

# 02 May 2024 - SDE Training

## 1. Identify the STATE, BEHAVIOUR and IDENTITY of the following Objects



### STATE

- title
- author
- pages

### BEHAVIOUR

- read
- lend
- borrow

### IDENTITY

- This is a book



### STATE

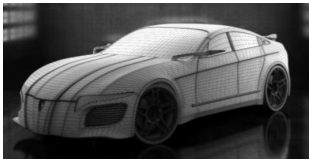
- name
- breed
- age

### BEHAVIOUR

- bark
- eat
- sleep

### IDENTITY

- This is Tom



### STATE

- make
- model
- year
- color
- engine type
- mileage
- price

### BEHAVIOUR

- start
- drive
- stop
- park

### IDENTITY

- This is a car



### STATE

- wingspan
- weight
- age
- color
- species

### BEHAVIOUR

- fly
- hunt
- sleep

### IDENTITY

- This is an eagle

---

2. Realize the concept of class in OOPs using an example of a class called **Sportsman** and explain what states and behaviors it might include to represent athletes from various sports.

STATE	BEHAVIOUR	IDENTITY
<ul style="list-style-type: none"><li>• name</li><li>• age</li><li>• height</li><li>• weight</li><li>• rank</li><li>• sport</li><li>• country</li><li>• medals</li><li>• records</li></ul>	<ul style="list-style-type: none"><li>• Profiling</li><li>• Training</li><li>• Compete</li><li>• BMI</li></ul>	<ul style="list-style-type: none"><li>• This is Cristiano Ronaldo</li></ul>

---

3. Can you explain how a letter, envelope, or courier service demonstrates the concept of encapsulation in ObjectOriented Programming (OOP)?

To illustrate how encapsulation works using the analogy of a letter, envelope, or courier service, let's consider a simplified scenario:

- **Letter:** Represents the data or state of an object. For example, a letter might have attributes like sender, recipient, message, and timestamp.
- **Envelope:** Acts as a class that encapsulates the letter. The envelope has methods that allow interaction with the letter, such as sealing the envelope (which might involve setting the sender and recipient information), opening the envelope (which might reveal the message), and tracking the delivery status (which might involve updating the timestamp).
- **Courier Service:** Represents the external system or user interacting with the envelope. The courier service uses the envelope's methods to send the letter, but it doesn't directly

manipulate the letter's attributes. Instead, it interacts with the envelope, which controls access to the letter's data.

In this analogy:

- The **letter** is the data or state of the object.
- The **envelope** is the class that encapsulates the letter, providing methods (like sealing, opening, and tracking) to interact with the letter's data.
- The **courier service** is the external entity that interacts with the envelope, using its methods to perform actions on the letter without directly accessing its data.

This example demonstrates encapsulation by showing how the envelope (class) acts as a protective barrier around the letter (data), controlling how the letter's data can be accessed and modified.

---

#### 4. You are about to board an airplane. What's your general view of the plane? How do you observe the concept of abstraction of the following individuals considering their distinct perspectives on airplanes?

- Passengers,
- Pilots,
- Aerospace engineers,
- Design engineers

#### Answer:

- **Passengers:** Passengers see the airplane as a mode of transportation that takes them from one place to another. They are concerned with the comfort, safety, and amenities provided during the flight. They may not be aware of the technical details of the airplane's design and operation.
- **Pilots:** Pilots view the airplane as a complex machine that they operate to fly from one location to another. They are trained to understand the aircraft's systems, controls, and instruments to ensure safe and efficient flight operations. They focus on navigation, communication, and handling emergency situations.
- **Aerospace Engineers:** Aerospace engineers see the airplane as a product of their design and innovation. They are involved in the research, development, and testing of aircraft components

and systems. They focus on aerodynamics, materials, propulsion, and structural design to optimize the airplane's performance and efficiency.

