Quiz# 2

Q1: For ages, your company used a greeting app to greet its employees. Recently, it realized that it is too formal to greet everyone like this. Some employees should be greeted informally. It decided to update the following code to make room for an `InformalGreetingService` (which greets as "Hi, Faridi") and renamed the existing `GreetingService` as `FormalGreetingService`. As a good software engineering practice, it decided to **use abstractions** to lower the coupling. Moreover, it realized that **dependencies should be injected** for low coupling. Help your company and update the following code.

```java
public class GreetingService {

    public String greet(String name) {

        return "Hello, Mr./Ms. " + name;

    }

}
```

```java
public abstract class GreetingService { 3

    public abstract String greet(String name);

}


public class FormalGreetingService extends GreetingService { 2

    @Override

    public String greet(String name) {

        return "Hello, Mr./Ms. " + name;

    }

}


public class InformalGreetingService extends GreetingService { 3

    @Override

    public String greet(String name) {
```

```java
public class GreetingApp {

    private final GreetingService greetingService;

    public GreetingApp() {

        this.greetingService = new GreetingService();

    }



    public void greetUser(String name) {

        String greeting = greetingService.greet(name);

        System.out.println(greeting);

    }


    public static void main(String[] args) {

        GreetingApp greetingApp = new GreetingApp();

        greetingApp.greetUser("Jamshed");

        greetingApp.greetUser("Faridi"); // TODO should be greeted

informally.

    }

}
```

```java
        return "Hi, " + name;

    }

}


public class GreetingApp {

    private final GreetingService greetingService; 1


    public GreetingApp(GreetingService greetingService) { 3

        this.greetingService = greetingService;

    }


    public void greetUser(String name) {

        String greeting = greetingService.greet(name);

        System.out.println(greeting);

    }


    public static void main(String[] args) {

        // Inject different greeting services

        GreetingService formalGreetingService = new

FormalGreetingService(); 1/2

        GreetingService informalGreetingService = new

InformalGreetingService(); 1/2
```

```
        GreetingApp formalGreetingApp = new
GreetingApp(formalGreetingService); 1/2
        GreetingApp informalGreetingApp = new
GreetingApp(informalGreetingService); 1/2

        formalGreetingApp.greetUser("Jamshed"); 1/2
        informalGreetingApp.greetUser("Faridi"); 1/2
    }
}
```