**SCEA Self Test(Q's & A's)**

**Chapter 2 - Common Architectures and Protocols**

**Certification Summary**

****

**Given an Architecture Described in Terms of Network Layout, List Benefits and Potential Weaknesses Associated with It**

                    *Architecture* refers to an abstract representation of a system's components and behaviors. A good system architecture leads to reusable components because each component is broken into parts that may be repeated and can therefore be reused. Abstraction naturally forms layers representing different levels of complexity.

                    *System* architecture corresponds to the concept of architecture as *a product.* It is the result of a design process for a specific system and must consider the functions of components, their interfaces, their interactions, and constraints. This specification is the basis for application design and implementation steps.

                    *Reference* architecture corresponds to architecture as a *style* or *method.* It refers to a coherent design principle used in a specific domain.

                    The key difference between the terms *architecture* and *design* is in the level of details. Architecture operates at a high level of abstraction with less detail. Design operates at a low level of abstraction, obviously with more of an eye to the details of implementation.

                    The *layers* of architecture are systems in themselves. They obtain input from their environment and provide output to their environment.

**TWO-MINUTE DRILL**

**Recognize the Effect on Each of the Following Characteristics of Two-tier, Three-tier and Multi-tier Architectures: Scalability Maintainability, Reliability, Availability, Extensibility, Performance, Manageability, and Security**

                    The attributes of a system based on solid architectural principles will include the following:

o                           **Availability** The degree to which a system is accessible. The term *24×7* describes total availability. This aspect of a system is often coupled with performance.

o                           **Reliability** The ability to ensure the integrity and consistency of an application and its transactions.

o                           **Manageability** The ability to administer and thereby manage the system resources to ensure the availability and performance of a system with respect to the other capabilities.

o                           **Flexibility** The ability to address architectural and hardware configuration changes without a great deal of impact to the underlying system.

o                           **Performance** The ability to carry out functionality in a time frame that meets specified goals.

o                           **Capacity** The ability of a system to run multiple tasks per unit of time.

o                           **Scalability** The ability to support the required availability and performance as transactional load increases.

o                           **Extensibility** The ability to extend functionality.

o                           **Validity** The ability to predict and confirm results based on a specified input or user gesture.

o                           **Reusability** The ability to use a component in more than one context without changing its internals.

o                           **Security** The ability to ensure that information is not accessed and modified unless done so in accordance with the enterprise policy.

**Self Test**

The following questions will help you measure your understanding of the material presented in this [chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

**Recognize the Effect on Each of the Following Characteristics of Two-tier, Three-tier and Multi-tier Architectures: Scalability Maintainability, Reliability, Availability, Extensibility, Performance, Manageability, and Security.**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is true about the requirements of a banking system?  A.         The need for security is a classic example of a functional service level requirement, and a checking account rule is an example of a nonfunctional requirement.  B.         Security and the mandatory checking account both illustrate functional service level requirements.  C.         Neither security nor the mandatory checking account is an example of any kind of requirement, theoretically speaking.  D.         Security is an architectural nonfunctional requirement and the mandatory checking accounts a functional design requirement.  E.         They are both examples of business use cases. |  |
| **2.** | Which of the following are nonfunctional requirements?  A.         Scalability, availability, extensibility, manageability, and security  B.         Performance, reliability, elaboration, transition, documentation, and security  C.         Specification, elaboration, construction, transition, use cases, and security  D.         Performance, availability, scalability, and security  E.         Reliability, availability, scalability, manageability, and security |  |
| **3.** | Which of the following is the most important item that should be considered when designing an application?  A.         Scalability  B.         Maintainability  C.         Reliability  D.         Meeting the needs of the customer  E.         Performance  F.         Ensuring the application is produced on time and within budget |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=444655476" \l "N246BD97378-0A32-4DE2-88CA-2E2476959169N246BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. Successful software architecture deals with addressing the nonfunctional service level requirements of a system. The design process takes all functional business requirements into account. Security is considered a nonfunctional requirement and specific business rules, such as the one described for the checking account, are considered functional requirements. Choice **D** is the only choice that accurately describes this.  [✗] **A, B, C**, and **E** are not true. Choice **A** is incorrect because the functional and nonfunctional requirements are switched. Choice **B** is incorrect because only one of them is a functional requirement. Choice **C** is incorrect because, as just described, one of them is a functional requirement and the other, a nonfunctional requirement. Finally, Choice **E** is incorrect because business analysis may start with use cases. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=444655476" \l "N966BD97378-0A32-4DE2-88CA-2E2476959169N966BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The nonfunctional service level requirements discussed are performance (I: The system needs to respond within 5 seconds); availability (II: The system needs to have a 99.9 percent uptime); scalability (III: An additional 200,000 subscribers will be added); and security (IV: HTTPS is to be used). Hence, choice **D** is correct.  [✗] **A, B, C**, and **E** are incorrect. There is no mention of extensibility (ability to easily add or extend functionality) and manageability (ability to monitor the health of the system). Hence, choice **A** is incorrect. Specification, elaboration, construction, transition, documentation, and use cases are not nonfunctional service level requirements. Hence, choices **B** and **C** are incorrect. While scalability and reliability may be related (Will the system perform as reliably when more users operate on it?), there is no mention of reliability in the question. Hence, choice **E** is incorrect. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=444655476" \l "N1686BD97378-0A32-4DE2-88CA-2E2476959169N1686BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The most important consideration when designing an application is that it meets the needs of the customer.  [✗] **A, B, C, E**, and **F** are incorrect. Ensuring the application is produced on time and within budget is something that should be done, but it is not the number one concern. The application does not have to be the best possible solution under the circumstances. As long as it meets the customer's needs, it is considered adequate. All of the other considerations are secondary to meeting the customer's needs. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Your have been contacted by a company to help them improve the performance of their e-commerce application. You have suggested that the hardware on which the application is currently deployed (two web servers and a database server) be migrated to three web servers, an application server, and a database server (all on different machines). You assure them that all the required software rewrites will be worth it in the long run. What are the characteristics of your suggested architecture?  A.         Fat clients  B.         Thin clients  C.         Good separation of business logic  D.         Good scalability  E.         Poor separation of business logic  F.         Poor scalability  G.        There is no difference in the separation of business logic |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=444655476" \l "N2316BD97378-0A32-4DE2-88CA-2E2476959169N2316BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **B, C**, and **D** are correct. The system you have suggested they migrate to is a three-tier system. The characteristics of a three-tier system are thin clients, good separation of business logic, and good scalability. This is due to the fact that each tier is separate from the other (for example, it would be possible to change the data store without affecting the business logic).  [✗] **A**, **E**, **F**, and **G** are incorrect. Choice **A** is incorrect; the suggested system has thin clients, the business logic residing on the application server, in the middle tier. Because there is a good separation of business logic, choices **E** and **G** are incorrect. Choice **F** is incorrect, as the three-tier nature of the system makes it very scalable. |

**Chapter 3 - Object-Oriented Analysis and Design**

**Certification Summary**

The UML is a language used for specifying, constructing, visualizing, and documenting the components of a software system. The primary design goals of the UML areas follow:

                    Provide users with a visual modeling language to develop and exchange comprehensive models.

                    Provide mechanisms for extensibility and specialization that extend the core concepts.

                    Create a standard specification that is independent of particular computing languages.

                    Provide a formal base for a modeling language.

                    Support high-level development concepts such as components, collaborations, frameworks, and patterns.

                    Integrate best practices.

**TWO-MINUTE DRILL**

**State the Effect of Encapsulation, Inheritance, and use of Interfaces on Architectural Characteristics.**

UML defines the following elements:

                    **Class** Any uniquely identified abstraction that models a single thing, where the term *object* is synonymous with *instance.* Classes have *attributes* and *methods*.

                    **Interface** A collection of operations that represents a class or specifies a set of methods that must be implemented by the derived class. An interface typically contains nothing but virtual methods and their signatures.

                    **Package** Used to organize groups of like kind elements. The package is the only group type element and its function is to represent a collection of functionally similar classes.

                    **Collaboration** Defines the interaction of one or more roles along with their contents, associations, relationships, and classes.

                    **Use Case** A description that represents a complete unit of functionality provided by something as large as a system or as small as a class.

                    **Component** Represents a modular and deployable system part. It encapsulates an implementation and exposes a set of interfaces.

                    **Node** A physical element object that represents a processing resource, generally having memory and processing capability, such as a server.

                    **State** A condition that can occur during the life of an object. It can also be an interaction that satisfies some condition, performs some action, or waits for some event.

UML defines the following relationships:

                    **Generalization** A specialized version of another class.

                    **Association** Uses the services of another class.

                    **Aggregation** A class "owns" another class.

                    **Composition** A class is composed of another class. Refers to an aggregation within which the component parts and the larger encompassing whole share a lifetime.

                    **Refinement** A refined version of another class.

                    **Dependency** A class dependent on another class.

**UML defines the following diagrams:**

                    **Use case diagram** Used to identify the primary elements and processes that form the system. The primary elements are termed as *actors* and the processes are called *use cases.* The use case diagram shows which actors interact with each use case.

                    **Class diagram** Used to define a detailed design of the system. Each class in the class diagram may be capable of providing certain functionalities. The functionalities provided by the class are termed *methods* of the class.

                    **Package diagram** Groups objects or classes.

                    **State diagram** Represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events.

                    **Activity diagram** Captures the process flow of the system. An activity diagram also consists of activities, actions, transitions, and initial and final states.

                    **Sequence diagram** Represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time ordered. Objects in the sequence diagram interact by passing messages.

                    **Collaboration diagram** Groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

                    **Component diagram** Represents the high-level parts that make up the system. This diagram depicts what components form part of the system and how they are interrelated. It depicts the components culled after the system has undergone the development or construction phase.

                    **Deployment diagram** Captures the configuration of the runtime elements of the application. This diagram is useful when a system is complete and ready for deployment.

**UML can be used to view a system from various perspectives:**

                    **Design view** Structural view of the system; class diagrams and package diagrams form this view of the system.

                    **Process view** Dynamic behavior of a system; state diagrams, activity diagrams, sequence diagrams, and collaboration diagrams form this view.

                    **Component view** Software and hardware modules of the system modeled using the component diagram.

                    **Deployment view** The deployment diagram of UML is used to combine component diagrams to depict the implementation and deployment of a system.

                    **Use Case view** View a system from this perspective as a set of activities or transactions; use case diagrams.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there may be more than one correct answer. Choose all correct answers for each question.

**Interpret UML Diagrams**

|  |  |  |
| --- | --- | --- |
| **1.** | Which one of the following items is *not* one of the phases of the Unified Process?  A.         Inception  B.         Design  C.         Construction  D.         Transition |  |
| **2.** | What is true about a use case?  A.         It is a complete end-to-end business process that satisfies the needs of a user.  B.         It is a description that represents a complete unit of functionality provided by something as large as a system or as small as a class.  C.         It defines the interaction of one or more roles along with their contents, associations, relationships, and classes.  D.         It is a collection of operations that represents a class or specifies a set of methods that must be implemented by the derived class. |  |
| **3.** | Which item is *not* true when speaking of a class?  A.         A class is a nonunique structure.  B.         An *instance* is one computer executable copy of a class, also referred to as an *object.*  C.         Multiple instances of a particular class can exist in a computer's main memory at any given time.  D.         A class is a structure that defines the attribute data and the methods or functions that operate on that data. |  |
| **4.** | What is *not* true about use cases?  A.         There are three types of use cases: essential, real, and virtual.  B.         A virtual use case describes the user's virtual view of the problem and is technology independent.  C.         A real use case describes the process in terms of its real design and implementation.  D.         Essential use cases are of importance early in the project. Their purpose is to illustrate and document the business process. |  |
| **5.** | What is not true about a sequence diagram?  A.         It has two dimensions.  B.         One sequence diagram dimension represents time.  C.         One sequence diagram dimension represents the different objects participating in a sequence of events required for a purpose.  D.         Sequence diagrams are static model views. |  |
| **6.** | Which item is *not* an example of things that a state diagram could effectively model?  A.         Life could be modeled: birth, puberty, adulthood, death.  B.         A computer system infrastructure.  C.         A banking transaction.  D.         A soccer match could be modeled: start, half time, injury time, end. |  |
| **7.** | What is not true about a collaboration diagram?  A.         A collaboration diagram models interactions among objects, and objects interact by invoking messages on each other.  B.         A collaboration diagram groups together the interactions among different objects.  C.         The interactions in a collaboration diagram are listed as alphabetically collated letters that help to trace the sequence of the interactions.  D.         The collaboration diagram helps to identify all the possible interactions that each object has with other objects. |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N2896BD97378-0A32-4DE2-88CA-2E2476959169N2896BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ B is correct because design is not a phase in the unified process.  [✗]A, C, and D are incorrect because the phases of the unified process include inception, whose focus is the scope of the project; elaboration, in which the architecture and the requirements of the product being built must be defined by the end of this phase; construction, during which the software must be developed or constructed; and transition, during which the software must be rolled out to users. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N3296BD97378-0A32-4DE2-88CA-2E2476959169N3296BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ A and B are correct because a use case is a complete end-to-end business process that satisfies the needs of a user. It is also a description that represents a complete unit of functionality provided by something as large as a system or as small as a class.  [✗]C and D are incorrect because a collaboration defines the interaction of one or more roles along with their contents, associations, relationships, and classes. A class diagram is a collection of operations that represents a class or specifies a set of methods that must be implemented by the derived class. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N3666BD97378-0A32-4DE2-88CA-2E2476959169N3666BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ A is correct because a class is unique.  [✗]B, C, and D are incorrect because they are true. A class is a unique structure that defines the attribute data and the methods or functions that operate on that data. An instance is one computer executable copy of a class, also referred to as an object. Multiple instances of a particular class can exist in a computer's main memory at any given time. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N4116BD97378-0A32-4DE2-88CA-2E2476959169N4116BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ A and B are correct because they are false. There are two types of use cases: essential and real.  [✗]C and D are incorrect because they are true. Essential use cases are expressed in an ideal form that remains free of technology and implementation detail. The design decisions are abstracted, especially those related to the user interface. A real use case describes the process in terms of its real design and implementation. Essential use cases are of importance early in the project. Their purpose is to illustrate and document the business process. Real use cases become important after implementation, as they document how the user interface supports the business process documented in the essential use case. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N4516BD97378-0A32-4DE2-88CA-2E2476959169N4516BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ D is correct because it is false. Class and object diagrams are static model views; sequence diagrams are dynamic.  [✗]A, B, and C are incorrect because they are true. The sequence diagram shows the explicit sequence of interactions as they flow through the system to affect a desired operation or result. It has two dimensions; one dimension represents time, and another dimension represents the different objects participating in a sequence of events required for a purpose. Class and object diagrams are static model views. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N4936BD97378-0A32-4DE2-88CA-2E2476959169N4936BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ B is correct because it is false. A computer system infrastructure does not have dynamic states; it is more or less static and the modeler would use a deployment diagram to depict the infrastructure.  [✗]A, C, and D are incorrect because they are true. Life could be modeled. A banking transaction and a soccer match could also be modeled. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N5336BD97378-0A32-4DE2-88CA-2E2476959169N5336BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ C is correct because it is false. The interactions in a collaboration diagram are listed as numbered interactions that help to trace the sequence of the interactions.  [✗]A, B, and D are incorrect because they are true. A collaboration diagram models interactions among objects, and objects interact by invoking messages on each other. A collaboration diagram groups together the interactions among different objects. The interactions in a collaboration diagram are listed as numbered interactions that help to trace the sequence of the interactions. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | What item is not true about a component?  A.         A component represents a modular and deployable system part. It encapsulates an implementation and exposes a set of interfaces.  B.         The component interfaces represent services provided by elements that reside on the component.  C.         A node may be deployed on a component.  D.         A component is shown as a rectangle with two smaller rectangles extending from its left side. A component type has a type name component-type. |  |
| **2.** | Which item(s) is not part of a class in a UML class diagram?  A.         Name  B.         Attributes  C.         Method  D.         Comments |  |
| **3.** | Which item is *not* one of the three kinds of relationships a class diagram can have?  A.         Association  B.         Aggregation  C.         Generalization  D.         Specialization |  |
| **4.** | In a class diagram, what does a line with an arrow from one class to another denote?  A.         Attribute visibility  B.         Class visibility  C.         Method visibility  D.         Global visibility |  |
| **5.** | What is *not a* type of visibility between objects?  A.         Local  B.         Method  C.         Attribute  D.         Global |  |
| **6.** | Which statement is *not* true about state machine and state diagrams?  A.         A state machine is basically a diagram of states and transitions that describes the response of an object of a given class to the receipt of external stimuli, and it is generally attached to a class or a method.  B.         The state diagram shows the sequences of states that an object passes through during its lifetime.  C.         A state diagram represents a state machine: a state being a condition during the life of an object or an interaction during which it satisfies some condition, performs some action, or waits for some event.  D.         State diagrams are used in situations for which all or most of the events represent the completion of internally generated actions (that is, procedural flow of control). |  |
| **7.** | Which of the following UML diagrams may be best suited for a business analyst?  A.         Deployment  B.         Class  C.         Use case  D.         Activity  E.         Collaboration  F.         Sequence |  |
| **8.** | In a UML class diagram, Private, Protected, and Public attributes are shown by which one of the following sets of symbols?  A.         −, +, *#*  B.         +,−,#  C.         #,−,+  D.         *−,#,+*  E.         +,#,−  F.         #, +, − |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N5766BD97378-0A32-4DE2-88CA-2E2476959169N5766BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ C is correct because it is false. A component may be deployed on a node.  [✗]A, B, and D are incorrect because they are true. A component represents a modular and deployable system part. It encapsulates an implementation and exposes a set of interfaces. The interfaces represent services provided by elements that reside on the component. A component is shown as a rectangle with two smaller rectangles extending from its left side. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N6136BD97378-0A32-4DE2-88CA-2E2476959169N6136BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ D is correct because it is false. A comment is not part of a UML class diagram.  [✗]A, B, and C are incorrect because they are true. UML class notation is a rectangle divided into three parts that include class name, attributes, and operations. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N6506BD97378-0A32-4DE2-88CA-2E2476959169N6506BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ D is correct because it is false. Specialization is not a relationship type.  [✗]A, B, and C are incorrect because they are true. Association is a relationship between instances of the two classes. An association exists between two classes if an instance of one class must know about the other to perform its work. In a diagram, an association is a link connecting two classes. Aggregation is an association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole. Generalization is an inheritance link indicating one class is a superclass of the other. A generalization has a triangle pointing to the superclass. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N6906BD97378-0A32-4DE2-88CA-2E2476959169N6906BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ A is correct.  [✗]B, C, and D are incorrect. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N7276BD97378-0A32-4DE2-88CA-2E2476959169N7276BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ B is correct.  [✗]A, C, and D are incorrect. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N7676BD97378-0A32-4DE2-88CA-2E2476959169N7676BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ D is correct because it is false. Activity diagrams are used in situations for which all or most of the events represent the completion of internally generated actions (that is, procedural flow of control). State diagrams, on the other hand, are used in situations for which asynchronous events predominate.  [✗]A, B, and C are incorrect because they are true. The state diagram shows the sequences of states through which an object passes during its lifetime. They correspond to prompts for input couples with the responses and actions. A state machine is basically a diagram of states and transitions that describe the response of an object of a given class to the receipt of external stimuli, and it is generally attached to a class or a method. A state diagram represents a state machine: a state being a condition during the life of an object or an interaction during which it satisfies some condition, performs some action, or waits for some event. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N8076BD97378-0A32-4DE2-88CA-2E2476959169N8076BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ C is correct because use case diagrams show a set of use cases and actors and their relationships. Use case diagrams show the static view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. Use case diagrams are frequently used by business analysts to capture business requirements of a system.  [✗]A, B, D, E, and F are incorrect. Deployment diagrams show the configuration of runtime processing nodes and the components that live within these nodes. Deployment diagrams address the static view of the architecture. Architects frequently use deployment diagrams. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams address the static design view of a system. Software designers frequently use class diagrams. Activity diagrams are a special kind of state chart diagram that shows the flow from activity to activity within the system. This type of diagram is important in modeling the function of a system and emphasizing the flow of control among objects. Designers and developers frequently use activity diagrams. A collaboration diagram is an interaction diagram that emphasizes the structural organization of objects that send and receive messages. Designers and developers frequently use interaction diagrams. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=553785348" \l "N8506BD97378-0A32-4DE2-88CA-2E2476959169N8506BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ D is correct because in UML notation, access modifiers are shown by the −, *#*, and + symbols to represent private, protected, and public, respectively.  [✗]A, B, C, E, and F are incorrect because they do not have the right combination. |

**Chapter 4 - Applicability of JEE Technology**

**Certification Summary**

As you have seen, the JEE platform is a multi-tiered distributed application model, where application logic is divided into components according to their function. The various components of a JEE application are installed on different machines. A component's location depends on which tier or layer in the multi-tiered JEE environment that component belongs to. These components will already exist (legacy, client/server databases, messaging) and must be integrated with the JEE components. The enterprise architect must be aware of the way in which the JEE application framework can be used to integrate seamlessly with the existing myriad of business components that make up the enterprise environment.

As you have seen in the chapter, these components reside at various tiers in the framework. The architect must understand the client tier components, web tier components, and business tier components that run on the JEE server, and, probably most important for the enterprise, the EIS tier.

**TWO-MINUTE DRILL**

**Explain the JEE Architecture and System Requirements**

                    While a JEE application can consist of three or more tiers or layers, JEE multi-tiered applications are generally considered to be three-tiered applications because they are distributed across three different locations: client machines, JEE server machine, and the database or legacy machines at the back end. JEE applications consist of client components, web components, and business components.

                    JEE applications are made up of components: self-contained functional software units assembled into JEE applications with their related classes and files. These components communicate with other components.

                    The component-based and platform-independent JEE architecture facilitates development, because business logic is organized into reusable components, and the JEE server provides underlying services in the form of a container for every component type.

                    A JEE application is usually assembled from two different types of modules: enterprise beans and web components. Both of these modules are reusable; therefore, new applications can be built from pre-existing enterprise beans and components. The modules are also portable, so the application that comprises them will be able to run on any JEE server conforming to the specifications.

**Explain the Use of Patterns in the JEE Framework**

                    The JEE framework employs design patterns to support these capabilities. JEE uses the following core patterns to enable flexible association of EJB classes with other components. The Proxy pattern provides a separate implementation of interface and working code for location transparency. The Decorator provides a similar contract for a class but with added functionality. The Factory Method provides ability to define a contract for creating an object but defers instantiation to subclasses. The Abstract Factory provides a contract for creating families of related or dependent objects without specifying concrete classes.

                    The use of best practices, design patterns, and guidelines is important for JEE architects. Successful architects and developers share their knowledge and pass on their proven techniques to others. The net result is productivity. The ultimate product is the implementation of solid applications.

**Describe the Concepts of "Best Practices" and "Guidelines"**

                    A best practice is an optimal process that is recognized and approved by peers in similar situations. It is applicable to a cross-section of scenarios with varying resources and sizes. It takes design requirements into consideration.

                    A guideline is a rule applied horizontally to the design. Guidelines reflect agreements on practices or operations by recognized professional associations. This includes formal, approved standards, as contrasted to de facto standards and proprietary standards that are exceptions to this concept.

**Illustrate the Use of JEE for Workflow**

                    A common method for designing applications is to organize them around an event-driven user interface. Utilizing the MVC design pattern best practice results in a separation of the application data from the ways that the data can be accessed or viewed as well as from the mapping between system events (such as user interface events) and application behaviors.

**Review Best Practices Applicable for All Tiers**

                    The Enterprise JavaBeans (EJB) tier hosts the application-specific business objects and the system-level services (such as transaction management, concurrency control, and security). The EJB tier is a critical link between the web tier and the EIS integration tier. It typically hosts the entity beans and session beans, data access objects and value objects, and perhaps master-detail modeling using enterprise beans.

**Review Best Practices for the Client Tier**

                    Thin-client solutions (HTML on a browser) are important to Internet-based applications. The browser acts as your client for rendering the presentation as encoded in HTML.

                    In addition to what can be rendered with static HTML, the following items can be used to create web content: JSPs, servlets, applets, and JavaScript can be used to enhance the browser interface.

**Enumerate the Components and Categories of the Web Tier**

                    The two types of components currently specified for the web tier are servlets and JSP pages.

                    Web components are hosted by servlet containers, JSP containers, and web containers.

                    In addition to standard container services, a servlet container provides network services by which requests and responses are sent and that decode requests and format responses. All servlet containers must support HTTP as a protocol for requests and responses, but they may also support additional request-response protocols such as HTTPS.

                    A JSP container provides the same services as a servlet container and an engine that interprets and processes a JSP page into a servlet.

                    A web container provides the same services as a JSP container and provides access to the JEE service and communication APIs.

**Explain How to Apply MVC to the Web Tier**

                    MVC is applied to the web tier by separating the application data from the ways that the data is accessed or viewed. The MVC pattern consists of three component types:

                    The Model, usually a JavaBean or an EJB, represents the application data along with methods that operate on that data.

                    The View component, usually a JSP, displays the data to the user.

                    The Controller, which is usually a servlet, translates user actions such as mouse movement and keyboard input and dispatches operations on the Model.

**Review the Best Practices for the Presentation Layer**

                    Separate HTML from Java.

                    Try to place business logic in JavaBeans.

                    Factor general behavior out of custom tag handler classes.

                    Favor HTML in Java handler classes over Java in JSPs.

                    Use an appropriate inclusion mechanism.

                    Use a JSP template mechanism.

                    Use style sheets.

                    Use the MVC pattern.

                    Use available custom tag libraries.

                    Determine the appropriate level of XML compliance.

                    Use JSP comments in most cases.

                    Follow HTML best practices.

                    Utilize the JSP exception mechanism.

**Review the Internationalization and Localization**

                    The set of political, cultural, and region-specific elements represented in an application is called a *locale.* Applications should customize data presentation to each user's locale. Internationalization, also known as *I18n*, is the process of separating locale dependencies from an application's source code. Examples of locale dependencies include messages and user interface labels, character sets, encoding, and currency and time formats. Localization (also called *L10n)* is the process of adapting an internationalized application to a specific locale. An application must first be internationalized before it can be localized. Internationalization and localization make a JEE application available to a global audience.

**Illustrate When to Use JEE Technology for Given Situations**

                    With respect to security, an entity is something that can have access rights applied to it. A principal is an entity to which privileges can be assigned. A role is a collection of privileges.

                    Authentication is a mechanism by which callers and service providers prove that they are acting on behalf of specific users or systems. Web-tier authentication consists of HTTP basic authentication, form-based authentication, and HTTPS mutual authentication.

                    Authorization entails applying security policies to regulate what specific users, or groups of users, can access in the system. An access control limits the resources a user can access based on permissions. Access control can also be used to limit the type of access a user has to a resource, such as read or write access. There are two approaches to defining access control rules: capabilities are examined to focus on what a caller can do, and permissions focus on who can do what.

