Integrated Systems Technology – 3%

Domain Scope

1. Scripting languages, their uses and architectures
2. Application programming interfaces
3. Programming practices to facilitate the management, integration and security of the systems that support an organization.

Domain Competencies

A. Illustrate how to code and store characters, images and other forms of data in computers and show why data conversion is often a necessity when merging disparate computing systems together. *(Data mapping and exchange)*

B. Show how a commonly used intersystem communication protocol works, including its advantages and disadvantages. *(Intersystem communication protocols)*

C. Design, debug and test a script that includes selection, repetition and parameter passing. *(Integrative programming and scripting)*

D. Illustrate the goals of secure coding, and show how to use these goals as guideposts in dealing with preventing buffer overflow, wrapper code, and securing method access. *(Defensible integration)*

Integrated Systems Technology Subdomains

01 Perspectives and impact
(Level 1 minimal degree of engagement)
Competencies:

a. Describe how integrating various modules can produce a working system.
b. Describe how integration is an important function of all IT professionals.

02 Data mapping and exchange
(Level 2 medium degree of engagement)
Competencies:

a. Produce a definition for the term ‘metadata.’
b. Describe how ASCII, EBCDIC, and Unicode are used to encode data, and show how each should be used.
c. Describe how XML and the document object model are being used to integrate and exchange data between systems.
d. Use DTD to create a document definition for a data structure. Given a DTD for data structure, create an XML document with real data.
e. Describe how XSL, XSLT and XPath are used to transform data streams.

03 Intersystem communication protocols
(Level 2 medium degree of engagement)
Competencies:

a. Describe how different types of architectures must be considered for integrating systems.
b. Demonstrate the role of DCOM, CORBA, and RMI in distributed processing.
c. Describe how web services are used to integrate disparate applications in an organization. Describe the role of the WSDL, SOAP, and UDDI architectures in creating and using web services.
d. Demonstrate the role of socket programming in communicating between systems. Contrast the protocols and uses of TCP/IP sockets and Datagram sockets.
e. Describe the purpose of message and queuing services and demonstrate how they work. Illustrate the protocol used by one messaging service (e.g., JMS).
f. List the commonly used low level data communications protocols (e.g., RS232), describe how to know when each protocol should be used, and illustrate the protocol for one low-level communication protocol.

04 Integrative programming
(Level 2 medium degree of engagement)
Competencies:

a. Describe how design patterns are useful in integrative programming.
b. Evaluate the motivation for using each of the following design patterns: MVC, singleton, factory method, façade, proxy, decorator, and observer.
c. Describe how a programming interface is used in programming, and illustrate with an example when the use of a programming interface simplified the development of a system.
d. Define the concept of inheritance and describe how it can be applied to encourage code reuse.
e. Design an abstract class and use inheritance to create a class that extends the abstract class.
f. Design, develop, and test an application that uses the abstract class.

05 Scripting techniques
(Level 2 medium degree of engagement)
Competencies:

a. Describe how scripting languages are used for web scripting, server-side scripting, and operating system scripting.
b. Write, debug, and test a script that includes selection, repetition, and parameter passing.

06 Defensible integration
(Level 1 minimal degree of engagement)
Competencies:

a. Contrast evidence-based security vs. code access security.
b. Define and evaluate several goals of secure coding.
c. Justify the guidelines for authenticating and defining permissions to systems services and resources.
d. For each of the following “best secure coding” practices, give an example of a problem that can occur when the practice is not followed and then describe how to overcome the problem.
   1. Preventing buffer overflow
   2. Securing state data
   3. Securing method access
   4. Wrapper code
   5. Unmanaged code
   6. Validation of user input
   7. Remoting considerations
   8. Protected objects
   9. Serialization
   10. Robust error handling.

Note: Level L1 (L1) used within a subdomain indicates a minimal degree of engagement associated with the learning proficiency of the fundamentals of the subdomain.

Levels 2 (L2) and 3 (L3) used within a subdomain indicate medium and large degrees of learning engagement associated with the application and transferring of learning to complex problems and situations.