```
import java.util.Optional;

class Address {
    private String city;
    private String state;

    public Address(String city, String state) {
        this.city = city;
        this.state = state;
    }

    public String getCity() {
        return city;
    }

    public String getState() {
        return state;
    }
}

class Employee {
```

```
    private String name;
    private String primaryPhone;
    private Optional<String> alternatePhone;
 // Optional for alternate phone number
    private Optional<Address> address;
// Optional for address

    public Employee(String name, String
primaryPhone, String alternatePhone,
Address address) {
        this.name = name;
        this.primaryPhone = primaryPhone;
        this.alternatePhone =
Optional.ofNullable(alternatePhone); //
Might be null
        this.address =
Optional.ofNullable(address);              //
Might be null
    }

    public Optional<String>
```

```
getAlternatePhone() {
        return alternatePhone;
    }

    public Optional<Address> getAddress() {
        return address;
    }
}

public class Main {
    public static void main(String[] args) {
        // Create an employee with no
alternate phone and no address
        Employee employee1 = new
Employee("John Doe", "1234567890", null,
null);

        // Create an employee with an
alternate phone and address
        Address address2 = new
Address("New York", "NY");
```

```
        Employee employee2 = new
Employee("Jane Smith", "9876543210",
"5551234567", address2);

        // Check if employee1 has an
alternate phone number

employee1.getAlternatePhone().ifPresent(p
hone ->
            System.out.println("Employee1 has
an alternate phone: " + phone)
        );

        // Check if employee2 has an
alternate phone number

employee2.getAlternatePhone().ifPresent(p
hone ->
            System.out.println("Employee2 has
an alternate phone: " + phone)
        );
```

```
        // Using map() to retrieve the
employee's city (address might be null)
        String city1 = employee1.getAddress()
                            .map(Address::getCity)
                            .orElse("Address not
available");
        System.out.println("Employee1 city: "
+ city1);

        // Using flatMap() to retrieve the
employee's state (address might be null)
        String state2 =
employee2.getAddress()
                            .flatMap(address ->
Optional.ofNullable(address.getState()))
                            .orElse("State not
available");
        System.out.println("Employee2 state: "
+ state2);
    }
```

```
}
```