1. (14 points) Of the following statements, identify all that hold about e-business concepts.

   A. Autonomy entails heterogeneity: two business partners that are mutually autonomous are obviously of independent construction as well

   > **Solution:** A is false: the two business partners might be autonomous but implementing copies of the same software package, e.g., of popular enterprise software packages such as the products of SAP and Oracle

   B. You would be well-advised to avoid a tightly integrated solution when you can—to simplify maintenance among other reasons

   > **Solution:** B is true: a tightly integrated solution would be difficult to maintain

   C. TP monitors apply well in settings such as banks that must ensure consistency of the account data they hold

   > **Solution:** C is true: a TP monitor can work well within an administrative domain to ensure consistency of the data; banking would be an ideal setting for such functionality

   D. Traditional techniques emphasize consistency at the cost of coherence and are therefore not suited to open environments

   > **Solution:** D is true: traditional techniques almost exclusively focus on consistency and disregard coherence

   E. The dynamism of an IT system refers to the fact that the staff in a real-life enterprise changes dynamically and therefore one can never presume that the same staff will be available for long

   > **Solution:** E is false:

   F. What makes e-commerce intellectually different from conventional computer science is the extensive use of XML in e-commerce settings

   > **Solution:** F is false: what makes e-commerce intellectually different from conventional computer science is having to deal with openness

   G. Some techniques created to deal with open environments may also be applied in settings that are not necessarily open in that they involve the same administrative domain

   > **Solution:** G is true:

2. (50 points) Of the following statements, identify all that hold about architecture.

   A. The purpose of an IT architecture is to identify the main uses of a system

   > **Solution:** A is false: the purpose of an architecture is to *support* the main uses of a system

   B. An architecture of a system provides a basis for how it is built and administered

Solution: B is true: that is the point of developing an architecture for a system

C. An architecture is largely determined by the physical components you wish to use to instantiate the architecture

Solution: C is false: an architecture applies at a higher level of description than the physical components

D. An IT architecture often maps naturally to the staffing of the resulting IT system in terms of the roles and skills of the people involved and how they are organized

Solution: D is true:

E. Adopting an architecture can often lead to additional organizational roles that you might not need if it weren't for the adopted architecture

Solution: E is true: an architecture may introduce a new type of component for which we might need new roles to administer and use it; for example, a DBA is needed only because we have databases

F. The primary benefit of an architecture is to specify the internal details of how a component would be implemented

Solution: F is false: the opposite, because an architecture helps us emphasize the interconnections among components and disregard the implementations of individual components

G. The openness of an architecture means that it doesn't specify the physical components but instead specifies the interconnections cleanly

Solution: G is true: an open architecture doesn't specify physical components but rather their logical traces as reflected in the interconnections

H. Something that is clearly a connector, such as a network, should not be modeled as if it were a component

Solution: H is false: we may model such an entity as a component or a connector depending on what else we want to express about it in our architecture

I. An architectural style specifies the patterns involving components and interconnections

Solution: I is true:

J. Protocols make sense as abstractions for networking architectures but do not make sense in the application layers

Solution: J is false: protocols make sense wherever we have interconnections among components from an IT standpoint

K. A benefit of using protocols is that they make interactions explicit and modular, thereby decoupling implementations and enhancing their maintainability

Solution: K is true:

L. We can view a relational database schema in architectural terms wherein the tables are components and foreign keys are interconnections

Solution: L is true:

M. The tiers in a three tier architecture represent a logical, rather than a physical, partitioning of an application

Solution: M is true:

N. It may sometimes be helpful to have the presentation tier directly connect with the data tier; therefore, the three-tier architecture supports direct connections between the presentation and the data tiers

Solution: N is false:

O. The three-tier architecture does not provide suitable high-level abstractions for describing service engagements among business partners

Solution: O is true: service engagements involve concepts founded on business relationships, not control or data flow, which three-tier architectures address

P. What motivate our choice of an architecture for a particular system are the requirements of the stakeholders of the system

Solution: P is true: the requirements of the stakeholders capture the business needs that the architecture is geared to accommodate

Q. Governance refers to the administration of a system according to the requirements of the stakeholders of the system

Solution: Q is true:

R. The stakeholders of a system impose consistent requirements on it

Solution: R is false: all too often, different stakeholders have conflicting requirements

S. The set of stakeholders involved in governance can change over time

Solution: S is true: real-life organizations change all the time—e.g., consider the set of students enrolled in our university at any time

T. Governance pertains to a system that is already implemented and operating; thus, governance is not concerned with the design or development of a system

