Summary & Highlights

Congratulations! You have completed this module. At this point, you know that:

Software architecture functions as a blueprint and represents the importance of a good architectural design.

Structured design breaks down a software problem into well-organized smaller solution elements whereas behavioral models describe the behavior of the system without explaining how the system implements the behavior.

Developing UML diagrams saves time and money by helping developers quickly get up to speed on a project, plan features in advance of coding, and navigate source code easily. Types of UML diagrams include state transition, interaction, and class diagrams.

Objects contain data, and they also have behaviors that prescribe the actions the object can take, whereas classes are blueprints for objects.

A service-oriented architecture (SOA) consists of loosely coupled services that interface with each other via a communication protocol over a network. Distributed systems run on multiple services on different machines, but they appear to the end-user as a single coherent system.

An architectural pattern is a repeatable solution to an architectural problem. Types of architectural patterns include 2-tier, 3-tier, event-driven, peer-to-peer, and microservices. Two or more patterns can be combined in a single system, but some are mutually exclusive.

Application environments include development, testing or QA, staging, and production. Production environments tend to be more complex than pre-production because they must take into account non-functional requirements like load, security, reliability, and scalability.

Application environments can be deployed either on-premises on traditional hardware, or on public, private, or hybrid cloud platforms.

Common components needed for a production environment include a firewall, a load balancer, web and application servers, proxy servers, and database servers.