1. What is AngularJS?

AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain. Following are the features of AngularJS framework.

AngularJS is a powerful JavaScript based development framework to create RICH Internet Application (RIA).

AngularJS provides developers options to write client side application (using JavaScript) in a clean MVC (Model View Controller) way.

Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.

AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache License version 2.0.

2. Why to use AngularJS?

There are following reasons to choose AngularJS as a web development framework:

It is based on MVC pattern which helps you to organize your web apps or web application properly.

It extends HTML by attaching directives to your HTML markup with new attributes or tags and expressions in order to define very powerful templates.

It also allows you to create your own directives, making reusable components that fill your needs and abstract your DOM manipulation logic.

It supports two-way data binding i.e. connects your HTML (views) to your JavaScript objects (models) seamlessly. In this way any change in model will update the view and vice versa without any DOM manipulation or event handling.

It encapsulates the behaviour of your application in controllers which are instantiated with the help of dependency injection.

It supports services that can be injected into your controllers to use some utility code to fulfil your need. For example, it provides $http service to communicate with REST service.

It supports dependency injection which helps you to test your angular app code very easily.

Also, AngularJS is mature community to help you. It has widely support over the internet.

3. Why this project is called "AngularJS"?

Html has angle brackets i.e. <,> and ng sound like Angular. That’s why it is called AngularJS.

4. What are the advantages of AngularJS?

There are following advantages of AngularJS:

Data Binding - AngularJS provides a powerful data binding mechanism to bind data to HTML elements by using scope.

Customize & Extensible - AngularJS is customized and extensible as per you requirement. You can create your own custom components like directives, services etc.

Code Reusability - AngularJS allows you to write code which can be reused. For example custom directive which you can reuse.

Support – AngularJS is mature community to help you. It has widely support over the internet. Also, AngularJS is supported by Google which gives it an advantage.

Compatibility - AngularJS is based on JavaScript which makes it easier to integrate with any other JavaScript library and runnable on browsers like IE, Opera, FF, Safari, Chrome etc.

Testing - AngularJS is designed to be testable so that you can test your AngularJS app components as easy as possible. It has dependency injection at its core, which makes it easy to test.

5. How AngularJS is different from other JavaScript Framework?

Today, AngularJS is the most popular and dominant JavaScript framework for professional web development. It is well suited for small, large and any sized web app and web application.

AngularJS is different from other JavaScript framework in following ways:

AngularJS mark-up lives in the DOM.

AngularJS uses plain old JavaScript objects (POJO).

AngularJS is leverages with Dependency Injection.

6. What IDEs you can use for AngularJS development?

AngularJS development can be done with the help of following IDEs:

Visual Studio 2012, 2013, 2015 or higher

Eclipse

WebStorm

Sublime Text

TextMate

7. Does AngularJS has dependency on jQuery?

AngularJS has no dependency on jQuery library. But it can be used with jQuery library.

8. How to use jQuery with AngularJS?

By default AngularJS use jQLite which is the subset of jQuery. If you want to use jQuery then simply load the jQuery library before loading the AngularJS. By doing so, Angular will skip jQLite and will started to use jQuery library.

9. Is AngularJS a library, framework, plugin or a browser extension?

AngularJS is a first class JavaScript framework which allows you to build well structured, easily testable, and maintainable front-end applications. It is not a library since library provides you limited functionality or has dependencies to other libraries.It is not a plugin or browser extension since it is based on JavaScript and compatible with both desktop and mobile browsers.

10. What browsers AngularJS support?

The latest version of AngularJS 1.3 support Safari, Chrome, Firefox, Opera 15+, IE9+ and mobile browsers (Android, Chrome Mobile, iOS Safari, Opera Mobile). AngularJS 1.3 has dropped support for IE8 but AngularJS 1.2 will continue to support IE8.

11. What is the size of angular.js file?

The size of the compressed and minified file is < 36KB.

12. What are AngularJS features?

Modules

Directives

Templates

Scope

Expressions

Data Binding

MVC (Model, View & Controller)

Validations

Filters

Services

Routing

Dependency Injection

Testing

13. How AngularJS handle the security?

AngularJS provide following built-in protection from basic security holes:

Prevent HTML injection attacks

Prevent Cross-Site-Scripting (CSS) attacks

Prevent XSRF protection for server side communication

Also, AngularJS is designed to be compatible with other security measures like Content Security Policy (CSP), HTTPS (SSL/TLS) and server-side authentication and authorization that greatly reduce the possible attacks.

14. What are Modules in AngularJS?

AngularJS modules are containers just like namespace in C#. They divide an angular app into small, reusable and functional components which can be integrated with other angular app. Each module is identified by a unique name and can be dependent on other modules. In AngularJS, every web page (view) can have a single module assigned to it via ng-app directive.

<script type="text/javascript">

// defining module angular.module('myApp', []);

//OR defining module which has dependency on other modules angular.module

('myApp', ['dependentModule1', 'dependentModule2']);

</script>

15. What components can be defined within AngularJS modules?

Directive

Filter

Controller

Factory

Service

Provider

Value

Config settings and Routes

16. What is core module in AngularJS?

ng is the core module in angular. This module is loaded by default when an angular app is started. This module provides the essential components for your angular app like directives, services/factories, filters, global APIs and testing components.

17. How angular modules load the dependencies?

An angular module use configuration and run blocks to inject dependencies (like providers, services and constants) which get applied to the angular app during the bootstrap process.

18. What is difference between config() and run() method in AngularJS?

Configuration block –

This block is executed during the provider registration and configuration phase. Only providers and constants can be injected into configuration blocks. This block is used to inject module wise configuration settings to prevent accidental instantiation of services before they have been fully configured. This block is created using config() method.

Run block –

This block is executed after the configuration block. It is used to inject instances and constants. This block is created using run() method. This method is like as main method in C or C++. The run block is a great place to put event handlers that need to be executed at the root level for the application. For example, authentication handlers.

19. When dependent modules of a module are loaded?

A module might have dependencies on other modules. The dependent modules are loaded by angular before the requiring module is loaded.

In other words the configuration blocks of the dependent modules execute before the configuration blocks of the requiring module. The same is true for the run blocks. Each module can only be loaded once, even if multiple other modules require it.

20. What is Angular Prefixes $ and $$?

To prevent accidental name collisions with your code, Angular prefixes names of public objects with $ and names of private objects with $$. So, do not use the $ or $$ prefix in your code.

21. How AngularJS expressions are different from the JavaScript expressions?

AngularJS expressions are much like JavaScript expressions but they are different from JavaScript expressions in the following ways:

Angular expressions can be added inside the HTML templates.

Angular expressions doesn't support control flow statements (conditionals, loops, or exceptions).

Angular expressions support filters to format data before displaying it.

22. What are Directives in AngularJS?

AngularJS directives are a combination of AngularJS template markups (HTML attributes or elements, or CSS classes) and supporting JavaScript code. The JavaScript directive code defines the template data and behaviors of the HTML elements. AngularJS directives are used to extend the HTML vocabulary i.e. they decorate html elements with new behaviors and help to manipulate html elements attributes in interesting way. There are some built-in directives provided by AngularJS like as ng-app, ng-controller, ng-repeat, ng-model etc.

23. What is the role of ng-app, ng-init and ng-model directives?

The main role of these directives is explained as:

ng-app - Initialize the angular app.

ng-init - Initialize the angular app data.

ng-model - Bind the html elements like input, select, text area to angular app model.

24. Is AngularJS a library, framework, plugin or a browser extension?

AngularJS fits the definition of a framework the best, even though it's much more lightweight than a typical framework and that's why many confuse it with a library. AngularJS is 100% JavaScript, 100% client side and compatible with both desktop and mobile browsers. So it's definitely not a plugin or some other native browser extension.

25. Explain what Angular JS routes does ?

Angular js routes enable you to create different URLs for different content in your application. Different URLs for different content enables user to bookmark URLs to specific content. Each such bookmarkable URL in Angular.js is called a route.

A value in Angular JS is a simple object. It can be a number, string or JavaScript object. Values are typically used as configuration injected into factories, services or controllers. A value should be belong to an Angular.js module.Injecting a value into an Angular.js controller function is done by adding a parameter with the same name as the value.