Solution: T is false: governance includes administration of all aspects of the life cycle of a system

U. Policies are motivated by requirements of flexibility and comprehensibility

Solution: U is true: policies facilitate understanding a system's strategies and updating them more easily than in conventional systems

V. One administrative aspect governance is not concerned with is accommodating change in a system as the requirements change

Solution: V is false: accommodating change, for example, by updating, creating, or removing services is indeed an aspect of governance

W. Governance is how a given system is managed

Solution: W is false: governance is a different way of administering a system than management

X. Governance relates stakeholders to the system architecture

Solution: X is true: governance concerns how a system is administered by its stakeholders and thus influences the architecture of the system

Y. The primary stakeholders of a system are those who pay for its construction and maintenance

Solution: Y is false: the stakeholders are those who have a stake in the functioning of the system; who pays or does not is not relevant

3. (46 points) Of the following statements, identify all that hold about architecture.

A. Adopting policies leads naturally to an architecture whose important components include the policy enforcement point and the policy decision point

Solution: A is true:

B. The infrastructure modules carry the most direct bearing on the users' requirements

Solution: B is false:

C. Applications, infrastructure, and systems form an hourglass view because many applications run on a few kinds of infrastructure, which maps to many systems modules

Solution: C is false: in general, there are many kinds of infrastructure that support a few key systems

D. The business processes of an enterprise ought to drive the technical architecture adopted by the enterprise

Solution: D is true:

E. The so-called ilities include nonfunctional requirements such as reliability and availability

Solution: E is true:

F. What distinguishes one architectural approach from another is whether and how it addresses the functional requirements that the system being architected must meet

Solution: F is false: all approaches must meet the functional requirements to even be considered, but whether and how they meet nonfunctional requirements is what distinguishes them (and their desirability for stakeholders) from each other

G. The end users of a system impose only the functional requirements on it

Solution: G is false: they also impose some of the nonfunctional requirements, e.g., delay in obtaining a result

H. Protocols specify a representation for the information they help exchange, including data formats and data types

Solution: H is true:

I. Protocols cannot be correctly specified using finite state machines

Solution: I is false:

J. Messaging middleware supports asynchronous communication between end points

Solution: J is true:

K. We should include local method calls in a sequence diagram that represents a business protocol

Solution: K is false: local method calls are internal to each party; we are primarily concerned with their interactions

L. Given a statechart state B is nested within state A, when the modeled entity is in state B, it is automatically in state A

Solution: L is true:

M. When a statechart state B is nested within state A, there can be no transition that begins from state B and ends outside state A

Solution: M is false: such transitions are possible and our example statechart of the commitment life cycle exemplifies them

N. An operational specification describes *what* we need, not *how* we achieve it

Solution: N is false: it is reverse

O. Our statecharts for delegating a commitment show two commitments in parallel, the original commitment and the delegated commitment

> **Solution:** O is true:

P. Reciprocal commitments, such as $\mathsf{C}(\text{buyer}, \text{seller}, \textit{ship}, \textit{pay})$ and $\mathsf{C}(\text{seller}, \text{buyer}, \textit{pay}, \textit{ship})$, entail that neither party has an incentive to act

> **Solution:** P is false: each party may have an incentive to act (and usually does for well-negotiated commitments); whether each party has an incentive to act depends on how it values the deal being described and if it trusts the other party to perform as it commits

Q. The only legal way to enact a commitment $\mathsf{C}(\text{buyer}, \text{seller}, \textit{ship}, \textit{pay})$ is for the seller to ship the goods first and the buyer to pay later

> **Solution:** Q is false:

R. A business partner in a service engagement can be viewed as a blackbox because its internal organization is irrelevant

> **Solution:** R is false: s business partner is better seen as a grey box because its internal organization is relevant in some scenarios, especially as dealing with exceptions

S. A formal model for contracts enables us to check for potential design flaws and anomalies such as having unbounded scope

> **Solution:** S is true:

T. An implementation clause in a contract specifies the quality of service guarantees given by a contracting party

> **Solution:** T is false: an implementation clause specifies only how a party's part in the contract is to be implemented, not the quality of service to be achieved

U. To transfer responsibility via a delegation, the original commitment should move to the pending state when the delegation occurs

> **Solution:** U is false: the original commitment is not needed any more and should move to the null state

V. Policies are the essential elements of contracts

> **Solution:** V is false:

W. A contract template, such as for leasing an apartment, would specify two or more roles along with the contract façades of each role

> **Solution:** W is true: a contract template specifies the roles it involves and what it expects from or grants to parties adopting each such role