                    For proper handling of transactions within the EIS integration tier, it is recommended that a component uses JTA whenever possible when accessing EIS systems. Using JTA transaction allows multiple components accessing EIS to be grouped in a single transaction. If a component marks the transaction as rollback only, all EIS work will be rolled back automatically.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

**Explain the JEE Architecture and System Requirements**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is not true about JEE containers?  A.         An EJB container manages the execution of all enterprise beans for a single JEE application. Enterprise beans and their accompanying containers run on the JEE server.  B.         A web container manages the execution of all JSP and servlet components for a single JEE application. Web components and their accompanying container run on the JEE server.  C.         An application client container manages the execution of all application client components for a single JEE application. Application clients and their accompanying containers run on the JEE server.  D.         An applet container is the web browser and Java plug-in combination that runs on the client machine. |  |
| **2.** | Which statement is *not* true when discussing the EJB tier?  A.         The Enterprise JavaBeans (EJB) tier hosts the application-specific business objects.  B.         The Enterprise JavaBeans (EJB) tier does not host system-level services (such as transaction management, concurrency control, and security); they are hosted on the EIS tier.  C.         The EJB tier is a link between the web tier and the EIS integration tier.  D.         The EJB tier hosts the entity beans and session beans, data access objects and value objects, and perhaps master-detail modeling using enterprise beans. |  |
| **3.** | Which of the following is *not* true when put in the context of JEE transaction processing?  A.         A compensating transaction is a transaction, or group of operations, used to undo the effect of a previously committed transaction.  B.         When choosing a transaction attribute, use *Required* for the default transaction attribute.  C.         When choosing a transaction attribute, use *RequiresNew* when the bean methods need to commit unconditionally.  D.         When using a compensating transaction, it is always possible to undo the effect of a committed transaction. |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N2586BD97378-0A32-4DE2-88CA-2E2476959169N2586BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. An application client container manages the execution of all application client components for a single JEE application. Application clients and their accompanying container run on the client's machine and not the JEE server.  [✗] **A, B**, and **D** are incorrect because they are true. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N3066BD97378-0A32-4DE2-88CA-2E2476959169N3066BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The Enterprise JavaBeans (EJB) tier does host system-level services such as transaction management, concurrency control, and security.  [✗] **A, C**, and **D** are incorrect because they are true. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N3576BD97378-0A32-4DE2-88CA-2E2476959169N3576BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. When using a compensating transaction, it is not always possible to undo the effect of a committed transaction, even if the server crashes.  [✗] **A, B**, and **C** are incorrect because they are true. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | JEE uses the core patterns to enable flexible association of EJB classes with other components. Which of the following is *not* used by JEE?  A.         Proxy  B.         Decorator  C.         Designer  D.         Factory |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N4226BD97378-0A32-4DE2-88CA-2E2476959169N4226BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. JEE uses the core patterns to enable flexible association of EJB classes with other components. The Designer pattern is not one of them.  [✗] **A, B**, and **D** are incorrect because they are true. Decorator, factory, and proxy are core patterns to enable flexible association of EJB classes with other components. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which statement is *not* true when discussing best practices?  A.         Data access objects are a useful best practice, as they encapsulate access to data and maintain a clean separation of bean and database access code.  B.         A session bean facade provides a simple, single point of entry to shared entity beans.  C.         A session bean facade does not shield the client from complex entity bean relationships and manages workflow on a client's behalf.  D.         A session bean facade avoids the problems associated with access to entity beans from the client layer—namely overabundance of network traffic and latency and awkward security management. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N4866BD97378-0A32-4DE2-88CA-2E2476959169N4866BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. A session bean facade *does* shield the client from complex entity bean relationships and manages workflow on the client's behalf.  [✗] **A, B**, and **D** are incorrect because they are true. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | In which of the following cases would an application not necessarily benefit from the use of Enterprise JavaBeans?  A.         Small-scale deployment  B.         Large-scale deployment  C.         Requirements transactional in nature  D.         No transactional requirements |  |
| **2.** | The JEE platform uses a multi-tiered distributed application model; which of the following is not considered a tier in this architecture?  A.         Client tier  B.         Web tier  C.         Enterprise information system (EIS) tier  D.         Security tier |  |
| **3.** | Which of the following are "best practices" for large distributed systems?  A.         Avoid business logic implementation in the display code; display inconsistencies can result because the logic can be copied and modified in one object and not another.  B.         Coding data manipulation logic, format and display code, and user event handling together can make application maintenance simple.  C.         Facilitate reuse of user interfaces by segregating application logic from the code for an existing interface.  D.         Utilizing the MVC design pattern results in a separation of the application data from the ways that the data can be accessed or viewed as well as from the mapping between system events (such as user interface events) and application behaviors. |  |
| **4.** | Which of the following are *not* benefits of using the MVC best practice?  A.         Clarifies application design through separation of data modeling issues from data display and user interaction  B.         Enhances reusability by separating application functionality from presentation  C.         Facilitates maintenance by encapsulating application functions behind trusted APIs  D.         Simplifies database design because only the View components access the database |  |
| **5.** | Which of the following is *not* true of using the MVC best practice?  A.         The Model in an MVC-based application can be divided into two parts: the internal state of the system and the actions that can be taken to alter that state.  B.         The Controller portion of the application focuses on receiving requests from the client (most often a user running a web browser), deciding what business logic function is to be performed, and delegating responsibility for producing the next phase of the user interface to an appropriate View component.  C.         The Model determines how the results should be displayed.  D.         The View transfers user input to the Controller. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N5486BD97378-0A32-4DE2-88CA-2E2476959169N5486BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** and **D** are correct. Enterprise JavaBeans are best used with large and complex enterprise applications with high deployment and transactional requirements.  [✗] **B** and **C** are incorrect. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N6006BD97378-0A32-4DE2-88CA-2E2476959169N6006BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct**.** The security tier is not considered a JEE tier.  [✗] **A, B**, and **C** are incorrect because they are true—client, web, and EIS are JEE tiers. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N6526BD97378-0A32-4DE2-88CA-2E2476959169N6526BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. Coding data manipulation logic, format and display code, and user event handling together can complicate and make application maintenance problematic and costly.  [✗] **A, C**, and **D** are incorrect because they are true. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N7036BD97378-0A32-4DE2-88CA-2E2476959169N7036BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. Does not necessarily simplify database design and typically, the model and not the view component accesses the database.  [✗] **A, B**, and **C** are incorrect because they are true. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N7546BD97378-0A32-4DE2-88CA-2E2476959169N7546BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. The View and not the Model determines how the results should be displayed.  [✗] **A, B**, and **D** are incorrect because they are true. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following are not components of the MVC?  A.         Model  B.         Calculator  C.         View  D.         Controller |  |
| **2.** | Which of the following is *not true* of the MVC?  A.         The View extrapolates data presentation and responds to users with data.  B.         The Controller extrapolates the user interaction/application semantic map and transforms user actions into application actions.  C.         The Model manages persistence.  D.         The Controller maintains the application state. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N8136BD97378-0A32-4DE2-88CA-2E2476959169N8136BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The Calculator is not a component of the MVC pattern.  [✗] **A, C**, and **D** are incorrect because they are true, as Model, View, and Controller are the components of the MVC pattern. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N8616BD97378-0A32-4DE2-88CA-2E2476959169N8616BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The Model, not the Controller, maintains the application state.  [✗] **A, B**, and **C** are incorrect because they are true. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is *not* typically considered a threat to enterprise-critical assets?  A.         The disclosure of confidential information  B.         The modification or destruction of information  C.         The misappropriation of protected resources  D.         A misappropriation that does not compromise availability |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N9206BD97378-0A32-4DE2-88CA-2E2476959169N9206BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. A misappropriation that does not compromise availability is not typically considered a threat to enterprise-critical assets.  [✗] **A, B**, and **C** are incorrect because they are true. The disclosure of confidential information, the modification or destruction of information, and the misappropriation of protected resources are typically considered threats to enterprise-critical assets. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following are not true about screen scrapers?  A.         Screen scrapers function as terminal emulators on one end and as object interfaces on the other.  B.         Screen scraping may be a useful tool when used in conjunction with the off-board servers.  C.         Changes to legacy UI have little or no impact on the new GUI.  D.         Screen scraping is best used when the legacy clients have loose coupling with other tiers. |  |
| **2.** | If the telephone company were to rewrite its existing legacy code using newer JEE technology, what technology would you choose to accommodate both the block purchase and the individual query?  A.         Java Applet technology for the CORBA call and custom socket programming for vanity number requests  B.         Java Servlet API for the CORBA call and JSP for the custom socket programming  C.         Entity EJBs for both  D.         Session EJBs for both  E.         JNDI for both  F.         MQ Series with a JMS-based solution for both |  |
| **3.** | Your company's web site offers the customers price comparisons on a variety of different products. You are in charge of converting the web-based solution over to the appropriate JEE technology. Which of the following should you use?  A.         JSP, servlets  B.         JSP, servlets, EJBs  C.         Applets, EJBs  D.         No need to change it  E.         Perl/CGI scripts is the best solution |  |
| **4.** | Regarding the JEE EIS integration, which of the following statements is *not* true?  A.         Before the JEE Connector architecture was defined, no specification for the Java platform addressed the problem of providing a standard architecture for integrating heterogeneous EISs.  B.         The JEE Connector architecture provides a Java solution to the problem of connectivity between the many application servers and only new EISs, not those already in existence.  C.         Application server vendors who conform to the JEE Connector architecture do not need to add custom code whenever they want to add connectivity to a new EIS. |  |
| **5.** | Regarding the JEE EIS integration contracts, which of the following statements is *not* true?  A.         A Connection Management contract allows an application server to pool connections to an underlying EIS.  B.         A Transaction Management contract lets an application server use a transaction manager to manage transactions across multiple resource managers.  C.         A Security Contract provides support for a secure application environment, which reduces security threats to the EIS and protects valuable information resources managed by the EIS.  D.         A Transaction Management contract does not support transactions that are managed internal to an EIS resource manager without the necessity of involving an external transaction manager. |  |
| **6.** | Which is the following is *not* true about enterprise applications and integration?  A.         Data integration focuses on integrating existing data with enterprise applications. For example, an integration might entail integrating a web-based order management system with an existing order and customer database.  B.         Legacy integration involves integrating new enterprise applications with applications and EISs that have been in operation for some time, often referred to as an enterprise's *legacy* systems.  C.         Application integration occurs when existing enterprise applications may be off-the-shelf bundled applications or they may be developed in-house.  D.         Enterprise application development is about building an enterprise application from scratch. |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N9796BD97378-0A32-4DE2-88CA-2E2476959169N9796BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **C** and **D** are correct. When using screen scrapers, any changes to the legacy user interface will also affect the new GUI. In addition, screen scraping is the best alternative only if the existing UI is tightly coupled with the business tier of the legacy application. Therefore, choices **C** and **D** are false and, therefore, the correct choices.  [✗] **A** and **B** are true about screen scrapers and, therefore, the incorrect choices. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N10396BD97378-0A32-4DE2-88CA-2E2476959169N10396BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **D** is correct. Session beans can be used for making both the CORBA call for block purchase of telephone numbers and the custom synchronous call to request a special vanity number.  [✗] **A, B, C, E**, and **F** are incorrect. Both operations represent business processes involving partner OSS integration. Applets are not used for modeling the business workflow of a system. Therefore, choice **A** is incorrect. JSP represents the view construction process in an MVC application. It should not be used for processing business logic. Therefore, **B** is incorrect. Entity beans represent the business model of an application and provide a representation of enterprise data. They are not to be used for workflow processing, which is better accomplished by using session beans. Therefore, **C** is incorrect. JNDI provides Naming and Directory interfaces, not workflow processing. Therefore, choice **E** is incorrect. The question specifically says that a synchronous mechanism is to be used for the vanity number request. The CORBA RPC call for TN reservation is also synchronous. MQ Series is a MOM used for messaging. Messaging is an inherently asynchronous communication mechanism. Therefore, choice **F** is incorrect. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N11186BD97378-0A32-4DE2-88CA-2E2476959169N11186BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **A** is correct as using JSP and servlets is the best option.  [✗] **B, C, D**, and **E** are incorrect. The important element to this question is that the revenue is generated by click-through sales. This implies that there are no transactions involved and you do not need to use EJBs. Therefore, choices **B** and **C** are not the best options. Perl/CGI scripts are harder to maintain than Java code. Therefore, choice **E** is not the best option. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N11816BD97378-0A32-4DE2-88CA-2E2476959169N11816BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **B** is correct. The JEE Connector architecture provides a Java solution to the problem of connectivity between the many application servers and most EISs, not just those already in existence.  [✗] **A, C**, and **D** are incorrect because they are true. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N12296BD97378-0A32-4DE2-88CA-2E2476959169N12296BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **D** is correct. A Transaction Management contract *does* support transactions that are managed internal to an EIS resource manager without the necessity of involving an external transaction manager.  [✗] **A, B**, and **C** are incorrect because they are true. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=884915591" \l "N12836BD97378-0A32-4DE2-88CA-2E2476959169N12836BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **D** is correct. Enterprise application development is about building an enterprise application from scratch or integrating new enterprise applications with applications and EISs that have been in operation for some time; they are often referred to as an enterprise's *legacy* systems.  [✗] **A, B**, and **C** are incorrect because they are true. |

**Chapter 5 - Design Patterns**

**Certification Summary**

By studying this chapter, you now have an understanding of the GoF design patterns and some introductory material on J2EE patterns. You should also understand which are the most appropriate patterns to use for given scenarios.

**TWO-MINUTE DRILL**

Here are some of the key points from each certification objective in [Chapter 5](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=477#477).

**Identify the Benefits of Using Design Patterns**

                    Help designers to focus on solutions quicker if they recognize patterns that have been successful in the past.

                    Give new ideas to designers who have studied patterns.

                    Provide a common language for design discussions.

                    Provide a solution to a real-world problem.

                    Capture knowledge and document the best practices for a domain.

                    Document decisions and the rationale that lead to the solution.

                    Reuse the experience of predecessors.

                    Communicate the insight already gained previously.

                    Describe the circumstances (when and where), the influences (who and what), and the resolution (how and why it balances the influences).

**Identify the Most Appropriate Design Pattern for a Given Scenario**

                    The Abstract Factory is most appropriate when the system needs to be independent of how its objects are created, composed, and represented.

                    The Adapter is most appropriate when you want to utilize an existing class with an incompatible interface.

                    The Bridge is most appropriate when you want to avoid a permanent binding between the functional abstraction and its implementation.

                    The Builder is most appropriate when the algorithm for creating a complex object needs to independent of the components that compose the object and how they are assembled.

                    The Chain of Responsibility is most appropriate when more than one object can handle a request and the handler is unknown.

                    The Command is most appropriate when you need to parameterize objects by an action to perform.

                    The Composite is most appropriate when you want to represent a full or partial hierarchy of objects.

                    The Decorator is most appropriate when you want to transparently and dynamically add responsibilities to objects without affecting other objects.

                    The Facade is most appropriate when you want to provide a simpler interface to a more complex subsystem.

                    The Factory Method is most appropriate when a class is not able to anticipate the class of objects it needs to create.

                    The Flyweight is most appropriate when the application uses a considerable number of objects.

                    The Interpreter is most appropriate when the grammar of the language is not complicated and efficiency is not a priority.

                    The Iterator is most appropriate when access to a collection object is required without having to expose its internal representation.

                    The Mediator is most appropriate when a set of objects communicates in complex but well-defined ways.

                    The Memento is most appropriate when a snapshot containing enough information regarding the state of an object can be saved so that it can be restored to the complete state using the snapshot information later.

                    The Observer is most appropriate when a change to an object requires changing other objects, and the number of objects that need to be changed is unknown.

                    The Prototype is most appropriate when the classes to instantiate are to be specified at runtime.

                    The Proxy is most appropriate when you need a more versatile or sophisticated reference to an object, rather than a simple pointer.

                    The Singleton is most appropriate when a single instance of a class is needed, and it must be accessible to clients from a well-known access point.

                    The State is most appropriate when the behavior of an object depends on its state and it must be able to change its behavior at runtime according to the new state.

                    The Strategy is most appropriate when multiple classes differ only in their behavior.

                    The Template Method is most appropriate when you want to implement the nonvarying parts of an algorithm in a single class and the varying parts of the algorithm in subclasses.

                    The Visitor is most appropriate when an object structure contains many objects with differing interfaces and you need to perform operations on these objects in a way that depends on their concrete classes.

**State the Name of a Gamma et al. Design Pattern Given the UML Diagram and/or a Brief Description of the Pattern's Functionality**

Review the GoF (Gamma et al.) diagrams and associated descriptions that appear earlier in the chapter:

**Identify Benefits of a Specified Gamma et al. Design Pattern**

Here are the benefits for each of the Gamma et al. design patterns:

**Identify the Gamma et al. Design Pattern Associated with a Specified Java EE Technology Feature**

Here is a list of Java EE technology features and the associated design patterns that are used to implement them:

                    The EJB Factory (javax.ejb.EJBHome, javax.ejb.EJBLocalHome) and JMS Connection Factory (javax.jms.QueueConnectionFactory, javax.jms. TopicConnectionFactory) use the Factory Method pattern.

                    The EJB remote reference (javax.ejb.EJBObject) uses the Proxy pattern

                    The JMS Publish/Subscribe Model uses the Observer pattern.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there may be more than one correct answer. Choose all correct answers for each question.

**Identify the Benefits of Using Design Patterns**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is not a benefit of using Design Patterns?  A.         They provide a common language for design discussions.  B.         They provide solutions to "real-world" problems.  C.         They communicate the insight already gained previously.  D.         They provide solutions to totally novel problems. |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N1956BD97378-0A32-4DE2-88CA-2E2476959169N1956BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. Design patterns do not address totally novel problems, so this cannot be a benefit gained.  [✗] **A, B, and C** are incorrect. These are benefits gained by using Design Patterns. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | The Factory Method design pattern is useful when a client must create objects having different  A.         Subclasses  B.         Ancestors  C.         Sizes  D.         Similarities |  |
| **2.** | What design pattern limits the number of instances a class can create?  A.         Command  B.         Limiter  C.         Strategy  D.         Singleton |  |
| **3.** | Iterators are useful when dealing with which of the following types of classes?  A.         Dynamic  B.         Collection  C.         Singleton  D.         Small |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N2516BD97378-0A32-4DE2-88CA-2E2476959169N2516BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** is correct. The Factory Method design pattern is useful when a client must create objects having different subclasses.  [✗] **B, C, and D** are incorrect. The Factory Method design pattern is not useful with these situations. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N2966BD97378-0A32-4DE2-88CA-2E2476959169N2966BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The Singleton pattern limits the number of instances a class can create.  [✗] **A, B, and C** are incorrect. These do not limit the number of instances a class can create. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N3416BD97378-0A32-4DE2-88CA-2E2476959169N3416BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. Iterators are useful when dealing with Collection classes.  [✗] **A, C, and D** are incorrect. These are not appropriate for the Iterator pattern. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | What is the Abstract Factory pattern also known as?  A.         Kit  B.         Wrapper  C.         Cursor  D.         Virtual Constructor |  |
| **2.** | Which pattern is shown in the diagram?  A.         Abstract Factory  B.         Factory Method  C.         Command  D.         Chain of Responsibility |  |
| **3.** | What pattern is also known as Virtual Constructor?  A.         Abstract Factory  B.         Memento  C.         Wrapper  D.         Factory Method |  |
| **4.** | Which pattern is shown in the diagram?  A.         Proxy  B.         Decorator  C.         Bridge  D.         Observer |  |
| **5.** | What is the Adapter pattern also known as?  A.         Surrogate  B.         Wrapper  C.         Token  D.         Proxy |  |
| **6.** | Which pattern is shown in the diagram?  A.         Proxy  B.         Facade  C.         Adapter  D.         Bridge |  |
| **7.** | What pattern is also known as Handle/Body?  A.         Proxy  B.         Adapter  C.         Abstract Factory  D.         Bridge |  |
| **8.** | Which pattern is shown in the diagram?  A.         Chain of Responsibility  B.         Command  C.         Memento  D.         Factory Method |  |
| **9.** | What is the Decorator pattern also known as?  A.         Wrapper  B.         Adapter  C.         Composite  D.         Strategy |  |
| **10.** | Which pattern is shown in the diagram?  A.         Template Method  B.         Command  C.         Singleton  D.         State |  |
| **11.** | What pattern is also known as Surrogate?  A.         Observer  B.         Bridge  C.         Proxy  D.         Decorator |  |
| **12.** | What is the Command pattern also known as?  A.         Action  B.         Transaction  C.         Wrapper  D.         Surrogate |  |
| **13.** | The Command design pattern\_\_\_\_\_a request in an object.  A.         Separates  B.         Encapsulates  C.         Processes  D.         Decouples |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N3946BD97378-0A32-4DE2-88CA-2E2476959169N3946BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** is correct. The Abstract Factory pattern is also known as Kit.  [✗] **B, C, and D** are incorrect. These are not valid aliases for Abstract Factory. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N4396BD97378-0A32-4DE2-88CA-2E2476959169N4396BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The diagram depicts the Factory Method pattern.  [✗] **A, C, and D** are are incorrect. These are not depicted in the diagram. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N5006BD97378-0A32-4DE2-88CA-2E2476959169N5006BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The Factory Method pattern is also known as the Virtual Constructor.  [✗] **A, B,** and **C** are incorrect. These are not valid aliases for Virtual Constructor. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N5496BD97378-0A32-4DE2-88CA-2E2476959169N5496BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. The diagram depicts the Bridge pattern.  [✗] **A, C, and D** are incorrect. These are not depicted in the diagram. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N6106BD97378-0A32-4DE2-88CA-2E2476959169N6106BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The Adapter pattern is also known as the Wrapper.  [✗] **A, C, and D** are incorrect. These are not valid aliases for Adapter. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N6556BD97378-0A32-4DE2-88CA-2E2476959169N6556BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The diagram depicts the Facade pattern.  [✗] **A, C, and D** are are incorrect. These are not depicted in the diagram. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N7176BD97378-0A32-4DE2-88CA-2E2476959169N7176BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **D** is correct. The Bridge pattern is also known as Handle/Body.  [✗] **A, B, and C** are incorrect. These are not valid aliases for Handle/Body. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N7626BD97378-0A32-4DE2-88CA-2E2476959169N7626BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** is correct. The diagram depicts the Chain of Responsibility pattern.  [✗] **B, C, and D** are incorrect. These are not depicted in the diagram. |
| **[9.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N8246BD97378-0A32-4DE2-88CA-2E2476959169N8246BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** is correct. The Decorator pattern is also known as the Wrapper  [✗] **B, C, and D** are incorrect. These are not valid aliases for Decorator. |
| **[10.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N8696BD97378-0A32-4DE2-88CA-2E2476959169N8696BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A** is correct. The diagram depicts the Template Method pattern  [✗] **B, C, and D** are incorrect. These are not depicted in the diagram. |
| **[11.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N9316BD97378-0A32-4DE2-88CA-2E2476959169N9316BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C** is correct. The proxy pattern is also known as Surrogate.  [✗] **A, C, and D** are incorrect. These are not valid aliases for Surrogate. |
| **[12.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N9766BD97378-0A32-4DE2-88CA-2E2476959169N9766BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **A and B** are correct. The Command pattern is also known as Action or Transaction  [✗] **C and D** are incorrect. These are not valid aliases for Command. |
| **[13.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N10216BD97378-0A32-4DE2-88CA-2E2476959169N10216BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **B** is correct. The Command design pattern encapsulates a request in an object.  [✗] **A, C, and D** are incorrect. These are not valid descriptions of the Command pattern. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following elements are parts of the Gang of Four (GoF) Design Pattern format?  A.         Problem  B.         Solution  C.         Consequences  D.         Intent |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N10746BD97378-0A32-4DE2-88CA-2E2476959169N10746BD97378-0A32-4DE2-88CA-2E2476959169)** | | ✓ **C and D** are correct. Consequences and Intent are valid elements in the (GoF) Design Pattern format.  [✗] **A and B** are incorrect. These are not valid elements in the (GoF) Design Pattern format. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | The Decorator pattern appears in which of the following Java packages?  A.         *java.io*  B.         *java.awt*  C.         *Java.long*  D.         *java.util* |  |
| **2.** | Which Java package contains classes that implement the Iterator design pattern?  A.         *Java. enumeration*  B.         *java.util*  C.         *Java.math*  D.         *java.text* |  |
| **3.** | What two methods are defined by the Enumeration interface?  A.         hasMoreElements()  B.         getElementO  C.         nextElement()  D.         nextelement() |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N11276BD97378-0A32-4DE2-88CA-2E2476959169N11276BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **A and B** are correct. The Decorator pattern appears in the *java.io* and *java.awt* packages  [✗] **C and D** are incorrect. These do not contain the Decorator pattern. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N11826BD97378-0A32-4DE2-88CA-2E2476959169N11826BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **B** is correct. The *java.util* package contains classes that implement the Iterator design pattern.  [✗] **A, C, and D** are incorrect. These do not implement the Iterator design pattern. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=104861818" \l "N12346BD97378-0A32-4DE2-88CA-2E2476959169N12346BD97378-0A32-4DE2-88CA-2E2476959169)** | ✓ **A and C** are correct. The Enumeration interface contains hasMoreElements()and nextElement()methods.  [✗] **B and D** are incorrect. These are not valid methods in the Enumeration interface. |

**Chapter 6 - Legacy Connectivity**

**Certification Objective 6.01: Distinguish Appropriate from Inappropriate Techniques for Providing Access to a Legacy System from Java Technology Code Given an Outline Description of That Legacy System**

The following ten exercises are in the form of practice essay questions:

1.          Read the question.

2.          Develop an essay-style answer.

3.          Review the draft and finalize your response.

4.          Review the answer in the book.

**EXERCISE 6-1: Techniques and Best Practices**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **1.** | **Question** As an enterprise architect who is commissioned to enable a set of existing legacy or EIS systems to handle JEE technology, what are some of the techniques and best practices that you might incorporate? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
| [**1.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968#N12346BD97378-0A32-4DE2-88CA-2E2476959169N12346BD97378-0A32-4DE2-88CA-2E2476959169) | | **Answer** When integrating existing EIS with any new technology, especially JEE, it's given that EIS integrates more easily when using proven guidelines and standards. Following are some tried-and-true concepts.  JEE systems that access existing or external information resources should avoid accessing those resources directly from multiple locations, i.e., use one point of access to facilitate changes and avoid potential data integrity problems. Otherwise, multiple access entwines your business logic with the implementation details of an external resource. If the API to that resource changes (when, for example, you change resource vendors, a new version is released, or for other reasons), changes to your JEE application source code will be necessary throughout the application, and the resulting testing burden can be considerable.  Per our desire to adhere to standards for some resources, an EIS resource vendor or some third party may provide a JEE connector extension, an adapter that allows JEE systems to interoperate with other EIS resources transparently and with transaction management capability.  For database access, a standard in the JDBC API makes vendors' proprietary technology accessible in an open way. Switching database implementations, even at runtime, is facilitated with JDBC, which provides a standard API to mask the vendor-specific implementation details in connection configuration data and access and transaction functionality.  In addition, it is a good practice to use a design pattern to encapsulate access to EIS resources and prepare for eventual migration to a JCA-based interface. If no JEE connector extension is available for your EIS resource, a good alternative is to use DAO classes to represent the EIS as an abstract resource. Instead of calling the EIS directly from the enterprise bean, create a DAO class that represents the services your bean needs. This is an application of the Bridge design pattern, which makes an interface's implementation transparently replaceable by decoupling the implementation from the interface. A DAO class that "wraps" an EIS resource insulates the enterprise bean from changes in that resource. New versions of the EIS resource can then be implemented and the change control will be the only necessary modifications to the DAO layer of the JEE application. Another benefit of this practice is experienced when a connector becomes available for your EIS resource. The enterprise can replace the existing DAO implementation with one that simply dispatches calls to the connector.  Imagine that the enterprise is using a custom legacy system to which your enterprise bean needs access. A DAO class can provide a "vendor-agnostic" API interface to the enterprise bean, while handling the details of service requests from the enterprise bean to the legacy system. This scheme is advantageous when a single service request from your JEE server's perspective requires access to a number of existing EIS resources—that is, a DAO can be used to facade multiple EIS resources. This use of the facade pattern facilitates changes to these EIS services. When an existing EIS service is replaced, the existing DAO class can be replaced with a new DAO class that presents the new service to the enterprise bean in terms of the existing DAO interface. Isolating your enterprise bean functionality with a DAO layer makes it easier for your JEE system design to evolve with time.  The DAOclass(es) should reflect the functional requirements of the services your enterprise beans need, not necessarily the structure of the existing system. A DAO class' interface should reflect a current view. Analyze what the existing EIS does, determine what needs to be done today and tomorrow, and create methods in the DAO classes that provide the most frequently required functionality. If multiple EIS resources are required to perform a single task, the DAO class can combine access to these systems and present them to the EJB as a single service. So as an EIS integration "best practice", we should avoid letting the structure of existing EIS resources dictate the structure of the integrated system. Instead, architect and design with your new requirements and goals in mind. Use existing legacy resources as services to meet those requirements.  A DAO class should be neither a collection of unrelated tools nor a tool designed for one application, but something that cleanly and completely represents a clear and reusable abstraction. UML diagrams such as collaboration, state chart, activity, and package diagrams can be a help in the analysis. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-2: Implementing Data Validation and Referential Integrity Contraints**

|  |
| --- |
|  |

These essay questions will help develop the ability to articulate and describe the JEE concepts and components used in parts 2 and 3 of the exam.

|  |  |  |
| --- | --- | --- |
|  | **Question** As an architect integrating a JEE system with an existing EIS database system, where should data validation and referential integrity constraints be implemented? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** This is a difficult call. The practical aspects of the decision revolve around the following:                      How much will it ultimately cost?                      How much is already invested in the database application?                      How long is it expected to be functional?  If the DBMS is relational and were implemented after the mid 1980s, it is typically best to use the DBMS functionality to enforce value and referential integrity. Sybase, Oracle, SQL Server, DB2, and Informix—to mention the most popular DBMSs—have had these abilities for many years. These DBMSs include declarative value and referential integrity constraint features, integrated with the Data Description Language (DDL), and they provide built-in declarative triggers to handle cascading actions required for referential integrity, such as deleting all item rows in a canceled order. Implementing these in the enterprise bean layer would duplicate logic, making maintenance difficult. Any change to the database constraints would require making the change to enterprise beans and to the database.  The architectural benefits and capabilities maintaining data integrity constraints in the database layer include the following:                      **Facilitated use by multiple applications** If for some reason multiple applications are responsible for maintaining database integrity, every application creates an opportunity for bugs that would violate that integrity. Furthermore, other applications that may want to access the database are relieved of the duty of maintaining integrity constraints. They still must, of course, deal with error conditions that result if they violate those constraints.                      **Centralization** If the constraints are maintained only in the database, the database is the one place where data can be considered consistent by definition. If data inconsistencies exist, either the integrity constraints are incorrect or the design has flaws.                      **Portability** Simple value and integrity constraints, such as primary keys, simple foreign keys, uniqueness, value range checking, and so on, are reasonably portable.                      **Performance and reliability** Database vendors that offer database constraints features have invested a great deal of time and money in ensuring that those features operate correctly and efficiently.  The drawbacks of using the DBMS built-in database integrity constraints mechanisms and the EJB can include the following:                      **Possible duplication of logic** Enterprise beans generally need reasonable data to perform properly. Therefore, most well-designed enterprise beans do a reasonably good job of checking data values and existence constraints. Database integrity violation errors usually indicate a bug or a problem with the design. Nevertheless, the logic enforcing value and referential integrity is necessarily duplicated. Changing the integrity rules in the database will usually also entail changes to the code, and keeping the two synchronized can be a problem.                      **Potential nonportability of DBMS constraints** While simple value and referential integrity constraints are fairly portable, databases differ in coverage and syntax for more involved mechanisms such as composite foreign keys, database triggers, and procedural triggers. Procedural triggers in particular  are portability concerns, because, when offered, they are often written in the database vendor's product-specific proprietary language. For example, Sybase Transact SQL is very different from the Oracle PL/SQL procedure language.                      **Database definition and configuration is uncontrolled** Because database constraint and trigger configuration are performed with the database vendor's tools, such constraints are maintained outside of the JEE server framework. Because the data model constraints are specified not in the deployment descriptor but in the persistence layer, such constraints are not part of a JEE server deployment. They must therefore be managed separately, complicating deployment and maintenance and providing another possible avenue for system flaws.  Another option is to use the EJB to handle constraints. Referential integrity constraints can be implemented in the EJB tier. The constraints required for an application may not be available in the DBMS chosen. The data model may have constraint requirements that cannot be satisfied using the DBMS constraint language. Such constraints can be implemented in the EJB tier. EJB-tier constraint management also provides portability, since the enterprise beans will operate identically in JEE-branded containers. Constraints in the EJB tier can also be controlled by way of environment settings in the application deployment descriptor, centralizing constraint management and making it controllable at deploy time.  Yet another option is to implement constraints in both the EJB and database tiers and configuring the constraint implementation at deploy time. This strategy is useful especially when an application must be portable to many different databases, and you want consistent behavior across vendors while optimizing performance by using each database's full power.  You could also create a persistence server for the EIS tier. Constraints should be expressed in a declarative constraint language provided by the database vendor. In their absence, the implementation should choose to wrap a layer of integrity management software around the database API. The EIS tier of your application can be an API that you create to wrap the database. Your application accesses the data store only through that server. This application of the decorator design pattern can provide a solution that is portable across databases, is declaratively configurable, and provides a consistent behavior across various clients. As a great deal of design, construction, validation, and maintenance are required, it should be the solution where ultimate flexibility and portability is required.  Finally, commercial transaction processing (TP) monitors provide the benefits of the persistence server just described. TP monitors can provide scalability and availability. Typically, they work with multiple database vendors. You avoid vendor "lock-in" by wrapping calls to the monitor in DAO classes. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-3: Legacy Mapping**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** What is legacy object mapping? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** Legacy object mapping builds wrappers around legacy system interfaces to access elements of the legacy business logic and database tiers directly. Legacy object mapping tools are used to create proxy objects that access legacy system functions and make them available in an object-oriented manner. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-4: Transaction Monitors**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** What is the purpose of a transaction monitor? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** Transaction monitors are programs, such as IBM CICS, that monitor transactions, to ensure that they are completed in a successful manner. They ensure that successful transactions are committed, that unsuccessful transactions are aborted, and that the in-flight data updates are rolled back to the status quo ante or the state it was before the attempted change. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-5: Off-Board Servers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** What is an off-board server? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** An off-board server is a server that executes as a proxy for a legacy system. It communicates with the legacy system using the custom protocols supported by the legacy system. It communicates with external applications using industry-standard protocols. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-6: JDBC vs. ODBC**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** How does Java Database Connectivity (JDBC) differ from the Microsoft database connectivity interface (Open Database Connectivity, or ODBC)? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** ODBC is the industry-standard interface by which database clients connect to database servers. JDBC is a pure Java solution that does not follow the ODBC standard. A bridge between JDBC and ODBC allows JDBC to access databases that support ODBC. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-7: Accessing Legacy System Software**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** How is Java Native Interface (JNI) used to access legacy system software? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** JNI is used to write custom code to interface Java objects with legacy software that does not support standard communication interfaces. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-8: Accessing COM Objects**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** How is Java-to-COM bridging used to access COM objects? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** A Java-to-COM bridge enables COM objects to be accessed as Java classes and Java classes to be accessed as COM objects, thereby providing some support for using Microsoft software with Java. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**EXERCISE 6-9: RMI vs. CORBA**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Question** What are the primary differences between RMI and CORBA, and for what is Internet Inter-ORB Protocol (IIOP) used? |  |

**Answers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Answer** RMI and CORBA are both distributed-object technologies that support the creation, maintenance, and accessibility of objects. CORBA supports a language-independent approach to developing and deploying distributed objects. RMI is a Java-specific approach. IIOP is used to support communication between object request brokers such as CORBA via TCP/IP. RMI uses a stub that is a proxy for a remote object that runs on the client computer. RMI and CORBA use a skeleton as a proxy for a remote object that runs on the server. Stubs forward a client's RMIs (and their associated arguments) to skeletons, which forward them on to the appropriate server objects. Skeletons return the results of server method invocations to clients via stubs. The difference between RMI and CORBA is that the CORBA stubs access the ORB, and then the CORBA skeleton. | |
|  | |  |  |
|  |  |  |  |

|  |
| --- |
|  |

**Certification Summary**

The JCA is a specification for the Java platform that addresses the need to provide a standard architecture for integrating EIS. It complements the use of JNI and RMI to create a Java interface to a process running in its native domain.

**Two-Minute Drill**

**Distinguish Appropriate from Inappropriate Techniques for Providing Access to a Legacy System from Java Technology Code Given an Outline Description of That Legacy System**

                    The EAI facilitates the integration of EISs, or legacy systems, as they are also known. The classic means of communicating with an existing EIS has been a specialized adapter, which implements the support for communication with the EIS and provides access to EIS data and functions. Communication between an adapter and the EIS typically uses a protocol specific to the EIS.

                    Another, more complex, form of an EIS adapter might do its "adaptation" work across diverse component models, distributed computing platforms, and architectures. For example, an EIS may develop a distributed adapter that includes the capability to perform remote communication with the EIS using Java RMI or CORBA.

                    The JCA puts EAI into mainstream use by establishing a standard.

                    The JCA comprises a resource adapter, connection management contracts, transaction management contract, security contract, and the CCI.

                    A JCA resource adapter is specific to an EIS (Tibco) and is contained in a RAR file. The RAR is composed of the JAR files and native libraries required to deploy the resource adapter on a JEE container.

                    A JCA adapter interacts with a JEE server via system contracts. Seven types of system contracts can be used:

o                           Connection management

o                           Transaction management

o                           Security

o                           Life Cycle Management (JCA 1.5)

o                           Work Management Contracts (JCA 1.5)

o                           Message Inflow (JCA 1.5)

o                           Transaction Inflow Contracts (JCA 1.5)

                    The connection management contract describes the interaction between a JEE container and the adapter with respect to pooling and tearing down connections. All JCA resource adapters supply two implementations with the adapter: a *ConnectionFactory* and a *Connection* class.

                    The transaction management contract provides a mechanism to propagate transactions that originate from inside an application server to an EIS. The transaction management contract can control transactions by creating local transactions that exist only on a particular EIS resource.

                    The security contract enables the application server to connect to an EIS using security properties composed of a principle (a user ID) and credentials (a password, a certificate).

                    Life Cycle Management is handled by the *ResourceAdapter* interface in the *javax.resource.spi* package. There are two methods in the *ResourceAdapter* interface that allow for life cycle management: start () and stop ().

                    The Work Management contract allows the resource adapter to submit work to the application server. It does this by creating an object that extends the *Work* interface in the *javax.resource.spi.work* package.

                    The Message Inflow contract allows the resource adapter to react to calls made by the application server to activate and deactivate message endpoints.

                    CCI: To retrieve and update data, JCA's CCI layer is used. The CCI APIs establishing a connection to an EIS cover command execution on an EIS to provide Record/ResultSet interfaces, which encapsulate the query results and allow EIS metadata (the type of data) to be queried.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

**Distinguish Appropriate from Inappropriate Techniques for Providing Access to a Legacy System from Java Technology Code Given an Outline Description of That Legacy System**

|  |  |  |
| --- | --- | --- |
| **1.** | For a system consisting of exclusively Java objects, which distributed technology would be most appropriate for communication?  A.         CORBA  B.         RMI  C.         JNDI  D.         JavaBeans |  |
| **2.** | Which of the following are true about the Interface Definition Language (IDL)?  A.         Interfaces between CORBA objects can be specified using IDL.  B.         Applications can be implemented using IDL.  C.         Interfaces described in IDL can be mapped to other programming languages.  D.         Stubs and skeletons are written in IDL. |  |
| **3.** | An object that implements the interfaces *java.rmi.Remote* and *java.io.Serializable* is being sent as a method parameter from one JVM to another. How would it be sent by RMI?  A.         RMI will serialize the object and send it.  B.         RMI will send the stub of the object.  C.         Both A and B throw an exception. |  |
| **4.** | The RMI compiler rmic runs on which of the following files to produce the stub and skeleton classes?  A.         On the remote interface class file  B.         On the remote service implementation class file  C.         On the remote service implementation Java file  D.         On the remote interface Java file |  |
| **5.** | Which distributed object technology is most appropriate for systems that consist of objects written in different languages and that execute on different operating system platforms?  A.         RMI  B.         CORBA  C.         DCOM  D.         DCE |  |
| **6.** | Which of the following are used by Java RMI?  A.         Stubs  B.         Skeletons  C.         ORBs  D.         IIOP |  |
| **7.** | Which of the following is not a tier of a three-tier architecture?  A.         Client interface  B.         Business logic  C.         Security  D.         Data storage |  |
| **8.** | Which of the following is *not* true about RMI ?  A.         RMI uses the Proxy design pattern.  B.         RMI uses object serialization to send objects between JVMs.  C.         The RMI Registry is used to generate stubs and skeletons.  D.         The RMI client can communicate with the server without knowing the server's physical location. |  |

**Answers**

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N17006BD97378-0A32-4DE2-88CA-2E2476959169N17006BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. RMI would be appropriate for communication between Java objects because it is built into the core Java environment. It is a built-in facility for Java, which allows you to interact with objects that are actually running on JVMs on remote machines on the network.  **A, C**, and **D** are incorrect. CORBA is more extensive than RMI. Unlike RMI, objects that are exported using CORBA can be accessed by clients implemented in any language with an IDL binding. RMI is much more simple and straightforward than CORBA because it supports only Java objects. So where the facilities of CORBA are not required, it is preferable to go for RMI. JNDI and JavaBeans are not distributed object technologies. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N17456BD97378-0A32-4DE2-88CA-2E2476959169N17456BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** and **C** are correct. Interfaces between CORBA objects can be specified using IDL, but it is a language that can be used only for interface definitions. It cannot be used to implement applications.  **B** and **D** are incorrect. We use other languages to implement the interfaces written in IDL. Interfaces written in IDL can be mapped to any programming language. CORBA applications and components are thus independent of the language used to implement them. Stubs and skeletons are not written; they are generated by the IDL compiler. Stubs and skeletons would be in the same language as the corresponding client or server. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N17946BD97378-0A32-4DE2-88CA-2E2476959169N17946BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. When you declare that an object implements the *java.rmi.Remote* interface, RMI will prevent it from being serialized and sent between JVMs as a parameter. Instead of sending the implementation class for a *java.rmi.Remote* interface, RMI substitutes the stub class. Because this substitution occurs in the RMI internal code, one cannot intercept this operation.  **A** and **C** are incorrect. If the object had not implemented *Remote*, it would have been serialized and sent over the network. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N18516BD97378-0A32-4DE2-88CA-2E2476959169N18516BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. The RMI compiler, rmic, can be used to generate the stub and skeleton files. The compiler runs on the remote service implementation class file.  **A, C**, and **D** are incorrect. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N18956BD97378-0A32-4DE2-88CA-2E2476959169N18956BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. CORBA is the most appropriate object technology for systems that use objects written in different languages, and it supports a variety of operating system platforms.  **A, C**, and **D;** each works with specific platforms. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N19446BD97378-0A32-4DE2-88CA-2E2476959169N19446BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** and **B** are correct. RMI uses stubs and skeletons.  **C** and **D** are incorrect because ORBs and IIOP are used with CORBA. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N19976BD97378-0A32-4DE2-88CA-2E2476959169N19976BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. Security is not a tier of a three-tiered architecture.  **A, B**, and **D** are tiers of a three-tiered architecture. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=979723968" \l "N20426BD97378-0A32-4DE2-88CA-2E2476959169N20426BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct because it is *not* true about RMI. RMI uses the proxy design pattern in the stub and skeleton layer. In the proxy pattern, an object in one context is represented by another (the proxy) in a separate context. The proxy knows how to forward method calls between the participating objects. In RMI's use of the proxy pattern, the stub class plays the role of the proxy. RMI uses a technology called *Object Serialization* to transform an object into a linear format that can then be sent over the network wire. The RMI compiler, rmic, is used to generate the stub and skeleton files.  **A, B**, and **D** are incorrect. |

**Self Test Answers**

**Distinguish Appropriate from Inappropriate Techniques for Providing Access to a Legacy System from Java Technology Code Given an Outline Description of That Legacy System**

**Chapter 7 - Enterprise JavaBeans and the EJB Container Model**

**Certification Summary**

If you have studied this chapter diligently, you should have an understanding of session and entity EJBs. You should also understand when it is appropriate to implement the different EJBs.

**TWO-MINUTE DRILL**

**List the Required Classes/Interfaces That Must Be Provided for an Enterprise JavaBeans Component**

                    Prior to EJB 3.0, the required classes/interfaces that must be provided for an EJB component are the home *(EJBHome)* interface, the remote *(EJBObject)* interface, business logic (bean) class, context objects, and the XML deployment descriptor. For EJB 3.0, the only required class is an annotated bean class. The business interface can be generated by default and the XML deployment descriptor is now optional and largely unnecessary for simple EJBs.

**Distinguish Between Session and Entity Beans**

                    A [*session bean*](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=1267#1267) is an EJB that is created by a client and usually exists only for the duration of a single client/server session.

                    An [*entity bean*](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=1251#1251) is an object representation of persistent data maintained in a permanent data store such as a database. A primary key identifies each instance of an entity bean.

**Recognize Appropriate Uses for Entity, Stateful Session, and Stateless Session Beans**

                    Use *stateful* session beans for functionality that requires data to be maintained across business logic method invocations.

                    Use *stateless* session beans for functionality that does not require data to be maintained across business logic method invocations.

**Distinguish Between Stateful and Stateless Session Beans**

                    Stateful session beans maintain data (state) across business logic method invocations.

                    Stateless session beans do not maintain data (state) across business logic method invocations.

                    Stateless session beans can utilize the bean-pooling feature of the EJB container.

**State the Benefits and Costs of Container-Managed Persistence**

                    The benefits of CMP include database independence and container-specific features (such as full-text search). CMP has drawbacks, as only container-supported algorithms persistence can be used, and portability to other EJB containers may be lost.

**State the Transactional Behavior in a Given Scenario for an Enterprise Bean Method with a Specified Transactional Deployment Descriptor**

The following transactional behaviors can be identified for an enterprise bean method:

                    In NotSupported [or @TransactionAttribute(NOT\_SUPPORTED) annotation in EJB 3.0] transactional behavior, existing transactions are suspended during method calls. An existing transaction is suspended until the method called in this bean completes.

                    In Required [or @TransactionAttribute(REQUIRED) annotation in EJB 3.0] transactional behavior, if an enterprise bean method already exists, it will be used. If one does not exist, it will be created.

                    In Supports [or @TransactionAttribute(SUPPORTS) annotation in EJB 3.0] transactional behavior, the container will not start a new transaction, but if a transaction already exists, the bean will be included in that transaction.

                    In RequiresNew [or @TransactionAttribute(REQUIRES\_NEW) annotation in EJB 3.0] transactional behavior, a new transaction is always started when the bean method is called. If a transaction already exists, that transaction is suspended until the new transaction completes.

                    In Mandatory [or @TransactionAttribute(MANDATORY) annotation in EJB 3.0] transactional behavior, if a transaction does not exist, an exception is thrown.

                    In Never [or @TransactionAttribute(NEVER) annotation in EJB 3.0] transactional behavior, if a transaction exists, an exception is thrown.

                    To encapsulate access to data, an application can use intermediate data access objects.

                    The benefits of bean pooling in an EJB container include lowered cost, specific rates of pool reuse, and increased request handling by the application server.

**Given a Requirement Specification Detailing Security and Flexibility Needs, Identify Architectures That Would Fulf ill Those Requirements**

The following is a list of some of the considerations when dealing with questions for the preceding objective:

                    For EJB systems, deciding on Container-Managed or Declarative Security (flexibility) vs. Bean-Managed or Procedural Security (fine-grained approach)

                    Which distributed object technology is most appropriate-RMI, CORBA, DCOM, or DCE

                    Protocols supported in the deployed environment

                    Ability for object serialization for transporting across a network

**Identify Costs and Benefits of Using an Intermediate Data Access Object Between an Entity Bean and the Data Resource**

To encapsulate access to data, an application can use intermediate data access objects. The use of separate objects to access data results in the following:

                    Keeps entity bean code clear and simple

                    Ensures easier migration to container-managed persistence for entity beans

                    Allows for cross-database and cross-schema portability

                    Provides a mechanism that supports tools from different vendors

                    Not useful for CMP entity beans

                    Adds an extra layer

                    Needs more class hierarchy design when using a factory strategy

**State the Benefits of Bean Pooling in an EJB Container**

                    The cost of creating and destroying an *EJBObject* can be expensive, so the concept of pooling is used to share the resources among multiple users.

                    The external deployment descriptor specifies the number of instances to pool and reuse.

                    This pool allows the application server to handle more requests, since the server does not have to spend time creating and destroy *EJBObjects.*

**State the Benefits of Passivation in an EJB container**

                    Passivation allows the EJB container to make the best possible use of server resources by passivating a bean to free up resources and then reactivating it when resources are available.

**Explain How the Enterprise JavaBeans Container Does Life Cycle Management and Has the Capability to Increase Scalability**

                    In the EJB container's life cycle management, while the container handles naming, management, transactional integrity, security, and persistence for the bean developer, the architect needs to determine the settings to tell the container how these concepts apply to a specific bean. This provides greater scalability.

                    The mechanism for creating the deployment descriptors varies from application server to application server, but they all contain the same basic information.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

