1. (10 points) Identify all of the following statements that are true about the basics of services.
   A. The Basic Profile 1.0 is a standard that specifies which standards to use for interoperable web services
      Solution: A is true:
   
   B. We should adopt object-oriented design approaches for service-oriented systems because do so yields services at a fine granularity
      Solution: B is false: O-O approaches yield services at a fine granularity but that is precisely the reason not to adopt them
   
   C. When we model legacy applications as services, we can end up with services viewed at an undesirably coarse granularity
      Solution: C is true: legacy applications tend to be monolithic and thus need to be refactored to facilitate reuse
   
   D. A technical requirement resulting from supporting revisions is that the services involved be sufficiently long-lived both to provide revisions and to consume revisions provided by others
      Solution: D is true:
   
   E. Asynchronous communications promote loose coupling and thus are unsuited for open environments
      Solution: E is false: loose coupling is indeed desirable for open environments

2. (4 points) Consider a value map between the set of letters, \(a = \{A \ldots Z\}\) (sorted with A as lowest and Z as highest) to the set of integers, \(b = \{1 \ldots 5\}\). As usual, we consider the value map as two functions, \(m_{ab}\) and \(m_{ba}\). Let \(m_{ab}\) map \(A, B, C, D\) to 1, 2, 3, 4 (respectively), and map all the letters \(\{E \ldots Z\}\) to 5.
   A. \(m_{ba}\) is necessarily order preserving
      Solution: A is false:
   
   B. If \(m_{ab}\) is consistently inverting, \(m_{ba}\) must map the number 5 to one of the letters \(\{E \ldots Z\}\)
      Solution: B is true: if the number 5 mapped to a letter in \(\{A \ldots D\}\), we would have \(m_{ab}(m_{ba}(m_{ab}(E))) \neq m_{ab}(E)\)—hence, \(m_{ab}\) would not be consistently inverting

3. (26 points) Identify all of the following statements that are true about knowledge modeling, RDF, and OWL
   A. An articulation axiom describes how concepts in one ontology map to concepts in another ontology
      Solution: A is true:
   
   B. A desirable property of a conceptualization is that it be elaboration tolerant
      Solution: B is true:
C. Logic-based conceptualizations make it difficult to perform reasoning about whatever they represent

**Solution:** C is false: logic mainly facilitates sound reasoning

D. In a well-formed RDF document, the domain and range of each property must be explicitly listed

**Solution:** D is false:

E. If we declare a resource as an instance of a class, e.g., via ⟨Building rdf:ID='MRC'⟩, then the resource is of rdf:type of the class, e.g., the resource named MRC has rdf:type equal to Building

**Solution:** E is true:

F. An RDF model naturally corresponds to a graph whose vertices correspond to resources and literals, and whose edges correspond to properties

**Solution:** F is true: a resource vertex may be a subject or an object; a literal vertex must be an object

G. In RDF, we can capture a two-party relationship with an attribute (on the relationship) via an association entity represented as a resource

**Solution:** G is true:

H. Reification in RDF is sometimes shown pictorially with an ellipse for a statement surrounding a triple; thus reification represents a situation where an RDF model does not correspond to a graph

**Solution:** H is false: reification simply means that a statement is modeled as a resource, and hence can be shown as an ellipse; drawing that ellipse around a statement is merely a convention employed in the lectures to emphasize reification; the subject, object, and predicate properties of a statement-as-a-resource are the only means to extract structure from a statement

I. We can use RDF Schema to define a custom vocabulary that our instance documents can use to specify models of particular web resources

**Solution:** I is true:

J. If we assert that medico is a superclass of physician, then we can send in a medico for a service (medical procedure) that requires a physician

**Solution:** J is false:

K. Consider that we have defined two classes, Person and Parent and properties hasParent and parentOf in OWL; then, it is impossible to assert that Person is a subclass of Parent

**Solution:** K is false:

L. In OWL, it is possible to define two classes, Foo and Bar, and assert that Foo is a subclass of Bar and Bar is a subclass of Foo
Solution: L is true: When Foo and Bar are equal, each is a subclass of the other

M. In OWL, subclass is a transitive property

Solution: M is true: