

# Programming Courseware

## Front End Programming

### 1,HTML5

#### HTML 5 COURSE SYLLABUS

**Overview:** HTML5 is a standard for structuring and presenting content on the Web. It incorporates features such as geolocation, video playback and drag-and-drop. HTML5 allows developers to create rich internet applications without the need for third party APIs and browser plug-ins.

#### Course Objectives:

- HTML5 comes with many new content-specific elements, like article, footer, header, nav, section.
- Improvements to HTML web forms where new attributes have been introduced for input tag with support for form controls like calendar, date, time, email, url, search etc...
- Introduction of canvas, which supports a two-dimensional drawing surface that you can program with JavaScript.
- Embedding audio or video to web pages without third-party plugins.
- Choice to the visitors to share their physical location with your web application.(Geo Location)
- CSS Support for additional selectors, Drop shadows, Rounded corners, multiple backgrounds, Gradients, Animation, Transparency and many more...
- The CSS3 specification is still under development by W3C. However, many of the new CSS3 properties have been implemented in modern browsers like New Selectors, Backgrounds and Borders, Gradient Effects, Text Effects and Fonts, Transformations, Transitions, Animations, Multiple Column Layout.

#### Pre-requisite / Target Audience:

- basic understanding of html and its tags

## **Module 1: HTML5 Introduction**

In this module, you will learn the evolution of html5. HTML5 is the latest standard for HTML. HTML5 was designed to replace both HTML 4, XHTML, and the HTML DOM Level 2. Limitations of HTML 4

- Introduction and Advantages of HTML 5
- First HTML5 Document
- Overview of New Features of HTML5
- List of HTML 4.01 elements removed from HTML5:

**Module 2: Page Layout Semantic Elements** In this module, you will learn what semantic elements is and how to use it. A semantic element clearly describes its meaning to both the browser and the developer.

- Header
- Navigation
- Section & Articles
- Footer
- Aside and more...

**Module 3: HTML5 Web Forms** In this module, you will learn about html web forms, HTML form on a web page allows a user to enter data that is sent to a server for processing. Forms can resemble paper or database forms because web users fill out the forms using checkboxes, radio buttons, or text fields.

- HTML 5 Global Attributes
- Displaying a Search Input Field
- Contact Information Input Fields
- Utilizing Date and Time Input Fields
- Number Inputs
- Selecting from a Range of Numbers
- Selecting Colors
- Creating an Editable Drop-Down

- Requiring a Form Field
- Autofocusing a Form Field
- Displaying Placeholder Text
- Disabling Autocomplete
- Restricting Values

**Module 4: Canvas API** In this module you will learn about canvas, The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images. Overview of HTML5 Canvas

- History
- What Is a Canvas?
- Canvas Coordinates
- Registering the Canvas dimensions
- Drawing on Canvas with paths, curves etc.
- Working with Solid colors, Gradients & Transparency
- Importing External Images & Setting the background
- Working with Color & Geometrical transformations
- Creating graphs & charts
- Working with Text
- Animating a Vertical Bar-Chart
- Fine tune animation with Acceleration & Easing.
- Working with Pixel Data
- CSS and Canvas
- Create High-Res, Retina-Display-Ready Media with Canvas
- Clipping Canvas drawings & saving them to a file.
- When Not to Use Canvas
- Fallback Content
- Implementing Canvas Security

- Ensuring backward compatibility Support of Canvas API to older versions of browsers

**Module 5: SVG API** In this module you will learn about svg, svg stands for Scalable Vector Graphics and it is an SVG viewer then renders a language for describing 2D-graphics and graphical applications in XML and the XML. SVG is mostly useful for vector type diagrams like Pie charts; Two-dimensional graphs in an X, Y coordinate system etc.

- Overview of SVG
- Understanding SVG
- Scalable Graphics
- Creating 2D Graphics with SVG
- Adding SVG to a Page
- Simple Shapes
- Transforming SVG Elements
- Reusing Content
- Patterns and Gradients
- SVG Paths
- Using SVG Text
- Putting the Scene Together
- Building an Interactive Application with SVG
- Adding the CSS Styles
- Ensuring backward compatibility Support of SVG API to older versions of browsers

**Module 6: HTML5 Media (Video & Audio)** In this module you will learn multimedia. HTML5 introduced two new elements that include playback functionality for supported media formats. The audio element can be used to add audio content to a web page.

- Adding Video & Audio to a page
- Supported Audio & Video formats
- Audio & Video codecs
- Loss & Lossless compression
- Media specific attributes Vs Global attributes
- Deployment challenges on Mobiles
- Converting Audio & Video to supported formats using open source & commercial software
- Using a Frame grabber
- Custom Controls, Seek bar, Progress bar with Javascript & CSS
- Applying CSS skins & transforms
- Working with multiple tracks, Subtitles & Captions with Captionator, Player & the Lean back Player
- Integrating Video with Canvas & SVG
- Applying Visual filters using Canvas & SVG
- Debugging, Browser support.
- Licensing issues.
- Ensuring backward compatibility Support of Media API's to older versions of browsers

**Module 7: Web Storage API** In this module you will learn web storage, web storage sometimes known as DOM storage (Document Object Model storage), provides web application software methods and protocols used for storing data in a web browser  
Goodbye cookies introducing Web Storage

- Browser Support
- Local Vs Session storage
- Using the HTML5 Web Storage API
- Setting and Retrieving Values

- Storing forms & caching events with local storage
- Storing & Accessing JSON data.
- Communicating Web Storage Updates
- Data Security
- Need more storage space - exploring Web SQL Database API & Indexed DB
- Creating a Grocery List with Web SQL DB
- Ensuring backward compatibility Support of Web Storage API to older versions of browsers

## Module 8: Geolocation

In this module, you will learn about geolocation. Geolocation is the identification or estimation of the real-world geographic location of an object, such as a radar source, mobile phone, or Internet-connected computer terminal. Comparing Geolocation techniques in the past & modern day Geolocation

- Understanding the pillars of Geolocation. i.e., GPS/ IP Address/ Cell IDs/ Wi-Fi and Bluetooth
- LBS (Location based services)
- Mobile & Augmented reality applications, which consume geolocation service.
- Understanding Latitude, Longitude, Speed, Course & Accuracy
- Getting you current location
- Browser compatibility & Fallbacks.
- Reverse geocoding
- Mapping location
- Getting Distance & Directions between two places.
- Following a moving location
- Combing geolocation with google maps
- Triggering the Privacy Protection Mechanism
- Saving Geographical information
- Geolocation usage – Geo Marketing, Geo social, Geo tagging, Geo social, Geo tagging & Geo applications.

- Building a Real-Time Application with HTML5 Geolocation
- Ensuring backward compatibility Support of Geolocation API to older versions of browsers

## **Module 9: Web Workers**

- In this module, you will learn about web worker. A web worker is a JavaScript that runs in the background without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.
- What are web workers?
- Possibilities & Limitations of web workers
- Inline, Dedicated & Shared Workers
- Creating a worker, Assign roles & deploying the same.
- Leveraging a Shared Worker
- Worker support in modern browsers
- Managing multiple workers
- Parsing data with workers
- Perform Heavy array computations
- Using timers in conjunction with worker
- Work with pixel manipulations
- Make twitter JSONP requests
- Connect to share workers at same time with multiple browser windows
- Transferable objects
- Debugging Your Workers
- Ensuring backward compatibility Support of Web Workers API to older versions of browsers

## **Module 10: HTML5-Server Sent Events**

- Server-sent events is a standard describing how servers can initiate data transmission towards clients once an initial client connection has been established. They are commonly used to send message updates or continuous data streams to a browser client and designed to enhance native, cross-browser streaming through a JavaScript API called Event Source.

## 2, CSS 3 COURSE SYLLABUS

**Overview:** CSS (Cascading Style Sheets) consist of a set of formatting rules that we use to control the layout and appearance of the content on a web page. One best feature of CSS is that you can store all the CSS rules in one document, keep that document separate from the HTML content, and link the two together. CSS3 offers some new and exciting features to enhance the appearance of the website. CSS3 makes it easier for the designers that will make the visitors go crazy over them to be implemented.

### Pre-requisite / Target Audience:

- basic understanding of html and its tags

### Module 1: INTRODUCING CSS3

In this module you will learn about css3, CSS3 is the latest upgrade in CSS levels. By Using CSS3, you can easily use old CSS element as well. There are a lot of new modules has been added in CSS3. But we must know that CSS3 is still under development and will be completed after sometimes. But most of its elements implemented in all major browsers. It has New Text effects .It has Transition effects. It has Animation effects and many other things.

- What CSS3 Is and How It Came to Be
- A Brief History of CSS3
- CSS3 Is Modular
- Module Status and the Recommendation Process
- CSS3 Is Not HTML5
- Let's Get Started: Introducing the Syntax
- Browser-Specific Prefixes
- Future-Proofing Experimental CSS
- Getting Started

**Module 2: BORDER AND BOX EFFECTS** In this module, you will learn this is the great incrementation in CSS3, We can add rounded border to any of the HTML elements. We don't need to use photoshop anymore for rounded corners. border-radius Shorthand

- Differences in Implementation Across Browsers
- Using Images for Borders
- Multicolored Borders
- Adding Drop Shadows
- Border and Box Effects: Browser Support

**Module 3: BACKGROUND IMAGES AND OTHER DECORATIVE** In this module, here we will learn how to set background-image and its properties to develop. The background-image property in CSS applies a graphic (e.g. PNG, SVG, JPG, GIF, WEBP) or gradient to the background of an element. There are two different types of images you can include with CSS: regular images and gradients.