**List the Required Classes/Interfaces That Must Be Provided for an Enterprise JavaBeans Component**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is not true about Enterprise JavaBeans (prior to EJB 3.0) objects?  A.         The home interface is responsible for locating or creating instances of the desired bean and returning remote references.  B.         The remote interface, or the *EJBObject* interface, typically provides method signatures for business methods.  C.         The bean implements either the *EntityBean* interface or the *SessionBean* interface but need not implement all the methods defined in the remote interface.  D.         The bean must implement one ejbCreate() method for each create() method in the home interface. |  |
| **2.** | Which of the following is true about Enterprise JavaBeans (EJB 3.*x*) objects?  A.         The home interface is no longer required.  B.         The remote interface, or the *EJBObject* interface, typically provides method signatures for business methods.  C.         The bean class implements the *EJBInterface* class.  D.         The bean must be defined in the XML deployment descriptor. |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N256BD97378-0A32-4DE2-88CA-2E2476959169N256BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **C** is correct for Enterprise JavaBeans (prior to EJB 3.0). The bean or enterprise bean is not the EJB object. It extends either the *EntityBean* interface or the *SessionBean* interface. It *must* implement all the methods defined in the remote interface.   **A, B**, and **D** are incorrect because the *EJBObject* or remote object is a wrapper residing inside the container, between the client and the code. It performs setup and shutdown tasks pre- and post–bean call. The *EJBObject* is generated by the EJB server tools. The developer must write another interface, called the remote interface or the *EJBObject* interface, that extends the interface *EJBObject* and provides method signatures for all the business methods. The server automatically generates a Java class that implements the remote interface. The home interface is a factory object responsible for locating and creating instances of the bean. The developer must code for the *EJBHome* interface (that is, extend the interface *EJBHome)*, and provide method signatures for all the desired create() and find() methods. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N126N126)** | |  A is correct for Enterprise JavaBeans (EJB 3.*x*). The home interface is no longer required.   **B, C**, and **D** are incorrect because the remote interface is no longer required, the bean class does not have to implement the methods of any interface, especially a nonexistent one *(EJBInterface)*, and the XML deployment descriptor entries for an EJB are now optional. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which statement is not true when contrasting the use of entity beans and JDBC for database operations?  A.         Entity beans represent real data in a database.  B.         The bean managed entity bean functionally replaces the JDBC API.  C.         The container-managed entity bean automatically retrieves the data from the persistent storage (database).  D.         When using JDBC, you must explicitly handle the database transaction and the database connection pooling. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N190N190)** | |  **B** is correct. Entity beans represent the data in a data store. Entity beans do not obviate the JDBC API; they merely provide an alternative.   **A, C**, and **D** are incorrect because in container-managed entity beans, when the bean is created, the container automatically retrieves the data from the persistent storage (for example, database) and assigns it to the bean's object variables for the user to manipulate or use it. The bean-managed entity bean for the class specifically has to obtain a database connection, retrieve the row/column values, and assign them to the objects in the ejbLoad(), which will be called by the container when it instantiates a bean object. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Suppose that the business logic of an existing application is implemented using a set of CGI programs. Which Java technologies can be used to implement the CGI programs as a Java-based solution?  A.         JMAPI  B.         Screen scrapers  C.         Enterprise JavaBeans  D.         Servlets |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N2516BD97378-0A32-4DE2-88CA-2E2476959169N2516BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **C** and **D** are correct. Both Enterprise JavaBeans and servlets may be used to upgrade CGI programs to Java-based solutions.   **A** and **B** are incorrect. JMAPI and screen scrapers are not Java technology. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following is not true about Enterprise JavaBeans (prior to EJB 3.0) session bean objects?  A.         A session bean can be defined without an ejbCreate() method.  B.         Stateful beans can contain multiple ejbCreate() methods as long as they match the home interface definition.  C.         The home interface of a stateless session bean must include a single create() method without any arguments.  D.         The stateless session bean class must contain exactly one ejbCreate() method without any arguments. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N3126BD97378-0A32-4DE2-88CA-2E2476959169N3126BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** is correct for Enterprise JavaBeans (prior to EJB 3.0). The Java Platform EE specification requires that the home interface of a Stateless session bean must include a single create() method without any arguments.   **B, C**, and **D** are all true statements. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | If you try to create an (prior to EJB 3.0) CMP-based entity bean for a table that does not have a primary key, which of the following statements is not true?  A.         You cannot create CMP entity beans without a database primary key.  B.         Duplicate records may be entered in the table.  C.         You can create CMP entity beans without primary keys, but the findByPrimaryKey() method will be unreliable. |  |
| **2.** | Can you update the primary key field in a CMP entity bean (prior to EJB 3.0)?  A.         No; you cannot update the primary key field in a CMP entity bean.  B.         Yes; you can update the primary key field in a CMP entity bean by using accessor methods for the primary key cmp-fields in the component interface of the entity bean.  C.         Yes; you can update the primary key field in a CMP entity bean by calling ejbStore(). |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N3816BD97378-0A32-4DE2-88CA-2E2476959169N3816BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** is correct for Enterprise JavaBeans (prior to EJB 3.0). Yes, you can create CMP entity beans without primary keys.   **B, C**, and **D** are incorrect because duplicate records may be entered in the table and the findByPrimaryKey() method may return varying rows. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N4316BD97378-0A32-4DE2-88CA-2E2476959169N4316BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** is correct for Enterprise JavaBeans (prior to EJB 3.0). You cannot change the primary key field of an entity bean.   **B** and **C** are incorrect because, according to the EJB specification (prior to EJB 3.0), "Once the primary key for an entity bean has been set, the Bean Provider must not attempt to change it by use of set accessor methods on the primary key cmp-fields. The Bean provider should therefore not expose the set accessor methods for the primary key cmp-fields in the component interface of the entity bean." You can affect an update of a primary key field by removing (deleting) and then recreating (inserting) the bean. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | In an application with several stateless session Enterprise JavaBeans (prior to EJB 3.0), in terms of performance ramifications of storing the remote reference to a stateless session bean, which of the following statements is *least* accurate?  A.         You can cache the stateless session bean reference using the EJBObject.getHandle().  B.         You can use the handle (SSB reference) when attempting to access the bean from here on.  C.         The cost for a remote lookup on a stateless session bean is insignificant and generally does not justify using a handle (SSB reference) to access the bean.  D.         The stateless session bean has no concurrency problems-that is, there is no shared data to be corrupted. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N4886BD97378-0A32-4DE2-88CA-2E2476959169N4886BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **C** is correct for Enterprise JavaBeans (prior to EJB 3.0). The cost for a remote lookup on a stateless session bean can be significant and can justify using a handle (SSB reference) to access the bean.   **A, B**, and **D** are all true statements. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which distributed object technology is most appropriate for systems that consist entirely of Java objects?  A.         RMI  B.         CORBA  C.         DCOM  D.         DCE |  |
| **2.** | Which distributed object technology is most appropriate for systems that consist of objects written in different languages and that execute on different operating system platforms?  A.         RMI  B.         CORBA  C.         DCOM  D.         DCE |  |
| **3.** | Which of the following are used by Java RMI?  A.         Stubs  B.         Skeletons  C.         ORBs  D.         IIOP |  |
| **4.** | Which of the following is not a tier of a three-tier architecture?  A.         Client interface  B.         Business logic  C.         Security  D.         Data storage |  |
| **5.** | Which of the following Java technologies implements transaction management?  A.         RMI  B.         JTS  C.         JMAPI  D.         JTA |  |
| **6.** | Which of the following is not true when discussing application servers and web servers?  A.         A web server understands and supports only the HTTP protocol.  B.         An application server supports HTTP, TCP/IP, and many more protocols.  C.         A web server does not support caching, clustering, and load balancing.  D.         We can configure application servers to work as web servers. |  |
| **7.** | Which statement is not true when discussing serialization in EJB?  A.         Serialization means that a machine A's object passed as part of a method call is flattened into a byte stream that can be sent over a network connection.  B.         All EJB methods arguments and return values must be serializable.  C.         Developers should make sure all objects passed as arguments implement *java.io.Serializable.*  D.         Serialization is not possible in EJB. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| [**1.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895#N548N548) | |  **A** is correct. RMI is the most appropriate distributed object technology for pure Java applications.   **B, C**, and **D** are incorrect because they can work with non-Java objects. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N5936BD97378-0A32-4DE2-88CA-2E2476959169N5936BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** is correct. CORBA is the most appropriate object technology for systems that use objects written in different languages, and it supports a variety of operating system platforms.   **A, C**, and **D** Each Works with specific platforms. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N6386BD97378-0A32-4DE2-88CA-2E2476959169N6386BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **B** are correct. RMI uses stubs and skeletons.   **C** and **D** are incorrect because ORBs and IIOP are used with CORBA. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N6916BD97378-0A32-4DE2-88CA-2E2476959169N6916BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **C** is correct. Security is not a tier of a three-tiered architecture.   **A**,**B**,and **D** are tiers of a three-tiered architecture. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N7446BD97378-0A32-4DE2-88CA-2E2476959169N7446BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** is correct.JTS provides an implementation of transaction management.   **A, C**, and **D**.These do not implement transaction management.JTA defines an API for transaction management; it does not implement it. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N789N789)** | |  **C** is correct. A web server understands and supports only the HTTP protocol, whereas an application server supports HTTP, TCP/IP, and many more protocols.   **A, B**, and **D** are incorrect because web servers and application servers both support features such as caching, clustering, and load balancing. We can also configure an application server to work as web server. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N8346BD97378-0A32-4DE2-88CA-2E2476959169N8346BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **D** is correct. Serialization is possible.   **A, B**, and **C** are incorrect because a good portion of EJB is the framework for underlying remote method invocation. To allow one JVM space A, the ability to invoke methods remotely on objects that are in JVM space B (objects running on another machine on the network), all arguments of each method call and their results must be serializable (that is, classes must implement *java.io.Serializable*). |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following are true about EJB components, containers, and application servers?  A.         Components run in containers.  B.         Containers are hosted by application servers.  C.         Containers run in components.  D.         Application servers run in containers. |  |
| **2.** | Which objects would you find in an Enterprise JavaBeans (prior to EJB 3.0) directory service?  A.         An EJB home interface  B.         An EJB component  C.         The EJB API  D.         An EJB object interface |  |
| **3.** | Containers and servers have the same function. What is the difference between an Enterprise JavaBeans container and an Enterprise JavaBeans server?  A.         Containers run within servers.  B.         Servers run within containers.  C.         Only one server can run in a container.  D.         Only one container can run in a server. |  |
| **4.** | Which of the following Enterprise JavaBeans (prior to EJB 3.0) CMP entity bean methods are used by the container to alert the bean when its state is synchronized with the database?  A.         ejbLoad()  B.         ejbStore()  C.         ejbCreate()  D.         ejbActivate() |  |
| **5.** | What happens if remove() is not invoked on a stateful Enterprise JavaBeans (prior to EJB 3.0) session bean?  A.         Nothing happens; the bean will last forever.  B.         The container will not honor any more requests for the bean.  C.         An exception occurs in the session bean.  D.         The bean is removed after the session time-out has been reached. |  |
| **6.** | With respect to stateful Enterprise JavaBeans (prior to EJB 3.0) session beans, which of the following statements is *not* true?  A.         Stateful session beans support instance pooling.  B.         The life cycle of a stateful session bean is strictly connected with its client.  C.         When the client removes the bean, it cannot be used by another client without being reinitialized.  D.         A stateful session bean has three states: Does not exist, Method Ready, and Passivated. |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N892N892)** |  **A** and **B** are correct because components run in containers that are hosted by application servers.   **C** and **D** are not true. Containers do not run in components. Application servers do not run in containers. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N9416BD97378-0A32-4DE2-88CA-2E2476959169N9416BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct for Enterprise JavaBeans (prior to EJB 3.0). *EJBHome* interfaces are placed in a directory service to facilitate access to an EJB component. The EJB home interface is used to obtain access to an *EJBObject* interface.   **B, C**, and **D** are incorrect because EJB components are never accessed directly, but only through their *EJBHome* and *EJBObject* interfaces. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N10026BD97378-0A32-4DE2-88CA-2E2476959169N10026BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. Enterprise JavaBeans containers run within the context of servers.   **B, C**, and **D** are incorrect. Servers do not run within containers. A server does not run in a container. Many containers can run in a server. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N10476BD97378-0A32-4DE2-88CA-2E2476959169N10476BD97378-0A32-4DE2-88CA-2E2476959169)** | **A** and **B** are correct for Enterprise JavaBeans (prior to EJB 3.0). The container notifies the bean using the ejbLoad()and ejbStore()methods. The ejbLoad()method alerts the bean that its container-managed fields have just been populated with data from the database. This gives the bean an opportunity to do any postprocessing before the data can be used by the business methods. The ejbStore()method alerts the bean that its data is about to be written to the database. This give the bean an opportunity to do any preprocessing to the fields before they are written to the database.   **C** and **D** have nothing to do with data persistence. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N1107N1107)** |  **D** is correct for Enterprise JavaBeans (prior to EJB 3.0). A stateful session bean would be put in an EXIST state until any of the following occurs:  Call remove() on the *EJBObjects'* stub from the client  Call remove(handleToEJBObject) on *EJBHome's* stub from the client System exception in bean Session time-out Container failure   **A, B**, and **C** are not true. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=966109895" \l "N11666BD97378-0A32-4DE2-88CA-2E2476959169N11666BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct for Enterprise JavaBeans (prior to EJB 3.0). There is no clear indication that a stateful session bean is or is not pooled, while for the stateless session bean there is a specific paragraph that discusses the sequence for adding or removing a pooled bean instance. Performance reasons may motivate some containers to pool stateful session beans and avoid the overhead of recreating the entire object. In the black box, it may be pooled, hence the methods ejbActivate() and ejbPassivate() are included.   **B, C**, and **D** are all true statements. |

**Chapter 8 – Messaging**

**Certification Objective 8.01: Identify Scenarios That Are Appropriate to Implementation Using Messaging**

The following table shows some example messaging implementations that can be used as solutions to the given scenarios.

| **SCENARIO & SOLUTION** | |
| --- | --- |
| You need to call a validation application to approve a customer's credit card purchases. Which type of messaging model is best used for this type of communication, and what type of communication works best in such a scenario? | Point-to-point model messaging is best because the message only needs to be processed one time by the validation system. Synchronous communication works best because the results are required before the customer is allowed to use the merchandise. |
| You are using an e-mail application, and you |  |

**Certification Objective 8.02: List Benefits of Synchronous and Asynchronous Messaging**

The benefits of synchronous messaging follow:

                    Because both parties must be active to participate in synchronous messaging, if either party is not active, the message transaction cannot be completed.

                    A message must be acknowledged before proceeding to the next message. If it is not acknowledged, the message cannot be considered processed.

The benefits of asynchronous messaging are as follows:

                    As the volume of traffic increases, asynchronous messaging is better able to handle the spike in demand by keeping a backlog of requests in its queue and then operating at maximum capacity over a period of time instead of needing to service the requests instantaneously.

                    Asynchronous messaging is less affected by failures at the hardware, software, and network levels.

                    When capacities are exceeded, information is not lost; instead, it is delayed.

**Certification Objective 8.03: Identify Scenarios That Are More Appropriate to Implementation Using Asynchronous Messaging, Rather Than Synchronous**

The following scenarios are more appropriate to implementation using asynchronous messaging:

| **SCENARIO & SOLUTION** | |
| --- | --- |
| You need to implement a messaging system in which a response is not required or not immediately required. Which messaging system is most appropriate? | Asynchronous messaging |
| You need a high-volume transaction processing capability for sending messages. Which type of messaging is best suited for this use? | Asynchronous messaging |
| You want a messaging system that uses your system hardware in an efficient manner. Which type of messaging should be used? |  |

**Certification Objective 8.04: Identify Scenarios That Are More Appropriate to Implementation Using Synchronous Messaging, Rather Than Asynchronous**

The following scenario is more appropriate to implementation using synchronous messaging:

| **SCENARIO & SOLUTION** | |
| --- | --- |
| You are using a credit card authorization/user login authentication system to send a message in which the response |  |

**Certification Objective 8.05: Identify Scenarios That Are Appropriate to Implementation Using Messaging, Enterprise JavaBeans Technology, or Both**

The following table shows messaging and EJB implementations that can be used as solutions for the given scenarios.

| **SCENARIO & SOLUTION** | |
| --- | --- |
| You need to perform a transaction that is distributed across multiple applications and systems; which technology is most appropriate for maintaining this type of distributed transaction? | The EJB container provides support for database updates, message processing, and connections to EIS systems using the Java EE Connector Architecture (JCA). This will allow all to participate in the same transaction context. Messaging by itself is not a complete solution for this scenario. |
| You need to broadcast stock prices to applications executing on a trader's desktop… | A publish/subscribe messaging solution will be sufficient. |
| You need to send an order request to another system… | Possibly use a combination of EJB for retrieving order data and messaging for sending the data to the other system. |
| What technology is appropriate for easier integration of incompatible systems? | Use a messaging solution to provide the interface between systems that are not able to communicate directly. |

**Certification Summary**

JMS provides a highly flexible and scalable solution for building loosely coupled applications in the enterprise environment. It brings all of the advantages of a messaging-based application into the Java language. JMS links messaging systems with all the benefits of Java technology for rapid application deployment and application maintenance.

This chapter should give you an understanding of the JMS and messaging in general and the appropriate scenarios for using messaging-in applications.

**TWO-MINUTE DRILL**

Here are some of the key points from each certification objective in [Chapter 8](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=934#934).

**Identify Scenarios That Are Appropriate to Implementation Using Messaging**

                    Scenarios appropriate to implementation using message include asynchronous communication, one-to-many communication, guaranteed messaging, and transactional messaging.

**List Benefits of Synchronous and Asynchronous Messaging**

                    Some benefits to synchronous messaging are that both parties must be active to participate and the message must be acknowledged before proceeding to the next.

                    Asynchronous messaging benefits are that as the volume of traffic increases, more bandwidth or additional hardware is not required; it is less affected by failures at the hardware, software, and network levels; and when capacities are exceeded, information is not lost but is instead only delayed.

**Identify Scenarios That Are More Appropriate to Implementation Using Asynchronous Messaging, Rather Than Synchronous**

                    Scenarios more appropriate to asynchronous messaging include those in which a response is not required or not immediately required.

                    Asynchronous processing is also more appropriate for high-volume transaction processing.

**Identify Scenarios That Are More Appropriate to Implementation Using Synchronous Messaging, Rather Than Asynchronous**

                    One scenario more appropriate to synchronous messaging includes that in which a response to the message is required before continuing, for example, for transactions requiring credit card or user login authentication.

                    A second scenario includes a transaction where both parties must be active participants.

**Identify Scenarios That Are Appropriate to Implementation Using Messaging, Enterprise JavaBeans Technology, or Both**

                    The scenarios appropriate for messaging technology include broadcasting stock prices to traders, instant messages, and in situations when integration of incompatible systems is necessary.

                    The scenarios appropriate for EJB technology include those that perform business logic and those that maintain persistent data.

                    The scenarios appropriate for messaging and EJB technology including those that require maintenance of distributed transactions and those that send an order to another system.

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there may be more than one correct answer. Choose all correct answers for each question.

**Identify Scenarios That Are Appropriate to Implementation Using Messaging**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following are characteristics of publish/subscribe message model?  A.         Always use a URL to identify publishers.  B.         Subject-based addressing.  C.         Location-independent publishers.  D.         Only synchronous communication between publishers and subscribers is possible. |  |
| **2.** | Which of the following are valid methods for a *TopicSubscriber?*  A.         receive()  B.         receiveNoWait()  C.         receiveWait()  D.         receiveSync() |  |
| **3.** | What are the types of messaging models supported in JMS?  A.         Point-to-point  B.         Send/receive  C.         Transmit/receive  D.         Publish/subscribe |  |
| **4.** | What is a message digest?  A.         A digital fingerprint value that is computed from a message, file, or byte stream  B.         A shortened summary of a message  C.         The subject line of a message  D.         A processing function of the mail server |  |
| **5.** | Which of the following scenarios are suitable for publish/subscribe messaging model?  A.         It is used to receive news stories.  B.         It is used for receiving sales forecasts.  C.         It is used for sending stock prices to traders on the trading floor.  D.         It is used to authorize a user ID and password. |  |
| **6.** | What deliver modes are available in JMS?  A.         *PERSISTENT*  B.         *NON\_PERSISTENT*  C.         *PERMANENT*  D.         *DURABLE* |  |
| **7.** | Which of the following are valid message acknowledgment types?  A.         *AUTO\_ACKNOWLEDGE*  B.         *CLIENT\_ACKNOWLEDGE*  C.         *DUPS\_OK\_ACKNOWLEDGE*  D.         *NO\_ACKNOWLEDGE* |  |
| **8.** | Which of the following are *not* valid message body formats?  A.         *MapMessage*  B.         *ObjectMessage*  C.         *TextMessage*  D.         *StringMessage* |  |
| **9.** | Which of the following are *not* valid JMS objects?  A.         *MessageProducer*  B.         *MessageConsumer*  C.         *MessageViewer*  D.         *MessageSelector* |  |
| **10.** | Which of the following would *not* be used in a client application performing point-to-point messaging?  A.         Topic  B.         *InitialContext*  C.         Queue  D.         Session  **List Benefits of Synchronous and Asynchronous Messaging** |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N1846BD97378-0A32-4DE2-88CA-2E2476959169N1846BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** and **C** are correct. Publish/subscribe messages use subject-based addressing and provide location-independence for publishers.   **A** and **D** are incorrect. URLs are not used to identify publishers. Publish/subscribe is not limited to synchronous communication. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N2386BD97378-0A32-4DE2-88CA-2E2476959169N2386BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **B** are correct. receive() and receiveNoWait() are valid methods for *TopicSubscriber.*   **C** and **D** are incorrect. receiveWait() and receiveSync() are not valid methods. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N3166BD97378-0A32-4DE2-88CA-2E2476959169N3166BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **D** are correct. Point-to-point and publish/subscribe are the messaging models supported in JMS.   **B** and **C** are incorrect. Send/receive and transmit/receive are not valid messaging models. |
| [**4.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479#N3666BD97378-0A32-4DE2-88CA-2E2476959169N3666BD97378-0A32-4DE2-88CA-2E2476959169) | |  **A** is correct. A message digest is a digital fingerprint value that is computed from a message, file, or byte stream.   **B, C**, and **D** are incorrect. These are not definitions of a message digest. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N4126BD97378-0A32-4DE2-88CA-2E2476959169N4126BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **D** is correct. Authorizing user IDs and passwords must use a synchronous process.   **A, B**, and **C** are incorrect. Receiving news stories, sales forecasts, and sending stock prices are suitable for asynchronous messaging. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N4586BD97378-0A32-4DE2-88CA-2E2476959169N4586BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **B** are correct. *PERSISTENT* and *NON\_PERSISTENT* are valid delivery modes.   **C** and **D** are incorrect. *PERMANENT* and *DURABLE* are invalid delivery modes. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N5286BD97378-0A32-4DE2-88CA-2E2476959169N5286BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A, B**, and **C** are correct. *AUTO\_ACKNOWLEDGE*, *CLIENT\_ACKNOWLEDGE*, and *DUPS\_OK\_ACKNOWLEDGE*, are valid.   **D** is incorrect. *NO\_ACKNOWLEDGE* is an invalid message acknowledgment type. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N5906BD97378-0A32-4DE2-88CA-2E2476959169N5906BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **D** is correct. *StringMessage* is not a valid message body format.   **A, B**, and **C** are incorrect. *MapMessage, ObjectMessage, TextMessage* are valid. |
| **[9.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N6496BD97378-0A32-4DE2-88CA-2E2476959169N6496BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **C** is correct. *MessageViewer* is not a valid JMS object.   **A, B**, and **D** are incorrect. *MessageProducer, MessageConsumer, MessageSelector* are valid. |
| **[10.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N7086BD97378-0A32-4DE2-88CA-2E2476959169N7086BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** is correct. Topics are used in publish/subscribe messaging.   **B, C**, and **D** are incorrect. These are valid classes in point-to-point messaging. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following are advantages of asynchronous messaging architectures?  A.         Better use of bandwidth  B.         Supports load balancing  C.         Provides sender with instant response  D.         Scalability |  |
| **2.** | Which of the following statements are true for asynchronous messaging?  A.         It decouples senders and receivers.  B.         It can increase performance.  C.         It is better suited to smaller message sizes.  D.         It only works with blocking calls.  **Identify Scenarios That Are More Appropriate to Implementation Using Asynchronous Messaging, Rather Than Synchronous** |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N7706BD97378-0A32-4DE2-88CA-2E2476959169N7706BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A, B**, and **D** are correct. Asynchronous architectures make better use of bandwidth, support leveling of workloads, and are more scalable.   **C** is incorrect. These architectures do not provide senders with instant response. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N8236BD97378-0A32-4DE2-88CA-2E2476959169N8236BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A, B**, and **C** are correct. Asynchronous messaging decouples senders and receivers, can increase performance, and is better suited to smaller message sizes.   **D** is incorrect. Asynchronous messaging does not work with blocking calls. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which method must be called to receive messages asynchronously?  A.         The receive method  B.         The processMessag method  C.         The readMessage method  D.         The onMessage method |  |
| **2.** | Which of the following are *not* features of asynchronous messaging?  A.         As the volume of traffic increases, it is better able to handle the spike in demand.  B.         A message must be acknowledged before the producer can send another.  C.         It is less affected by failures at the hardware, software, and network levels.  D.         When capacities are exceeded, information is not lost; instead, it is delayed. |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N8886BD97378-0A32-4DE2-88CA-2E2476959169N8886BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **D** is correct. The onMessage method must be implemented to receive messages asynchronously.   **A, B**, and **C** are incorrect. The processMessage and readMessage methods do not exist. The receive method is used for synchronous messaging. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N9646BD97378-0A32-4DE2-88CA-2E2476959169N9646BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** is correct. A message is not acknowledged before a producer can send another in asynchronous messaging.   **A, C**, and **D** are incorrect. These are valid features. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which method must be called to receive messages synchronously?  A.         The receive method  B.         The processMessage method  C.         The readMessage method  D.         The onMessage method |  |
| **2.** | Which of the following cases are better suited to synchronous messaging?  A.         Electronic mail  B.         Credit card authorization  C.         Electronic processing of tax returns  D.         Validation of data entered |  |
| **3.** | Which of the following are features of synchronous messaging?  A.         Both parties must be active to participate.  B.         Messages must be acknowledged before proceeding.  C.         It decouples senders and receivers.  D.         It does not work with blocking calls. |  |
| **4.** | Which of the following are *not* features of synchronous messaging?  A.         Both parties must be active to participate.  B.         It is unaffected by increases in traffic volume.  C.         Message must be acknowledged before proceeding to the next.  D.         Message is queued until it is ready for processing.  **Identify Scenarios That Are Appropriate to Implementation Using Messaging, Enterprise JavaBeans Technology, or Both** |  |

**Answers**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| [**1.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479#N10216BD97378-0A32-4DE2-88CA-2E2476959169N10216BD97378-0A32-4DE2-88CA-2E2476959169) | |  **A** is correct. The receive method must be implemented to receive messages synchronously.   **B, C**, and **D** are incorrect. The processMessage and readMessage methods do not exist. The onMessage method is used for asynchronous messaging. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N10976BD97378-0A32-4DE2-88CA-2E2476959169N10976BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** and **D** are correct. Credit card authorization and validation of data entered are better suited to synchronous messaging because of the need for an instant response.   **A** and **C** are incorrect. Electronic mail and electronic processing of a tax return do not need instant responses. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N11476BD97378-0A32-4DE2-88CA-2E2476959169N11476BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **B** are correct. Both parties must be active to participate, and messages must be acknowledged before proceeding.   **C** and **D** are incorrect. Synchronous messaging does not decouple senders and receivers and only works with blocking calls. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N12016BD97378-0A32-4DE2-88CA-2E2476959169N12016BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **A** and **C** are correct. Synchronous messaging is affected by volume increase, and synchronous messages are not queued.   **B** and **D** are incorrect. These are not valid features for synchronous messaging. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following scenarios are not suitable for publish/subscribe messaging model?  A.         Sending an instant message  B.         Sending an order to another system  C.         Sending news stories to interested parties  D.         Sending a transaction to another system |  |
| **2.** | What method must be implemented to receive messages in a message-driven bean (MDB)?  A.         The receive method  B.         The onMessage method  C.         The readMessage method  D.         The processMessage method |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N12706BD97378-0A32-4DE2-88CA-2E2476959169N12706BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A, B**, and **D** are correct. Sending an instant message, an order to another system, or a transaction to another system is not suitable for the publish/subscribe message model.   **C** is incorrect. Sending news stories to interested parties is suitable for the publish/subscribe message model. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=144201479" \l "N13166BD97378-0A32-4DE2-88CA-2E2476959169N13166BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. The onMessage() method is the correct method.   **A, C**, and **D** are incorrect. These are incorrect methods to receive messages in a message-driven bean (MDB). |

**Chapter 9 - Internatonalization and Localization**

**Certification Summary**

An application must be internationalized, and then it can be localized. The Java APIs provide tools that enable the developer to internationalize a JEE application. A JEE application requires the correct behavior for locale and character-set encoding when accepting input, communicating data between tiers, and presenting data back to the client. An application also needs to report system errors in all tiers in a language appropriate for support personnel.

**TWO-MINUTE DRILL**

Here are some of the key points from each certification objective in [Chapter 9](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=1025#1025).

**State Three Aspects of any Application That Might Need to Be Varied or Customized in Different Deployment Locales**

                    Presentation of text, dates, numbers

                    Labels on presentation components

                    Sounds

                    Colors

                    Images or icons

                    Input and output routines that read and write text files

                    Collation or ordering of data presented in a list

**List Three Features of the Java Programming Language That Can Be Used to Create an Internationalizable/Localizable Application**

                    *java.util.Properties* for obtaining localized values using the same key

                    *java.text.NumberFormat* to handle numbers and currencies

                    *java.text.DateFormat* to handle date and time

                    *java.text.Collator* and *java.text.CollationKey* for ordering data

                    *java.text.MessageFormat, java.util.ResourceBundle*, or *java.util. PropertyResourceBundle* to handle text

                    *java.io.InputStreamReader* and *java.io.OutputStreamWriter* for reading and writing files

                    *java.util.Locale* and contentType and pageEncoding attributes for JSPs

                    *java.util.Locale* and ServletResponse.setContentType() and ServletResponse.setLocale() methods for servlets

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there may be more than one correct answer. Choose all correct answers for each question.