- **Design Engineers:** Design engineers see the airplane as a combination of form and function. They are responsible for creating the visual and structural design of the aircraft, considering factors such as aesthetics, ergonomics, and manufacturability. They work closely with aerospace engineers to translate conceptual designs into detailed specifications for production.
  - **Abstraction:** Abstraction is the process of simplifying complex systems by focusing on the essential aspects while ignoring unnecessary details. In the context of airplanes, passengers, pilots, aerospace engineers, and design engineers each have their unique perspective on the aircraft, emphasizing different aspects based on their roles and expertise. Abstraction allows them to understand and interact with the airplane at their respective levels of detail without being overwhelmed by the complexity of the entire system.
- 

## 5. Identify and define Encapsulation and Abstraction in the following pictures.



### Encapsulation

Encapsulation is the bundling of data (state) and methods (behavior) that operate on the data into a single unit (class) while restricting access to some of the object's components. In the context of a TV, encapsulation can be seen in the internal components and circuits that are



### Abstraction

Abstraction is the process of simplifying complex systems by focusing on the essential aspects while ignoring unnecessary details. In the context of a TV, abstraction can be seen in the user interface and remote control, which provide users with a simplified way to interact with the TV without needing to understand the

enclosed within the TV casing, protecting them from external interference.

intricate details of its internal components and operations.

---

## 6. Realize the concept of inheritance in object-oriented programming (OOP) by considering different versions of ChatGPT.

- **ChatGPT v1:** Basic chatbot with text-based interactions.
  - **ChatGPT v2:** Features of ChatGPT v1 + Chatbot with voice recognition and synthesis capabilities.
  - **ChatGPT v3:** Features of ChatGPT v2 + Chatbot with natural language understanding and generation.
  - **ChatGPT v4:** Features of ChatGPT v3 + Chatbot with emotional intelligence and personality customization.
  - **ChatGPT v5:** Features of ChatGPT v4 + Chatbot with multi-modal capabilities (text, voice, images, videos).
  - **ChatGPT v6:** Features of ChatGPT v5 + Chatbot with advanced context awareness and memory.
  - **ChatGPT v7:** Features of ChatGPT v6 + Chatbot with self-learning and adaptation capabilities.
  - **ChatGPT v8:** Features of ChatGPT v7 + Chatbot with human-level conversational skills and empathy.
- 

## 7. Explain how the concept of a pen adapting its behavior based on the surface it writes on relates to the concept of polymorphism in Object-Oriented Programming (OOP)?

- **Pen:** A pen is a class that has a behavior (write) which can adapt based on the surface it writes on. For example, a pen can write smoothly on paper, but it may require more pressure on a rough surface like cardboard.
- **Polymorphism:** Polymorphism is the ability of an object to take on multiple forms or behaviors based on the context in which it is used. In the context of OOP, polymorphism allows objects of different classes to be treated as objects of a common superclass, enabling them to exhibit different behaviors based on their specific implementations.

- **Relation:** The behavior of a pen adapting to different writing surfaces exemplifies the concept of polymorphism in OOP. The pen object can exhibit different writing behaviors based on the surface it interacts with, demonstrating the flexibility and adaptability of polymorphism in handling diverse scenarios.
- 

## 8. Realize Association and its types

- Association using book and author
- Aggregation using wallet and money
- Composition using drummer and drums

### Answer:

- **Association:** Book and author represent association, where a book class is associated with an author class. Association is a relationship between two classes that describes how they are connected or interact with each other. In this case, a book can have an author, and an author can write multiple books.
  - **Aggregation:** Wallet and money demonstrate aggregation, where the wallet class contains money objects. Aggregation is a type of association where one class (wallet) contains another class (money) as a part of its state. The money objects can exist independently of the wallet and can be shared among multiple wallets.
  - **Composition:** Drummer and drums illustrate composition, where the drummer class is composed of drum objects. Composition is a stronger form of aggregation where the composed objects (drums) are part of the whole (drummer) and cannot exist independently. The drums are created and destroyed along with the drummer object.
-