- PROPERTIES
- Background Images
- Multiple Background Images
- Background Size
- Background Clip and Origin
- background-repeat
- Background Image Clipping
- Image Masks
- Background Images: Browser Support

**Module 4: 2D TRANSFORMATIONS** In this module, you will learn about transformation .transformation is an effect that lets an element change shape, size and position.

- The transform Property
- rotate
- Position in Document Flow
- transform-origin
- translate
- skew

- scale
- Multiple Transformations
- Transforming Elements with Matrices
- Reflections with WebKit
- 2D Transformations: Browser Support

### **3D TRANSFORMATIONS**

- 3D Elements in CSS
- Transform Style
- The Transformation Functions
- Rotation Around an Axis
- Translation Along the Axis
- Scaling
- The Transformation Matrix
- Perspective
- The perspective and perspective-origin Properties
- The Transformation Origin
- Showing or Hiding the Backface
- 3D Transformations: Browser Support

## **Module 5: TRANSITIONS AND ANIMATIONS**

In this module, By the Help of Transition Property of CSS3, We can transit our element in hover status.

Actually you can display the hover effects in the transitive way or animated way.

Animation is another and power full technique in CSS3; by using it, we can animate any element in frames.

- Transitions
- Property
- Duration
- Timing Function
- Delay

- Shorthand
- The Complete Transition Example
- Multiple Transitions
- Triggers
- More Complex Animations
- Key Frames
- Animation Properties
- The Complete Animations Example
- Multiple Animations
- Transitions and Animations: Browser Support

**Module 6: CSS3-Multi Column Layout** In this module, using css3 we can divide columns into multiple parts according to web page viewport. Let's check how to do multiple column layout

- CSS3 Multi-column Properties
- CSS3 Create Multiple Columns
- CSS3 Specify the Gap Between Columns
- CSS3 Column Rules
- How Many Columns an Element Should Span
- The Column Width
- CSS3 Multi-columns Properties

**Module 7: MEDIA QUERIES** In this module, Media Queries are a key component of responsive design, which make it possible for CSS to adapt based on various parameters or device characteristics. The @media at-rule is used to conditionally apply styles to a document. The Advantages of Media Queries

- Syntax
- Media Features
- Width and Height
- Device Width and Height
- Using Media Queries in the Real World
- Orientation
- Aspect Ratio
- Pixel Ratio
- Multiple Media Features
- Mozilla-Specific Media Features
- Media Queries: Browser Support

**At the end of the course, participants will be able to get:**

1. Create Web Page with HTML(5) & CSS(3)
2. How to set Headers, Paragraph for web page
3. How to set pages for webpage
4. Create animation elements
5. How to create a responsive website for all devices such as (Mobile, Tablet, Computer)
6. How to create Box and set Positions for elements
7. How To create buttons and use for pages or send forms
8. How to create to insert Video and Audio in webpage
9. How to create Vertical, Horizontal, Dropdown Navigation Bar (menus)
10. Create attractive different Forms
11. How to create Circle, Thumbnail and set Text on images

## 3, Javascript

**Overview:** JavaScript is a loosely-typed client side scripting language that executes in the user's web browser. A web page without JavaScript is unimaginable today. There are many open source application development frameworks based on JavaScript.

### Course Objectives:

- Understand the JavaScript language & the Document Object Model.
- Alter, show, hide and move objects on a web page.
- Check information inputted into a form.
- Javascript allows programming to be performed without server interaction.
- Javascript can respond to events, such as button clicks.
- Javascript can validate data before sending out a request.
- Javascript can adjust an HTML document for special effects.
- Javascript can create cookies! Cookies can be used to store and retrieve information from the user's computer.

**Pre-requisite / Target Audience:** An intermediate knowledge on HTML.

### Module 1: Introduction

In this module we will learn about what is the JavaScript and benefits of the language. We can also learn how write the first JavaScript program.

- JavaScript Introduction

### Module 2: Language Syntax

In this module we will learn about JavaScript language syntax and how to use the program level. And we can also learn understanding the arrays.

- Variable declaration
- Operators
- Control Statements
- Error Handling
- Understanding arrays
- Function Declaration

## Module 3: Built In Functions

In this module we will learn about what are the built in functions in JavaScript and how to use the JavaScript functions.

- Built In Functions
- Standard Date and Time Functions

## Module 4: HTML Forms

In this module we will learn about how to use the forms. What are the properties in JavaScript. HTML form validation can be done by JavaScript.

- HTML Document object Model
- Working with HTML form and its elements

### • **Module 5: HTML DOM**

In this module we will learn about JavaScript can access and change all the elements of an HTML document.

- HTML Document object Model
- Working with HTML form and its elements
- Other Document Object Model

## Module 6: Cookies

In this module we will learn about how to use the cookies. What are the advantages of the cookies and how to create cookies?

- Working with cookies

## Module 7: Working with Objects and Classes

- In this module we will learn about what is the use of objects and how to call the data. We can also learn about inheriting data.
- Working with Objects
- Call method in JavaScript
- Inheritance in JavaScript using prototype

**At the end of the course participants will be able to:**

- 1. HTML to define the content of web pages
- 2. CSS to specify the layout of web pages
- 3. JavaScript to program the behavior of web pages

## Note:

After completion of html, css,javascript there will be a project will assign to members.. Who ever members will complete the project they will promote to next level frontend Frameworks

# Frontend Frameworks

## 1,Angular JS

### AngularJS Syllabus

**Overview:** AngularJS is a structural framework for creating dynamic web applications. HTML is a great declarative language for static pages. It does not contain much for creating a dynamic applications. So Angular will be filling that gap. Angular's data binding and dependency injection eliminate much of the code than we would actually write. The best part is that it all happens in the browser by making it an ideal partner with any server technology.

### Course objectives:

- Reduce the amount of code you write to build rich user interface applications.
- Increase the reliability and maintainability of UI by using data binding.
- Retrieve data from back-end server, manipulate it and display it with ease.
- Modularise your code with the custom services and directives.
- Providing two way binding of data.
- Create Single Page Applications (SPA).

### Pre-requisite / Target Audience:

Working knowledge of HTML and JavaScript.

### Module 1: Introduction

In this module you will learn brief introduction to angular js and MVC pattern i.e. used in angular. And you will learn how to setup angular framework in local environment. You can also learn how to create your first application using angular framework.

- Introduction to AngularJS
- MVC Architecture
- Conceptual Overview
- Setting up the Environment

- First Application
- Understanding ng attributes

## **Module 2: Expressions and Data Binding**

In this module you will learn what expressions are and how to implement expressions , and what are the different types expressions In angular like Number and String Expressions, Object Binding and Expressions, and also use arrays and understanding data binding .

- Number and String Expressions
- Object Binding and Expressions
- Working with Arrays
- Forgiving Behavior
- Understanding Data binding

## **Module 3: Working with Directives**

In this module you will learn how to make use of built-in elements of angular Core Directives, how to use directives and different types of directives like styles directives,,mouse and keyboard events directives .

- Conditional Directives
- Styles Directives
- Mouse and Keyboard Events Directives

## **Module 4: Controllers**

In this module you will learn about \$scope context and controllers and how to use controllers and \$scope object ,passing parameters to the methods and also we will learn how to create nested controllers and Scope Inheritance , Multiple Controllers and their scopes.

- Understanding Controllers
- Programming Controllers & \$scope object
- Adding Behavior to a Scope Object
- Passing Parameters to the Methods

- Having Array as members in Controller Scope.
- Nested Controllers and Scope Inheritance.
- Multiple Controllers and their scopes

## **Module 5: Filters**

In this module you will learn what are pipes ,what is the use of pipes ,and also we will learn built-in pipes In angular,how to create custom pipes and how to use those custom pipes in our code.

- Built-In Filters
- Uppercase and Lowercase Filters
- Currency and Number Formatting Filters
- OrderBy Filter
- Filter Filter
- Creating Custom Filter

## **Module 6: Forms**

In this module you will learn how to deal with forms and its controls using angular framework, you will learn how to perform validations using HTML5 attributes (minlength, maxlength, pattern etc...) and how to make use of javascript events.

- Using Simple Form
- Working with Select and Options
- Input Validations
- Using CSS classes
- Form Events
- Custom Model update triggers
- Custom Validations

## **Module 7: Modules**

- In this module you will learn what are modules an why they are used. And also what is the Module Loading and Dependencies ,how to setup of application , and also we going to learn what is creation and retrieval.
- Why Module?

- Module Loading and Dependencies
- Recommended Setup of Application
- Creation vs Retrieval.

## Module 8: Services

In this module you will learn about services and Dependency Injection (DI). And also we will learn how to creating services , and advantages of services , how to inject dependencies in a service.

- Understanding Services
- Developing Creating Services
- Using a Service
- Injecting Dependencies in a Service

## Module 9: Ajax in AngularJS

In this module you will learn how to use built in services of angular framework and ajax implementation using \$http and \$q services. And also we will learn Ajax Implementation using \$http and \$q Service.

- \$http Service
- \$q Service
- Ajax Impl using \$http and \$q Service

## Module 10: Routing

In this module you will learn and introduction to SPA (Single page application) and how to perform routing in angular framework.

- Introduction to SPA
- Creating HTML Templates
- Configuring Route Provider.

## Real-time Project involving most of the above concepts with following will be provided

- Product Abstract Document
- Requirement Specification Document
- **Step-by-Step procedure for building the project from ground up**
- Complete Source Code

- Database Script with Sample data
- Instructions to Setup the Project on a Development box
- Instruction to Deploy the project on Production Box / Microsoft Azure

**At the end of the course participants will be able to**

1. Build native mobile apps for Android, iOS and using Angular 1.x
2. Understand the fundamentals of Angular Forms and its architecture
3. Present data in beautiful, interactive lists
4. Build forms and setting pages
5. Implement Single page application(SPA)

## 2,React JS

### **Module 1: Introduction to React**

In this module, You will learn about the introduction of react Js, Advantages and Limitation of react is. You can understand how to setup the react js environment using ASP.NET MVC.

- What is React Js
- Advantages of React Js
- Limitation of Re

### **Module 2: JSX**

- In this Module, You will learn about the Jsx syntax,Jsx Parser,Comments in jsx and naming Conventions in jsx.
- Overview of JSX
- Rendering an Element into the DOM
- Naming Conventions

### **Module 3: Components**

In this module, You will learn about the What is Components,Props and state

- Overview of Components
- Props
- State

### **Module 4: Components Life cycle**

In this module, You will learn about the Life Cycle of component and reusing of Component.

- Life Cycle methods
- Reusable Components

### **Module 5: Forms**

In this module, You will learn about submitting the form data to sever using react component and updating state.

## **Module 6: Props Validation**

In this module, You will learn about the Validating Props.

## **Module 7: Events**

In this module, You will learn about the Call back events.

## **Module 8: API Calls**

In this module, You will learn about the calling the services using react and Rendering the element into DOM.

- Using WebApi

## **Module 9: Using Flux**

In this Module, You will learn about the Flux pattern, Limitations of Flux and Flux Elements.

- Overview of Flux
- Flux Elements
- Limitations of Flux
- Advantages of Flux

## **Module 10: Animations**

In this Module, You will learn about the Animation using react Components.

### **Note:**

After completion of angular JS, React Js there will be a project will assign to members.. Whoever members will complete the project they will promote to next level Backend Programming and also we will refer them to real time jobs

# Back End Programming

## 1,Java

### Module 1- Introduction

- ✚ Types of Programming language and Paradigms.
- ✚ Java – what, where and why?
- ✚ Platform independency
- ✚ Comparison in Java with C and C++
- ✚ Role of Java Programmer in Industry.
- ✚ Java Evolution and History
- ✚ Features of Java Language.
- ✚ The Java Virtual Machine (JVM) – The heart of Java.
- ✚ Java's Magic Byte code
- ✚ JDK , JRE and JIT

## **Module 2- Language Fundamentals or Grammar of Java**

- ⊕ The Java Environment:
- ⊕ Installing Java in WINDOWS and LINUX.
- ⊕ Java Program Development in different environment.
- ⊕ Java Source File Structure
- ⊕ Introduction to VI, notepad, edit plus editor and Net beans, Eclipse IDE.
- ⊕ Compilation and Executions procedure using different editor and IDE.
- ⊕ Reference parameters, Output parameters.
- ⊕ Access specifiers and its requirement in java.
- ⊕ Naming conventions

## **Module 3- Reserve / Keywords present in Java**

- ⊕ Lexical Tokens, Identifiers

Abstract	Const	For	Implements	switch
Assert	Default	Go to	Package	super
Boolean	Do	If	Private	this
Break	Double	new	protected	throw
Byte	Else	Import	Instanceof	throws
Case	Enum	public	Return	static
Catch	Extends	Int	Interface	try
Char	Final	short	transient	void
volatile	Finally	Long	Strictfp	Class
Continue	Float	Native	synchronized	while

## Module 4- Primitive Data types and Block in java

- ✚ Data types
- ✚ int , char , float , double , Boolean , short , long , byte
- ✚ UNICODE system
- ✚ Value type, Reference type.
- ✚ Types and Scope of variables
- ✚ Static variables, Instance variable, Local variables, final variable, transient variable, volatile variable.
- ✚ Static block and Non-static block.
- ✚ Static,non-static,final,abstract, native and synchronized
- ✚ Communicate java application with other language using java native interface.

## Module 5- Java Operators

- ⊕ Arithmetic operators,
  - ⊕ Relational operators,
  - ⊕ Logical operators,
  - ⊕ Shift operators
  - ⊕ Assignment Operators,
  - ⊕ Unary operator
  - ⊕ Bitwise operators,
  - ⊕ Special operators.
  - ⊕ Ternary operator
- 
- ⊕ instanceof operator and typecasting.

## Module 6- Wrapper Class

- ⊕ Integer
- ⊕ Character
- ⊕ Float
- ⊕ Double
- ⊕ Boolean
- ⊕ Short
- ⊕ Long
- ⊕ Type conversions
- ⊕ Implicit conversion, Explicit conversion

## Module 7- Decision making and branching PROGRAMMING WITH JAVA

- ⊕ If statement
- ⊕ If....Else statement and if....else ladder.
- ⊕ Nested if
- ⊕ Multiple if
- ⊕ Switch... case statement
- ⊕ Conditional operator vs. if statement
- ⊕ Break and continue in java

## **Module 8- Decision making and looping**

- ⊕ While
- ⊕ Do
- ⊕ For
- ⊕ For each

## **Module 9- Object Oriented Programming**

- ⊕ Class Fundamentals.
- ⊕ Object & Object reference.
- ⊕ Life time of object & Garbage Collection.
- ⊕ Creating with Operating reference and Objects.
- ⊕ Constructor & initialization code block.
- ⊕ Access Control, Modifiers, methods
- ⊕ Nested, Inner Class & Anonymous Classes
- ⊕ Abstract Class & Interfaces
- ⊕ Defining Methods, Argument Passing Mechanism
- ⊕ Method Overloading, Recursion.
  
- ⊕ Dealing with Static Members.
- ⊕ Finalize () Method.
- ⊕ Native Method.
- ⊕ Use of "this" reference.
- ⊕ Use of Modifiers with Classes & Methods.
- ⊕ Design of Accessors and Mutator Methods
- ⊕ Cloning Objects, shallow and deep cloning
- ⊕ Generic Class Types

## **Module 10- Extending Classes and Inheritance**

- ⊕ Aggregation (HAS-A) and its uses.
- ⊕ Use and Benefits of Inheritance (IS-A) over aggregation in OOP.
- ⊕ Types of Inheritance in Java
- ⊕ Role of Constructors in inheritance
- ⊕ Polymorphism in OOP.
- ⊕ Overriding Super Class Methods.
- ⊕ Use of "super" keyword.
- ⊕ Restriction in case of method overriding.
- ⊕ Type Compatibility and Conversion
- ⊕ Implementing interfaces.
- ⊕ Dynamic method dispatching by down-casting and up-casting.

## **Module 11- Package**

- ⊕ Organizing Classes and Interfaces in Packages.
- ⊕ Package as Access Protection
- ⊕ Defining Package.
- ⊕ Advantage of package
- ⊕ Sub-Package
- ⊕ CLASSPATH Setting for Packages.
- ⊕ Making JAR Files for Library Packages
- ⊕ Import and Static Import
- ⊕ Creating .EXE and jar executable file.

## Module 12- Exception Handling

- The Idea behind Exception
- Exceptions & Errors
- Types of Exception
- Checked and Un-Checked Exceptions
- Control Flow in Exceptions
- Use of try and catch block
  
- Multiple catch block
- Nested try
- finally block
- throw keyword
- Exception Propagation
- throws keyword
- Exception Handling with Method Overriding
- In-built and User Defined Exceptions
- Exception handling rule in case of method overriding.
- How to handle unreachable statements using finally.

## Module 13- Array & String

- ↳ Defining an Array
- ↳ Single-Dimensional Array
- ↳ Initializing & Accessing Array
- ↳ Multi -Dimensional Array
- ↳ Jagged Array
- ↳ Arrays class
- ↳ Methods in Arrays class
- ↳ Sorting the elements of Array
- ↳ Searching, insert, delete dynamically.
- ↳ Matrix multiplication, addition, transpose, upper triangular, lower triangular, sparse matrix.
- ↳ String – what and why
- ↳ Operation on String
- ↳ Immutable String
- ↳ String comparison and concatenation
- ↳ Method of String class
- ↳ StringBuffer class and its methods.
- ↳ StringBuilder class in java.
- ↳ Creating Immutable class like String.
- ↳ Using Collection Bases Loop for String
- ↳ Tokenizing a String
- ↳ Object comparisons using Comparator and comparable interface.

## Module 14- Dancing and Singing together "Multithreading" "In Java

- ↳ Understanding Threads and process.
- ↳ Multithreading – what and why
- ↳ Creating Thread
- ↳ Thread Life-Cycle

- ↳ Thread Priorities
- ↳ Daemon thread
- ↳ Performing multiple job by multiple Thread.
- ↳ Runnable class.
- ↳ Synchronizing Threads - what and why
- ↳ Synchronized method
- ↳ Synchronized block
- ↳ Inter Communication of Threads
- ↳ Producer & Consumer problem without balancing
- ↳ Producer & Consumer problem with balancing using wait() & notify().

#### **Module 15 - Transformation from CUI to GUI "Applet"**

- ↳ Applet and its use
- ↳ Design Patterns using Applet and JApplet.
- ↳ Run Applet application by browser and applet tool.
- ↳ Applet Architecture.
- ↳ Parameters to Applet
- ↳ Life Cycle of Applet
- ↳ Embedding Applets in Web page.
- ↳ Graphics in Applet
- ↳ Displaying image in Applet
- ↳ Animation in Applet
- ↳ Painting in Applet
- ↳ Applet Communication
- ↳ Digital Clock in Applet
- ↳ Analog Clock in Applet

#### **Module 16- Input/output Operation in Java (java.io Package)**

- ↳ Streams and the new I/O Capabilities
- ↳ Understanding Streams
- ↳ File class and its methods.
- ↳ Creating file and folder using java code.
- ↳ The Classes for Input and Output
- ↳ FileOutputStream & FileInputStream
- ↳ FileWriter & FileReader
- ↳ Input from keyboard by InputStreamReader
- ↳ Input from keyboard by Console
- ↳ Input from keyboard by Scanner
- ↳ PrintStream class
- ↳ PrintWriter class