**State Three Aspects of Any Application That Might Need to Be Varied or Customized in Different Deployment Locales**

|  |  |  |
| --- | --- | --- |
| **1.** | Why is Internationalization called I18N?  A.         Because the internationalization is just too long to write in a presentation.  B.         The number of characters between the first and last character is 18.  C.         Because I18N is the encoded UTF version of internationalization.  D.         Because I18N is the default ISO code for internationalization. |  |
| **2.** | Which of the following application aspects can be customized for different locales?  A.         Labels  B.         Reading a text file  C.         Ordering of data presented in a list  D.         Writing a text file  **List Three Features of the Java Programming Language That Can Be Used to Create an Internationalizable/Localizable Application** |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N11796BD97378-0A32-4DE2-88CA-2E2476959169N11796BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** is correct. The number of characters between the first and last character is 18.   **A, C**, and **D** are untrue. |
| [**2.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063#N12296BD97378-0A32-4DE2-88CA-2E2476959169N12296BD97378-0A32-4DE2-88CA-2E2476959169) | |  **A, B, C** and **D**. All four application aspects can be customized for different locales. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | What statement is true with respect to Unicode?  A.         Unicode provides a standard encoding for the character sets of different languages.  B.         Unicode is an encoding that is dependent of the platform, program, or language used to access said data.  C.         Unicode provides a non-unique number for every character.  D.         The Unicode Standard has not been adopted by Microsoft. |  |
| **2.** | What statement is true with respect to a *ResourceBundle?*  A.         A *ResourceBundle* allows you to have various lookup tables based on the locale (language/ country) upon which the server but not the client is running.  B.         Resource bundles support a fourth-level descriptor beyond language/country variants, such that you can have customized messages for people in the northern and people in the southern part of a country, for example.  C.         A *ResourceBundle* is a Hashtable that maps strings to values.  D.         Resource bundles contain locale-specific objects. When your program needs a locale-specific resource, it can load the resource from the resource bundle that is appropriate for the current user's locale. |  |
| **3.** | When using *ResourceBundle*, what is the procedure the system uses to determine which bundle to bind?  A.         The resource bundle lookup searches for classes with various suffixes on the basis of the desired locale and the current default locale as returned by Locale.getDefault(), and the root resource bundle (base class), in the following order: from parent level to lower level.  B.         The resource bundle lookup searches for classes with various suffixes on the basis of the desired locale and the current default locale as returned by Locale.getHelp(), and the root resource bundle (base class), in the following order: from lower level (more specific) to parent level (less specific).  C.         The resource bundle lookup searches for classes with various suffixes on the basis of the desired locale and the current default locale as returned by Locale.getDefault(), and the root resource bundle (base class), in the following order: from lower level (more specific) to parent level (less specific).  D.         The resource bundle lookup searches for classes with various suffixes on the basis of the desired locale and the current default locale as returned by Locale.getDefault(),and the root resource bundle as returned by Locale.getBase(), in the following order: from parent level (less specific) to lower level (more specific). |  |
| **4.** | How do you determine the default character encoding for file operations, JDBC requests, and so on?  A.         You can identify the default file encoding by checking the *Jvm* property named *default.properties*, as follows:  B.              System.out.println(Jvm.getProperty("default.encoding"))  C.         The default encoding used by locale/encoding-sensitive API in the Java libraries is determined by the system property *defaultfile.encoding.*  D.         You can identify the default file encoding by checking the system property named *file.properties*, as follows:  E.              System.out.println(System.getProperty("file.encoding"))  F.         You can set the default file encoding by checking the *Jvm* property named *default.properties*, as follows:  G.              System.setProperty("default.encoding")) |  |
| **5.** | What does UTF stand for?  A.         Universal Technical Frontend  B.         Unicode Transformation Format  C.         United Text Format  D.         Universal Transformation Formula |  |
| **6.** | What internationalization areas does Java not support?  A.         Locales such as country, regional, or area/cultural identifiers  B.         Localized resources by virtue of the *ResourceBundle* series of classes  C.         Formatting for dates, numbers and decimals, and messages  D.         Planetary variants |  |
| **7.** | How can you handle input of different decimal symbols—for example, *343,4* as opposed to *343.4*?  A.         Use *NumberFormat* and its methods format() and parse(). This will handle the default locale for you.  B.         Use *Format* and its methods format() and parse(). This will handle the default locale for you.  C.         Use *DecimalFormat* and its methods format() and parse(). This will handle the default locale for you.  D.         Use *Format* and its methods numberformat() and numberparse(). This will handle the default locale for you. |  |
| **8.** | What is the difference between UTF-8 and UTF-16?  A.         UTF-16 represents every character using two bytes. UTF-8 uses the one-byte ASCII character encodings for all languages except English.  B.         UTF-16 represents every character using two bytes. UTF-8 uses three bytes per character for all languages except English.  C.         UTF-16 represents every character using two bytes. UTF-8 uses the one-byte ASCII character encodings for ASCII characters and represents non-ASCII characters using variable-length encoding.  D.         UTF-16 represents every character using one byte. UTF-8 uses the two-byte ASCII character encodings for ASCII characters and uses three bytes per character for all languages except English. |  |
| **9.** | What is a locale and how is it used for I18N?  A.         A locale is an object that represents and provides information about a specific geographical, political, or cultural region. An operation that requires a locale to perform its task is called locale-sensitive and uses the locale to format information correctly for the user.  B.         A locale is an object that represents and provides information about a specific geographical, political, or cultural region. A globale is an object that represents and provides information about a geographical, political, or cultural region.  C.         A locale is an object that Java calls to present information to the user based upon the locale location of the browser.  D.         A locale is an object that represents the supported geographical, political, or cultural regions. An operation that requires a locale to perform its task is called locale-intensive and uses the locale to display information for the user. |  |
| **10.** | Which of the following are logical fonts in Java?  A.         Sans-serif  B.         Time New Roman  C.         Monospaced  D.         Dialog |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N12786BD97378-0A32-4DE2-88CA-2E2476959169N12786BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. Unicode provides a standard encoding for the character sets of different languages.   **B, C**, and **D** are incorrect. Unicode is an encoding that is independent of the platform, program, or language used to access said data. Unicode provides a unique number for every character. The Unicode Standard has been adopted by Microsoft, as well as by Apple, HP, IBM, JustSystem, Oracle, SAP, Sun, Sybase, Unisys, and others. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N13246BD97378-0A32-4DE2-88CA-2E2476959169N13246BD97378-0A32-4DE2-88CA-2E2476959169)** |  **D** is correct. Resource bundles contain locale-specific objects. When your program needs a locale-specific resource, your program can load it from the resource bundle that is appropriate for the current user's locale.   **A, B**, and **C** are incorrect. A *ResourceBundle* allows you to have various lookup tables based upon what locale (language/country) the client's browser is running in. Resource bundles also support a third-level descriptor beyond language/country, such that you can have customized messages for presentation beyond just language and country. A *ResourceBundle* is analogous to a Hashtable that maps strings to values. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N13886BD97378-0A32-4DE2-88CA-2E2476959169N13886BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. The resource bundle lookup searches for classes with various suffixes on the basis of the desired locale, the current default locale as returned by Locale. getDef ault (), and the root resource bundle (base class), in the following order: from lower level (more specific) to parent level (less specific).   **A, B**, and **D** are incorrect, as they are at odds with the correct answer, **C**. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N14656BD97378-0A32-4DE2-88CA-2E2476959169N14656BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. You can identify the default file encoding by checking the *System* property named *file.properties*, as follows:  System.out.println(System.getProperty("file.encoding"))   **A, B**, and **D** are incorrect, as they are at odds with the correct answer, **C**. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N15506BD97378-0A32-4DE2-88CA-2E2476959169N15506BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. UTF stands for Unicode Transformation Format.   **A, C**, and **D** are incorrect. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N16046BD97378-0A32-4DE2-88CA-2E2476959169N16046BD97378-0A32-4DE2-88CA-2E2476959169)** |  **D** is correct. Calendar and planetary variants.   **A, B**, and **C** are true. Java internationalization supports locales such as country, regional, or area/cultural identifiers, as well as localized resources by virtue of the *ResourceBundle* series of classes and formatting for dates, numbers and decimals, and messages. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N16566BD97378-0A32-4DE2-88CA-2E2476959169N16566BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** and **C** are correct. Use *NumberFormat* or *DecimalFormat* and its methods format() and parse(). This will handle the default locale for you.   **B** and **D** are incorrect, as they are at odds with the correct answers, **A** and **C**. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N17786BD97378-0A32-4DE2-88CA-2E2476959169N17786BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. UTF-16 represents every character using two bytes. UTF-8 uses the one-byte ASCII character encodings for ASCII characters and represents non-ASCII characters using variable-length encoding.   **A, B**, and **D** are incorrect, as they are at odds with the correct answer, **C**. UTF-16 represents every character using two bytes. UTF-8 uses the one-byte ASCII character encodings for ASCII characters and represents non-ASCII characters using variable-length encodings. Note that while UTF-8 can save space for Western languages, which are the most common, it can actually use up to three bytes per character for other languages. |
| **[9.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N18286BD97378-0A32-4DE2-88CA-2E2476959169N18286BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. A locale is an object that represents and provides information about a specific geographical, political, or cultural region. An operation that requires a locale to perform its task is called locale-sensitive and uses the locale to refine and properly format the date and numeric information for the user.   **B, C**, and **D** are incorrect. |
| **[10.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=405442063" \l "N18786BD97378-0A32-4DE2-88CA-2E2476959169N18786BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A, C**, and **D** are correct. Java recognizes five font names—Serif, Sans-serif, Monospaced, Dialog, and DialogInput—along with four font styles—plain, bold, italic, and bolditalic. These are physical fonts but are standard names mapped to physical fonts known to be installed by default on the platform. The mapping is handled by *font.properties.* See the lib/fonts subdirectory of the Java JDK.   **B**. Java does not recognize Times New Roman as platform-standard. |

**Chapter 10 – Security**

**Certification Summary**

The security mechanisms covered in this chapter show how the features of Java Platform, Enterprise Edition provide a robust solution for interoperable and distributed security.

**TWO-MINUTE DRILL**

Here are some of the key points from each certification objective in [Chapter 10](http://skillport.books24x7.com/viewer.asp?bkid=22148&destid=1079#1079).

**Identify Security Restrictions That Java Technology Environments Normally Impose on Applets Running in a Browser**

                    An unsigned applet can make network connections only to the host from which it was downloaded.

                    An unsigned applet can utilize only its own code and is not allowed to load libraries or define native methods.

                    An unsigned applet cannot change thread priority.

                    An unsigned applet cannot execute any native code.

                    An unsigned applet cannot install software.

                    An unsigned applet cannot issue an RMI call to a remote object running on a different server than the applet's.

                    An unsigned applet cannot monitor mouse motion.

                    An unsigned applet cannot programmatically read from or write to the clipboard.

                    An unsigned applet cannot read or write local files on the host that is executing it.

                    An unsigned applet cannot read the following system properties: *java.home, java.class.path, user.name, user.home, and user.dir.*

                    An unsigned applet cannot send e-mail to a server other than the host from which it was downloaded.

                    An unsigned applet cannot start any program on the local host.

                    An unsigned applet cannot talk to a serial or parallel port.

                    An unsigned applet cannot use the System.setOut() or System.setErr() method to redirect the console.

                    An unsigned applet cannot use the Preferences API.

                    An unsigned applet cannot use the Reflection API.

**Given an Architectural System Specification, Identify Appropriate Locations for Implementation of Specified Security Features and Select Suitable Technologies for Implementation of Those Features**

                    Authentication

o                           Authentication method: BASIC, FORM, DIGEST, and CLIENT-CERT

o                           Digital certificates, certificate authorities

o                           Secure Sockets Layer (SSL)

o                           Common Secure Interoperability (CSIv2)

o                           Identity selection: *<run-as>* or *<use-caller-identity>* or *@RunAs* annotation

o                           Security roles, including *@DeclareRoles* and *@SecurityRoles* annotations

                    Authorization

o                           Authorization enforced by the container (declarative), defined in the deployment descriptor and/or within the component itself via annotations

o                           Authorization enforced by the component (programmatic), defined within the application code

**Self Test**

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there may be more than one correct answer. Choose all correct answers for each question.

**Identify Security Restrictions That Java Technology Environments Normally Impose on Applets Running in a Browser**

|  |  |  |
| --- | --- | --- |
| **1.** | Which of the following properties cannot be read by an unsigned applet?  A.         *os.name*  B.         *file.separator*  C.         *java.home*  D.         *java.version* |  |
| **2.** | Which of the following is possible with an unsigned applet?  A.         Connect to the host from which it was downloaded.  B.         Draw a button.  C.         Load a library.  D.         Read from the clipboard. |  |
| **3.** | A company is building an application that allows its sales force to access and process sales information via a web browser. As part of the application, the plan is to develop an applet that will upload data read from a directory on the salesperson's machine. What are your recommendations on the use of applets for this purpose?  A.         The Applet technology is not a viable solution because it is not allowed to access local resources.  B.         The Applet technology is a viable solution provided that it is packaged as a signed Applet and the salesperson explicitly allows (trusts) the Applet to be run in its signed state.  **Given an Architectural System Specification, Identify Appropriate Locations for Implementation of Specified Security Features and Select Suitable Technologies for Implementation of Those Features** |  |

**Answers**

|  |  |  |
| --- | --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N1736BD97378-0A32-4DE2-88CA-2E2476959169N1736BD97378-0A32-4DE2-88CA-2E2476959169)** | |  C is correct. An unsigned applet cannot read the *java.home* system property.   **A, B**, and **D** are incorrect. *os.name, file.separator*, and *java.version* can be read by an unsigned applet. |
| [**2.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262#N2316BD97378-0A32-4DE2-88CA-2E2476959169N2316BD97378-0A32-4DE2-88CA-2E2476959169) | |  B is correct. An unsigned applet is allowed to draw a button.   **A, C**, and **D** are incorrect because they are actions that are not permitted by an unsigned applet. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N2726BD97378-0A32-4DE2-88CA-2E2476959169N2726BD97378-0A32-4DE2-88CA-2E2476959169)** | |  **B** is correct. A signed applet is permitted to access local resources.   **A** is incorrect because the signed type of applet is allowed to access local resources. |
|  |  | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **1.** | What is a message digest?  A.         A digital fingerprint value that is computed from a message, file, or byte stream  B.         A shortened summary of a message  C.         The subject line of a message  D.         A processing function of the mail server |  |
| **2.** | What method can be used to help programmatically determine the caller's identity within enterprise bean code?  A.         getIdentity()  B.         getCallerPrincipal()  C.         getCallerIdentity()  D.         getUserId() |  |
| **3.** | Within enterprise bean code, what method can be used to determine whether the caller is in a security role and authorized to execute the method?  A.         inRole()  B.         isAuthorized()  C.         isCallerInRole()  D.         isValid() |  |
| **4.** | What method can be used to help programmatically determine the caller's identity within a JSP?  A.         getUserPrincipal()  B.         getPrincipal()  C.         getUser()  D.         getIdentity() |  |
| **5.** | Within a JSP, what method can be used to determine whether the caller is programmatically authorized to execute its functionality?  A.         inRole()  B.         okToExecute()  C.         isValid()  D.         isUserInRole() |  |
| **6.** | What role maps the declarative authorization rules to the target environment?  A.         Deployer  B.         Component provider  C.         Application assembler  D.         Authorizer |  |
| **7.** | What role maps the programmatic authorization rules to the target environment?  A.         Application assembler  B.         Component provider  C.         Coder  D.         Deployer |  |
| **8.** | For Enterprise JavaBeans (EJBs), where are the declarative authorization rules defined?  A.         Application properties  B.         EJB deployment descriptor  C.         JNDI  D.         Enterprise bean code |  |
| **9.** | For web resources, where are the declarative authorization rules defined?  A.         EJB deployment descriptor  B.         Application deployment descriptor  C.         In the web resource  D.         JMS |  |
| **10.** | For Enterprise JavaBeans (EJBs), where are the programmatic authorization rules implemented?  A.         JNDI  B.         EJB deployment descriptor  C.         In the enterprise bean code  D.         JMS |  |
| **11.** | For web resources, where are the programmatic authorization rules defined?  A.         Java Security Manager  B.         Security policy file  C.         JNDI  D.         Within the JSP or servlet |  |
| **12.** | Which of the following is not a valid authentication method (auth-method)?  A.         FORM  B.         HTTP  C.         DIGEST  D.         CLIENT-CERT |  |

**Answers**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[1.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N3206BD97378-0A32-4DE2-88CA-2E2476959169N3206BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. A message digest is a digital fingerprint value that is computed from a message, file, or byte stream.   **B, C**, and **D** are not definitions of a message digest. |
| **[2.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N3696BD97378-0A32-4DE2-88CA-2E2476959169N3696BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. The getCallerPrincipal() method returns the principal object. The getName() method can then be used to determine the caller's name (identity) from within enterprise bean code.   **A, C**, and **D** are incorrect because getCallerIdentity() is a deprecated method, and getIdentity() and getUserId() are not valid methods. |
| **[3.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N4426BD97378-0A32-4DE2-88CA-2E2476959169N4426BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. The isCallerInRole() method can be used to determine if the caller is within the specified role and therefore able to execute the EJB functionality.   **A, B**, and **D** are incorrect because inRole(), isAuthorized(), and isValid() are not valid methods. |
| **[4.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N5116BD97378-0A32-4DE2-88CA-2E2476959169N5116BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. The getUserPrincipal() method returns the principal object. The getName() method can then be used to determine the caller's name (identity) from within a JSP.   **B, C**, and **D** are incorrect because getPrincipal(), getUser(), and getIdentity() are not valid methods. |
| **[5.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N5846BD97378-0A32-4DE2-88CA-2E2476959169N5846BD97378-0A32-4DE2-88CA-2E2476959169)** |  **D** is correct. The isUserInRole() method can be used to determine whether the caller is within the specified role and therefore able to execute the JSP functionality.   **A, B**, and **C** are incorrect because inRole(), okToExecute(), and isValid()are not valid methods. |
| **[6.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N6576BD97378-0A32-4DE2-88CA-2E2476959169N6576BD97378-0A32-4DE2-88CA-2E2476959169)** |  **A** is correct. The deployer is responsible for mapping declarative authorization rules to the target environment.   **B, C**, and **D** are incorrect because the Component Provider and Application Assembler are not responsible for providing this mapping. Authorizer is not a JEE role, so it can't be correct either. |
| **[7.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N7026BD97378-0A32-4DE2-88CA-2E2476959169N7026BD97378-0A32-4DE2-88CA-2E2476959169)** |  **D** is correct. The deployer is responsible for mapping programmatic authorization rules to the target environment.   **A, B**, and **C** are incorrect because the application assembler and component provider are not responsible for providing this mapping. Coder is not a JEE role, so it can't be correct either. |
| **[8.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N7516BD97378-0A32-4DE2-88CA-2E2476959169N7516BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. The declarative authorization rules for EJBs are defined within the EJB deployment descriptor.   **A, C**, and **D** are incorrect because EJB authorization rules are not declaratively defined in application properties, JNDI, or Enterprise bean code. |
| **[9.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N7966BD97378-0A32-4DE2-88CA-2E2476959169N7966BD97378-0A32-4DE2-88CA-2E2476959169)** |  **B** is correct. The declarative authorization rules for web resources are defined within the application deployment descriptor.   **A, C**, and **D** are incorrect because authorization rules for web resources are not declaratively defined in the EJB deployment descriptor, in the web resource, or in JMS. |
| **[10.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N8416BD97378-0A32-4DE2-88CA-2E2476959169N8416BD97378-0A32-4DE2-88CA-2E2476959169)** |  **C** is correct. The programmatic authorization rules for EJBs are implemented within the enterprise bean code.   **A, B**, and **D** are incorrect because EJB authorization rules are not programmatically implemented in JNDI, the EJB deployment descriptor, or JMS. |
| **[11.](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262" \l "N8866BD97378-0A32-4DE2-88CA-2E2476959169N8866BD97378-0A32-4DE2-88CA-2E2476959169)** |  **D** is correct. The programmatic authorization rules for web resources are implemented within the JSP or servlet.   **A, B**, and **C** are incorrect because authorization rules for web resources are not programmatically implemented in the Java Security Manager, security policy file, or JNDI. |
| [**12.**](http://skillport.books24x7.com/book/id_22148/viewer_r.asp?bookid=22148&chunkid=916258262#N9316BD97378-0A32-4DE2-88CA-2E2476959169N9316BD97378-0A32-4DE2-88CA-2E2476959169) |  **B** is correct. HTTP is not a valid authentication method.   **A, C**, and **D** are incorrect. FORM, DIGEST, and CLIENT-CERT are valid authentication methods. |

**SCEA part 1 mock questions and Ans**

These are few questions from SCEA 1.5 exam (that i found on the net and repeated in the exam):  
1.What are the three primary roles in a web service interaction? (Choose three.)  
A.Broker  
B.Facade  
C.Provider  
D.Decorator  
E.Requestor  
F.Interceptor  
Correct:A C E  
  
2.A stock trading company is writing a new application for stock market forecasting. A significantportion of the work required by the business logic involves navigating through the persistentobject model. As lead architect on this project, you have chosen JPA over EJB2 entity beans toimplement these persistent objects. You have done this to maximize performance whennavigating through the model. Why does JPA offer better performance for this task?  
A.JPA guarantees referential integrity at the object level.  
B.JPA allows the application to specify lazy or eager retrievals.  
C.JPA simplifies the source code that implements the object model.  
D.The guaranteed referential integrity in EJB2 entity beans is expensive.  
Correct:B  
  
3.A developer creates a Java web service to be used by consumers in an SOA. This SOA uses aUDDI service registry. How can the developer make the new service available to consumers?

A.deploy to the registry using JAXR  
B.publish to the registry using JAXR  
C.query the registry using JAXRPC  
D.target the registry using JAXRPC  
Correct:B  
  
4.With the release of a new product line, there has been a significant increase in the volume oftransactions on your web site. You need to scale your application and manage session failover.What is the best option for scalability?  
A.add additional web servers and application servers  
B.introduce a High Availability pair and utilize sticky sessions  
C.add additional application servers and implement DNS round robin  
D.add additional application servers and use clustered HttpSession  
Correct:D  
  
5.You are asked to architect an SOA solution that leverages Java web services. The architectureneeds to be flexible and allow for the SOAP 1.1, SOAP 1.2, and REST implementations. Which JavaEE technology should you use?  
A.JAXP  
B.JAXB  
C.JAXWS  
D.JAXRPC  
Correct:C

6. You are architecting an online ordering application with these requirements: Users access the system over the Internet using HTML. An email message is sent to the user confirming the order.  
Users must log in and are validated using LDAP. The product catalog is stored in a relationaldatabase. All orders are logged to the internal fulfillment system. Orders must not be lost.  
  
WhichJava EE technology should be used to send orders to the fulfillment system?  
A.JNDI  
B.JMS  
C.JAXWS  
D.RMI IIOP  
  
Correct:B  
  
7.An online sporting goods store's web application uses HTTPSession to store shopping carts.When the application is initially deployed, the business plan predicts only a few customers willaccess the site. Over time, the store projects a steady increase in volume. The deployment plancalls for a single web container in the initial deployment. As demand increases, the plan calls formultiple web containers on separate hardware with clustered HTTPSession objects. Which twoprinciples will help the application meet the requirements and optimize performance? (Choosetwo.)  
  
A.The application should store as much as possible in HTTPSession objects.  
B.The application should NOT make frequent updates to HTTPSession objects.  
C.The application should make coarsegrainedupdates to HTTPSession objects.  
D.The application should create new HTTPSession objects instead of updating existing objects.  
  
Correct:B C  
  
8.You are writing a utility that searches for existing web services provided by large companiesthrough UDDI. Your web site allows the user to input search criteria using eventdriven,statemanagedGUI screens, performs the search, and displays them in a formatted HTML page.Which technologies would you use for this application?  
  
A.JSP and JAXB  
B.JSF and JAXR  
C.JSF and JAXWS  
D.JSP and JAXWS  
  
Correct:B  
  
9.A company has a web service that provides the most recent price for stocks, mutual funds, andcommodities. The company has the only web service that allows a person to check prices on allthree financial assets with one call. Its system does not store this information but sends individualcalls to each of the primary vendors for an asset and then aggregates the response to therequester. The company has committed to support a nonfunctionalrequirement (NFR) forperformance that states it must process all requests within three seconds and each of the threevendors is obligated to support the NFR as dictated by the company.  
  
Where, in the message flow,is it appropriate to measure whether all the NFRs are met?  
A.when a request is received and a response is sent to the requester  
B.when a request is received, first call to vendor, last response from vendors, response is sent to arequester  
C.when a requester sends a request, the request is received, each call to vendor, each response fromvendor, requester receives response  
D.when a request is received, each call to vendor, each response from a vendor, a response is sent to arequester  
  
Correct:D  
  
10.A Java web component, EJB component, or another web service can be a client to a webservice. Which Java API can the client use to access the web service through a Service EndpointInterface?  
  
A.JAXB  
B.JAXR  
C.JDBC  
D.JAXWS  
  
Correct:D  
  
11.Which three are parts of a SOAP message? (Choose three.)  
  
A.SOAP body  
B.SOAP endpoint  
C.SOAP headers  
D.SOAP handlers  
E.SOAP attachments  
  
Correct:A C E  
  
12.You are integrating with a single legacy Enterprise Information System. You are interested inthe transaction management capabilities of the Java Connector Architecture. This new systemneeds the capability to invoke multiple operations against this single legacy system. Theseoperations succeed together or fail together as a group. To which minimum level of transactionmanagement are you going to set your resource adapter?  
  
A.No transaction  
B.Local transaction  
C.Distributed transaction  
D.Container Managed transaction  
  
Correct:B  
  
13.What is an advantage of XML over HTTP, as compared to SOAP over HTTP, for web services?  
  
A.guaranteed delivery  
B.more security options  
C.smaller message size  
D.strongly typed parameters  
  
Correct:C  
  
14.An application needs to invoke a service on a remote system over a low latency connection,and then wait for a response. Which two are best for this type of invocation? (Choose two.)  
  
A.JMS topic  
B.JMS queue  
C.RMI over IIOP  
D.synchronous web service  
E.asynchronous web service  
  
Correct:C D  
  
15.Your new architecture needs to access the business logic of an Enterprise Information Solution(EIS). What are three benefits of using the Java Connector Architecture to connect to EIS insteadof implementing a proprietary solution? (Choose three.)  
  
A.security  
B.performance  
C.loose coupling  
D.connection pooling  
E.Common Client Interface  
  
Correct:A D E  
  
16.Your web application requires access to several different services, so you created a ServiceLocator class to be used by the UI developers on the team. New services with different interfacesare occasionally added. Unfortunately, the caching benefits of the Service Locator class are NOTbeing realized because a new instance of this class is being created in every backing beanmethod that requires a service.  
  
Which pattern should you apply to eliminate this problem?  
A.Bridge  
B.Singleton  
C.Prototype  
D.Factory Method  
E.Business Delegate  
  
Correct:B  
  
17.What are two benefits of using the Value List Handler pattern? (Choose two.)  
  
A.improves network performance  
B.facilitates exposing existing services  
C.provides an alternative to potentially inefficient EJB finders  
D.facilitates postprocessingacross heterogeneous requests  
E.provides a mechanism to support shared elements of composite views  
  
Correct:A C  
  
18.What are two capabilities of the Abstract Factory pattern? (Choose two.)  
  
A.creates wholeparthierarchies  
B.creates families of related objects  
C.enforces dependencies between concrete classes  
D.specifies the types of objects to create using a sample instance  
E.separates the construction of a complex object from its representation  
  
Correct:B C  
  
19.A teenage fashion web site, includes a set of pages for displaying and browsing their catalog,as well as pages for making fashion suggestions that also display tables of catalog entries.Currently, the JSP code uses scriptlets that perform database SELECT statements and format theresults in HTML tables. You have been hired to help reduce the maintenance overhead when eitherthe look is modified or the database schema changes.  
  
