

Write a Java program to create a class called "Event" with attributes for event name, date, and location. Create subclasses "Seminar" and "MusicalPerformance" that add specific attributes like number of speakers for seminars and performer list for concerts. Implement methods to display event details and check for conflicts in the event schedule.

Sample Solution:

Java Code:

Event.java

```
import java.util.Date;

// Define the Event class
class Event {
    String eventName;
    Date date;
    String location;

    public Event(String eventName, Date date, String location) {
        this.eventName = eventName;
        this.date = date;
        this.location = location;
    }

    public void displayDetails() {
        System.out.println("Event Name: " + eventName);
        System.out.println("Date: " + date);
        System.out.println("Location: " + location);
    }

    public boolean isConflict(Event otherEvent) {
        return this.date.equals(otherEvent.date) && this.location.equals(otherEvent.location);
    }
}
```

Explanation:

- Import statement: Imports the Date class from java.util.
- Class definition: Defines the Event class.

- Attributes: Declares three attributes: eventName (String), date (Date), and location (String).
- Constructor: Initializes the eventName, date, and location attributes.
- displayDetails() method: Prints the event's name, date, and location.
- isConflict() method: Checks if the current event conflicts with another event by comparing their dates and locations.

Seminar.java

```
import java.util.Date;
import java.util.List;
// Define the Seminar class that extends Event
class Seminar extends Event {
    int numberOfSpeakers;

    public Seminar(String eventName, Date date, String location, int number
        super(eventName, date, location);
        this.numberOfSpeakers = numberOfSpeakers;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Number of Speakers: " + numberOfSpeakers);
    }
}
```

Explanation:

- Import statements: Imports the 'Date' and 'List' classes from java.util.
- Class definition: Defines the Seminar class, which extends the Event class.
- Additional attribute: Declares 'numberOfSpeakers' (int) specific to the Seminar class.
- Constructor: Initializes eventName, date, location (inherited from Event), and numberOfSpeakers.
- Override displayDetails() method: Calls the displayDetails() method of the Event class and adds the number of speakers to the output.

MusicalPerformance.java

```

import java.util.List;
import java.util.Date;
// Define the MusicalPerformance class that extends Event
class MusicalPerformance extends Event {
    List performers;

    public MusicalPerformance(String eventName, Date date, String location) {
        super(eventName, date, location);
        this.performers = performers;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Performers: " + String.join(", ", performers));
    }
}

```

Explanation:

- Import statements: Imports the 'List' and 'Date' classes from java.util.
- Class definition: Defines the MusicalPerformance class, which extends the Event class.
- Additional attribute: Declares performers (List of Strings) specific to the MusicalPerformance class.
- Constructor: Initializes eventName, date, location (inherited from Event), and performers.
- Override displayDetails() method: Calls the displayDetails() method of the Event class and adds the list of performers to the output, formatted as a comma-separated string using String.join().

Main.java

```

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

// Main class to test the Event, Seminar, and MusicalPerformance classes
public class Main {
    public static void main(String[] args) {

        // Create different dates to avoid conflict
        // Create a date for testing
    }
}

```

```

Date date = new Date();

// Create an instance of Seminar
Seminar seminar = new Seminar("Space Conference", date, "Convention");

// Create an instance of MusicalPerformance
List performers = new ArrayList<>();
performers.add("Band A");
performers.add("Band B");
MusicalPerformance concert = new MusicalPerformance("Winter Fest",

// Display details of the seminar
System.out.println("Seminar Details:");
seminar.displayDetails();

// Display details of the musical performance
System.out.println("\nMusical Performance Details:");
concert.displayDetails();

// Check for scheduling conflict
if (seminar.isConflict(concert)) {
    System.out.println("\nConflict detected: Both events are scheduled at the same time.");
} else {
    System.out.println("\nNo conflict: Events are scheduled at different times.");
}
}
}

```

Explanation:

- Import statements: Import the 'ArrayList', 'Date', and 'List' classes from java.util.
- Main class definition: Defines the Main class to test the Event, Seminar, and 'MusicalPerformance' classes.
- main method: Contains the main logic for testing.
 - Import statements: Import the 'ArrayList', 'Date', and 'List' classes from java.util.
 - Create a date instance: Initializes a Date object for testing.
 - Create an instance of Seminar: Instantiates a Seminar object with specific attributes.
 - Create an instance of MusicalPerformance:

- Import statements: Import the 'ArrayList', 'Date', and 'List' classes from java.util.
 - Initializes a list of performers.
 - Instantiates a MusicalPerformance object with specific attributes.
- Display details of the seminar: Calls the 'displayDetails()' method on the seminar object.
- Display details of the musical performance: Calls the 'displayDetails()' method on the concert object.
- Check for scheduling conflict: Uses the isConflict() method to determine if the seminar and concert are scheduled at the same time and location, and prints an appropriate message based on the result.

Output:

```
Seminar Details:
Event Name: Space Conference
Date: Sat Jun 08 08:05:33 IST 2024
Location: Convention Center
Number of Speakers: 5

Musical Performance Details:
Event Name: Winter Fest
Date: Sat Jun 08 08:05:33 IST 2024
Location: Convention Center
Performers: Band A, Band B
Conflict detected: Both events are scheduled at the same time and location.
```

Java Code Editor: