1. **What is the recommended way of implementing navigation within an Ionic 4 app?**
Ionic 4 depends on Angular's router module called RouterModule for navigation. The NavController service is still there but it will be deprecated very soon and is therefore not recommended to be used in new Ionic apps built with Ionic 4.

   The ion-router-outlet component is used to manage the animations that show up as the user navigates to-from a component within the app. This is same as the router-outlet that Angular has but with some animation related sugar added.

2. **How is Ionic Framework v4 different from the previous versions of the Ionic Framework?**
Ionic 4 is one the biggest updates that the Ionic team at Drify Co. has made to the framework. It is a complete re-write of the framework from scratch. The primary changes are:

- Ionic 4 is built with web-components. This essentially means that the Ionic Framework is available as drop-in UI components ready to be used in our apps. These components are self-contained and do not require any external dependencies. All the HTML, CSS and JS required for a web-component to work is included with the web-component. To use the framework, you just need to add the CSS and JS files required to your project and you are good to go. No build process required anymore.
- Ionic 4 makes use of CSS variables. CSS variables are entities defined by developers that contain specific values to be reused throughout an app. The values of these entities can be overridden by app developers to achieve custom colors and designs in their Ionic Apps to make their app standout. Earlier, this was done with the help of a CSS preprocessor like SASS.
- Ionic 4 uses Angular's RouterModule to implement navigation. Earlier, NavController was used to implement push-pop style navigation within the app which has now been deprecated and the much more powerful and flexible RouterModule is the recommended way. Using RouterModule enables lazy-loading for the Ionic components.

3. **What is Lazy Loading?**
Lazy loading is a general concept where we only load things as we need them. In Ionic 3, we would use the @IonicPage decorator to lazy load our pages, and now with Ionic 4 we lazy load our Angular components with Angular routing. The idea behind lazy loading pages is that we don't need to load every page in the application as soon as the application is launched.

   Lazy loading web components in Ionic 4 takes this a step further. In Ionic 4, every Ionic component that we use is a web component. Each of these web components will be lazily loaded only when they are accessed in the application.

   This means that the code for <ion-button> or <ion-card> will only be loaded when it is accessed in the application, even each <ion-icon> is loaded individually rather than having to load the entire icon library. This isn't even something we need to set up, it is just the way it works by default.

4. **How would you save data in an Ionic app so that it can be accessed later or on the next launch?**
Ionic Framework offers a Storage module that resides in the @ionic/storage package. It allows us to store key-value pairs and JSON objects in a secure reliable storage, however, it has its own set of limits.
Storage uses a variety of storage engines underneath, picking the best one available depending on the platform. When running in a native app context, Storage will prioritize using SQLite, as it's one of the most stable and widely used file-based databases, and avoids some of the pitfalls of things like localstorage and IndexedDB, such as the OS deciding to clear out such data in low disk-space situations.

   When running in the web or as a Progressive Web App, Storage will attempt to use IndexedDB, WebSQL, and localstorage, in that order.

5. **The size of your Ionic application's generated APK is huge. What can you do to reduce the size?**
First off, we can always get rid if the static assets that are not in use within the app. That reduces the size of the generated executable.

We can also optimize static assets like images, audio and video for mobile devices. That way the app size will reduce considerably. It does not makes sense to use huge images, audio and videos files as a static asset within the app.

Finally, we can get rid of any external modules and cordova plugins that are there but are not being used. Having unnecessary code in the app is an overkill and add a lot of extra bytes to the app's size.

6. **What is a Web View?**
Ionic apps are built using web technologies and are rendered using Web Views, which are a full screen and full-powered web browser.

Modern Web Views offer many built-in HTML5 APIs for hardware functionality such as cameras, sensors, GPS, speakers, and Bluetooth, but sometimes it may also be necessary to access platform-specific hardware APIs. In Ionic apps, hardware APIs can be accessed through a bridge layer, typically by using native plugins which expose JavaScript APIs.

7. **What is CORS? How would deal with it?**
Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to tell a browser to let a web application running at one origin (domain) have permission to access selected resources from a server at a different origin. A web application makes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, and port) than its own origin.

An example of a cross-origin request: The frontend JavaScript code for a web application served from http://domain-a.com uses XMLHttpRequest to make a request for http://api.domain-b.com/data.json.

For security reasons, browsers restrict cross-origin HTTP requests initiated from within scripts. Web Views also enforce CORS, and generally it needs to be handled on the server. If CORS is not implemented on the server, there is a native plugin that performs HTTP requests in the native layer which bypasses CORS. So it's important that external services properly handle cross-origin requests.

8. **What is ion-app?**
App is a container element for the whole Ionic application. There should only be one <ion-app> element per project. An app can have many Ionic components including menus, headers, content, and footers. The overlay components get appended to the <ion-app> when they are presented.

9. **What are the color names that are bundled in when we created a new Ionic 4 project?**
Ionic has nine default colors that can be used to change the color of many components. Each color is actually a collection of multiple properties, including a shade and tint, used throughout Ionic. These nine colors are, primary, secondary, tertiary, danger, warning, success, dark, medium and light.

10. **Name all the page lifecycle hooks in Ionic 4.**
There are 3 different lifecycle hooks in Ionic 4. They are as following.

ionNavDidChange - Event fired when the nav has changed components

ionNavWillChange - Event fired when the nav will components

ionNavWillLoad - Event fired when Nav will load a component

11. **An Ionic 4 app, integrated with Angular, is using Angular's class ActivatedRoute. What is the alternative way of sharing data between different pages/components in an Ionic 4 app using Angular?**
We can also use Angular Services to share data between multiple pages. An Angular service is a singleton and thus does not get destroyed when the user loads and unloads components within the app. Therefore, the data can be set from one component and it can be retrieved from another component. This is what we need to pass data from one component to another during navigation.

12. **How would compare Ionic 4 and Apache Cordova?**
Apache Cordova is an open-source mobile development framework. It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development. Applications execute within

wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc

Ionic on the other hand is a web framework with some additional tools and services that lets you build a responsive mobile app using web technologies like HTML, CSS and JS. Once the web is done, it is packaged into a native app for supported platforms like Android, iOS and Windows using Apache Cordova. So, in short, Ionic uses Apache Cordova to create apps with the help of web technologies for various mobile operating systems.

13. **Name some Ionic components with brief descriptions of each.**
ion-list - The list component is used to display rows of information, such as a contact list, playlist, or menu. It can display cards, list items and more. Lists are also very customizable and match the design of the native Android and iOS lists by default.

ion-card - Card components are a great way to display important pieces of content, and are quickly emerging as a core design pattern for apps. Cards can contain images, buttons, text and more. Cards can also be used standalone, in a list or in a grid.

ion-grid - Grid is a powerful mobile-first system for building custom layouts. The Grid component can be used to ensure your app looks great on any device. It is heavily influenced by Bootstrap's grid system.

ion-button - Buttons are an essential way to interact with and navigate through an app, and should clearly communicate what action will occur after the user taps them. Buttons are highly customizable with color, shape and size being fully customizable. We support round buttons, FAB buttons, outline buttons and more!

ion-input - Inputs allow users to enter data into a UI. They typically appear in forms and dialogs. They should always be wrapped with an ion-item.

14. **Explain the working of ion-grid component.**
ion-grid is a powerful mobile-first flexbox system for building custom layouts.

It is composed of three units — a grid, row(s) and column(s)). Columns will expand to fill the row, and will resize to fit additional columns. It is based on a 12 column layout with different breakpoints based on the screen size. The number of columns can be customized using CSS.

Here are some pointers that we need to keep in mind while working with grids in Ionic.

Grids act as a container for all rows and columns. Grids take up the full width of their container, but adding the fixed attribute will specify the width per screen size.
Rows are horizontal groups of columns that line the columns up properly.
Content should be placed within columns, and only columns may be immediate children of rows.
The size-{breakpoint} attributes indicate the number of columns to use out of the default 12 per row. So, size="4" can be added to a column in order to take up 1/3 of the grid, or 4 of the 12 columns.
Columns without a value for size will automatically have equal widths. For example, four instances of size-sm will each automatically be 25% wide for the small breakpoint and up.
Column widths are set as a percentage, so they're always fluid and sized relative to their parent element.
Columns have padding between individual columns, however, the padding can be removed from the grid and columns by adding no-padding on the grid.
There are five grid tiers, one for each responsive breakpoint: all breakpoints (extra small), small, medium, large, and extra large.
Grid tiers are based on minimum widths, meaning they apply to their tier and all those larger than them (e.g., size-sm="4" applies to small, medium, large, and extra large devices).

15. **Explain the purpose of config.xml file in an Ionic project.**
config.xml is a global configuration file that controls many aspects of a cordova application's behavior. This platform-agnostic XML file is arranged based on the W3C's Packaged Web Apps (Widgets) specification, and extended to specify core Cordova API features, plugins, and platform-specific settings.

For projects created with the Ionic CLI, this file can be found in the top-level directory. It contains vital information about the project like the package name, version, author's information and a lot of permissions and specifications as well.

16. **What is the purpose of ion-infinite-scroll component in Ionic?**
    ion-infinite-scroll component is very useful component. It allows to load data into the app (typically in a list) as the users scrolls the page. The most common use case is where the app loads more content to display to the user as the user keeps scrolling the page. This concept is used widely nowadays in almost all modern apps.

    The Infinite Scroll allows us to perform an action when the user scrolls a specified distance from the bottom or top of the page. The expression assigned to the infinite event is called when the user scrolls to the specified distance. When this expression has finished its tasks, it should call the complete() method on the infinite scroll instance.

17. **What is the purpose of ion-refresher component in Ionic?**
    The Refresher provides pull-to-refresh functionality on a content component. It is a functionality that allows the users to pull the page down from the top and as they let it go, the content on the page is refreshed.

    To use it, we place the ion-refresher as the first child of our ion-content element. Pages can then listen to the refresher's various output events. The refresh output event is fired when the user has pulled down far enough to kick off the refreshing process. Once the async operation has completed and the refreshing should end, call complete().

18. **How would you get the list of all the available startup templates for an Ionic app?**
    ionic start --list command displays the list of all the available templates that can be used while creating a new Ionic application. As for Ionic 4, as of now, only blank and sidemenu are available

19. **What are different page lifecycle events in Ionic?**
    These are the events:

    ionNavDidChange event is fired when the nav has changed components.

    ionNavWillChange event is fired when the nav will components.

    ionNavWillLoad event is fired when Nav will load a component.

20. **Can we use Firebase with Ionic?**
    Yes, absolutely. Firebase SDK is also available as a Node module and we can install it within our Ionic project, import it and it is ready to be used.

21. **What programming language do you need to know if you are planning to build an Ionic 4 app with Angular?**
    TypeScript.

22. **How can you detect a platform (Android or iOS) at runtime in Ionic application?**
    Ionic provides platform classes: when the application is loaded, Ionic adds CSS classes to the <body> tag. For example, on iOS devices, Ionic adds platform-ios class to <body> tag. Ionic also adds OS version classes such as platform-ios8 (for iOS 8) and platform-android4_4 (for Android 4.4).

23. **How can you access mobile phone native functionality in Ionic applications, for example the camera?**
    Ionic does not provide a camera API out of the box. However, since Ionic uses plugins architecture, and because it is based on Cordova, we can use Cordova plugins in our application. Ionic team provides a set of Cordova extensions with Angular wrappers, and they can be found at ngCordova.

    To use Cordova plugins, we need to install the plugin using Ionic command install <plugin name>. In some cases, we will additionally need to add the plugin's Angular module to your Angular application too.

To use a mobile phone's camera in the Ionic application, we can call the camera API by using cordova-plugin-camera that is hosted on GitHub. This plugin defines a global navigator.camera object, which provides an API for taking pictures and for choosing images from the system's image library.

24. **What's the difference between "ionic build" and "ionic prepare"?**
ionic prepare <platform> copies all files from the www folder into the target platform's www folder.

ionic build <platform> also does this, but also builds the app's source code so that it can be run on a simulator/emulator or a device.

25. **Describe the Project structure of an Ionic 4 app.**
Ionic 4 app has a structure similar to an angular application. e2e , node_modules, src folders will be present in root directory along with individual files like angular.json, ionic.config.json, package-lock.json, package.json, tsconfig.json, tslint.json . e2e/ folder has files for running end to end integration tests, node_modules contains all the dependencies. src/app/ folder contains most of the app code that you will be writing. App folder essentially has app.module.ts, app.component.ts, app.component.html, app.component.spec.ts and app-routing.module.ts. any services, components, pages created can be stored in their respective folders. assets/ folder can be used to store images and other static content.

As its name suggests, angular.json hosts all of the angular settings like project name, root path for app folder, source root, paths for index file, main.ts, polyfills, assets folder path, css stylesheet paths, script paths, build settings etc. ionic.config.json can be used to set proxy settings for API to avoid CORS errors during development. package.json contains a list of dependencies and serve , build , test and e2e commands.

26. **Explain Ionic lifecycle hooks.**
Every component in Ionic has a lifecycle. Ionic creates, renders the component, checks it when its data-bound properties change and destroys it finally. Ionic offers lifecycle hooks that provide a way to tap into these key moments and trigger an action when they occur.