Which two patterns, used together, do youapply to reduce this maintenance overhead? (Choose two.)  
A.View Helper  
B.Front Controller  
C.Composite View  
D.Data Access Object  
  
Correct:A D  
  
20.A new security feature has been requested for an existing web application with the followingrequirements: All requests must be logged to a secure database. Each request must betimestampedwith the start and completion times. Each request must contain the IP address ofthe client that made the request.  
  
Which pattern is most applicable for this new feature?

A.Strategy  
B.Front Controller  
C.Abstract Factory  
D.Intercepting Filter  
E.Model View Controller  
  
Correct:D  
  
21.Which two are benefits of using the Intercepting Filter pattern? (Choose two.)  
  
A.allows the recombination of filters  
B.provides efficient data sharing between filters  
C.facilitates creating a generic command interface  
D.facilitates common processing across heterogeneous requests  
E.helps to minimize coupling between the presentation and business tiers  
  
Correct:A D  
  
22.You are building a subsystem that has several complex components, but you want to hide thatcomplexity from the client code.  
  
Which pattern can you apply to hide this complexity?

A.Proxy  
B.Bridge  
C.Adaptor  
D.Facade  
E.Decorator  
  
Correct:D  
  
23.Some media players use a virtual proxy for image loading.  
  
What is the benefit of using a virtualproxy?  
A.It controls access to the original object.  
B.It defers creation of expensive objects until necessary.  
C.It provides a local representation for an object in a different address space.  
D.It is a replacement for a bare pointer that performs additional actions when an object is accessed.  
  
Correct:B  
  
24.Your company's web site is supported with a cluster of loadbalancedweb servers and adatabase server. To reduce expenses, your company must replace your current cluster of webservers with a single web server. All servers under consideration have the same specification.  
  
Which three items will be negatively impacted by this rearchitecture?(Choose three.)  
A.security  
B.reliability  
C.scalability  
D.availability  
E.manageability  
F.maintainability  
  
Correct:B C D  
  
25.A company manufactures widgets for sale to distributors. Distributors call this company when they want to order more widgets. The company wants the distributors to send orders using XML documents over the Internet to reduce the number of data entry personnel needed. It has no control over the distributor's technologies. The company does not want the orders to impact the performance of the other users. You have been assigned the task of designing the new API.  
  
Which approach do you take?  
A.design the API as a JMS queue  
B.design the API as an RMI interface  
C.design the API as a synchronous web service  
D.design the API as an asynchronous web service  
  
Correct:D  
  
26.You have been tasked with improving the availability of an existing threetierapplication. Whatis your first step in evaluating what changes should be made to the architecture to achieve thegoal?  
  
A.monitor network traffic between tiers  
B.separate presentation from business logic  
C.identify and document all single points of failure  
D.cluster the presentation tier without session replication  
  
Correct:C  
  
27.Which nonfunctionalrequirement is a disadvantage for a twotierarchitecture?  
  
A.security  
B.reliability  
C.availability  
D.manageability  
  
Correct:D  
  
28.A travel company is designing an application to allow customers to browse for information onany flights operating domestically and to place new reservations on any of those flights. Thecompany makes the following assumptions: significant read volume, in terms of operations thecustomers will perform significant overlap, in the search criteria of customers simple processingof each customer browse/update request .  
  
What advice can you give this company?

A.use a twotierarchitecture (rich client directly accessing the database) because running copies of thebusiness logic in each client provides significant advantages in terms of processing time per request

B.use a threetierarchitecture (thin client >application server >database) because executing businesslogic remotely on a central location results in better performance per request

C.use a threetierarchitecture (thin client >application server >database) because the shared businessserver allows them to cache information with high likelihood of cache hits, which reduces the load on thedatabase

D.use a twotierarchitecture (rich client directly accessing the database) because each client can operateon its own business objects, independently of others, which provides significant advantages from reducedlatency due to synchronization  
  
Correct:C  
  
29.A company provides call center support for corporations worldwide.Each agent in the callcenter can be assigned to multiple call center customers. All of the company's customers useWindowsbased user interfaces and it has just signed a new customer that uses a Java EEbackend and wants a rich interface. The company is developing a user interface for the newcustomer with the following requirements: Customer service representatives (CSRs) must be ableto work with minimal training on the application. CSRs must be able to switch between call centersystems quickly. Screens must have a Windows "look and feel." 2000 agents spread across fourlocations must be able to use the system.  
  
What advice would you give this company on the userinterface (UI)?  
A.write the UI using JSP and JSTL  
B.write the UI using JSPs with embedded scriptlets  
C.write the UI using Ajax, accessing servlets directly  
D.write the UI using Java Swing and distribute using JNLP  
  
Correct:D  
  
30.A travel company decides to rearchitecttheir twotierapplication (where each client ran itsown copy of the application against a shared database) and hires you as their lead architect. Yousuggest they rearchitecttheir application as a browserbased,threetiersolution: presentation,business logic, and persistence. You also suggest they deploy each of the three tiers on its owncomputer.  
  
Why is the three tier solution more scalable than the twotiersolution?  
A.Every client runs its own GUI application. Clients do not compete for resources for presentationpurposes.  
B.Clients share the same business logic tier. Clientspecificobjects can be stored centrally, optimizingaccess.  
C.Every client shares the same business logic tier. Each client competes with each other for resources onthat JVM.  
D.Clients share the same business logic tier. Duplicate effort can be avoided by sharing objects, reducingthe load on the database.

31. Which two can be used to maintain convesational state ? (Choose two.)  
  
A. Entity beans  
B. Http session  
C. Stateful session beans  
D. message-driven beans  
E. stateless session beans  
  
Correct B,C  
  
32. You are the architect of a project that will provide an external, low latency, scalable, and highly- available sevice for handling string transactions. Each request consists of a short string ID and laguage key, limited to "EN", "FR", "ES", "DE", and "JP". Each response is a simple unicode string averaging 256 bytes in size, and there will be no more than 50,000 records for each language. All the records have aleady been translated and changes to the records will be rare.  
  
What should you do to ensure that your service wll scale and perform well as new clients are added ?  
A - store all the records in an LDAP server and use JNDI to access them from the web tier  
B - deploy a standard 3-tier solution that is supported by a fast and reliable relational database  
C - deploy a single service on many servers in the web tier, each storing all the records in memory  
D - store all of the records in a network attached file system so they can be served directly from the file system  
  
Correct: C  
  
33.Tou are the architect of a web application that uses JSF as a presentation tier for business processes coded as stateless session beans. When you add new code to the stateless session beans to address new accounting requirements, without changing the interface, you discover that the new business processes are being ignored by some of the JSF components.  
  
Which might be the cause of this problem?  
A - The presentation tier is relying on validation logic in the business tier.  
B - The broswer is caching out-of-date versions of the JSF components.  
C - The business processes are not rigorously encpsulated in the session beans.  
D - The new session beans have been deployed incorrectly, and proper deployment will resolve the problem.  
  
Correct: C  
  
34. You have refactored your legacy Java application into a three-tiered architecture. Your security audit group is concerned that your architecture may be vulnerable to security threats in the separate tiers. Which two methods can you use to reduce those threats? (Choose two).  
  
A - programmatic security in the EJB entities  
B - interecepting filters between the view and the controller  
C - intercepting filters between the controller and the model  
D - role-based security for the EJBs in the deployement descriptor  
  
Correct: B,D  
  
35. Drag and drop the question. Drag the items to the proper locations.

Correct:

36. The current architecture of a fashion website consists of one web server, three application servers, and a database. You, as the lead architect, recommend adding more web servers. What are two valid justifications for new architecture? (Choose two.)

A. New web servers will decrease latency for I/O-bound requests.  
B. Adding multiple web servers will have a positive impact on scalability.  
C. Adding new web servers will increase the overall availability of the web site.  
D. New web servers will increase the number of user accounts that can be supported.

Correct: B,C

37. Which three statements are true about delegation as on OO design technique? (Choose three.)

A. It is applied to a system only at compile time.  
B. It is an essential element of the State pattern.  
C. It is an essential element of the Singleton pattern.  
D. It allows you to replace inheritance with composition.  
E. In Java technology, it is always implemented through the use of generics.  
F. It always requires that at least two objects are involved in handling a request.

Correct: B, D, F

38. Which two statements are true about the Flyweight pattern? (Choose two.)

A. It allows a single instance of a class to virtually represent many instances.  
B. When used approximately it can reduce memory demands on your servers.  
C. It allows for many instances of a class to be controlled by a single instance.  
D. It allows many tightly related, homogeneous objects to each maintain their own state.

Correct: A, B

39. Which two techniques can used to provide polymorphic behavior? (Choose two.)

A. extending a class and adding a new method  
B. implementing two interfaces in the same class  
C. extending a class and overriding an existing method  
D. implementing an interface with several different classes

Correct: C, D

40. As a project architect, you are selecting technologies for a complex, n-tier web application's virtual platform. At this stage in the project, which two technologies should be of primary consideration? (choose two.)

A. RMI  
B. Linux  
C. JDBC  
D. Firefox  
E. Tomcat

Correct: A, C

41. Which design pattern is usefull for hiding the construction and implementation details of an object?  
  
A. Flyweight  
B. Singleton  
C. Abstract Factory  
D. Chain of Command  
Correct Answer: C  
  
42. Some media players use a virtual proxy for image loading. What is the benefit of using a virtual proxy?  
  
A. It controls access to the original object.  
B. It defers creation of expensive objects until necessary.  
C. It provides a local representation for an object in a different address space.  
D. It is a replacement for a bare pointer that performs additional actions when an object is accessed.  
  
Correct Answer: B  
  
43. What are two capabilities of the Decorator pattern ? (Choose two.)  
  
A. Provides a unified interafe to a subsystem  
B. Converts the interface of a class into another interface  
C. Is used when the base class is unavailable for subclassing  
D. Promotes loose coupling by keeping objects from referring to each other  
E. Modifies responsibilities to individual objects dynamically and transparently  
  
Correct Answers: C, E  
  
44. You are building a web application that must integrate to a content management system(CMS). Your company currently has a homegrown CMS, but management is considering purchasing a new CMS.  
Unfortunately, you have little confidence that their lates choice, BigCMS, is likely to be the final decision. After analyzing the interface to BigCMS, you find that its interface is different from the homegrown CMS. Furthermore, you suspect that any other third-party CMS will have yet another interface.  
  
What is the simplest pattern that would isolate your web application from the interface of the CMS tool ?  
A. Proxy  
B. Bridge  
C. Adapter  
D. Service Locator  
E. Business Delegate  
  
Correct Answer: C  
  
45. What is a benefit of using the Transfer Object pattern ?  
  
A. Reduces requests across the network  
B. Avoids the overhead of using EJB finder methods for large searches  
C. Separates the business state and related behavior from the rest of the appliation  
D. Implements parent-child relationships efficiently when imlementing Business Objects  
  
Correct Answer: A  
  
46. What are two benefits of using the Value List Handler pattern? (Choose two.)  
  
A. Improves network performence  
B. Facilitates exposing existing services  
C. Provides an alternative to petentially inefficient EJB finders.  
D. Facilitates post-processing across heterogeneous requsts  
E. Provides a mechanism to support shared eleemnts of composite views  
  
Correct Answers: A, C  
  
47. A company created its own MVC-like framework in the years before struts and JSF. Unfortunately, the company's Front Controller has become bloated with too many features including fine-grained authorization, view dispatching, and business logic invocation. Which three patterns could be applied to reduce the complexity of the Front Controller? (choose three.)  
  
A. Mediator  
B. Command  
C. View Helper  
D. Intercepting Filter  
E. Composite View.  
F. Application controller  
  
Correct Answers: B, D, F  
  
48. What are two advantages of the Business Delegate pattern? (Choose two.)  
  
A. Increases the scalability of remote services  
B. Decouples presentation logc from business logic  
C. Avoids unnecessary invocation of remote services  
D. Hides underlying communication details of the service  
E. Enables transparent presistent storage of the business entity  
  
Correct Answers: C, D  
  
49. What are two advantages of a thin client, three tier architecture over a thick client, two-tier architecture ? (choose two.)  
  
A. It is more secure.  
B. It is more reliable  
C. It is easier to maintain.  
D. It makes it easier to manage the application deployment.  
  
Correct Answers: C, D  
  
50. A teenage fashion website, has a multi-tier web application with 103 web servers, 12 middle-tier servers, and a large RDBMS server with more than enough capacity to support peak loads. You are the architect of the system, and you are concerned about reliability of the web application. Which change could you make to improve reliability ?  
  
A. Add additional web servers.  
B. Add additional database servers  
C. Add additional middle-tier servers  
D. Reduce the number of web servers  
E. Reduce the number of middle-tier servers.  
  
Correct Answer: B  
  
51. Which is NOT a valid reason to separate presentation from business logic ?  
  
A. Improved scalability  
B. Improved performence  
C. Separation of cencerns  
D. Improved maintainability  
  
Correct Answer: B  
  
52. A company is considering re-architecting their application from a two-tier to a three-tier architecture. To see what impact the new architecture would have on their non-funtional requirements (NFRs), they created a prototype.  
When they tested the prototype based on their three-tier architecture they noticed, contrary to expectations, that the prototype was less scalable than the original two-tier solution.  
Which two statements explain the result? (Choose two.)  
  
A. Clients end up competing for CPU cycles on the common business logic tier.  
B. Clients end up competing for resources on the back-end database used by the application.  
C. Clients did NOT share interest in any domain objects, but the business tier spent too much time coordinating among clients anyway.  
D. Clients did NOT share interest in any domain objects and the business tier ran out of memory to represent all domain objects required by the clients.  
  
Correct Answers: A, D  
  
53. Which two statements are true only when implementing rich client applications, and NOT when implementing web browser-based clients that support only HTML? (Choose two.)

A. Information can be sent to the client by the server, without client polling  
B. Information can be encrypted prior to delivering it to the client and decrypted when received to the client.  
C Information can be compressed prior to delivering it to the client and decompressed when received ro the client.  
D. Information can be delivered to the client incrementally, without requiring that the server deliver all the information to be presented to the client on each update.  
  
Correct Answers: A, D  
  
54. A bank designed its first-generation web-based banking system aroung a Java technology rich client application that interacts with server-side service objects implemented as stateful session beans in a portable Java EE application. For their second-generation system, the company wants to open the architecture to other types of clients. The company is considering exposing its existing stateful session bean service as a web service. Which statement is true ?  
  
A. Session beans cannot be exposed as web services.  
B Stateful session beans cannot be exposed as web services.  
C. Stateful session beans are automatically exposed as web services.  
D. Stateful session beans annotated with @WebService are exposed as web services.  
  
Correct Answer: B  
  
55. Brokers at a firm currently use a two-tier application to execute stock transactions on behalf the their customers. Business componens within the application enforce a number of complex business rules that ensure that stock transactions are executed properly.  
Management has decided that clients should be able to execute their own transactions to streamline operations. Management also wants clients to run the same existing two-tier applications from their home computers. They have hired you to advise them on how to proceed to unsure that no illegal stock transactions are executed once the application is available directly to clients.  
  
Which two recommendations should you give to this brokerage firm? (Choose two.)

A. The code already checks for correct execution so they can deliver the application to clients "as is"  
B. Checks for correctness should be rewritten as database constraints because the application running on the client might be modified.  
C. The application should be re-architected as a thee-tier solution. That way, validation checks can be moved to a server-side business tier, which remains trustworthy.  
D. The application should be obfuscated before it is delivered to the client. That way, clients cannot modify it. Therefore, the validation checks currently implemented will remain trustworthy.  
  
Correct Answers: B, C  
  
56. A travel company re-architected its application from a two-tier to a three-tier architecture. To see what impact the new architecture would have on its non-functional requirements(NFR), the company intends to build a prototype based on the new architecture. The company compared the MFR metrics associated with the new prototype against the metrics from their original two-tier solution. What is an advantage of the original two-tier solution?  
  
A. It has better availability because it has fewer single point of failure.  
B. It has better manageability because each client has its own copy of the application.  
C. It has better performence because each client must use is own set of domain objects.  
D. It has better scalability because each client can access the database independently of other clients.

57. The Java system you are enhancing needs an integration point to an external system to gain access to data stored in a database. The application uses an open database connectivity data source to access data. What do you use to connect to the database ?  
  
A. XML over HTTP  
B. RPC-style SOAP  
C. RMI-IIOP and EJB  
D. JDBC-ODBC bridge driver  
  
Correct Answer: D  
  
58. Java Connector Architecture (JCA) as a technology solution addresses certain needs for your Java applications. What is the best description of a JCA solution. ?  
  
A. asynchronuous message-based interfaces  
B. integration of slow responders in a loosely-coupled way  
C. access tightly-coupled business logic of legacy systems  
D. integration of systems/components and guaranteing message delivery  
  
Correct Answer: C  
  
59. Your client is interested in the benefits or integration with an external system using RMI-IIOP, RMI-JRMP, and CORBS for external inegration. What should you tell your client?  
  
A. An RMI-JRMP client can call a CORBA server.  
B. A CORBA client can call an RMI\_JRMP server.  
C. An RMI-JRMP client can call an RMI\_IIOP server.  
D. A CORBA client CANNOT call an RMI-IIOP server.  
  
Correct Answer: C  
  
60. Your online e-commerce application has a message driven bean (MDB) that calls an email server. Which statement about invoking the MDB is true?  
  
A. The client can access the MDB directly.  
B. The client accesses the MDB using an interface.  
C. A message driven bean is simply a JMS message provider.  
D. A JMS message is sent to the destination to which the MDB is listening.  
  
Correct Answer: A  
  
61. SOAP was selected as an integration technology for the flexibility of messaging styles it supports (in particular, how an XML payload can be presented in a SOAAP message). Which message style statement is correct?  
  
A. The body of an RPC-style SOAP message cannot be a literal message.  
B. The XML payload for a document-style message cannot be an encoded message.  
C. The XML payload for RPC and document-style messages guarantee XML payload delivery.  
D. An RPC-style message has the XML payload wrapped inside an operation element in a SOAP body  
  
Correct Answer: D  
  
62. As part or you Java application, you are required to integrate with an external system that has a Java web service. Tha java web service is using synchronous communication and exposes several methods with varying method signatures. Which technology do you use for this solution?  
  
A. JMS  
B. SOAP  
C. RMI-IIOP  
D. RMI-JRMP  
  
Correct Answer: B  
  
63. Your company is going through an extensive security audit and it has been identified that your internet-facing web site is vulnerable to SQL injection from authenticated users. Which two are appropriate for mitigating this threar? (Choose two.)  
  
A. Using security roles in the deployement descriptor  
B. In stored procedures called with prepared statements  
C. Adding an intercepting validation filter to your syste,  
D. Requiring SSL in the deployement descriptor transport guarantee.  
  
Correct Answers: B, C.  
  
64. Your competitor is trying to crash your web site by using various Denial of Service attacks. Which two flaws should you protect against for this specific threat? (Choose two.)  
  
A. SQL injection  
B. buffer overflow  
C. Man in the middle  
D. session hijacking  
E. weak password exploits  
  
Correct Answers: A, B  
  
65. Which is an appropriate technique for minimizing the consequences of a successful attack?  
  
A. Input validation  
B. Principle of least privilege  
C. Encryption os wire transmissions  
D. Use of strong/two-factor authentication  
  
Correct Answer: B  
  
66. What is the appropriate location to configure a JSP based application to require secure communication between a broswer and particular resources?  
  
A. In the application code  
B. In the business-tier code  
C. In the broswer configuration  
D. In the deployement descriptor  
E. In the web server configuration  
  
Correct Answer: D.  
  
67. Service methodA(), implemented in a session bean, performs a highly sensitive operation. This operation must be available in limited ways to low privilege users to support a low sensitivity operationB()  
  
Which approach addresses the requirements most securely?

A. mark the methodA() as accessible to all necessary roles  
B. mark the methodA() as accessible to all appropriate roles, and use the programmatic security model to impose the necessary additional restrictions  
C. mark the methodA() as accessible to all appropriate roles, and use the deployment desciptor to indicate the conditions under which each role can invoke the high priority method  
D. mark methodA() as accessible only to a special role, then use a run-as element to invoke A() from B(); before making the call to A(), B() checks that conditions are appropriate for the call  
  
Correct Answer: D.

**SCEA1.5 - Architecture Concepts**

**Common Architecture Concepts:**  
**Scalability:**  
Scalability is the ability to economically support the required quality of service as the load increases.  
Two types: Vertical and Horizontal  
  
Vertical: Achieved by adding capacity (memory, CPUs, etc.) to existing servers.  
Requires few to no changes to the architecture of a system.  
Increases: Capacity, Manageability  
Decreases: Reliability, Availability (single failure is more likely to lead to system failure)  
Vertical scalability is usually cheaper than horizontal scalability.  
J2EE supports vertical scaling because of automatic lifecycle management. Adding more capacity to a server allows it to manage more components (EJBs, etc.).  
  
Horizontal:Achieved by adding servers to the system.Increases the complexity of the system architecture.Increases: Reliability, Availability, Capacity, Performance (depends on load balancing), FlexibilityDecreases: Manageability (more elements in the physical architecture)  
J2EE supports horiz. scaling because the container and server handle clustering and load-balancing.  
Availability and reliability are obtained through scalability.  
Scalability affects capacity. The more scalable the system is the more capacity it can support. This must be traded-off against the complexity & manageability costs.  
  
**Flexibility:**  
Flexibility is the ability to change the architecture to meet new requirements in a cost-efficient manner.  
A flexible system should be more maintainable in the face of changes to the environment and/or to the application itself.  
Flexibility improves: Availability, Reliability, ScalabilityFlexibility slightly decreases: Performance, Manageability  
Flexibility is achieved via code that can be distributed across servers with load balancing that prevents one system from being overburdened. The use of a multi-tier architecture also helps achieve flexibility.  
  
**Reliability:**  
The ability to ensure the integrity and consistency of the application and all of its transactions.  
You increase reliability through the use of horizontal scalability, i.e., by adding more servers. This only works up to a certain point, though.  
When you increase reliability you increase availability.  
  
**Availability:**  
Availability is about assuring that services are available to the required number of users for the required proportion of time.  
  
**Extensibility:**   
The ability to modify or add functionality without impacting the existing functionality.  
The key to an extensible design is to make an effective OO design. Extensibility pays the most towards the font end of a system.  
Some rough guidelines:More than 25 top-level classes will lead to problems  
Every use case should be able to be implemented using domain model methods  
J2EE supports extensibility because it is component-based and allows you to separate the roles of an app. JSPs can handle presentation. Servlets can handle routing, and EJBs can handle business logic.  
  
**Performance:**  
Architectural performance is concerned with creating an architecture that forces end-to-end performance.  
The purpose of an architecture that ensures performance is to control expensive calls and to identify bottlenecks.  
If you know the boundaries of the various parts of the system, the technologies, and the capabilities of the technologies you can do a good job of controlling performance.  
You want to minimize the number of network calls your distributed app makes – make a few “large” calls that get a lot of data vs. lots of calls that get small amounts of data.  
Try to minimize process-to-process calls because they are expensive.  
Use resource pooling to reduce the number of expensive resources that need to be created like network connections, database connections, etc.  
  
**Manageability:**  
Manageability refers to the ability to manage a system to ensure the health of the system.  
A single tier or monolithic app would be more manageable from a management perspective than a multi-tier system but this must be weighed against the possibility of a change rippling through a monolithic app.  
A simple architecture may not be as flexible or available as a more complex system but the amount of effort required to keep the system up & functioning will be less.  
A component-based architecture like J2EE offsets some of the manageability problems caused by a multi-tier system.  
  
**Security:**  
Security ensures that info is neither modified nor disclosed except in accordance with the security policy.  
Tradeoffs: personal privacy, ease of use, and expense.  
A highly secure system is: More costly, Harder to define and develop, Requires more watchdog activities  
Principles of Security:  
Identity – The user is correctly ID’d thru an authentication mechanism  
Authority – The user can perform only allowed activities  
Integrity – Data can only be modified in allowed ways  
Privacy – Data is disclosed to authorized entities in authorized ways  
Auditability – The system maintains logs of actions taken for later analysis

**SCEA1.5 - UML**

**UML Terms:**  
Dependency: a change in one element can affect the semantics of another element.  
Represented by: dashed line with arrow  
  
Association: represents set of connections between objects  
Represented by: solid line with arrow or without arrow may have multiplicity  
  
Aggregation: represents the relationship between whole and the part  
Represented by: solid line with open diamond arrowdiamond side one is whole  
  
Composition: whole part relationship but its a stronder form of aggregation  
Represented by: solid line with filled diamond, diamond side is part  
  
Generalization:is parent and child relationship  
Represented by: solid line with triangular open arrow, extends keyword  
  
Realization: interface and implemenation class  
Represented by: dashed arrow with triangular open arrow, implements keyword  
  
Include relationship in usecase - that the other use case can reuse the coommon factored out use case.  
  
**Difference between Aggregation & Composition:**   
Aggregation defines a part of relationship but both objects can exist independently. But with composite aggregation if one part is removed then the other part will be removed. Think of a plane the wings have a composite aggregation relationship with the body of the plane.

**SCEA1.5 - Internationalization**

**Internatinalization & Localization:**  
Internationalization: Adapting a program for use in any country is called Internationalization. Localization: The process of adapting a program for use in a particular country is referred to as Localization.  
  
**Other classes related to Internationalization:**  
Locale - Language (en, es), Regional (GB,US,), Variant (WIN, POSIX)  
java.util.Properties,  
java.util.ResourceBundle,  
java.util.PropertyResourceBundle,  
InputStreamReader(getEncoading()),  
OutputStreamWriter,  
java.text.NumberFormat,  
java.text.MessageFormat(format()),  
java.text.DateFormat,  
java.text.SimpleDateFormat,  
java.text.Collator  
  
Servlet - setContentType(), setLocale();  
JSP - pageEncoading, contentType() (for default encoading file.proprties)  
  
Java supported font types - Serif, Sans-serif, Monospaced, Dialog and DialogInput (font.properties)

**SCEA1.5 - Design Patterns**

**Benefits of using design patterns:**  
Improves communication between designers by use of pattern names vs. the details of the patterns.  
Captures experience of solving a type of problem.  
Provide a way of reusing design.  
Provide a mechanism for making designs more reusable.  
Provides a mechanism for systematizing the reuse of things that have been seen before.  
Can be used to teach good design.  
  
**Abstract Factory:**  
The Abstract Factory pattern is used for creating many objects that are dependent on each other.  
Also known as Kit  
Used in J2EE - DAO and VO assembler, J2SE-java.awt.Toolkit  
J2EE technology uses this pattern for the EJB Home interface, which creates new EJB objects.  
Related patterns-Factory method,Prototype, concrete factpty often Singleton.  
It isolates concrete classes.  
It makes exchanging product families easy.  
It promotes consistency among products.Supporting new kinds of products is difficult.  
  
