

NAME:PRUTHVI RAJ S M

USN:INT22CS149

FULL STACK DEVELOPMENT

Question 4)

Dependency Injection: Design a java code for Injecting dependencies into classes, making them loosely coupled and easier to test.

CODE:

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>6.0.3</version>
  </dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>6.0.3</version>
</dependency>
<dependency>
  <groupId>javax.annotation</groupId>
  <artifactId>javax.annotation-api</artifactId>
  <version>1.3.2</version>
</dependency>
</dependencies>
```

```

public class SMSService implements MessageService {
    @Override
    public void sendMessage(String message, String recipient) {
        System.out.println("☎ SMS sent to " + recipient + " with message: " +
message);
    }
}
package lab4;

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

@Component
public class NotificationService {
    private final MessageService messageService;

    // Use @Qualifier to specify which implementation to inject
    public NotificationService(@Qualifier("smsService") MessageService
messageService) {
        this.messageService = messageService;
    }

    public void notifyUser(String message, String recipient) {
        messageService.sendMessage(message, recipient);
    }
}
package lab4;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class DependencyInjectionApplication {

    public static void main(String[] args) {
        ApplicationContext context =new
AnnotationConfigApplicationContext(AppConfig.class);

        NotificationService notificationService =
context.getBean(NotificationService.class);

        // Will use the service specified in @Qualifier
        notificationService.notifyUser("Hello, World!", "user@example.com");
    }
}

```

```

package lab4;

public interface MessageService {
    void sendMessage(String message, String recipient);
}

package lab4;

import org.springframework.stereotype.Service;

@Service("emailService")
public class EmailService implements MessageService{
    @Override
    public void sendMessage(String message, String recipient) {
        System.out.println("✉ Email sent to " + recipient + " with message: " + message);
    }
}

package lab4;

import org.springframework.stereotype.Service;

@Service("smsService") // Bean name: smsService

```

OUTPUT:

```

[INFO] -----< com.example:spring-demo >-----
[INFO] Building spring-demo 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- exec:3.6.2:java (default-cli) @ spring-demo ---
[INFO] [stdout] 📧 SMS sent to user@example.com with message: Hello, World!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.223 s
[INFO] Finished at: 2025-11-05T22:23:20+05:30
[INFO] -----
PS C:\Users\Pruthvi Raj\OneDrive\Desktop\spring-demo\spring-demo> 

```