1) What is AngularJS?

AngularJS is a JavaScript framework i.e. used to create single web page application. It follows MVC (Model View Controller) pattern. It is open source, cross browser compliant and easy to maintain.

2) What are the advantages of AngularJS?

- allows us to create single page application
- follows MVC pattern
- predefined form validations
- supports animation
- open source
- cross browser compliant
- supports two way data binding
- its code are unit testable

3) What are the disadvantages of AngularJS?

- **JavaScript Dependent**: If end user disables JavaScript, AngularJS will not work.
- **Not Secured**: It is JavaScript based framework so it is not safe to authenticate user through AngularJS only.

4) Is AngularJS dependent on JQuery?

No

5) What are the features of AngularJS?

1. MVC, Validations
2. Modules, Directives, Templates
3. Scope, Expressions, Data Binding
4. Filters, Services, Routing
5. Dependency Injection, Testing

6) Explain what is scope in AngularJS?

Scope refers to the **application model**, it acts like glue between application controller and the view. Scopes are arranged in hierarchical structure and impersonate the DOM (Document Object Model) structure of the application.

The $scope is a special JavaScript object. Both View and controller have access to the scope object. It can be used for communication between view and controller. Scope object contains both data and functions.

Every AngularJS application has a $**rootScope** that is the top most scope created on the DOM element which contains the ng-app directive.

Each AngularJS application can have only one root scope but can have multiple child scopes. It can watch expressions and propagate events. Using **rootScope** we can set the value in one controller and read it from the other controller.

7) What are directives in AngularJS?

Directives are the markers on DOM element that is used to specify behavior on that DOM element. All AngularJS directives start with the word "ng". There are many in-built directives in AngularJS such as "ng-app", "ng-model", "ng-controller", "ng-repeat" etc.

Let's see a simple example of AngularJS directive.

```
<input type="text" id="empName"  ng-model="EmpName"/>
```

8) Which are core directives of AngularJS?

Following are the three core directives of AngularJS.

- **ng-app** − This directive defines and links an AngularJS application to HTML.
- **ng-model** − This directive binds the values of AngularJS application data to HTML input controls.
- **ng-bind** − This directive binds the AngularJS Application data to HTML tags.

9) What are controllers in AngularJS?

A controller is a set of JavaScript functions which is bound to a specified scope, the **ng-controller** directive. Angular will instantiate the new controller object, and injects the new scope as a dependency. It contains business logic for the view and avoids using controller to manipulate the DOM.

ng-controller directive tells AngularJS what controller to use with this view. AngularJS application mainly relies on controllers to control the flow of data in the application.

AngularJS Controllers are used to:

- Set up initial state of the $scope object, and
- Add behavior to the $scope object.

```
<div ng-app="mainApp" ng-controller="SimpleController">
</div>
```

10) What is data binding in AngularJS?

Data Binding is the automatic synchronization of data between model and view. ng-model directive is used in data binding. There are two ways of data binding:

- One way data binding (used in classical template)
- Two way data binding (used in AngularJS template)

11) What are services in AngularJS?

Services are singleton objects that can be used to store and share data across the application. AngularJS offers many built-in services such as $http i.e. used to make XMLHttpRequest.

The AngularJS Module defines three methods for defining services: **factory, service and provider**. The result of using these methods is the same – a service object that provides functionality that can be used throughout the AngularJS application. E.g.

- *$window* Provide a reference to a DOM object.
- *$Location* Provides reference to the browser location.
- *$timeout* Provides a reference to window.settimeout function.
- *$Log* Used for logging.
- *$sanitize* Used to avoid script injections and display raw HTML in page.
- *$Rootscope* Used for scope hierarchy manipulation.
- *$Route* Used to display browser based path in browser URL.
- *$Filter* Used for providing filter access.

The simplest method of defining a service is to use the Module.factory method, passing an argument, the name of the service and a factory function that returns the service objects.

AngularJS services are designed based on two principles.

1. **Lazily instantiated**
   Angular only instantiates a service when an application component depends on it using dependency injection for making the Angular codes robust and less error prone.

2. **Singletons**
   Each component is dependent on a service that gets a reference to the single instance generated by the service factory.

12) What are expressions in AngularJS?

Expressions are the code snippets that resolves to a value. AngularJS expressions are placed inside {{expression}}. For example: {{1+1}}

AngularJS supports one-time binding expressions. Expressions are used to bind application data to html. The key difference between the JavaScript expressions and Angular expressions

- **Context :** In Angular, the expressions are evaluated against a scope object, while the Javascript expressions are evaluated against the global window
- **Forgiving:** In Angular expression evaluation is forgiving to null and undefined, while in Javascript undefined properties generates TypeError or ReferenceError
- **No Control Flow Statements:** Loops, conditionals or exceptions cannot be used in an angular expression
- **Filters:** To format data before displaying it you can use filters

13) What is the use of filter in AngularJS?

