1. (24 points) Identify all of the following statements that are true about the basics of services.
   A. If you know that two parties implement SOAP, then you can safely conclude they will interoperate at least in terms that SOAP addresses, namely, at the level of exchanging requests and responses
   B. Programmers need to learn the SOAP schema so as to offer and use Web services.
   C. Every URI is a URL
   D. Screen scraping reflects the recommended best practice for extracting information from a web site
   E. UDDI not only provides a registry but also provides hosting for services that a consumer can discover and execute on the UDDI engine
   F. UDDI provides a natural basis for trusting services
   G. The definition of services we have adopted in this course is that services are objects accessed remotely
   H. The classical “triangle” architecture for services defines a way for a service consumer to send a request to a service
   I. The classical “triangle” architecture for services defines three main roles: a service consumer, a service provider, and a service monitor
   J. WSDL provides a way to describe a service based on its input and output signatures, but largely ignores the functionality the service provides
   K. In important use cases, a service provider and consumer carry out conversations, which the basic Web services standards (SOAP, WSDL, UDDI) do not support
   L. A fundamental assumption of services is that though the participants may be autonomous, they are never selfish and will do always do right by others

2. (8 points) Consider a value map between the set of letters, \( a = \{ A \ldots Z \} \) (sorted with \( A \) as lowest and \( Z \) as highest) and a set of integers, \( b = \{ 1 \ldots 4 \} \). As usual, we consider the value map as two functions, \( m_{ab} \) and \( m_{ba} \). Let \( m_{ab} \) map \( \{ A, B \} \) to 1; \( \{ C, D \} \) to 2; \( \{ E, F \} \) to 3; and map all the remaining letters \( \{ G \ldots Z \} \) to 4.
   The following are mutually independent problems based on the above setting.
   A. \( m_{ab} \) is order preserving and total
   B. If \( m_{ba} \) is order preserving and total, then \( m_{ab} \) is necessarily consistently inverting
   C. If \( m_{ba} \) maps any two numbers to the same letter, it cannot be consistently inverting
   D. No matter what \( m_{ba} \) is, if it is total and if \( m_{ab} \) and \( m_{ba} \) are consistent inverses of each other, then \( m_{ba} \) must be order preserving

3. (24 points) Identify all of the following statements that are true about knowledge modeling and RDF
   A. When author in one conceptual model is creator in another conceptual model, this is an example of the process dimension of abstraction
   B. That the expected shipping time is always positive is an example of the data dimension of abstraction
   C. Whether you can cancel a booking or handle other contingencies is an example of the policy dimension of abstraction
   D. A desirable property of a conceptualization is that it be modular
E. A major advantage of a formal conceptualization is that it enables us to infer appropriate relationships without having to state each of the relationships explicitly

F. We can apply an articulation axiom in both directions: to map concepts in one ontology map to concepts in another ontology, and in the reverse direction

G. In RDF, we can capture a multiparty relationship via an association entity represented as a resource

H. An RDF validator (i.e., reasoner) would flag conflicting assertions about the same resource

I. In a well-formed RDF document, you cannot have two (different) assertions regarding the range of the same property

J. A well-formed RDF Schema document is necessarily a well-formed XML document

K. A possible use for reification in RDF is to enable asserting comments about descriptions of services

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

4. (44 points) Identify all of the following statements that are true about RDF and OWL

A. OWL is nothing but a vocabulary for ontologies specified using RDF with constraints on how the terms in the vocabulary are to be interpreted

B. Anything you can represent in OWL DL, you can represent in RDF with the appropriate meaning conventions established

C. Anything you can represent in RDF, you can represent more precisely in OWL DL

D. Using the OWL DL restriction construct, we can define a class based on a property (and incorporating suitable additional constraints)

E. Let $F$ be a functional property in OWL DL with domain $D$ and range $R$. Then for each instance $d$ of $D$, there must be an instance $r$ of $R$ such that $d$ relates to $r$ through property $F$

F. Let $F$ be a functional property in OWL DL with domain $D$ and range $R$. Then for each instance $r$ of $R$, there must be an instance $d$ of $D$ such that $d$ relates to $r$ through property $F$

G. If an OWL DL object property is both symmetric and transitive, it is necessarily reflexive, meaning that each instance of the property’s domain relates to itself through the property

H. If you assert in OWL that both Class-A and Class-B are domains of the same property, then the OWL reasoner would conclude that Class-A is equivalent to Class-B

I. In OWL, it is possible to define a property whose domain and range are the same class

J. Because OWL has no variables, it is not possible to define a well-formed model in OWL that includes a cycle when rendered as a graph (with classes as vertices and properties as edges)

K. We cannot correctly declare a resource to be an instance of two OWL classes in the same OWL document—for example, via ⟨Person rdf:ID='Pradeep'/> ⟨Student rdf:about='Pradeep'/⟩

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

M. In OWL owl:FunctionalProperty is actually a class, i.e., having rdf:type owl:Class

N. If we assert that Medico is a superclass of Physician, then any property whose domain is Physician may potentially be asserted for an instance of Medico

O. If we assert that Medico is a superclass of Physician, then any property whose range is Medico may potentially be asserted mapping an instance of its domain to an instance of Physician

P. 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

Q. In OWL, it is acceptable to define two classes, Foo and Bar, and assert that Foo is a subclass of Bar and Bar is a subclass of Foo
R. In OWL DL, subclass is a property whose domain is owl:Class
S. In OWL DL, we can assert that Class-A and Class-B are mutually exclusive subclasses of Class-C
T. In OWL DL, we cannot assert that Class-A and Class-B are mutually exhaustive subclasses of Class-C
U. An OWL DL data property cannot be symmetric
V. An OWL DL data property can be transitive

5. (14 points) Problems on dependencies and events (here ⌜ and ⌝ are delimiters)
   A. Our formal syntax is such that the disjunction of two or more dot expressions involving different events,
      for example, \((e \cdot f \lor d \cdot \overline{f} \lor \overline{h} \cdot k)\), is a (well-formed) dependency;
   B. Our formal syntax is such that the disjunction of two events is itself an event
   C. Our formal syntax is such that the conjunction of two events is itself an event
   D. The requirement ⌜If \(e\) then \(f\) and if \(f\) then \(\overline{e}\)⌝ may be expressed as \(\overline{e}\)
   E. The two dependencies, \(\overline{e} \lor e \cdot f\) and \(\overline{f} \lor f \cdot e\), cannot both be satisfied by the same schedule
   F. Given a dependency, we can allow any event whose residual with that dependency is nonzero
   G. It is not possible to have a dependency \(D\), where \(D \neq \top\), such that \(D/e = D/\overline{e}\)