**Introduction :**

**Objectives**

* Explain the characteristics of a Single Page Application (SPA)
  + Works inside a browser, no page reload, example multi page application (MPA) sites, example SPA sites, technologies to develop SPA (HTML, CSS, JavaScript, jQuery, ajax, AngularJS, Angular, ReactJS, Ember.js)

* Describe the Single Page Application Lifecycle
  + Traditional page life cycle in Servlet/ASP.NET, initial request, single time loading of HTML resources, AJAX, JSON
    - SPA Lifecyle Diagram: https://cdn.goconqr.com/uploads/media/image/20395412/desktop\_6edf2ce8-99ea-4dc1-80b6-01a2b7b2f9ab.png

* Demonstrate usage of JavaScript objects
  + Objects, Property, Property Value
    - Ref - https://www.w3schools.com/js/js\_objects.asp

* Demonstrate usage of JavaScript Object Notation (JSON)
  + Lightweight format for storing and transferring data, used when data is sent from a server to web page, self-describing, easy to understand, syntax, name value pair, separator is comma, curly braces hold objects, square brackets hold arrays, JSON value data types (string, number, object, array, boolean)
    - Intro - https://www.w3schools.com/js/js\_json\_intro.asp
    - Syntax - https://www.w3schools.com/js/js\_json\_syntax.asp
    - Data Types - https://www.w3schools.com/js/js\_json\_datatypes.asp
    - Parsing - https://www.w3schools.com/js/js\_json\_parse.asp

* Explain the need and benefits of Angular
  + Limitations of implementing SPA using JavaScript, Limitations of implementing SPA using jQuery, how to implement using angular, better performance since UI resources loads only once, gives better user experience, clear separation of view, action and web service call

* Describe the difference between AngularJS and Angular
  + Angular JS is version 1, called as Angular from version 2, how Angular JS works, key differences between Angular JS and Angular, Angular JS uses JavaScript, Angular uses TypeScript, Angular is command line based, Angular has a compilation step

**TypeScript:**

**Objectives**

* List key features of ECMAScript6 (ES6)
  + European Computer Manufacturers Association (ECMA) standards organization, ECMAScript, scripting language specification, let, const, arrow functions, classes, TypeScript ES6 is related to TypeScript and Angular
    - Ref - https://www.w3schools.com/js/js\_es6.asp

* Explain the need and benefits of TypeScript
  + static typing, superset of JavaScript, TypeScript code compiles to JavaScript, browser compatible, open source, primary language of angular
    - Ref - https://medium.com/swlh/the-major-benefits-of-using-typescript-aa8553f5e2ed

* Demonstrate compilation and execution of a TypeScript file
  + Install typescript npm package, write TypeScript program, compile and run
    - Ref - https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html

* Demonstrate defining objects using data types, interfaces and classes
  + Data Types (boolean, number, string, array, tuple, enum, any, void) interfaces, class
    - Data Types - https://www.typescriptlang.org/docs/handbook/basic-types.html
    - Interface - https://www.typescriptlang.org/docs/handbook/interfaces.html
    - Classes - https://www.typescriptlang.org/docs/handbook/classes.html
    - Modules - https://www.typescriptlang.org/docs/handbook/modules.html

* Interpret how to implement functions and for .. of in TypeScript
  + Defining functions with parameters' data type and return data type defined, using for .. of to iterate through arrays
    - Functions Ref - https://www.typescriptlang.org/docs/handbook/functions.html
    - for .. of Ref - https://www.typescriptlang.org/docs/handbook/iterators-and-generators.html

**Displaying Data:**

**Objectives**

* Demonstrate creation of angular application
  + Create new angular application in command line (ng new), execute the application (ng serve), understanding app component and app component html, importance of app module
    - New Application - https://angular.io/cli/new
    - Start Server - https://angular.io/cli/serve App Component
    - File Structure - https://angular.io/guide/file-structure

* Demonstrate displaying data using interpolation, ngIf and ngFor
  + Property definition in component, using double curly brackets, ngIf directive, ngFor directive
    - Ref - https://angular.io/guide/displaying-data

* Demonstrate using pipes to format number and date
  + Data to view transformation, pipe character, built-in pipes (Date, UpperCase, LowerCase, Currency, Percent), parameterizing pipes, chained pipes
    - Ref - https://angular.io/guide/pipes

* Demonstrate karma test case creation with DOM reading
  + ng test, TestBed, fixture, component, debugElement, nativeElement, detectChanges(), it(), expect(), toEqual(), toBe()
    - Ref - https://angular.io/guide/testing

**Components and Routing:**

**Objectives**

* Demonstrate creation of component
  + ng generate component, component.html, component.ts, @Component decorator, selector, templateUrl, style
    - Generate component - https://angular.io/cli/generate
    - Component Architecture - https://angular.io/guide/architecture-components

* Demonstrate routing
  + enables navigation from one view to another, <base href="/">, import router modules, appRoutes, Routes, path, component, router-outlet, routerLink, routerLinkActive
    - Ref - https://angular.io/guide/router

* Demonstrate event handling and two way binding
  + click event, two way binding with [(ngModel)], handling event in component
    - Ref - https://angular.io/guide/template-syntax

**Forms:**

**Template Driven Form:**

**Objectives**

* Demonstrate implementation of forms using Template Driven Forms
  + template reference variable, [(ngModel)], FormModule, Validation Properties (invalid, touched, dirty, errors.required, errors.minlength, json pipe, binding ngModel with radio buttons using [value], binding ngModel with select drop dwon using [value], [selected], (change) and $event.target.value, karma testing with createNewEvent() and dispatchNewEvent()
    - Forms Overview - https://angular.io/guide/forms-overview
    - Template Driven Form - https://angular.io/guide/forms
    - Form Validation - https://angular.io/guide/form-validation

**Reactive Form:**

**Objectives**

* Demonstrate implementation of forms using Reactive Forms with validations
  + ReactiveFormsModule, FormControl, [formControl], formControlName, set form control values, FormGroup, Validators (required, minlength, maxlength), read value from FormGroup, definition of FormGroup in <form>, nested form groups, FormBuilder, Dynamic Forms, FormArray, formArrayName, FormBuilder.array(), FormBuilder.control(), push new form control
    - Reactive Form - https://angular.io/guide/reactive-forms
    - Dynamic Form - https://angular.io/guide/dynamic-form
    - Form Validation - https://angular.io/guide/form-validation

**Component Interaction:**

**Objectives**

* Demonstrate implementation of interaction between components
  + app selector with passing parameters, @Input decorator, filter() method, defining router link with parameters in app-routing.module.ts, defining router link in HTML template with parameter, ActivatedRoute, paramMap, router.navigate() method
    - Master / Detail components - https://angular.io/tutorial/toh-pt3
    - Routing and Navigation - https://angular.io/guide/router

**Service and Guards:**

**Objectives**

* Demonstrate implementation of service
  + generate service, dependency injection
    - Service & Dependency Injection - https://angular.io/guide/architecture-services

* Demonstrate protecting routes using auth guard
  + generate guard, define protected routes in app-routing.module.ts with mapping to guard, define authentication service to retain the status of authentication, implement redirection in guard based on authentication status
    - Ref - https://angular.io/guide/router

**HttpClient:**

**Objectives**

* Demonstrate using Service and HttpClient to invoke web service
  + generate service, dependency injection, HttpClientModule, HttpClient, HttpHeaders, Observable, get(), post(), put(), delete(), subscribe, error handling
    - Service & Dependency Injection - https://angular.io/guide/architecture-services
    - HttpClient - https://angular.io/guide/http

**Protractor:**

**Objectives**

* Demonstrate end to end testing using Protractor
  + Protractor and WebDriver Manager installation, describe(), it(), browser.get(). expect(), toEqual(), element(), by.model(), click(), getText(), testing multiple scenarios
    - Ref - https://www.protractortest.org/#/tutorial

* Demonstrate building and deploying an angular application in web server
  + ng build, deploy distribution folder in web server
    - Ref - https://angular.io/start/deployment