- ⊕ BufferedReader and BufferedWriter class.
- ⊕ Compressing and Uncompressing File.
- ⊕ Reading and Writing data simultaneously
- ⊕ DataInputStream and DataOutputStream
- ⊕ The Standard Streams
- ⊕ Working with File Object
- ⊕ Java & XML Data Binding
  - Marshalling
  - Unmarshalling

## **Module 17- GUI Programming and Designing Graphical User Interfaces in Java**

- ⊕ Components and Containers
- ⊕ Basics of Components
- ⊕ Using Containers
- ⊕ Layout Managers and user-defined layout.
- ⊕ BorderLayout , FlowLayout , GridLayout , GridbagLayout, BoxLayout.
- ⊕ AWT Components
- ⊕ Adding a Menu to Window
- ⊕ Extending GUI Features Using SWING Components
- ⊕ Designing GUI using Netbeans.
- ⊕ Advanced swing components like JProgressbar , JSlider, JRadioButton , JTree, JTable, JToggleButton, etc.

## **Module 18- Java Data Structure by the help of java.util Package.**

- ⊕ Collections of Objects
- ⊕ Stack
- ⊕ Queue & Deque
- ⊕ Use of HashSet & TreeSet
- ⊕ Sets
- ⊕ Map
- ⊕ Understanding Hashing
- ⊕ Use of ArrayList & Vector
- ⊕ Use of LinkedList.
- ⊕ Use of HashMap & TreeMap
- ⊕ LinkedHashMap class
- ⊕ Hashtable class
- ⊕ Generics

## Module 19 - Event Handling

- Event-Driven Programming in Java
- Event- Handling Process with AWT.
- Working with Listeners
- Event-Handling Mechanism with SWING.
- Event Classes and its methods.
- Adapter Classes as Helper Classes in Event Handling
- Applet with Event-Handling.

## Module 20 - Networking Programming

- Process and Networking Basics
- Client-Server Architecture.
- InetAddress class
- Communicate between two processes in single or different system.
- Two way communication
- Socket Overview
- Networking Classes and Interfaces
- Network Protocols
- Read and write operation between client and server.
- PrintWriter and BufferedReader class for read and write operation.
- Developing Networking Applications in Java
- Developing a chatting application.

## **Module 21- Database Programming using JDBC**

- ↳ Introduction to JDBC
- ↳ Steps to connect to the database
- ↳ JDBC Drivers & Architecture
- ↳ Types of JDBC Drivers.
- ↳ Connectivity with Oracle
- ↳ Connectivity with MySQL

## **Module 22- Projects on J2SE**

- ↳ A application just like Notepad
- ↳ A application like a Calculator
- ↳ A application like Address book
- ↳ Puzzle game
- ↳ Snake game
- ↳ A chatting Application.
- ↳ Paint Application
- ↳ Develop any editor.
- ↳ Library information System

## 2,Python

### **GETTING STARTED**

- History & need of Python
- Application of Python
- Advantages of Python
- Disadvantages of Python
- Installing Python
- Program structure
- Interactive Shell
- Executable or script files.
- User Interface or IDE

### **PYTHON FUNDAMENTALS**

- Working with Interactive mode
- Working with Script mode
- Python Character Set
- Python Tokens, Keywords, Identifiers, Literals, Operators
- Variables and Assignments
- Input and Output in Python

### **DATA HANDLING**

- **Data Types**
  - Numbers
  - Strings
  - Lists
  - Tuples
  - Dictionary
  - Set
  - Frozenset
  - Bool
  - Mutable and Immutable

## **STRING MANIPULATION**

- Introduction to Python String
- Accessing Individual Elements
- String Operators
- String Slices
- String Functions and Methods

## **LIST MANIPULATION**

- Introduction to Python List
- Creating List
- Accessing List
- Joining List
- Replicating List
- List Slicing

---

## **TUPLES**

- Introduction to Tuple
- Creating Tuples
- Accessing Tuples
- Joining Tuples
- Replicating Tuples
- Tuple Slicing

## **DICTIONARIES**

- Introduction to Dictionary
- Accessing values in dictionaries
- Working with dictionaries
- Properties

## **SET AND FROZENSET**

- Introduction to Set and Frozenset
- Creating Set and Frozenset
- Accessing and Joining
- Replicating and Slicing

## OPERATORS

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Membership Operators
- Identity Operators
- Bitwise Operators
- Assignment Operators
- Operators Precedence
- Evaluating Expression
- Type Casting

## PROGRAM CONTROL FLOW

- **Conditional Statements**
  - The if Statement
  - The if-else Statement
  - The if-elif Statement
  - Nested if Statements
  - Python Indentation
- **Looping and Iteration**
  - The For Loop
  - The While Loop
  - Loop else Statement
  - Nested Loops
  - Break and Continue
- **The Range Function**
  - Introduction to range()
  - Types of range() function
  - Use of range() function

## INTRODUCTION TO FUNCTIONS

- **Built-In Functions**
  - Introduction to Functions
  - Using a Functions
  - Python Function Types
  - Structure of Python Functions
  - E.g. - map, zip, reduce, filter, any, chr, ord, sorted, globals, locals, all, etc.