A filter is used to format the value of expression to display the formatted output. AngularJS enables us to write our own filter.

We've added currency filter to print fees using currency format.

```
Enter fees: <input type = "text" ng-model = "student.fees">
fees: {{student.fees | currency}}
```

Uppercase & lowercase example:

```
Enter first name:<input type = "text" ng-model = "student.firstName">
Enter last name: <input type = "text" ng-model = "student.lastName">
Name in Upper Case: {{student.fullName() | lowercase}}
Name in Upper Case: {{student.fullName() | uppercase}}
```

Orderby filter orders the array based on provided criteria.

In below example, to order subjects by marks, we've used orderBy marks.

```
Subject:
<ul>  <li ng-repeat = "subject in student.subjects | orderBy:'marks'">
        {{ subject.name + ', marks:' + subject.marks }}
      </li>
</ul>
```

14) What is Module?

AngularJS module is nothing but a container of all angular components like controller, services, directive, filter, config etc. Angular module is an entry point to application. Using module we can decide how the AngularJS application should be bootstrapped.

We can create a simple module using the following code. In the above code myModuleApp is the module name and if this module is dependent on other modules we can inject in "[]".

```
var myApp = angular.module('myModuleApp',[]);
```

15) What is service method?

Using service method, we define a service and then assign method to it. We've also injected an already available service to it.

```
mainApp.service ('CalcService', function (MathService) {
   this.square = function (a) {
      return MathService.multiply (a, a) ;}   });
```

16) What is factory method?

Using factory method, we first define a factory and then assign method to it.

```
var mainApp = angular.module("mainApp", []);
mainApp.factory('MathService', function() {
   var factory = {};
   factory.multiply = function(a, b) {
      return a * b
  }
   return factory;
});
```

17) Explain ng-repeat directive?

It iterates over a collection of items and creates DOM elements. It constantly monitors the source of data to re-render a template in response to change. The ng-repeat directive creates a new scope for each element of a collection.

Syntax of ng-repeat

```
<table class="table table-bordered">
    <tr ng-repeat="empin empDetails">
        <td>{{emp.name}}</td>
        <td>{{emp.salary}}</td>
    </tr>
</table>
```

Below are the some important variables created by ng-repeat

1. $index
2. $first
3. $middle
4. $last

18) How to make an ajax call using Angular JS?

AngularJS provides $https: control which works as a service to make ajax call to read data from the server. The server makes a database call to get the desired records. AngularJS needs data in JSON format. Once the data is ready, $https: can be used to get the data from server in the following manner:

```
function studentController ($scope, $https:) {
   var url = "data.txt";
   $https:.get (url).success (function (response) {
      $scope.students = response;
   });
}
```

19) How to implement routing in AngularJS?

It is concept of switching views. AngularJS based controller decides which view to render based on the business logic.

Routing is a core feature in AngularJS. This feature is useful in building SPA (Single Page Application) with multiple views. In SPA application, all views are different Html files and we use

Routing to load different parts of the application and it's helpful to divide the application logically and make it manageable.

In other words, Routing helps us to divide our application in logical views and bind them with different controllers.

```
var app = angular.module("AngularApp", ['ngRoute']);
app.config(['$routeProvider',
    function($routeProvider)
    {
        $routeProvider.
        when('/page1',
             {
                templateUrl: 'Modules/Page1/page1.html',
                controller: 'Page1Controller'
                  })
                 .
              when('/page2',
                  {
                      templateUrl: 'Modules/Page2/page2.html',
                      controller: 'Page2Controller'
                  })
                 .
              otherwise
              ({
                  redirectTo: '/page1'
              });
          }
      ]);
```

$**routeProvider** is the key service which set the configuration of urls, maps them with the corresponding html page or ng-template, and attaches a controller with the same.

20) How are validations implemented in AngularJS?

AngularJS enriches form filling and validation. We can use $dirty and $invalid flags to do the validations in seamless way. Use novalidate with a form declaration to disable any browser specific validation.

Following can be used to track error.

- **$dirty** – states that value has been changed.
- **$invalid** – states that value entered is invalid.
- **$error** – states the exact error.

```
<p>
<input type="date" name="BirthDate" ng-model="BirthDate"
required placeholder="yyyy-MM-dd">

<span style="color:red" ng-show="myForm.BirthDate.$dirty
&& myForm.BirthDate.$invalid">

<span ng-how="myForm.BirthDate.$error.required">
Birth Date is required.</span>

      <span ng-show="myForm.BirthDate.$error.date">
      Not a Valid Date.</span>

      <span ng-show="myForm.email.$error.email">
      Invalid email address.</span>
      </span>
      </p>
```

22) What is SPA (Single page application) in AngularJS?

