafe.api.JuiceOrderListener (used by order screens to display orders).

**Manifest (Waiter):**

plaintext

CopyEdit

Bundle-Name: Waiter

Bundle-SymbolicName: com.cafe.waiter

Bundle-Version: 1.0.0

Export-Package: com.cafe.waiter;version="1.0.0"

Import-Package: com.cafe.api;version="1.0.0"

1. **CoffeeOrderScreen (Consumer) Manifest:**
   * Imported Services: com.cafe.api.CoffeeOrderListener (to receive coffee orders).
   * Exported Services: None.

**Manifest (CoffeeOrderScreen):**

plaintext

CopyEdit

Bundle-Name: CoffeeOrderScreen

Bundle-SymbolicName: com.cafe.coffeeorderscreen

Bundle-Version: 1.0.0

Import-Package: com.cafe.api;version="1.0.0"

1. **JuiceOrderScreen (Consumer) Manifest:**
   * Imported Services: com.cafe.api.JuiceOrderListener (to receive juice orders).
   * Exported Services: None.

**Manifest (JuiceOrderScreen):**

plaintext

CopyEdit

Bundle-Name: JuiceOrderScreen

Bundle-SymbolicName: com.cafe.juiceorderscreen

Bundle-Version: 1.0.0

Import-Package: com.cafe.api;version="1.0.0"

**c. Commands Required to Install and Run Bundles in OSGi Framework**

1. **Installing Bundles:**
   * First, ensure you have all the bundles (JAR files) in the proper directory.
   * To install the **CoffeeBrewer** bundle:

plaintext

CopyEdit

g! install file:/path/to/CoffeeBrewer.jar

* + To install the **JuiceMaker** bundle:

plaintext

CopyEdit

g! install file:/path/to/JuiceMaker.jar

* + To install the **Waiter** bundle:

plaintext

CopyEdit

g! install file:/path/to/Waiter.jar

* + To install the **CoffeeOrderScreen** bundle:

plaintext

CopyEdit

g! install file:/path/to/CoffeeOrderScreen.jar

* + To install the **JuiceOrderScreen** bundle:

plaintext

CopyEdit

g! install file:/path/to/JuiceOrderScreen.jar

1. **Starting Bundles:**
   * After installation, you can start the bundles in the following order:

**Start the producer bundles first** (they provide the services that the consumers depend on):

* + Start the **CoffeeBrewer** bundle:

plaintext

CopyEdit

g! start 2

* + Start the **JuiceMaker** bundle:

plaintext

CopyEdit

g! start 4

**Then, start the consumer bundles** (these will depend on the services provided by the producers):

* + Start the **Waiter** bundle:

plaintext

CopyEdit

g! start 3

* + Start the **CoffeeOrderScreen** bundle:

plaintext

CopyEdit

g! start 5

* + Start the **JuiceOrderScreen** bundle:

plaintext

CopyEdit

g! start 6

1. **Stopping Bundles:**
   * To stop the bundles:

plaintext

CopyEdit

g! stop <bundle-id>

Replace <bundle-id> with the corresponding ID for each bundle.