1. Describe how a Naming Service is similar to and different from a Directory Service. Using DNS and LDAP as respective examples, explain their different features and use ca

   Common characteristics between DNS and LDAP:
   namespace is hierarchical;
   transmit protocol is TCP/IP;
   the scale is global

   Differences between DNS and LDAP:
   DNS store data in files as resource records; LDAP store data in directories in indexed database
   The servers of DNS is in the mode of Master/Slave; The servers of LDAP is in the mode of Mater/Replica
   LDAP use SSL to increase security; while DNS do not

2. Discuss the problem associated with name services in a distributed system. How can this be solved?

   Name services are designed to bind names to addresses in a single management domain, which is simple initially. But the interconnection of networks and the increased scale of distributed systems produce problems in name services. The problems include:
   a) the number of names and administrative organizations is arbitrary.
   b) long lifetime.
   c) high availability
   d) fault isolation
   e) tolerance of mistrust

   We can use hierarchical name spaces to handle the growth of names and organizations; aliases can be used to substitute the complicated names; naming domains can manage and update the names stored by the name service; replicate heavily used name services to achieve high availability; each subset of database is replicated in at least two failure-independent servers; maintain a cache of the results of previous name resolutions.

3. Describe the heterogeneity of the Distributed Computing Environment (DCE) name space. What is its cell and junction in this context? Give an example.

The distributed Computing Environment(DCE) name space allows heterogeneous name spaces to be embedded within it.
A DCE cell is an administrative and security domain within the cell name space. DCE names may contain junctions, which allow heterogeneous name spaces to be mounted. For the DCE name /.../cs.univ.edu/fs/usr/ziggy, the first part of this name, /.../cs.univ.edu is a cell, the next component fs is a junction. And the final component usr/ziggy may be looked up in the distinct syntax. The junction is the root of heterogeneous name spaces, implemented by heterogeneous name services.

4. A DNS client is called a resolver. What is its role?

A DNS client is normally implemented as library software, which accepts queries from users, formats the queries into messages in DNS protocol , and communicates with one or more name servers to satisfy the queries. The resolver can be configured to contact a list of initial name servers to increase the availability. The resolver specifies the required types of navigation to be recursive navigation or iterative navigation when contacting a name server.

5. What is mobile computing? What are the different problems associated with wireless communication?

Mobile computing is exploiting the connectedness of portable devices such as laptops, mobile phones to transmit data with no physical link.
The set of users, hardware and software in mobile and ubiquitous systems is highly dynamic and changes unpredictably. Failures of devices and communication links happen frequently; changes in the characteristics of communication like bandwidth are unpredictable; the creation and destruction of association is unpredictable.
Problems associated with wireless communication:

a) Disconnection: wireless disconnections happen far more frequently. There are many factors leading to disconnection, like distance from devices to wireless connection, and multi-hop wireless routing between devices.
b) Variable bandwidth and latency: the changing error rate is high in wireless connection.

6. What is meant by the “physically driven spontaneity of associations”? How is this different from associations that are data-driven?

In volatile systems, components routinely change the set of components they communicate as they move or when other components appear in their environment. As devices change their physical locations, they might appear in different associations (the logical relationship formed when at least one of a given pair of components communicates with others). This is “physically driven spontaneity of associations”.

Components in data-driven association are mainly dependent on data. For example, in file-sharing programs, the value of the data makes a peer to make associations with a peer to transmit the file. While physically-driven spontaneity of association are made and broken according to the current physical circumstances of the components, and in particular their proximity.

7. Explain the data-oriented interoperability features of tuple space systems with an example.

Tuple space systems is an indirect communication paradigm that supports the addition and retrieval of tuple structured data, called tuples, to and from a tuple space. The basis for association and interoperation is the components' agreement about structures for tuples and values contained within them. For example, we can discover the tuple space to record the information of students, assume we need to record the name, age and a photo of a student, then we can use a tuple as:

<'Aric Black', 25, <jpeg data>>

This tuple space creates a model to store the information of students, so data of students should be placed in the certain format.

8. What does the association problem comprise? What are the two important aspects of the association problem? What is the boundary principle?

Association problem: how to associate appropriately within the association.

The two important aspects of the association problem are: scale and scope. There are many devices in the smart space, we need to decide which components on the appearing device to interoperate. And we also need to consider constraining the scope when solving the problem.

The boundary principle states that smart spaces need to have system boundaries that correspond accurately to meaningful spaces which are system-defined criteria that scope but do not necessarily constrain association.

9. Discuss the applicability to mobile and ubiquitous systems of techniques drawn from the areas of:

i) peer-to-peer systems (Chapter 10);

ii) coordination and agreement protocols (Chapter 15);

iii) replication (Chapter 18).

i) Peer-to-peer systems have some common characteristics with mobile and ubiquitous systems. Peer-to-peer systems are volatile, object locations can be randomized. Peer-to-peer systems ensure each user contributes resources to the system, and can address more objects. Peer-to-peer systems have some benefits, like the ability to exploit unused resources in the host computers, the scalability to support large numbers of clients and hosts and the self-organizing properties of the middleware platforms.

When apply peer-to-peer systems to mobile and ubiquitous systems, we should focus on the anonymity problem and reliability.

ii) Coordination and agreement protocols are designed for handling failure and reaching consensus. In mobile and ubiquitous systems, it might be expensive to apply the protocols since the systems are unreliable and failures happen frequently. We can use weaker protocols like Cooltown’s eSquirt protocol to interoperate between devices in volatile systems.

iii) Replication of data can improve the speed of getting responses and data sharing in mobile and ubiquitous systems. Also, when one server disconnects, the replica can handle the request. Although it might need more space to store the replication, the replica can make the systems more reliable and efficient.