**Competency Question G1-D1**

*"What layout systems maintain consistency across both portrait and landscape orientations for in-vehicle interfaces?"*

**1. Ontology Modeling**

**Core Classes & Subclasses**

| **Class** | **Hierarchy** | **Purpose** |
| --- | --- | --- |
| LayoutSystem | UIComponent ➔ LayoutSystem | Models UI layout systems (e.g., grid, flex). |
| InVehicleContext | Context ➔ EnvironmentalContext ➔ InVehicleContext | Represents in-vehicle interfaces (e.g., car dashboards). |

**Data Properties**

| **Property** | **Domain** | **Range** | **Purpose** |
| --- | --- | --- | --- |
| supportsOrientation | LayoutSystem | xsd:string | Specifies supported screen orientations ("Portrait", "Landscape"). |
| maintainsConsistencyIn | LayoutSystem | xsd:boolean | Indicates if the layout adapts seamlessly to orientation changes. |

**Object Property**

| **Property** | **Domain** | **Range** | **Purpose** |
| --- | --- | --- | --- |
| appliesToContext | LayoutSystem | InVehicleContext | Links layout systems to in-vehicle contexts. |

**2. Example Instances**

turtle

Copy

Download

# Layout systems

:FlexLayout a :LayoutSystem ;

:supportsOrientation "Portrait", "Landscape" ;

:maintainsConsistencyIn true ;

:appliesToContext :CarDashboardContext .

:GridLayout a :LayoutSystem ;

:supportsOrientation "Portrait" ;

:maintainsConsistencyIn false ;

:appliesToContext :CarDashboardContext .

# In-vehicle context

:CarDashboardContext a :InVehicleContext ;

rdfs:comment "Interfaces in car dashboards requiring orientation consistency." .

**3. SPARQL Query**

sparql

Copy

Download

PREFIX : <http://www.semanticweb.org/asifcomputer/ontology/2025/ui\_ux\_automation#>

SELECT ?layout

WHERE {

# Get layout systems for in-vehicle contexts

?layout a :LayoutSystem ;

:appliesToContext :CarDashboardContext ;

:supportsOrientation "Portrait" ;

:supportsOrientation "Landscape" ;

:maintainsConsistencyIn true .

}

**Result**:

| **layout** |
| --- |
| :FlexLayout |

**4. Step-by-Step Validation**

**Step 1: Check Layout Orientations**

sparql

Copy

Download

SELECT ?layout ?orientation

WHERE {

?layout a :LayoutSystem ;

:supportsOrientation ?orientation .

}

**Result**:

| **layout** | **orientation** |
| --- | --- |
| :FlexLayout | "Portrait" |
| :FlexLayout | "Landscape" |
| :GridLayout | "Portrait" |

**Step 2: Verify Consistency**

sparql

Copy

Download

SELECT ?layout

WHERE {

?layout a :LayoutSystem ;

:maintainsConsistencyIn true .

}

**Result**:

| **layout** |
| --- |
| :FlexLayout |

**Step 3: Combine Conditions**

Use the final query to filter layouts that meet al