1-) SOAP vs REST:

SOAP stands for simple access protocal. A messaging protocol specification for exchanging structered data.

REST is an arcihtectural structure that specifically designed to work with components such as media, files and etc. Which stands for Representational stage transfer.

For first difference, SOAP is design that is based on pre-defined rules to follow. REST is an architectural style with loose guidelines.

Second difference would be as in the ways of approach, SOAP is functional driven, which means data and services are available with functions such as "get user".

The third difference would be state. SOAP is stateless by default even though it is possible to make an API with SOAP that is stateful. In REST, no server side sessions really occur. Therefore it is stateless.

Fourth difference is on caching; SOAP API calls cannot be cached.On the other hand REST API calls can be cached.

Fifth difference is about security. SOAP has WS- Security with SSL support and alsa has a built-in acidcompliance which stands for atomicity, consistency, isolation and durability. Therefore it is very useful in sensetive data transaction. REST supports HTTPS and SSL.

For the last difference, REST requires fewer resources.

2-)Difference between acceptance test and functional test:

Functional test covers the possible scenarios and is done with all possible scenarios, even the scenario is unlikely to occur in real world. Maximum code coverage is aimed.

In Acceptance testing, typical scenarios that the software would expected to face. It may be called beta testing aswell.

3-) What is Mocking?

Creating a dummy version of a service which can stand in for real one.

4-) What is reasonable code coverage % for unit tests (and why)?:

Reasonable code coverage would be 80% of frequently used code- paths. In big projects it wouldn't be efficient to cover all of the code. 80% is mean percentage that can provide good results and efficiency.

5-) HTTP/POST vs HTTP/PUT?

"PUT" accepts the body of the reqyest and stores it; similiar to a file upload. "POST" is more general. It innitiaets an action on the server.

6-) Safe and Unsafe methods of HTTP?:

HTTP method is said to be safe if it does not alter the state of the server. If the mehod leads to a read- only operation, it is refered to as "safe". GET, HEAD, OPTIONS are common safe methods.

Unsafe methods may lead ab attacjer tı steak credentials. Common unsafe methods are; PUT, DELETE, POST.

7-) HTTP Basic Authentication

A simple challange and response mechanism. A server requests a user ID and password for authenticaiton. Then encodes it with Base64 and puts it on header with key authorization.

8-) RestTemplate in Spring:

RestTemplate is the core Spring Class for client-side HTTP access. It is customizable and thread-safe once constructed.

9-) idempotant

Refers to methods that can be called several times without changing the state of the server. GET, HEAD, PUT and delete methods are idempotent.

10-) DNS Spoofing:

DNS cache poisining. Online traffic is redirected with altered DNS records. Users that are promted to login into their accounts, sensitive information and access credentials can be stolen. Several types of dns spoofing includes, Man in the middle, dns server compromise.

To prevent it;

DNSSEC can be used to protect the server’s register from outside tempering.

DNS traffic-filtering can be used to prevent dns spoofing.

11-) Content Negotiation:

Process of negotiation between the server and the client for the format of the transfered data.

12-) Statelessness

Since in REST architacture, server does not hold any state information about the instance, this restiriction is refered to as statelessness. Each client request to the server must include necessary information for server to proceed with the request. Each HTTP request occurs in total isolation.

13-) CSRF:

Cross-site request forgery is a form of vulnerability that allows the attacker to manipulate the actions of the users. Attacker causes the user to take unintended action. It can be for example, changin the email on their account. The circumstances while the action is taken may give the attacker the chance to access full control over the account.If the “contaminated” user has a priveleged role within the application; the attacker might access all the data.

The most robust way to prevent such attacks is to include a CSRF token within the relevant requests.

14-) Core components of HTTP Request and Response?:

For response:

1- Status/Response Code

2- HTTP Version

3- Response Header

4- Response Body

For Request:

1- Method/Verb

2- URI

3- HTTP Version

4- Request Header

5- Request Body