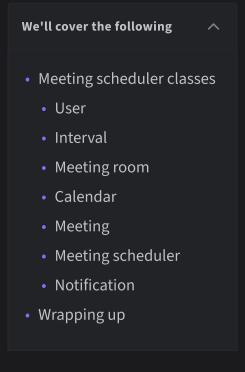
Code for the Meeting Scheduler

Write the object-oriented code to implement the design of the meeting scheduler problem.



problem using various UML diagrams. Let's explore the more practical side of things, where we will work on implementing the meeting scheduler using multiple languages. This is usually the last step in an objectoriented design interview process. We have chosen the following languages to write the skeleton code of the different classes present in the meeting scheduler:

We've reviewed the different aspects of the meeting scheduler and observed the attributes attached to the

Java • C#

- Python
- C++
- JavaScript
- Meeting scheduler classes

Note: For simplicity, we are not defining getter and setter functions. The reader can assume that all

modified only through their public method functions.

class attributes are private and accessed through their respective public getter methods and

In this section, we will provide the skeleton code of the classes designed in the class diagram lesson.

User The User class refers to a participant taking part in a meeting. A user can either accept or reject an

invitation. The definition of this class is given below:

1 public class User {

private String name;

private Date endTime;

private String email; public void respondInvitation(Notification invite); 4

```
The User class
Interval
The Interval class denotes the meeting interval (the start and end time).
```

1 public class Interval { private Date startTime;

```
The Interval class
Meeting room
```

```
The MeetingRoom class
The Calendar class contains a list of meetings to keep track of all the scheduled meetings. The definition of
```

1 public class MeetingScheduler { private User organizer; private Calendar calendar; private List<MeetingRoom> rooms;

The Notification class is responsible for sending notifications to users about any new meetings or cancelations. The definition of this class is provided below: 1 public class Notification { private int notificationId; 3 private string content;

The MeetingScheduler class

← Back Activity Diagram for the Meeting Scheduler

Notification

private Date creationDate;

public boolean sendNotification(User user); public boolean cancelNotification(User user);

The Notification class Wrapping up We've explored the complete design of the meeting scheduler in this chapter. We've looked at how the meeting scheduler can be visualized using various UML diagrams and designed using object-oriented principles and design patterns.

Complete

Getting Ready: Movie Ticket Booking System

Next \rightarrow