26. Why is this project called "AngularJS"? Why is the namespace called "ng"?

Because HTML has Angular brackets and "ng" sounds like "Angular".

27. Can you explain what is testability like in Angular?

Very testable and designed this way from ground up. It has an integrated dependency injection framework, provides mocks for many heavy dependencies (server-side communication).

28. Is AngularJS a templating system?

At the highest level, Angular does look like a just another templating system. But there is one important reason why the Angular templating system is different, that makes it very good fit for application development: bidirectional data binding. The template is compiled in the browser and the compilation step produces a live view. This means you, the developers, don't need to write code to constantly sync the view with the model and the model with the view as in other templating systems.

29. What is a scope in AngularJS?

scope is an object that refers to the application model. It is the glue between application controller and the view. Both the controllers and directives have reference to the scope, but not with each other. It is an execution context for expressions and arranged in hierarchical structure. Scopes can watch expressions and propagate events.

30. Do I need to worry about security holes in AngularJS?

Like any other technology, AngularJS is not impervious to attack. Angular does, however, provide built-in protection from basic security holes including cross-site scripting and HTML injection attacks. AngularJS does round-trip escaping on all strings for you and even offers XSRF protection for server-side communication.

AngularJS was designed to be compatible with other security measures like Content Security Policy (CSP), HTTPS (SSL/TLS) and server-side authentication and authorization that greatly reduce the possible attack vectors and we highly recommended their use.

31. Explain what is linking function and type of linking function?

Link combines the directives with a scope and produce a live view. For registering DOM listeners as well as updating the DOM, link function is responsible. After that template is cloned it is executed.

Pre-linking function: Pre-linking function is executed before the child elements are linked. It is not considered as the safe way for DOM transformation.

Post linking function:Post linking function is executed after the child elements are linked. It is safe to do DOM transformation by post-linking function.

32. Explain what is injector?

An injector is a service locator. It is used to retrieve object instances as defined by provider, instantiate types, invoke methods and load modules. There is a single injector per Angular application, it helps to look up an object instance by its name.

33. Explain what is factory method in angular.js?

For creating the directive, factory method is used. It is invoked only once, when compiler matches the directive for the first time. By using $injector.invoke the factory method is invoked.

34. Mention what are the advantages and disadvantage of using Angular.js framework ?

Following are the advantages of AngularJS:

AngularJS provides capability to create Single Page Application in a very clean and maintainable way.

AngularJS provides data binding capability to HTML thus giving user a rich and responsive experience

AngularJS code is unit testable.

AngularJS uses dependency injection and make use of separation of concerns.

AngularJS provides reusable components.

With AngularJS, developer writes less code and gets more functionality.

In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.

AngularJS applications can run on all major browsers and smart phones including Android and iOS based phones/tablets

Following are the disadvantages of AngularJS:

Not Secure: Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.

Not degradable: If your application user disables JavaScript then user will just see the basic page and nothing more.

35. Explain the concept of scope hierarchy? How many scope can an application have?

Each angular application consist of one root scope but may have several child scopes. As child controllers and some directives create new child scopes, application can have multiple scopes. When new scopes are formed or created they are added as a children of their parent scope. Similar to DOM, they also creates a hierarchical structure.

36. Explain what is DI (Dependency Injection ) and how an object or function can get a hold of its dependencies ?

DI or Dependency Injection is a software design pattern that deals with how code gets hold of its dependencies. In order to retrieve elements of the application which is required to be configured when module gets loaded , the operation “config” uses dependency injection.

These are the ways that object uses to hold of its dependencies

Typically using the new operator, dependency can be created

By referring to a global variable, dependency can be looked up

Dependency can be passed into where it is required

37. Who created Angular JS ?

Intially it was developed by Misko Hevery and Adam Abrons. Currently it is being developed by Google.

38. Brief about how Angular JS Framework is developed by Google

Angular is built and maintained by dedicated Google engineers. This one may seem obvious, but it’s important to remember that many (not all) frameworks are made by hobbyists in the open source community. While passion and drive have forged frameworks, like Cappucino and Knockout, Angular is built and maintained by dedicated (and highly talented) Google engineers. This means you not only have a large open community to learn from, but you also have skilled, highly-available engineers tasked to help you get your Angular questions answered.