**Builder:**  
The builder pattern separates the construction and representation of an object. The client is shielded from the objects construction only needing to specify it's content and type.  
Related patterns-Abstract factory is similar to builder in that it too may construct complex objects,The primary difference is that the builder pattern focuses on constructing a complex object step by step, AbstractFactory's emphasis is on families of product objects(either simple or complex), Builder returns the product as a final step,but as far as the Abstract pattern is concerned, the product gets returned immediately. A composite is what is builder often builds.  
  
**Factory Method:**  
Factory Method pattern provides an interface for creating an object that allows either sub classes or helper classes to create that object.  
Aslo knowna as: virtual constructor  
Used in J2EE-EJBHome, EJBLocalHome, QueueConnectionFactory, TopicConnectionFactory, J2SE- Collator, ContentHandlerFactory, InitialContextFactory, SocketFactory  
PrototypeJ2EE technology uses this pattern for the EJB Home interface, which creates new EJB objects.  
Aslo knowna as: virtual constructor  
Related patterns: absract factory,within Template Methods,  
Eliminates the need to bind application-specific classes into your code.  
Gives subclasses a hook for providing an extended version of an object being constructed.  
  
**Prototype:**  
The Prototype pattern is used to create new objects by copying its prototype.specify the kinds of objects to create using a prototypical instance,and create new objects by copying this prototype.  
Used in J2SE-java.lang.Object  
Related patterns-Prototype and abstract factory are competing is some ways.  
Design that make heavy use of the composite and decorator patterns often can  
benefit from prototype as well.  
  
**Singleton:**  
Ensure a class only had one instance and provide a global point of access to it.The Singleton doesn't just create a single instance it can also be used to create a variable number of instances of a class.J2SE-java.lang.Runtime  
Related patterns:Abstract factory,builder and prototye can be implemented using singleton  
  
**Adapter:**  
The Adapter pattern implements an interface known to its clients and provides an instance of a class not known to its clients.  
Also known as: Wrapper  
Used in JCA architecture, J2SE-java.awt.event.ComponentAdapter  
Related patterns-Bridge,Decorator,Proxy  
  
**Bridge:**  
The Bridge pattern creates a separation between abstractions and classes that implement those abstractions  
Also known as Handle/Body  
Related patterns-An abstract factory can create and configure a particular bridge.  
  
**Composite:**   
Compose objects into tree structure to represent part-whole hierarchies.Composite lets clients treat individual objects and compositions of objects uniformly  
Related patterns: Chain of responsibility, Decorator, Flyweight, Iterator, Visitor  
  
**Decorator Pattern:**  
The Decorator pattern isn't used to build objects. It adds extra functionality to existing objects  
Also known as: Wrapper  
Used in J2EE-EJBObject, J2SE-BufferedReader  
Related patterns: Adapter,Composite,Strategy  
In J2EE technology, The EJB object is a decorator for the bean because the bean’s functionality is expanded to include remote dehavior.  
  
**Facade:**  
Provide a unified interface to a set of interfaces in a subsystem.Facade defines a higher level interface that makes the sub system easies to use  
Used in J2SE-java.net.URL  
Related patterns: AbstractFactory, Mediator,Single tons  
  
**Flyweight:**  
Use sharing to support large numbers of fine grained objects efficientlyWhen the instances of your class can be used interchangeably and you want to reduce the number of instances created in order to improve performance  
Used in J2SE-java.lang.String  
Related patterns: Composite,State and Strategy  
  
**Proxy:**  
Provide a surrogate or placeholder for another object to control access to it In this scenario what you are essentially trying to do is filter all packets that don't meet a certain set of requirements. This behavior is just like a Proxy server dropping packets from certain IP address etc  
Also known as Surrogate  
The EJB’s remote interface(EJBObject) acts as a proxy for the bean. Proxy is also used in RMI.  
Related patterns: Adapter,Decorator  
  
**Chain of responsibility:**  
Avoid coupling the sender of a request to its receiver by giving mmore than one object a chance to handle the request.Chain the receiving objects and passthe request along the chain until an object handles it.  
Uused in J2EE-ReqestDispatcher in the servlets/JSP API  
Related patterns: composite  
  
**Command:**  
Encapsulate a request as an object,there by letting you parameterize clients with different requests,queue or log requests and support undoable operations  
Also known as Action of Transaction  
Used in J2EE-Message Beans,servlets and JSPs  
Related patterns: Composite,Memonto  
  
**Interpreter:**  
Given a language,define a representation for its grammer along with an interpreter that uses the representation to interpret sentences in the language.  
Related patterns:composite,iterator,visitor  
Iterator:Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.  
Also known as- Cursor  
Used in J2SE- Iterator,Enumaration  
Related patterns: Composite,Factory method,Memento  
  
**Mediator:**  
Define an object that encapsulates how a set of objects interact. mediator promotes loose coupling by keeping objects from referring to each other explicitly  
and it lets you vary their interaction independentlyThe Mediator pattern allows you to co-ordinate state changes between other objects by using one object.  
Related patterns: Facade Observer  
  
**Memento:**  
Without violating encapsulation, capture and externalise an object's internal state so that the object can be restored to this state later.  
Also known as Token  
J2EE-Entiry Bean using Bean managed persistence  
Related patterns: Command, Iterator  
  
**Observer:**  
Define a one to many dependency between objects so that when one object changes its state,all its dependents are notified and updated automatically.  
When you need classes to be notified of events but you don't know which classes or if you will need to add more at a later date.  
Also known as: Dependents,Publish-subscribe  
Used in J2SE-Observable,Observer  
Related patterns:Mediator.Singleton  
  
**State:**  
Allow an object to alter its behavior when its itnernal state changes.The object will appear to change its class.  
Also known as: Objects for states  
Related patterns:Flyweight, singletons  
  
**Strategy:**  
Define a family of algorithms, encapsulate each one,and make them interchangable. Strategy lets the algorithm vary independently from clients that uses it.  
Also known as-Policy  
Related patterns: Flyweight  
  
**Template method:**  
Define the skeleton of an algorithm in an operation,deferring spme steps to sub classes.Template method lets sub classes redefine certain steps of an algorithm without changing the algorithms structure  
Related patterns: Factory method,Strategy  
  
**Visitor:**  
Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates  
Related patterns: Composite, Interpreter

**SCEA1.5 - Few Terms related to JAX-WS, JAX-RPC and JAXB**

**Reasons you may want to stay with JAX-RPC 1.1:**  
If you want to stay with something that's been around a while, JAX-RPC will continue to be supported for some time to come. If you don't want to step up to Java 5. If you want to send SOAP encoded messages or create RPC/encoded style WSDL.  
**Reasons to step up to JAX-WS 2.0:**If you want to use the new message-oriented APIs. If you want to use MTOM to send attachment data. If you want better support for XML schema through JAXB. If you want to use an asynchronous programming model in your Web service clients. If you need to have clients or services that can handle SOAP 1.2 messages. If you want to eliminate the need for SOAP in your Web services and just use the XML/HTTP binding. If you like playing with leading edge technology.  
JAX-WS style of web services is built on JSR-224 specification.It uses annotations (JSR-181) and new data binding stack JAXB. JbossWS 2.0 >= series whichis used in this porting exercise is JAX-WS compliant.It supports SOAP 1.1 and SOAP 1.2.Their are quiet a few notable differences between JAX-WS and JAX-RPC.  
**1.2.1.1 SOAP 1.2**JAX-RPC and JAX-WS support SOAP 1.1. JAX-WS also supports SOAP 1.2.  
**1.2.1.2 XML/HTTP**  
The WSDL 1.1 specification defined an HTTP binding, which is a means by which you can send XML messages over HTTP without SOAP. JAX-RPC ignored the HTTP  
binding. JAX-WS adds support for it.  
**1.2.1.3 WS-I's Basic Profiles**JAX-RPC supports WS-I's Basic Profile (BP) version 1.0. JAX-WS supports BP 1.1. (WS-I is the Web services interoperability organization.)  
**1.2.1.4 New Java features**  
JAX-RPC maps to Java 1.4. JAX-WS maps to Java 5.0. JAX-WS relies on many of the features new in Java 5.0.Java EE 5, the successor to J2EE 1.4, adds support for JAX-WS, but it also retains support for JAX-RPC, which could be confusing to today's Web services  
novices.  
**1.2.1.5 The data mapping model**JAX-RPC has its own data mapping model, which covers about 90 percent of all schema types. Those that it does not cover are mapped to  
javax.xml.soap.SOAPElement.  
JAX-WS's data mapping model is JAXB. JAXB promises mappings for all XML schemas.  
**1.2.1.6 The interface mapping model**  
JAX-WS's basic interface mapping model is not extensively different from JAX-RPC's; however:  
JAX-WS's model makes use of new Java 5.0 features.JAX-WS's model introduces asynchronous functionality.  
**1.2.1.7 The dynamic programming model**JAX-WS's dynamic client model is quite different from JAX-RPC's. Many of the changes acknowledge industry needs:  
It introduces message-oriented functionality.It introduces dynamic asynchronous functionality.JAX-WS also adds a dynamic server model, which JAX-RPC does not have.  
**1.2.1.7 MTOM** (Message Transmission Optimization Mechanism)  
JAX-WS, via JAXB, adds support for MTOM, the new attachment specification. Microsoft never bought into the SOAP with Attachments specification; but it  
appears that everyone supports MTOM, so attachment interoperability should become a reality.  
**1.2.1.8 The handler model**The handler model has changed quite a bit from JAX-RPC to JAX-WS.JAX-RPC handlers rely on SAAJ 1.2. JAX-WS handlers rely on the new SAAJ 1.3 specification.  
  
**Difference between original WSDL and modified WSDL**No import statement required hereHelloWorld is not a complex type but just an element, if it is complex type it adds another layer in SOAP PacketSimilarly HelloWorldRespons is not a complex type but just an element  
  
**JAXB:** JAXB provides methods for unmarshalling XML instance documents into Java content trees,and then marshalling Java content trees back into XML instance documents. JAXB alsoprovides a way to generate XML schema from Java objects  
JAXB 2.0 includes several important improvements to JAXB 1.0:¦ Support for all W3C XML Schema features. (JAXB 1.0 did not specify bindings for some ofthe W3C XML Schema features.)¦ Support for binding Java-to-XML, with the addition of the javax.xml.bind.annotationpackage to control this binding. (JAXB 1.0 specified the mapping of XML Schema-to-Java,but not Java-to-XML Schema.)  
A significant reduction in the number of generated schema-derived classes.  
Additional validation capabilities through the JAXP 1.3 validation APIs.  
Smaller runtime libraries.  
  
**Unmarshalling:** Unmarshalling provides a client application the ability to convert XML data into JAXB-derivedJava objects  
**Marshalling:** Marshalling provides a client application the ability to convert a JAXB-derived Java object treeback into XML data.

**SCEA1.5 - Transaction Attributes**

Transactional attributes of bean methods are specified in the deployment descriptor. Here are the attributes and what they mean :  
  
**TX\_BEAN\_MANAGED:** The bean programmatically controls it’s own txEJB 1.0 Only boundaries via JTA.  
  
**NotSupported:** The bean CANNOT be involved in atransaction at all. When a bean method is called, any existing tx is suspended.  
  
**Required:** The bean must ALWAYS run in a transaction. If a tx is already running, the bean joins in that tx. If not, the container starts a tx for you.  
  
**RequiresNew:** The bean must ALWAYS run in a NEW transaction. Any current tx is suspended.  
  
**Supports:** If a transaction is underway, the bean joins in that tx, otherwise runs with no tx at all.  
  
**Mandatory:** Mandates that a transaction must already be running when the bean method is called or an exception is thrown back to the caller.  
  
**Never:** If a tx is underway the bean will throw aEJB 1.1 Only RemoteException, otherwise the methodRuns normally without a tx.

**SCEA1.5 - Legacy Connectivity**

**Upgrading Client-Tier GUIs:**In cases where the GUI is loosely coupled to the other legacy tiers you can use an applet or a small application to replace the GUI.  
Applets can communicate with the other tiers via TCP sockets. The applet can be signed and trusted, if necessary to access resources.  
Applets can also communicate with COM and CORBA objects (using bridge or Java IDL).  
  
**Screen Scrapers :**  
Screen scrapers may be used to integrate applet (or other) interface with an existing system. They are particularly useful when the client interface is  
tightly coupled to the other tiers of the system.  
A screen scraper is an application that translates an existing client interface into a set of objects.  
Screen scrapers usually function as a terminal emulator on one end and an object interface on the other. The screen scraper is configured to read data from  
terminal fields of the legacy interface and make them available via objects.  
Screen scrapers have the following advantages:  
Provides a low-level object-based interface to the legacy app.  
Allows you to build a new GUI over the existing client interface.  
Disadvantages of screen scrapers:  
Any changes to the legacy interface can break the new GUI.  
Prone to causing errors in the new GUI because of unexpected outputs from the legacy interface.  
Prone to causing the new GUI to “freeze” when the legacy interface is expecting input that the screen scraper in unaware of.  
  
**Object Mapping Tools:**Object mapping tools can be used if you choose to ignore the existing legacy interface and access the underlying tiers directly.  
These tools are used to create proxy objects that access legacy system functions and make them available in an object-oriented form.  
Object mapping tools are usually more effective than screen scrappers because they are not dependent on the format generated by the existing legacy  
interface.  
  
When u have access to mainframe source code use Object-mapping, if u dont have access to mainframe source code use screen scrapping.  
  
**Upgrading Application Business Logic:**  
Java servlets provide a capability to make existing applications available via an intranet or the Internet.  
Clients (browsers and/or applets) access servlets via HTTP or HTTPS. The servlets take the requests and communicate with the legacy system.  
EJBs provide a component-based approach to upgrading legacy applications.  
Java’s support for CORBA enables CORBA objects to be accessed from Java and Java objects to be accessed as CORBA objects.  
Microsoft’s JVM provides (or used to provide) a bridge between Java and COM objects.  
JNI may be used to write custom code to interface new business logic with an existing legacy system.  
  
**Upgrading the Data Storage Tier:**  
JDBC may be used to access relational databases in a legacy system.  
In many cases the legacy database will not support a pure JDBC driver. If the database provides ODBC support the JDBC-ODBC bridge can be used.  
If the existing legacy database is hierarchical or flat-file then it may be able to be imported into an RDBMS.  
  
**Securing Legacy System Components:**Retrofitting a system with security is generally more expensive and less productive the redesigning and redeveloping the system to operate in a secure  
manner. However, budget constraints may prevent this.  
Legacy systems may be isolated from threats by placing them behind a firewall.  
Access control to legacy systems can be controlled by requiring users and external applicationsto authenticate themselves with the firewall before they can  
access the legacy system. Auditing features of the legacy system should be used to determine who is accessing the legacy system and when.  
A VPN may be used to secure all communications with a legacy system.  
Check what is needed for VT100 terminal? Given ans JDBC with SQL, for MQseries - EJB session bean with JMS message, JNDI allows to connect to LDAP server  
When u have access to mainframe source code use Object-mapping,if u dont have access to mainframe source code use screen scrapping.  
  
**Offboard server:**An off-board server is simply a proxy server for legay system  
Enable secure remote access to a mainframe by forwarding SSL requests to serial connections  
  
**Fast Lane Reader:**  
The Fast Lane Reader design pattern provides a more efficient way to access tabular, read-only data. A fast lane reader component directly accesses  
persistent data using JDBCTM components, instead of using entity beans. The result is improved performance and less coding, because the component represents  
data in a form that is closer to how the data are used.  
  
**How to connec to VT100 terminal?**   
Use JDBC with SQL  
**How to connect to MQseries ?**  
Use EJB session bean with JMS message  
**How to connect to LDAP server?**  
JNDI allows to connect to LDAP server

**SCEA1.5 - Messaging**

**Java Messaging Service (JMS):**JMS provides a common way for Java programs to create, send, receive and read an enterprise messaging system’s messages.  
JMS defines a set of message interfaces.  
JMS provides client interfaces for point-to-point (PTP) and publish-subscribe systems.  
PTPbuilt around the concept of message queueseach message is addressed to a specific queue; clients get messages from the queue(s) created tohold their messages  
  
**Publish-Subscribe:**  
Publishers address messages to a node or addressSystem distributes the messages arriving from a publisher to the subscribers of that publisher  
Nothing prevents a JMS application from combining PTP and publish-subscribe but JMS focuses on applications that use one approach or the other.  
  
**Asynchronous messaging:**  
Loose coupling between sender and receiver  
Does not block sender  
Network does not need to be available, messages can be queuedLeast demanding on comm. mechanisms  
Good for publish-subscribe  
  
**Synchronous messaging:**  
Tight coupling between sender and receiver  
Blocks sender until receiver is finished processing  
Network must be availableMore demanding on comm.  
Mechanisms Good for transaction processing, Fail-safe comm.  
Coping with error situations  
  
**JMS does NOT include the following:**Load balancing/fault tolerance  
Error/advisory notification  
AdministrationSecurity  
Wire protocol  
Message Type Repository

**SCEA1.5 - Protocols**

**HTTP Properties:**  
Client-Server Architecture The HTTP protocol is based on a request/response paradigm. The communication generally takes place over a TCP/IP connection on the Internet. The default port  
is 80, but other ports can be used. This does not preclude the HTTP/1.0 protocol from being implemented on top of any other protocol on the Internet, so long  
as reliability can be guaranteed.  
The HTTP protocol is connectionless and stateless After the server has responded to the client's request, the connection between client and server is dropped  
and forgotten. There is no "memory" between client connections. The pure HTTP server implementation treats every request as if it was brand-new, i.e. without  
context.  
An extensible and open representation for data types HTTP uses Internet Media Types (formerly referred to as MIME Content-Types) to provide open and  
extensible data typing and type negotiation. When the HTTP Server transmits information back to the client, it includes a MIME-like (Multipart Internet Mail  
Extension) header to inform the client what kind of data follows the header. Translation then depends on the client possessing the appropriate utility (image  
viewer, movie player, etc.) corresponding to that data type.  
  
  
**HTTPS(Secure Hypertext Transfer Protocol) :**HTTPS (Secure Hypertext Transfer Protocol) is a Web protocol developed by Netscape and built into its browser that encrypts and decrypts user page requests  
as well as the pages that are returned by the Web server. HTTPS is really just the use of Netscape's Secure Socket Layer (SSL) as a sublayer under its  
regular HTTP application layer. (HTTPS uses port 443 instead of HTTP port 80 in its interactions with the lower layer, TCP/IP.) SSL uses a 40 or 128-bit key  
size for the RC4 stream encryption algorithm, which is considered an adequate degree of encryption for commercial exchange.  
Suppose you use a Netscape browser to visit a Web site such as NetPlaza (<http://www.netplaza.com/>) and view their catalog. When you're ready to order, you  
will be given a Web page order form with a URL that starts with https://. When you click "Send," to send the page back to the catalog retailer, your  
browser's HTTPS layer will encrypt it. The acknowledgement you receive from the server will also travel in encrypted form, arrive with an https:// URL, and  
be decrypted for you by your browser's HTTPS sublayer.  
HTTPS and SSL support the use of X.509 digital certificates from the server so that, if necessary, a user can authenticate the sender. SSL is an open,  
nonproprietary protocol that Netscape has proposed as a standard to the World Wide Consortium (W3C). HTTPS is not to be confused with SHTTP, a security-  
enhanced version of HTTP developed and proposed as a standard by EIT.  
Resource:Whatis.com/https.htm  
  
  
**IIOP:**  
CORBA and IIOP assume the client/server model of computing in which a client program always makes requests and a server program waits to receive requests  
from clients. When writing a program, you use an interface called the General Inter-ORB Protocol (GIOP). The GIOP is implemented in specialized mappings for  
one or more network transport layers. Undoubtedly, the most important specialized mapping of GIOP is IIOP, which passes requests or receives replies through  
the Internet's transport layer using the Transmission Control Protocol (TCP). Other possible transport layers would include IBM's Systems Network  
Architecture (SNA) and Novell's IPX.  
For a client to make a request of a program somewhere in a network, it must have an address for the program. This address is known as the Interoperable  
Object Reference (IOR). Using IIOP, part of the address is based on the server's port number and Internet Protocol (IP) address. In the client's computer, a  
table can be created to map IORs to proxy names that are easier to use. The GIOP lets the program make a connection with an IOR and then send requests to it  
(and lets servers send replies). A Common Data Representation (CDR) provides a way to encode and decode data so that it can be exchanged in a standard way.  
CORBA is not the only architecture that uses IIOP. Because a TCP/IP-based proxy is usable on almost any machine that runs today, more parties now use IIOP.  
When another architecture is IIOP-compliant, it not only establishes a well-proven communication transport for its use, but it also can communicate with any  
ORB implementation that is IIOP-compliant. The possibilities are endless.  
reference: <http://www.blackmagic.com/people/gabe/iiop.html>  
  
  
**JRMP :**  
The Transport layer employs JRMP, also known as the RMI Wire Protocol, to send method invocations and associated parameters and to return values and  
exceptions from one Java virtual machine (JVM) to another. JRMP is a simple protocol consisting of five messages, plus an extra five for multiplexing flow  
control.  
All JRMP sessions consist of a header followed by one or more messages. The header contains just the ASCII codes for the characters JRMI , the protocol  
version, and the "subprotocol" to be used. There are three subprotocols: SingleOpProtocol, StreamProtocol, and MultiplexProtocol. SingleOpProtocol signifies  
that only one message follows a header before the end of a session (i.e., the connection closes). StreamProtocol and MultiplexProtocol can transfer one or  
more messages. The latter is used when multiplexing calls from both client and server on a single socket, as described below.  
Communicating clients and servers typically each open a socket to the other (i.e., both systems connect and listen for connections). The client's socket  
typically invokes methods on server-side objects, and the server's socket calls client-side objects (e.g., callbacks). The figure shows a hypothetical  
StreamProtocol situation. The client sends the Call message to invoke a server object's method; the server then invokes this method and replies with a Return  
containing any results. Assuming that a remote object is returned, the client then sends a DgcAck message to let the server's garbage collector know that it  
has received the remote object. On another socket, the server sends a Ping to find out whether the client is alive, which replies with a PingAck.  
Default applet security restrictions deny applets the right to open sockets back to any server other than their originating host; they also block any attempt  
to listen for socket connections. This being the case, how do clients listen for server connections?  
Enter the MultiplexProtocol and its group of five messages: Open , Close , CloseAck , Request, and Transmit. They allow client and server to simulate the  
StreamProtocol's two-way communication using a single socket. In the current implementation, up to 256 virtual connections can be opened, each identified by  
a unique ID.  
Unfortunately, connecting via a socket back to the server is not always possible for applets running behind firewalls (e.g., on a corporate intranet), which  
typically block any attempt to open a socket back to the Internet. Should it fail to open a connection, an RMI client wraps its method invocation inside the  
body of an HTTP request (which is the protocol browsers use to communicate with Web servers), and the RMI server sends any results as an HTTP response.  
This workaround is a smart solution, since HTTP is a firewall-trusted protocol. Still, performance takes a hit due to the time needed to convert messages to  
HTTP requests. In addition, no multiplexing of invocations can be accomplished, because keeping the connection open between client and server is not part of  
HTTP 1.0. The primary reason for SingleOpProtocol's existence is to encapsulate RMI through HTTP.  
Reference: <http://www.byte.com/art/9802/sec4/art3.htm>  
  
RMI:  
RMI This is a possible if the objects in the user interface and the business layers are all Java objects. The persistence layer is mostly  
accessed through JDBC. Other relational object mapping of the data layer is also possible. Advantage of RMI Object are passed by value. The server/ client  
can reconstitute the objects easily. Data type can be any Java objects. Any Java objects can be passed as arguments. Arguments has to implement the  
serializable interface Disadvantage of RMI Heterogeneous objects are not supported. Corba If the objects in the client layer and the business layer are  
heterogeneous, i.e. the objects are implemented in C, C++ Java, Smalltalk then Corba is most suitable. Advantage of Corba Heterogeneous objects are  
supported. Disadvantage of Corba Objects are not passed by value, only the argument data is passed. The server/ client has to reconstitute the objects with  
the data. Only commonly accepted data types can be passed as arguments. Dcom This works best in windows environment. Distributed Object Communication Advantages Disadvantages HTTP Simple, Established Has to communicate to a Servlet, Java Server pages Cannot communicate to a Java class directly RMI Object are passed by value. The server/ client can reconstitute the objects easily. Object are passed by reference Data type can be any Java objects. Any  
Java objects can be passed as arguments. Arguments has to implement the Serializable interface Heterogeneous objects are not supported. Corba Heterogeneous objects are supported. Objects are not passed by value, only the argument data is passed. The server/ client has to reconstitute the  
objects with the data. Only commonly accepted data types can be passed as arguments Dcom If windows is the deployment platform suits well with the operating system This works in windows environment at best Distributed Object Frameworks Distributed Object Frameworks are RMI, Corba, Dcom, EJB. Basic Three-Tier Java Technology Architecture The three-Tier Java  
Technology Architectureis achieved by HTML, Applet, Java Application on the client. Servlet, Java Server Pages on the Middle Tier. JDBC communication to the  
persistence or Database layer Client C to M comm. Middle M to P comm. Persistence HTML HTML with applet HTTP Servlet Java Server Pages JDBC RDBMS Legacy File Java Application JRMP RMI Server JDBC RDBMS Legacy File Java Application RMI- II0P EJB JDBC RDBMS Legacy File Java Application ( Not a Java 3 tier) IIOP Corba JDBC RDBMS Legacy File  
  
**Conclusions:** HTTP and HTTPS are very similar protocols with only the fact that HTTPS provides a layer of security(the SSL). They are both capable of passing a variety of  
data types but there is no logic, objects may only be executed if there's another protocol to handle them. HTTP is the lowest layer of logic and can only be  
used as a delivery mechanism for other protocols. JRMP is a robust object server that communicates well when working with JAVA based objects. It is capable  
of passing objects refrences rather than just values that have to be reconstituted so that the object may be executed by the client rather than the server.  
In the even that the server is secure or cannot communicate in the most efficient manner JRMP falls back to HTTP. JRMP is only capable of passing JAVA  
objects. IIOP is the most flexible of the transport mechanisms, it can communicate objects created in C, C++, JAVA, and smalltalk but only passes data by  
value requiring the server to do all the work and requiring that only common data types be passed as arguments making it more restrictive than JRMP which  
allows any JAVA data type.  
  
