

W1D5: Quiz

CS544 Enterprise Architecture

1. Give two things that change when @Id is moved from a field to a getter method:

1. access
2. Where it finds other annotations

2. What SQL datatype do you get when you do not add @Temporal on a java.util.Date field:

Timestamp

3. What is the output of the following application:

```
@Configuration
@ComponentScan("edu.mum.cs544")
@EnableAspectJAutoProxy
public class Config {
}
```

```
public class App {
    public static void main(String[] args) {
        ConfigurableApplicationContext context = new
            AnnotationConfigApplicationContext(Config.class);
        System.out.println("Testing Spring Startup");
        AClass a = context.getBean(AClass.class);
        a.getAText();
        a.getBText();
        context.close();
    }
}
```

```
@Component
public abstract class MySuper {
    @Value("Test")
    private String text;
    public MySuper() {
        System.out.println("MySuper Constructor - text: " + text);
    }
    public void setText(String text) {
        System.out.println("Setting text to: " + text);
        this.text = text;
    }
    public String getText() { return text; }
}
```

```
@Component
public class AClass extends MySuper {
    private BClass bClass;
    @Autowired
    public AClass(BClass b) {
        System.out.println("MyClass Constructor");
        setText("Hello");
        this.bClass = b;
    }
    public void getAText() {
        System.out.println("This is a " + getText());
    }
    public void getBText() {
        System.out.println("This is a " + bClass.getText());
    }
}
```

Name: _____ StudentID: _____

```
@Lazy
@Component
public class BClass {
    @Value("Other")
    private String text;
    public BClass() {
        System.out.println("BClass constructor");
    }
    public String getText() {
        return text;
    }
    @PreDestroy
    public void done() {
        System.out.println("BClass done!");
    }
}

@Aspect
@Component
public class AwesomeAspect {
    @Value("Test")
    private String text;
    public AwesomeAspect() {
        System.out.println("AwesomeAspect Constructor - text: " + text);
    }
    @PostConstruct
    public void start() {
        System.out.println("AwesomeAspect start method - text: " + text);
    }
    @Around("execution(* edu.mum.cs544.*.get*(..))")
    public Object beforeTrace(ProceedingJoinPoint pjp) throws Throwable {
        String name = pjp.getTarget().getClass().getSimpleName();
        if (name.equals("BClass")) {
            return "Something";
        }
        return pjp.proceed();
    }
}
```

AwesomeAspect Constructor - text: null
AwesomeAspect start method - text: Test

BClass Constructor

MySuperConstructor text: null

MyClass Constructor

Setting text to Hello

Testing Spring Startup

This is a Test

This is a something

BClass done!