Class Exercise:

OOP v3

You are tasked with creating a ticket booking system using object-oriented programming (OOP) principles in Python. The system should allow users to book tickets for various events, display event details, manage ticket availability, and calculate total ticket prices. Implement inheritance and composition to create specialized classes for different types of events and ticket categories.

In this exercise, we have defined three classes: Event, **TicketCategory**, and **Ticket**. The **Event** class represents an event with attributes for its name, venue, date, and time. It also maintains a list of available ticket categories. The **TicketCategory** class represents a category of tickets with attributes for its name and price. The **Ticket** class represents a ticket for a specific event and ticket category.

The **TicketBookingSystem** class is responsible for managing events, ticket categories, and ticket bookings. It has methods to add events, display events, book tickets, display tickets, and calculate the total price of all booked tickets.

In the example usage section, we create a **TicketBookingSystem** object and add two events with their ticket categories. We display the event list, book tickets for different events and ticket categories, display the ticket list, and calculate the total price of all booked tickets.

class Event:

    def \_\_init\_\_(self, name, venue, date, time):

# Your code here

    def \_\_str\_\_(self):

# Your code here

    def add\_ticket\_category(self, ticket\_category):

# Your code here

    def display\_ticket\_categories(self):

# Your code here

class TicketCategory:

# Your code here

class Ticket:

# Your code here

    def \_\_str\_\_(self):

# Your code here

class TicketBookingSystem:

    def \_\_init\_\_(self):

# Your code here

    def add\_event(self, event):

# Your code here

    def display\_events(self):

# Your code here

    def book\_ticket(self, event, ticket\_category):

# Your code here

    def display\_tickets(self):

# Your code here

    def calculate\_total\_price(self):

# Your code here

# Example usage

booking\_system = TicketBookingSystem()

event1 = Event("Music Concert", "Stadium A", "2023-07-15", "20:00")

event2 = Event("Theater Play", "Theater B", "2023-07-20", "19:30")

category1 = TicketCategory("General", 50)

category2 = TicketCategory("VIP", 100)

event1.add\_ticket\_category(category1)

event2.add\_ticket\_category(category1)

event2.add\_ticket\_category(category2)

booking\_system.add\_event(event1)

booking\_system.add\_event(event2)

booking\_system.display\_events()

booking\_system.book\_ticket(event1, category1)

booking\_system.book\_ticket(event2, category1)

booking\_system.book\_ticket(event2, category2)

booking\_system.display\_tickets()

total\_price = booking\_system.calculate\_total\_price()

print(f"Total Price: {total\_price}")

Expected output:

Event List:

Event: Music Concert, Venue: Stadium A, Date: 2023-07-15, Time: 20:00

Event: Theater Play, Venue: Theater B, Date: 2023-07-20, Time: 19:30

Ticket booked successfully.

Ticket booked successfully.

Ticket booked successfully.

Ticket List:

Event: Music Concert, Ticket Category: General, Price: 50

Event: Theater Play, Ticket Category: General, Price: 50

Event: Theater Play, Ticket Category: VIP, Price: 100

Total Price: 200