Angular Interview

**What is Angular?**

1. A Platform and framework for building single page application using HTML and typescript.

**Angular Architecture:**

1. Basic building blocks – components organized into *NgModules*.
2. **NgModules** keep related code into functional sets. NgModules can import functionality from other NgModules and export their own functionality to be used by other NgModules. Ex. Router module is imported into Root Module (AppModule). 2 types:
   1. Root Module – provides bootstrap mechanism that launches the app, an app has only one – imports BrowserModule
   2. Feature Modules – can be many in an app – imports CommonModule
3. Modules, Components and Services use decorators which mark their type and provide metadata
   1. Component class metadata associates it with a template that defines a view. A template consists of HTML, angular directives and binding markup
   2. Service class metadata provides information Angular needs to make it available to components through dependency injection.
   3. Decorators are basically functions, there are many decorators defined by angular for different purpose
4. Components:
   1. Every app has atleast one component – root component (app.component)
   2. *@Component()* decorator identifies class it is associated with as component that contains data and logic and provides the template, stylesheet and other metadata.
   3. Template:
      1. Template directives provide logic
      2. Binding markup connects application data and DOM
      3. 2 types of data binding:
         1. Event Binding – user input
         2. Property binding – data from component in HTML
      4. 2 way binding – data changes in component are reflected in DOM
      5. Pipes – can be used to transform values for display such as currency and date pipes.
5. Services:
   1. Used to share data across components
   2. Decorator: *@Injectable()* – provides metadata allowing service classes to be injected in component classes.
6. An App can have many views arranged hierarchically
7. Routing:
   1. Router NgModule provides routing service which can be used to define navigation paths among views
   2. Router maps URL like paths to views instead of pages. It intercepts the browser’s behavior when loading a page and shows or hides view hierarchies.
   3. Router can lazy-load modules on demand as required

**Lifecycle Hooks**

1. Components and Directives have lifecycles and lifecycle hooks are the functions that can be used to tap into different lifecycle hooks. It is not required to implement any or all the lifecycle hooks, just implement them as needed.
2. Lifecycle
   1. ngOnChanges()
      1. Called whenever any of the data input properties change. This method receives a *SimpleChanges* object of current and previous values.
      2. If component doesn’t has any inputs or no inputs have been provided for a component then this hooks is not called.
      3. Because this hook gets called frequently, therefore we should keep in mind that any operation performed here impacts app performance significantly
   2. ngOnInit