**Default Portnumbers for the above protocols:**  
Http : 80, 8080  
HTTPS,SSL: 443, 8443  
JRMP: 1099  
LDAP: 389  
LDAP Over SSL - IIOP: 636

**[SCEA1.5 - Security](http://mycollectivematerial.blogspot.com/2009/02/scea15-security.html)**

The Java 2 security model is policy-based and has superseded the sandbox/trusted approach of Java 1.1.  
In Java 1.1 remote code (applets, for example) that was not trusted was constrained to the sandbox. If the remote code was signed and trusted then it could  
access local resources.  
**Code Source:**A combination of a set of signers (certificates) and a code base URLBy default, Java 2 uses a policy file to associate permissions with code sources  
Security Policy File:A permission is the right to access a protected resource or guarded objectFor Java 2 permissions are specified in the security policy fileOnly one policy is in effect at a timeA policy file consists of a number of grant entriesEach grant entry describes the permissions (one or multiple) granted to a code source  
**Policy class** - You can use java.security.Policy to create your own security policy.  
java.security package  
The following are some of the classes in the java.security package:  
**CodeSource** – This class extends the concept of a codebase to encapsulate not only the location (URL) but also the certificate(s) that were used to verify  
signed code originating from that location.  
**KeyStore** – This class represents an in-memory collection of keys and certificates. It manages keys and trusted certificates.  
**MessageDigest** – The MessageDigest class provides applications the functionality of a message digest algorithm, such as MD5 or SHA.  
**Permission** – Abstract class for representing access to a system resource.  
**Policy** – This is an abstract class for representing the system security policy for a Java application environment (specifying which permissions are available  
for code from various sources).  
**ProtectionDomain** – The ProtectionDomain class encapulates the characteristics of a domain, which encloses a set of classes whose instances are granted the  
same set of permissions.  
**Security** – Centralizes all security properties and common security methods.  
Given an architectural system specification, identify appropriate locations for implementation of specified security features, and select suitable  
technologies for implementation of those features.  
Exposure to threats can be mitigated by using:  
AuthenticationAuthorization (ACLs)Protecting MessagesAuditing  
  
**Web tier authentication :  
Basic HTTP** – the web server authenticates a principal with user name & password from Web client  
**Form-based** – lets developers customize the authentication user  
**HTTPS mutual authentication** – the client and server use X.509 certificates to establish identity over a SSL channel.  
**EJB/EIS tier authentication:**  
For EJBs can use protection domains. Thus the EJB tier could entrust the web tier to vouch for the identity of users.  
Put a protected web resource in front of a protected EJB resource  
Have every web resource that calls an EJB resource route through a protected web resource  
For access to EIS tier resources authentication is usually carried out by the component accessing the EIS resource.  
You can have the container manage the EIS resource authentication or have the app do this itself.  
**Authorization:**  
In J2EE a container serves as an authorization boundary between callers and its components. The authorization boundary is inside the authentication boundary  
so authorization occurs within the context of successful authentication.  
For component to component invocations inside the container the calling component must make its credentials available to the called component.  
You can have file-based & code-based security in J2EE.  
Access control policy is set a deployment time.  
Controlling access to resources in the container (deployment descriptor)  
To control access to web resources, specify constraint in the deployment descriptor.To control access to EJB, specify roles in the deployment descriptor.You can specify methods of the remote & home interface that each security role is allowed to invoke  
**Protecting Messages:**  
To ensure message integrity you can use:  
**Message signature** – a enciphered digest of the message contents (costly in terms of CPU cycles)  
**Message confounder** – ensures message authentication is useful only once  
A deployer must configure the containers involved in a call to implement integrity mechanisms either because the call will traverse open or unprotected  
networks or because the call will be made between components that do not trust each other.  
Auditing  
When security is breached it is usually more important to know who has been allowed access than who has not.  
Audit records need to be well protected – tapes or logging to a printer vs disk drive  
  
**Firewalls:**Basic Services:• Block incoming data that might contain hacker attack.• Hide information about topology of the network. Make it seems like all requests come from one IP address.• Screen outgoing traffic.  
3 basic types.  
**Packet Filter Firewall:** Looks at the information related to IP address of a packet, types of connections, etc. and then provides filtering based on that.  
Uses this info. to decide which packets to let through and which to deny. IP spoofing may fool some of these.  
**Application-Level Proxies:** Work at the application level to provide proxy services. Allows more specific inspection of the packets. Can use application level  
knowledge to decide what to filter. Usually requires separate proxy for each type of application you want to filter.  
**Stateful Packet Inspection Firewall:** Examines and remembers outgoing packets so that when incoming packets come in, that information will be used to  
determine whether or not to let the incoming packets through. For example, if an incoming packet wasn’t requested by any outgoing packets, it will be  
filtered.  
  
**Security:**  
Java sandbox consists of following elements:• Byte code verifier• Access controller• Security manager• Class loader (applet class loader, url class loader, rmi class loader, default internal class loader, custom built class loaders)• Security package (security provider interface, message digests, keys, certificates, digital signatures, encryption)  
AccessControllerChecking permission is done by checking the permissions associated with the protection domain for each method on the stack starting from the top. If each  
protection domain on the stack allows access, then it is granted.  
Using PrivilegedAction and PrivilegedExceptionAction, protection domains can grant privileges to code that has called it but not to code that it calls.  
GuardedObjectAllows you to embed another object within it such that all access to that object will first have to go through a guard, usually the AccessController.  
MessageDigestSmall sequence of bytes that represents the actual input data. In order to use the digest, you also need a copy of the original data so that you can  
calculate the digest on it again and compare it to the digest that was given to you. For example, to do authentication, the user needs to enter his  
id/password but you don’t want that to be sent in clear text over the network so they send you the digest instead. Then you take that and compare it to a  
digest that you calculate on their password and if it matches the digest they sent to you, then you can authenticate them. Message digests do not need any  
key to calculate. Also you can’t derive anything about the actual data from the digest.  
Digital SignatureUsed to uniquely identify an entity, non-repudiation. The way it works is you calculate a message digest on some piece of data and then you encrypt that  
digest with your private key. Then you send that data along with the encrypted digest to the other party. The other party then uses that data to calculate  
another digest and then encrypts it with your public key. Then they compare it to the signature that you sent them and if it matches, then your identity is  
verified.  
CertificateContains 3 pieces of information:• Name of entity for whom certificate has been issued, known as the “subject”• Public key associated with the subject• Digital signature of the issuer (some CA) of the certificate which verifies the information in the certificate.  
Java Security  
AccessController introduced in 1.2.  
In 1.2, classes on the CLASSPATH can also be subject to a security model.  
Bytecode verifier verifies the Java language safety constraints of the bytre code:• In 1.1, all non-local classes are sent thru byte code verification.• In 1.2, all classes except core Java classes are sent thru verification.  
Classloaders work with security manager to enforce security roles:• Classloader knows where class was loaded from.• Knows whether or not the class came with a digital signature.• Different instances of classloaders group classes into different namespaces based on which instance of the classloader loaded it.  
In 1.2, SecureClassLoader was introduced.  
In 1.2, URLClassLoader was introduced.  
Classloaders have to load the system classes first.  
  
**Trusted vs. Untrusted Classes:**In JDK 1.0, classes loaded from CLASSPATH are considered trusted while those loaded from a class loader are untrusted.  
In JDK 1.1, same rules apply but a class loaded from a jar file may have a digital signature giving it more privileges.  
In JDK 1.2, classes form core API are trusted and other classes are given privileges based on where they were loaded (codebase, codesource?). However, this  
requires special command-line args. BY default, classes from CLASSPATH are considered trusted.  
  
**Thread Security:**  
Threads are grouped into a hierarchy---in theory, the policy of security should be such that threads may only manipulate threads that are below them in the  
hierarchy.  
In JDK 1.1, this isn’t true, each applet is given an individual thread group and threads within that group can manipulate other threads within that group  
without respect to any hierarchy.  
In JDK 1.2, thread hierarch operates as expected.  
Untrusted classes may only manipulate threads that they have created.Untrusted classes may only manipulate thread groups that they have created.  
Threads of untrusted classes must belong to specified groups .  
AccessController  
CodeSource: encapsulation of location from which classes were obtained.  
Permission: encapsulation of request to perform a particular operation.  
Policies: encapsulation of all the specific permissions that should be granted to specific code sources.  
ProtectionDomain: encapsulation of a codesource and the permissions granted to that particular code source.  
Security Policy  
Policy file:• Collection of policy entries.• Each entry is specific to one code source and should list all permissions for that code source.• Single policy file can have multiple entries.• May contain an additional entry to specify the location of the keystore in which public keys for the signers listed the policy file should be found.• Each grant entry represents a protection domain.  
Protection Domain  
Each class in the VM may belong to one and only one protection domain. Set by the class loader when the class is defined.  
The permissions for any particular operation can be considered to be the intersection of all permissions of each protection domain on the stack at the time  
the operation is called.  
Using the “doPrivileged” method of the AccessController, you can temporarily allow a class to perform an action that it normally would not be allowed to do.  
Keys  
Key factory and key specifications available only in Java 1.2. They allow for exporting and importing keys using various specifications.  
Keys from the Sun provider use DSA algorithm.  
Key pair generation is done by KeyPairGenerator, a standard engine of Java security.  
Key Management  
KeyCertificateIdentities  
“Keytool” stores individual private and public keys with retrieval subject to a password.  
“Keystore” is the database of the keytool.  
Keytool works on a file that contains a set of private keys and certificates for those keys.  
Each entry in the keystore has:• Alias: name for referencing that entity• One or more certificates for that entity’s identify.• Optionally, a private key which can be protected by a password.  
Represented by the “KeyStore” class.  
There are 2 types of entries: Key entry and Certificate entry.  
Key entries contain both public and private keys and may contain multiple certificates in a certificate chain.  
Certificate entries contain only public keys in a certificate.  
  
**Signed Classes:**  
Delivered as signed jar files.  
In JDK 1.1, use “javakey” to sign it.  
In JDK 1.2, use “jarsigner” to sign it.  
Each file in a jar file may be signed by a different group of identities and some may not be signed.  
Encryption  
“KeyGenerator” class used for generating new secret keys.  
“SecretKeyFactory” converts from algorithmic or encoded key specifications to actual key objects and translates keys from one implementation to another.  
“KeyAgreement” class can also be used to generate secret key between multiple people. SunJCE provider uses “Diffie-Hellman” protocol for generation.  
  
**Applet Security:**  
Most browsers limit a lot of things that applets can do. Sun’s appletviewer allows more access to applets.  
Two ways in which applets can be considered trusted:• Applet is installed on local disk in a directory in the CLASSPATH.• Applet is signed by identity marked as trusted in your identity database (keystore?).  
Applets cannot do the following things with files:• Check for the existence of the file.• Read the file• Write the file.• Rename the file.• Create a directory on the client file system• List the files in this file (as if it were a directory)• Check the file’s type.• Check the timestamp of when the file was modified.• Check the file’s size.  
Using Sun’s appletviewer, you can grant applets special privileges to perform those file operations.  
Applets cannot open network connection to any host other than the one it came from (the host where the html page was obtained or the host specified in the  
codebase parameter of the applet tag.  
To open network connection, the host name has to be specified exactly the same. If you used an IP address, you can’t use a name now and vice versa.  
Applets loaded through client’s local file system using CLASSPATH can do the following:• Read and write files.• Load libraries on the client.• Execute processes.• Exit the VM.• Are not passed through bytecode verifier.

**[SCEA1.5 - Few Queries](http://mycollectivematerial.blogspot.com/2009/02/scea15-few-queries.html)**

**How many tiers are there in J2EE Application ?**  
J2EE applications have the following tiers: Client (Browsers, Applications, Applets, Mobile clients and so on), Web (presentation tier consisting of JSP as  
view and Servlets as controllers), EJB (Business Tier, consisting of EJB and supporting classes), EIS Integration (Java classes that integrate to the  
Enterprise Information System tier) and finally the EIS tier (relational databases, XML databases, ERP systems and so on.)  
  
**Confused with UserInRole and CallerInRole ?**Servlet - getUserPrincipal() and isUserInRole() - Servlet is called by user so UserInRole comes in servlet code  
EJB - getCallerPrincipal() and isCallerInRole() - EJB is called by servlet not by the user directly, so CallerInRole comes in EJB code  
  
**What is the use of DAO ?**  
EJBs are remote objects that consume significant system resources and network bandwidth. You can use Data Access Objects to encapsulate the logic required to  
access databases.  
Data Access Objects:  
Allow EJBs to delegate responsibility for database access and free them from complex data access routines.  
Make code more maintainable.  
Provide an easier migration path to CMP  
Allow you to adapt data access to different schemas and different databases.

**[SCEA1.5 - Few Terms for exam](http://mycollectivematerial.blogspot.com/2009/02/scea15-few-terms-for-exam.html)**

**Session failover:** ----- In a clustered environment, all requests for a particular session are directed to the same WebSphere Portal server instance in the cluster. In other words, after a user establishes a session (for example, by logging in), the user is served by the same WebSphere Portal server instance for the duration of the session. To verify which server is handling user requests for a session, you can view the global settings portlet in WebSphere Portal, which displays the node name of the WebSphere Portal server handling requests. If one of the WebSphere Portal servers in the cluster fails, the request is rerouted to another WebSphere Portal server in the cluster. If distributed sessions support is enabled (either by persistent sessions or memory-to-memory session replication), the new server can access session data from the database or another WebSphere Portal server instance  
  
**JMS delivery modes:** ---The message delivery semantics cover a range of once-and-only-once to at-most-once delivery. In the once-and-only-once delivery mode, a message is guaranteed by the JMS provider to always arrive at the intended destination no matter what, and it's sent only once. Even in the pub/sub model in which multiple receivers may consume a copy of a broadcasted message, the rules still apply within the relative view of each consumer. Once-and-only-once delivery guarantee is accomplished by the JMS provider through the combination of a store-and-forward mechanism and a rigidly defined set of message acknowledgments  
At-most-once delivery is a less stringent QoS setting on a message - the JMS provider is allowed to occasionally lose a message. A classic example I like to use is a stock feed application. If the broadcast of a particular ticker symbol doesn't reach its intended destination, another one will be along shortly.  
Whether it's once-and-only-once or at-most-once, the key word is once. Regardless of the guaranteed-ness of the delivery mode, the JMS provider is responsible for ensuring that the messages are delivered in the exact order in which they are sent.  
  
**Polling ---- Server Polling - (Reverse Ajax)**Keeping the displayed information up-to-date was always difficult in web world. Before AJAX, one had to use JavaScript or META Refresh tag to get the page refreshed. This was quite annoying from the user experience point of view. However it was not as annoying as something that I experienced few days ago on one of the banks website (a bank in Australia). I was filling out the form and there were couple of select boxes on the page. I selected an option in the select box and moved onto next field just to realize that as I was typing, the page has been reloaded and all data entered past that select-box was gone and had to be re-typed again. Very, very annoying - and it's AJAX age already!  
Server polling, in my humble opinion, is a great feature of AJAX. There is no need to refresh the whole page to obtain the required information. With AJAX, it is possible to:  
update the forms with information as the user moves through the form (e.g. country - state - city)get the feedback about a long server-side or transport process (e.g. progress bar showing the percentage of the uploading file)fake the push of the updated data from the server (think stock prices, weather, traffic info)  
  
**SEI (Service end point implementation):** ----  
JAX-WS technology enables the implementation of Web services based on both the standard service endpoint interface and a new Provider interface. JAX-WS service endpoints are similar to the endpoint implementations in the Java API for XML-based RPC (JAX-RPC) specification. Unlike JAX-RPC, the requirement for a service endpoint interface (SEI) is optional for JAX-WS Web services. JAX-WS services that do not have an associated SEI are regarded as having an implicit SEI, whereas services that have an associated SEI are regarded as having an explicit SEI. The service endpoint interfaces required by JAX-WS are also more generic than the service endpoint interfaces required by JAX-RPC. With JAX-WS, the SEI is not required to extend the java.rmi.Remote interface as required by the JAX-RPC specification.  
The JAX-WS programming model also leverages support for annotating Java classes with metadata to define a service endpoint implementation as a Web service and define how a client can access the Web service. JAX-WS supports annotations based on the Metadata Facility for the Java Programming Language (JSR 175) specification, the Web Services Metadata for the Java Platform (JSR 181) specification and annotations defined by the JAX-WS 2.0 (JSR 224) specification, which includes Java Architecture for XML Binding (JAXB) annotations. Using annotations, the service endpoint implementation can independently describe the Web service without requiring a WSDL file. Annotations can provide all of the WSDL information necessary to configure your service endpoint implementation or Web services client. You can specify annotations on the service endpoint interface used by the client and the server, or on the server-side service implementation class.  
<<http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/twbs_devjaxwsendpt.html>>  
  
**EJB timer service:** ---- Consider a reporting Application, that will send report in the form of mails, every Monday, or a Billing Service that sends credit or debit bills on the 1st of every month. These applications depend on time-based events. To be more precise, these applications should allow developers to schedule some business logic or process so that they can be executed at some regular intervals of time. This is the core concept behind EJB Timers.  
EJB Timer Services are services that are provided by the container (or the Application Server) and developers can take advantage of the timer services by registering one or more enterprise beans for time-based notification.  
Different Types of Timers:EJB basically supports two forms of Timer objects:  
Single Action Timer Interval Timer  
  
**Streaming API for XML (StAX):** --- a streaming Java-based,event-driven, pull-parsing API for reading and writing XML documents. StAX enables you tocreate bidrectional XML parsers that are fast, relatively easy to program, and have a lightmemory footprint.StAX is the latest API in the JAXP family, and provides an alternative to SAX,DOM, TrAX, andDOMfor developers looking to do high-performance stream filtering, processing, andmodification, particularly with low memory and limited extensibility requirements.To summarize, StAX provides a standard, bidirectional pull parser interface for streaming XML

**[SCEA1.5 - Few Design patterns](http://mycollectivematerial.blogspot.com/2009/02/sun-certified-enterprise-architect_04.html)**

**Service Activator:** ----  
Problem:a business service such as a session or entity bean provides only synchronous processing and thus presents a challenge to implementing asynchronous processingSolution: Use a Service Activator to receive asynchronous client requests and messages. On receiving a message, the Service Activator locates and invokes the necessary business methods on the business service components to fulfill the request asynchronously.  
The ServiceActivator is a JMS Listener and delegation service that requires implementing the JMS message listener-making it a JMS listener object that can listen to JMS messages. The ServiceActivator can be implemented as a standalone service. Clients act as the message generator, generating events based on their activity.  
Any client that needs to asynchronously invoke a business service, such as an enterprise bean, may create and send a message to the Service Activator. The Service Activator receives the message and parses it to interpret the client request. Once the client's request is parsed or unmarshalled, the Service Activator identifies and locates the necessary business service component and invokes business methods to complete processing of the client's request asynchronously.  
The Service Activator may optionally send an acknowledgement to the client after successfully completing the request processing. The Service Activator may also notify the client or other services on failure events if it fails to complete the asynchronous request processing.  
The Service Activator may use the services of a Service Locator to locate a business component.  
  
**Virtual Proxy:** ---Application is often a collection of components and in most of the situations it may be the case that a component should be loaded until it is first accessed by the Client. Reasons may be that the component in consideration may be using most of the system resources. For example, consider the Microsoft Word Application which is providing Printer Service and the Help System. It should not be the case that both the Printer Service and the Help System should be loaded during the Application startup. Imagine what would be the case if the Client after starting the Word Application is accessing other part of the System and not the Printer and the Help Service. Definitely it will lead to a slow-response time because the Components are un-necessarily loaded.  
Virtual Proxy Pattern comes into picture here as it defers the Object Creation process of memory-intensive components thereby speeding up the Application. Now, let us see how to design the Virtual Proxy.  
  
**Composite View:** --- ProblemInstead of providing a mechanism to combine modular, atomic portions of a view into a composite whole, pages are built by embedding formatting code directly within each view.modification to the layout of multiple views is difficult and error prone, due to the duplication of code.  
SolutionUse composite views that are composed of multiple atomic subviews. Each component of the template may be included dynamically into the whole and the layout of the page may be managed independently of the content.  
This solution provides for the creation of a composite view based on the inclusion and substitution of modular dynamic and static template fragments. It promotes the reuse of atomic portions of the view by encouraging modular design. It is appropriate to use a composite view to generate pages containing display components that may be combined in a variety of ways. This scenario occurs, for example, with portal sites that include numerous independent subviews, such as news feeds, weather information, and stock quotes on a single page. The layout of the page is managed and modified independent of the subview content.  
  
**View Helper :** --- ProblemPresentation tier changes occur often and are difficult to develop and maintain when business data access logic and presentation formatting logic are interwoven. This makes the system less flexible, less reusable, and generally less resilient to change.SolutionA view contains formatting code, delegating its processing responsibilities to its helper classes, implemented as JavaBeans or custom tags. Helpers also store the view's intermediate data model and serve as business data adapters.  
  
**Service to Worker:** ---The system controls flow of execution and access to business data, from which it creates presentation contentThe Service to Worker pattern, like the Dispatcher View pattern, describes a common combination of other patterns from the catalog. Both of these macro patterns describe the combination of a controller and dispatcher with views and helpers. While describing this common structure, they emphasize related but different usage patterns.  
Combine a controller and dispatcher with views and helpers to handle client requests and prepare a dynamic presentation as the response. Controllers delegate content retrieval to helpers, which manage the population of the intermediate model for the view. A dispatcher is responsible for view management and navigation and can be encapsulated either within a controller or a separate component.

**[SCEA1.5 - Few Terms in Security](http://mycollectivematerial.blogspot.com/2009/02/sun-certified-enterprise-architect.html)**

**Man in the middle** ---- Man-In-The-Middle attack is the type of attack where attackers intrude into an existing connection to intercept the exchanged data and inject false information. It involves eavesdropping on a connection, intruding into a connection, intercepting messages, and selectively modifying data.  
Also known as:  
Bucket-brigade attack  
Fire brigade attack  
Monkey-in-the-middle attack  
Session hijacking  
TCP hijacking  
TCP session hijacking  
  
**SQL Injection:** ----SQL injection attacks are attacks against websites that rely on relational databases.  
In this type of site, parameters are passed to the database in the form of an SQL query. As such, if the designer does not verify the parameters passed in the SQL query, a hacker can modify the query in order to access the entire database and even to modify its content.  
Some characters make it possible to string together several SQL queries or to ignore the rest of the query. By inserting this type of character in the query, a hacker can potentially execute the query of his choice.  
Given the following query, waiting for a user name as a parameter:  
SELECT \* FROM users WHERE name="$name";A hacker simply needs to enter a name such as "toto" OR 1=1 OR name ="titi" for the query to become as follows:  
SELECT \* FROM users WHERE name="toto" OR 1=1 OR name ="titi";With the above query, the WHERE clause is always performed, which means it will return records that correspond to all users.  
Stored proceduresMoreover, some database management systems such as Microsoft SQL Server have stored procedures that make it possible to launch administration commands. These stored procedures are potentially dangerous in that they can make it possible for a malicious user to execute system commands that may lead to a possible intrusion.  
CountermeasuresA number of rules can help you protect yourself against SQL injection attacks:  
Verify the format of input data and particularly the presence of special characters; Do not display explicit error messages displaying the query or a part of the SQL query; Delete unused user accounts, and particularly default accounts; Avoid accounts without passwords; Keep the privileges of used accounts to a minimum; Delete stored procedures.  
  
**Cross site scripting:** ----Cross site scripting (also known as XSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another website, instant message, or simply just reading a web board or email message. Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked on. After the data is collected by the web application, it creates an output page for the user containing the malicious data that was originally sent to it, but in a manner to make it appear as valid content from the website.  
Often attackers will inject JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable application to fool a user (Read below for further details) in order to gather data from them. Everything from account hijacking, changing of user settings, cookie theft/poisoning, or false advertising is possible. New malicious uses are being found every day for XSS attacks. The post below by Brett Moore brings up a good point with regard to "Denial Of Service", and potential "auto-attacking" of hosts if a user simply reads a post on a message board.  
  
**Distributed denial of service:** ----Distributed denial-of-service attacks are ones in which the hacker plants malicious code on numerous, scattered and usually unwitting, servers.Those servers, known as zombies then flood a single IP address with packets so it is driven offline, unable to handle the volume.  
A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted, malevolent efforts of a person or persons to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root nameservers.  
  
**Broken anuthentication:** ---Authentication and session management includes all aspects of handling user authentication and managing active sessions. Authentication is a critical aspect of this process, but even solid authentication mechanisms can be undermined by flawed credential management functions, including password change, forgot my password, remember my password, account update, and other related functions. Because “walk by” attacks are likely for many web applications, all account management functions should require reauthentication even if the user has a valid session id.  
User authentication on the web typically involves the use of a userid and password. Stronger methods of authentication are commercially available such as software and hardware based cryptographic tokens or biometrics, but such mechanisms are cost prohibitive for most web applications. A wide array of account and session management flaws can result in the compromise of user or system administration accounts. Development teams frequently underestimate the complexity of designing an authentication and session management scheme that adequately protects credentials in all aspects of the site. Web applications must establish sessions to keep track of the stream of requests from each user. HTTP does not provide this capability, so web applications must create it themselves. Frequently, the web application environment provides a session capability, but many developers prefer to create their own session tokens. In either case, if the session tokens are not properly protected, an attacker can hijack an active session and assume the identity of a user. Creating a scheme to create strong session tokens and protect them throughout their lifecycle has proven elusive for many developers. Unless all authentication credentials and session identifiers are protected with SSL at all times and protected against disclosure from other flaws, such as cross site scripting, an attacker can hijack a user’s session and assume their identity.  
  
**Forced browsing:** ---- Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the application, but are still accessible.  
An attacker can use Brute Force techniques to search for unlinked contents in the domain directory, such as temporary directories and files, old backup and configuration files. These resources may store sensitive information about web applications and operational systems, such as source code, credentials, internal network addressing, and so on, thus being considered a valuable resource for intruders.  
This attack is performed manually when the application index directories and pages are based on number generation or predictable values, or using automated tools for common files and directory names.  
This attack is also known as Predictable Resource Location, File Enumeration, Directory Enumeration, and Resource Enumeration  
  
**Session hijacking:** --- The term session hijacking refers to the exploitation of a valid computer session - sometimes also called a session key - to gain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server. It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many web sites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim's computer  
  
**Insecure direct object reference**: --- A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. An attacker can manipulate direct object references to access other objects without authorization, unless an access control check is in place.  
For example, in Internet Banking applications, it is common to use the account number as the primary key. Therefore, it is tempting to use the account number directly in the web interface. Even if the developers have used parameterized SQL queries to prevent SQL injection, if there is no extra check that the user is the account holder and authorized to see the account, an attacker tampering with the account number parameter can see or change all accounts.  
  
**WS-Security:** ---(Web Services Security) is a communications protocol providing a means for applying security to Web services.officially called WSS and developed via committee in Oasis-Open.  
The protocol contains specifications on how integrity and confidentiality can be enforced on Web services messaging. The WSS protocol includes details on the use of SAML and Kerberos, and certificate formats such as X.509.  
WS-Security describes how to attach signatures and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.  
WS-Security incorporates security features in the header of a SOAP message, working in the application layer. Thus it ensures end-to-end security.  
  
**Principle of least privilege:** --- In information security, computer science, and other fields, the principle of least privilege, also known as the principle of minimal privilege or just least privilege, requires that in a particular abstraction layer of a computing environment, every module (such as a process, a user or a program on the basis of the layer we are considering) must be able to access only such information and resources that are necessary to its legitimate purpose.[1][2]  
When applied to users, the terms least user access or least-privileged user account (LUA) are also used, referring to the concept that all users at all times should run with as few privileges as possible, and also launch applications with as few privileges as possible