This isn’t Google’s first attempt at a JavaScript framework; they first developed their comprehensive Web Toolkit, which compiles Java down to JavaScript, and was used by the Google Wave team extensively. With the rise of HTML5, CSS3, and JavaScript, as both a front-end and back-end language, Google realized that the web was not meant to be written purely in Java.

AngularJS came about to standardize web application structure and provide a future template for how client-side apps should be developed.

Angular JS is being used by a host of applications, ranging from hobby to commercial products. Adoption of AngularJS as a viable framework for client-side development is quickly becoming known to the entire web development community.

Because AngularJS is built by Google, you can be sure that you’re dealing with efficient and reliable code that will scale with your project. If you’re looking for a framework with a solid foundation, Angular is your choice.

39. What makes angular.js better ?

Followings are makes angular js better:

Registering Callbacks:There is no need to register callbacks . This makes your code simple and easy to debug.

Control HTML DOM programmatically:All the application that are created using Angular never have to manipulate the DOM although it can be done if it is required

Transfer data to and from the UI:Angular.js helps to eliminate almost all of the boiler plate like validating the form, displaying validation errors, returning to an internal model and so on which occurs due to flow of marshalling data

No initilization code: With angular.js you can bootstrap your app easily using services, which auto-injected into your application in Guice like dependency injection style

40. 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.

41. Mention the steps for the compilation process of HTML happens?

Compilation of HTML process occurs in following ways:

Using the standard browser API, first the HTML is parsed into DOM

By using the call to the $compile () method, compilation of the DOM is performed. The method traverses the DOM and matches the directives.

Link the template with scope by calling the linking function returned from the previous step

42. Explain what is directive and Mention what are the different types of Directive?

During compilation process when specific HTML constructs are encountered a behaviour or function is triggered, this function is referred as directive. It is executed when the compiler encounters it in the DOM.

Different types of directives are:

Element directives

Attribute directives

CSS class directives

Comment directives

43. Explain what is the difference between link and compile in angular.js?

Compile function: It is used for template DOM Manipulation and collect all of the directives.

Link function: It is used for registering DOM listeners as well as instance DOM manipulation. It is executed once the template has been cloned.

44. What is factory method in angular.js?

For creating the directive, factory method is used. It is invoked only once, when compiler matches the directive for the first time. By using $injector.invoke the factory method is invoked.

45. Mention what are the styling form that ngModel adds to CSS classes ?

ngModel adds these CSS classes to allow styling of form as well as control:

ng- valid

ng- invalid

ng-pristine

ng-dirty

46. Mention what are the characteristics of “Scope”?

To observer model mutations scopes provide APIs ($watch)

To propagate any model changes through the system into the view from outside of the Angular realm

A scope inherits properties from its parent scope, while providing access to shared model properties, scopes can be nested to isolate application components

Scope provides context against which expressions are evaluated

47. Mention what are the advantages of using Angular.js framework ?

Advantages of using Angular.js as framework are:

Supports two way data-binding

Supports MVC pattern

Support static template and angular template

Can add custom directive

Supports REST full services

Supports form validations

Support both client and server communication

Support dependency injection

Applying Animations

Event Handlers

48. Explain the concept of scope hierarchy? How many scope can an application have?

Each angular application consist of one root scope but may have several child scopes. As child controllers and some directives create new child scopes, application can have multiple scopes. When new scopes are formed or created they are added as a children of their parent scope. Similar to DOM, they also creates a hierarchical structure.

49. Explain what is the difference between angular.js and backbone.js?

Angular.js combines the functionalities of most of the 3rd party libraries, it supports individual functionalities required to develop HTML5 Apps. While Backbone.js do their jobs individually.

50. Which are the 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.

51. 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.

52. Explain ng-app directive?

ng-app directive defines and links an AngularJS application to HTML. It also indicate the start of the application.

53. Explain ng-model directive?

ng-model directive binds the values of AngularJS application data to HTML input controls. It creates a model variable which can be used with the html page and within the container control( for example, div) having ng-app directive.