Single-Page Applications (SPAs) are web applications that load a single HTML page and dynamically update that page as the user interacts with the app. SPAs use AJAX and HTML to create fluid and responsive web apps, without constant page reloads. This means much of the work happens on the client side, in JavaScript.

A SPA does not require the page to be reloaded as the user navigates to various parts of the application. This results in faster navigation, more efficient network transfers, and better overall performance for the end user.

Applications written in AngularJS are cross-browser compliant. Angular automatically handles the JavaScript code suitable for each browser.

23) Explain what is string interpolation in Angular.js ?

In Angular.js the compiler during the compilation process matches text and attributes using interpolate service to see if they contains embedded expressions. As part of normal digest cycle these expressions are updated and registered as watches.

24) Explain AngularJS boot process?

When the page is loaded in the browser, following things happen:

- HTML document is loaded into the browser, and evaluated by the browser. AngularJS JavaScript file is loaded; the angular *global* object is created. Next, JavaScript which registers controller functions is executed.

- Next AngularJS scans through the HTML to look for AngularJS apps and views. Once view is located, it connects that view to the corresponding controller function.
- Next, AngularJS executes the controller functions. It then renders the views with data from the model populated by the controller. The page gets ready.

There are two ways to bootStrap Angular Application.

- Automatic BootStrap (Coffee by Machine)
- Manual BootStrap (Handmade coffee, you can face some trouble)

Automatic:

When DOM content is loaded, Angular looks for the ngApp directive which designates application root.
If it finds ngApp directive. It loads the module associated with this directive.

```
<html ng-app='myApp'>

<head>
    <script src='angular.js'>
    </script>
    <script>
        var app = angular.module('myApp', []);
        app.controller('myController', function($scope) {
            $scope.message = 'Dear';
              });
          </script>
      </head>

      <body>
          <div ng-controller="myController">
              <p> Hi {{ message}} </p>
          </div>
```

Manual:

There is a big difference in Automatic and manual Bootstrap.

1. You do not need to attach ng-app directive with the html element.
2. You call function angular.bootstrap(document,['myApp']).

```
<!doctype html>
<html>

<body>
    <div ng-
controller="myController"> Best movie: {{MovieName}}! </div>
```

```
  <script src='angular.js'>
  </script>
  <script>
    angular.module('myApp', []).controller('MyController', ['$scope', function($scope) {
         $scope.MovieName = 'The IRON MAN';
       }]);
       angular.element(document).ready(function() {
         angular.bootstrap(document, ['myApp']);
       });
     </script>
   </body>

   </html>
```

Angular.bootstrap cannot create a module for you until you make a custom module to give it as a parameter inside.

Before bootstrapping the process you need to add controllers, directives and services etc.

Manual bootstrap comes into picture when you need more control over the initialization process, like if you want to perform an operation before Angular compiles a page.

25) Explain what is Dependency Injection in AngularJS?

It is a software design pattern in which objects are passed as dependencies. It helps us to remove hard coded dependencies and makes dependencies configurable. Using Dependency Injection, we can make components maintainable, reusable and testable.

AngularJS uses dependency with several types

- Value
- Factory
- Service
- Provider
- Constant

A simple case of dependency injection in Angular js

```
AppModule.controller("AppController", function($scope, $window, $log,$http)
{

});
```

26) What are the differences between service and factory methods?

Factory method is used to define a factory which can later be used to create services as and when required whereas service method is used to create a service whose purpose is to do some defined task.

27) What is provider?

Provider is used by AngularJS internally to create services, factory etc. During config phase(phase during which AngularJS bootstraps itself). Below mention script can be used to create MathService that we've created earlier. Provider is a special factory method with a method get() which is used to return the value/service/factory.

```
//define a module
var mainApp = angular.module("mainApp", []);
//create a service using provider which defines a method square to return
square of a number.
mainApp.config(function($provide) {
   $provide.provider('MathService', function() {

      this.$get = function() {
         var factory = {};
         factory.multiply = function(a, b) {
            return a * b;
         }
         return factory;
      };
   });
});
```

28) What is Constants in AngularJS?

Constant are like services in AngularJS in which we can define our global data. It is declared using "constant" keyword.

As we define our app-keys in Web.Config file for ASP.NET application, which further we can use anywhere in the application, likewise we can declare constant data in AngularJS globally that can be used throughout the application.

We can inject **Constant** everywhere in controller or service like any other dependency (e.g. $http).AngularJS uses Singleton structure for creating Constant dependency in our Angular application.

So, using the Constant you can create your Config.js file and it can be injected anywhere in your application.

App.js

```
var app = angular.module('ConstantApp', [])
```

Config.js file

```
app.constant('config',
{
    appName: 'Constants',
    appVersion: 2.0
});
```

Controller.js

```
app.controller('mainController', function ($scope,config) {

$scope.ApplicationName = config.appName;
}
```

View

```
<title>{{ApplicationName}}</title>
```