Ionic 2 & 3 had these lifecycle events : ionViewDidLoad, ionViewWillEnter, ionViewDidEnter, ionViewWillLeave, ionViewDidLeave, ionViewWillUnload, ionViewCanEnter, ionViewCanLeave.

ionViewDidLoad : Fired only when a view is stored in memory, will not fire if view has been already cached.

ionViewWillEnter : fired when entering a page, before it becomes active one. this event is triggered every time user enters in the view.

ionViewDidEnter : fired when entering a page, after it becomes active page.

ionViewWillLeave : fired when user leaves a page, before it stops being the active page.

ionViewDidLeave : fired when user leaves a page, after it stops being the active page.

ionViewWillUnload: fired when a view is going to be completely removed.

ionViewCanEnter: this is a nav guard. fired before entering a view. Useful when you want to control access to pages based on access credentials.

ionViewCanLeave: this is also a nav guard, fired before leaving a view. Allows you to control whether user can exit the view or not.

Ionic 4 provides the Angular lifecycle hooks in addition to the above listed

Ionic lifecycle hooks. All the angular lifecycle hooks are available.

ngOnChanges , ngOnInit, ngDoCheck, ngAfterContentInit, ngAfterContentChecked,

ngAfterViewInit, ngAfterViewChecked, ngOnDestroy .

27. **What is the difference between ionic polymerization and free-radical?**

IONIC POLYMERIZATION	FREE-RADICAL IONIC
• Ionic polymerization is a chain polymerization that has ions and ion pairs in the centre. • The initiation of ionic polymerization consumes very less activation energy than radical polymerization. • It is an alternative to free radical polymerization.	• Free radical polymerizations are those polymerizations that are formed by adding free-radical building blocks. • Free radical polymerization uses initiators and coinitiators. • It is the only way to obtain different types of polymers and material composites.

28. **How many types of storage available in Ionic framework?**

The easy way to store key or values and JSON objects is known as storage in Ionic Framework. In this various storage, engines are used. While on the web application, the storage will tend to use IndexedDB, WebSQL, and localstorage. Various types of storage are available in ionic framework. Some of them are –

- HTML5 local storage
- Cookie and session storage
- indexedDB
- WebSQL
- PouchDB
- Webservice/api storage
- Cordova storage.

29. **What is the advantage of caching the views in Ionic apps? Provide examples.**

Ionic by default caches up to ten views, which improves performance and also maintains different states in the views at the same time. For example, the cache can maintain scroll position in the views or active state of buttons.
Caching can be disabled per view by using cache: false in UI-router’s state config, like in the following example:

```
$stateProvider.state(‘myState’, {
cache: false,
url : ‘/myUrl’,
templateUrl : ‘my-template.html’
})
```

Caching can be also disabled globally, by setting maxCache to 0:
`$ionicConfigProvider.views.maxCache(0);`

30. **How can you detect a platform (Android or iOS) at runtime in Ionic application?**

Ionic provides platform classes: when the application is loaded, Ionic adds CSS classes to the <body> tag. For example, on iOS devices, Ionic adds platform-ios class to <body> tag. Ionic also adds OS version classes such as platform-ios8 (for iOS 8) and platform-android4_4 (for Android 4.4).

31. **What are the benefits of caching views in Ionic applications?**

In Ionic, caching of up to ten views is done by default. This improves performance and assists in maintaining different states simultaneously in the views. For instance, cache maintains scroll position in the views or the active state of buttons.

32. **Why is the performance of Ionic apps bad on older Android devices? Can this be improved?**

Ionic uses the default web browser available in the device to run hybrid applications. The older versions of Android devices use Android's default browser, which has less performance and compliance as compared to modern browsers.

This can be resolved by using crosswalk with Ionic. Crosswalk will enable you to pack a Chrome webview so that your application is not relying on native Android browser.

33. **What Is MVC, MVP And MVVM Design Patterns?**

-MVC Stands For Model-View-Controller. It Is A Software Design Pattern Which Was Introduced In 1970s. Also, MVC Pattern Forces A Separation Of Concerns, It Means Domain Model And Controller Logic Are Decoupled From User Interface (View). As A Result Maintenance And Testing Of The Application Become Simpler And Easier.

-MVP Stands For Model-View-Presenter.This Pattern Is Similar To MVC Pattern In Which Controller Has Been Replaced By The Presenter. This Design Pattern Splits An Application Into Three Main Aspects: Model, View And Presenter.The Presenter Is Responsible For Handling All UI Events On Behalf Of The View.

-MVVM Stands For Model-View-View Model. This Pattern Supports Two-Way Data Binding Between View And View Model. This Enables Automatic Propagation Of Changes, Within The State Of View Model To The View. Typically, The View Model Uses The Observer Pattern To Notify Changes In The View Model To Model.

34. **What Are The Tips And Tricks To Improve Mobile App Performance?**

-Cache Images
-Compress And Resize Images
-Re-Use Data Templates
-Reduce HTTP Requests
-Use Loading Validations
-Load Data As You Need It
-Create An Offline Mode
-Use Lazy Loading