54. Explain ng-bind directive?

ng-bind directive binds the AngularJS Application data to HTML tags. ng-bind updates the model created by ng-model directive to be displayed in the html tag whenever user input something in the control or updates the html control's data when model data is updated by controller.

55. Explain ng-controller directive.

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. A controller is a JavaScript object containing attributes/properties and functions. Each controller accepts $scope as a parameter which refers to the application/module that controller is to control.

56. How AngularJS integrates with HTML?

AngularJS being a pure javaScript based library integrates easily with HTML.

Step 1:Include angularjs javascript libray in the html page

<head>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</head>

Step 2: Point to AngularJS app

Next we tell what part of the HTML contains the AngularJS app. This done by adding the ng-app attribute to the root HTML element of the AngularJS app. You can either add it to html element or body element as shown below:

<body ng-app="myapp">

</body>

57. Explain ng-init directive ?

ng-init directive initializes an AngularJS Application data. It is used to put values to the variables to be used in the application.

58. Explain ng-repeat directive ?

ng-repeat directive repeats html elements for each item in a collection.

59. What are AngularJS expressions ?

Expressions are used to bind application data to html. Expressions are written inside double braces like {{ expression}}. Expressions behave in same way as ng-bind directives. AngularJS application expressions are pure JavaScript expressions and outputs the data where they are used.

60. Explain uppercase filter ?

Uppercase filter converts a text to upper case text.

In below example, we've added uppercase filter to an expression using pipe character. Here we've added uppercase filter to print student name in all capital letters.

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() | uppercase}}

61. Explain lowercase filter ?

Lowercase filter converts a text to lower case text.

In below example, we've added lowercase filter to an expression using pipe character. Here we've added lowercase filter to print student name in all lowercase letters.

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}}

62. Explain currency filter ?

Currency filter formats text in a currency format.

In below example, we've added currency filter to an expression returning number using pipe character. Here we've added currency filter to print fees using currency format.

Enter fees: <input type="text" ng-model="student.fees">

fees: {{student.fees | currency}}

63. Explain filter filter ?

filter filter is used to filter the array to a subset of it based on provided criteria.

In below example, to display only required subjects, we've used subjectName as filter.

Enter subject: <input type="text" ng-model="subjectName">

Subject:

<ul>

<li ng-repeat="subject in student.subjects | filter: subjectName">

{{ subject.name + ', marks:' + subject.marks }}

</li>

</ul>

64. Explain orderby filter ?

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>

65. Explain ng-disabled directive ?

ng-disabled directive disables a given control.

In below example, we've added ng-disabled attribute to a HTML button and pass it a model. Then we've attached the model to an checkbox and can see the variation.

<input type="checkbox" ng-model="enableDisableButton">Disable Button

<button ng-disabled="enableDisableButton">Click Me!</button>

66. Explain ng-show directive ?

ng-show directive shows a given control.

In below example, we've added ng-show attribute to a HTML button and pass it a model. Then we've attached the model to a checkbox and can see the variation.

<input type="checkbox" ng-model="showHide1">Show Button

<button ng-show="showHide1">Click Me!</button>

67. Explain ng-hide directive ?

ng-hide directive hides a given control.

In below example, we've added ng-hide attribute to a HTML button and pass it a model. Then we've attached the model to a checkbox and can see the variation.

<input type="checkbox" ng-model="showHide2">Hide Button

<button ng-hide="showHide2">Click Me!</button>

68. Explain ng-click directive ?

ng-click directive represents a AngularJS click event.

In below example, we've added ng-click attribute to a HTML button and added an expression to updated a model. Then we can see the variation.

<p>Total click: {{ clickCounter }}</p></td>

<button ng-click="clickCounter = clickCounter + 1">Click Me!</button>

69. How angular.module works ?

angular.module is used to create AngularJS modules along with its dependent modules. Consider the following example:

var mainApp = angular.module("mainApp", []);

Here we've declared an application mainApp module using angular.module function. We've passed an empty array to it. This array generally contains dependent modules declared earlier.

