# Quiz 7: Service-Oriented Architecture III

**Due** Nov 11, 2016 at 11:59pm **Points** 100 **Questions** 4
**Available** Nov 7, 2016 at 8am - Nov 11, 2016 at 11:59pm 5 days **Time Limit** 60 Minutes

This quiz was locked Nov 11, 2016 at 11:59pm.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 52 minutes | 90 out of 100 |

Score for this quiz: **90** out of 100
Submitted Nov 11, 2016 at 5:57pm
This attempt took 52 minutes.

### Question 1

**25 / 25 pts**

1. What is the form of data abstraction that is defined with constructor-oriented designs?
   1. procedural Data Abstra
2. What are the four ways that data abstraction may be defined in observer-oriented designs?
   1. Abstract Data Types via
   2. Abstract Data types via
   3. Abstract Data Types via
   4. Partially Abstract Types

**Answer 1:**

Correct! Procedural data abstraction

**Answer 2:**

You Answered Abstract Data Types via Subtypes

Correct Answer Subtypes

**Answer 3:**

You Answered — Abstract Data types via opaque Types

Correct Answer — Opaque Types

**Answer 4:**

You Answered — Abstract Data Types via Sealing

Correct Answer — Sealing

**Answer 5:**

Correct! — Partially Abstract Types

## Question 2

25 / 25 pts

Match up each concept with its description.

Correct! — **Type name for a resource, such as a part of memory or a lock, so that the compiler can analyze the use of the resource in the program.** — Region

Correct! — **The potential uses of resources in a procedure or method, reflected in the type signature for the procedure.** — Effect

Correct! — **Encapsulation of the state of a domain entity with the domain logic for that entity** — Object

Correct! — **Type name for the state of an object or service, used to determine which of the service operations may be invoked.** — Typestate

Correct! — **Encapsulation of a program in an execution context, that allows the program to use state (e.g. stateful IO) without explicitly getting a handle on the state.** — Monad

## Question 3

20 / 25 pts

1. What are the three relationships between interfaces and classes in Java (or C#)?
2. What is the type rule for subtyping of procedure (arrow) types?

Your Answer:

1. Relationship between interfaces and classes:

a. Type inheritance or subtyping relationship

b. Implementation inheritance relationship.

c. The Implementation relationship.

2. Type rule for subtyping of procedure Array types:

If T1 is a subtype of T2,

then T1[] is a subtype of T2[]

2.Subtyping rule for procedure types: (T1 -> T2) < (T3 -> T4) if T3 < T1 and T2 < T4.

## Question 4

20 / 25 pts

1. What is a covariant change in a **data** contract, i.e., what does such a change to a type make to the *set of possible values* for that type?
2. What is a contravariant change in a **data** contract, i.e., what does such a change to a type make to the *set of possible values* for that type?
3. When is it safe to make a covariant change to a type in a data contract for a service interface (i.e., when it is input data or output data)?
4. When is it safe to make a contravariant change to a type in a data contract for a service interface (i.e., when it is input data or output data)?

Your Answer:

1. Covariant Change in data contract: Compatible input data model changes

2. Contravariant Change in data contract: Compatible Output data model changes

A procedure subtyping (T1 -> T2) <= (T1' funtyop T'2) is safe if:

a) The subtype is covariant within the range.

b) The subtype is contravariant within the domain.

where: T1 and T2 are the two procedure types.

1.Covariant change: Change the type to allow more values for that type.
2.Contravariant change: Change the type to allow less values for that type. 3.Safe to make covariant changes to the input data model 4.Safe to make contravariant changes to the output data model.

Quiz Score: **90** out of 100