70. How to validate data 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.

71. Explain ng-include directive ?

Using AngularJS, we can embed HTML pages within a HTML page using ng-include directive.

72. How to make an ajax call using Angular JS ?

AngularJS provides $http 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, $http can be used to get the data from server in the following manner:

function studentController($scope,$http) {

var url="data.txt";

$http.get(url).success( function(response) {

$scope.students = response;

});

}

73. What is use of $routeProvider in AngularJS ?

$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.

74. What is $rootScope ?

Scope is a special JavaScript object which plays the role of joining controller with the views. Scope contains the model data. In controllers, model data is accessed via $scope object. $rootScope is the parent of all of the scope variables.

75. What is scope hierarchy in AngularJS ?

Scopes are controllers specific. If we define nested controllers then child controller will inherit the scope of its parent controller.

Following are the important points to be considered in above example.

We've set values to models in shapeController.

We've overridden message in child controller circleController. When "message" is used within module of controller circleController, the overridden message will be used.

76. What is a service ?

Services are JavaScript functions and are responsible to do specific tasks only. Each service is responsible for a specific task for example, $http is used to make ajax call to get the server data. $route is used to define the routing information and so on. Inbuilt services are always prefixed with $ symbol.

77. 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);

}

});

78. 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;

});

79. 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.

80. Which components can be injected as a dependency in AngularJS ?

AngularJS provides a supreme Dependency Injection mechanism. It provides following core components which can be injected into each other as dependencies.

value

factory

service

provider

constant

81. 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;

};

});

});

82. What is constant ?

constants are used to pass values at config phase considering the fact that value cannot be used to be passed during config phase.

mainApp.constant("configParam", "constant value");

83. Is AngularJS extensible ?

Yes! In AngularJS we can create custom directive to extend AngularJS existing functionalities.

Custom directives are used in AngularJS to extend the functionality of HTML. Custom directives are defined using "directive" function. A custom directive simply replaces the element for which it is activated. AngularJS application during bootstrap finds the matching elements and do one time activity using its compile() method of the custom directive then process the element using link() method of the custom directive based on the scope of the directive.

84. On which types of component can we create a custom directive ?

AngularJS provides support to create custom directives for following type of elements.

Element directives - Directive activates when a matching element is encountered.

Attribute - Directive activates when a matching attribute is encountered.

CSS - Directive activates when a matching css style is encountered.

Comment - Directive activates when a matching comment is encountered.

85. What is internationalization ?

Internationalization is a way to show locale specific information on a website. For example, display content of a website in English language in United States and in Danish in France.

86. How to implement internationalization in AngularJS ?

AngularJS supports inbuilt internationalization for three types of filters currency, date and numbers. We only need to incorporate corresponding js according to locale of the country. By default it handles the locale of the browser. For example, to use Danish locale, use following script

87. What are scopes in AngularJS ?

A) Scopes define the accessibility of a javascript variable.

B) Scopes are objects that refer to the model. They act as glue between controller and view.

C) Scopes are objects that refer to the controller.

D) Scopes are objects that refer to the views.

88. Which of the following is true about ng-app directive ?

A) ng-app directive defines and links an AngularJS application to HTML.

B) ng-app directive indicates the start of the application.

C) Both of the above.

D) None of the above.

89. AngularJS expressions are written using.

A) Double braces like {{ expression}}

B) single braces like {expression}

C) small bracket like (expression)

D) capital bracket like [expression]

90. Which of the following is true about ng-show directive ?

A) ng-show directive can show a given control.

B) ng-show directive can hide a given control.

C) Both of the above.

D) None of the above.

91. Which of the following is true about ng-include directive ?

A) Embed HTML pages within a HTML page using ng-include directive

B) Embed JS files within a HTML page using ng-include directive

C) Both of the above.

D) None of the above.

92. Which components can be injected as a dependency in AngularJS?

A) Application Module

B) constant

C) value

D) factory

93. Services are singleton objects which are instantiated only once in app.

A) false

B) true

94. With AngularJS, developer writes less code and gets more functionality.

A) false

B) true

95. AngularJS Expressions are used to bind application data to html.

A) true

B) false

96. In AngularJS, Scope contains the model data.

A) true

B) false