- **User Defined Functions**
  - Structure of a Python Program w.r.t. UDF
  - Types of Functions
  - Invoking UDF
  - Flow of Execution
  - Arguments and Parameters
  - Default Arguments, Named Arguments
  - Scope of Variables
  - Lambda function
- **Recursion Function**
  - Use of recursion function

## **MODULES AND PACKAGES**

- **Built-in Modules**
  - Importing Modules in Python Programs
  - Working with Random Modules
  - E.g. - builtins, os, time, datetime, calendar, sys, etc.
- **User Defined Functions**
  - Structure of Python Modules

## **FILE OPERATIONS**

- **Text and Bytes files**
  - Opening a file
  - Reading and Writing Files
  - Other File tools
- **MS Excel files**
  - Introduction to MS Excel files

## **CLASSES AND OBJECTS**

- Classes as User Defined Data Type
- Objects as Instances of Classes
- Creating Class and Objects
- Creating Objects By Passing Values
- Variables & Methods in a Class

## **EXCEPTION HANDLING**

- Default Exception and Errors
- Catching Exceptions
- Raise an exception
- Try.... except statement
- Raise, Assert, Finally blocks
- User defined exception

## **INTRODUCTION TO OOPS**

- Procedural Vs Modular Programming
- The Object Oriented Programming
- Data Abstraction
- Data Hiding
- Encapsulation
- Modularity
- Inheritance
- Polymorphism

## **DATABASE**

- Introduction to MySQL
- PYMYSQL Connections
- Executing queries
- Transactions
- Handling error

## **GUI PROGRAMMING**

- Introduction
- Tkinter programming
- Tkinter widgets
- Frame
- Button
- Label
- Entry

## **TURTLE PROGRAMMING**

- Introduction to Turtle
- Controlling Turtle
- Animation Programming

## **MULTITHREADING**

- Thread and Process
  - Starting a thread
  - Threading module
  - Synchronizing threads
  - Multithreaded Priority Queue
- 

## **NETWORKING**

- Socket Module
- Server-client-socket
- Connecting client server
- Client-server chatting program

## **REGULAR EXPRESSIONS**

- Match function
- Search function
- Grouping
- Matching at Beginning or End
- Match Objects
- Flags

## **ADVANCED CONCEPTS\***

- Decorators
- Generators
- Iterators
- Co-routines

## CGI\*

- Architecture
- CGI environment variable
- GET and POST methods
- Cookies
- File upload

## Projects:

1. Console Project
2. GUI Project

## Data Analytics Using Python

### REVISITING PYTHON

- List and dictionary comprehension
- Programming assignment

### INTRODUCTION TO DATA

#### ANALYTICS

- Why Analytics?
- Traditional Data Management
- Analytical tools
- Types of Analytics
- Hind sight, ore sight and insight
- Dimensions and measures
- Why learn Python for data analysis?
- Using the IPython notebook

---

#### LIBRARIES FOR DATA ANALYTICS

- Anaconda
- Numpy
- Scipy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn

#### JUPYTER NOTEBOOK

- Create Documentation
- Code mode
- Markdown mode

## STATISTICS:

- Mean, Median, Mode
- Z-scores
- Bias -variance dichotomy
- Sampling and t-tests
- Sample vs Population statistics
- Random Variables
- Probability distribution function
- Expected value
- Binomial Distributions
- Normal Distributions
- Central limit Theorem
- Hypothesis testing
- Z-Stats vs T-stats
- Type 1 type 2 error
- Chi Square test
- ANOVA test and F-stats

## NUMPY:

- Creating NumPy arrays
- Indexing and slicing in NumPy
- Downloading and parsing data
- Creating multidimensional arrays
- NumPy Data types
- Array tributes
- Indexing and Slicing
- Creating array views copies
- Manipulating array shapes I/O

## SCIPY:

- Introduction to SciPy
- Create function
- modules of SciPy

<p><b><u>MATPLOTLIB:</u></b></p> <ul style="list-style-type: none"> <li>• Scatter plot</li> <li>• Bar charts, histogram</li> <li>• Stack charts</li> <li>• Legend title Style</li> <li>• Figures and subplots</li> <li>• Plotting function in pandas</li> <li>• Labelling and arranging figures</li> <li>• Save plots</li> </ul> <p><b><u>PANDAS:</u></b></p> <ul style="list-style-type: none"> <li>• Using multilevel series</li> <li>• Series and Data Frames</li> <li>• Grouping, aggregating</li> <li>• Merge DataFrames</li> <li>• Generate summary tables</li> <li>• Group data into logical pieces</li> <li>• Manipulate dates</li> <li>• Creating metrics for analysis</li> <li>• Data wrangling</li> <li>• Merging and joining</li> <li>• Analytics Vidhya dataset- Loan Prediction Problem</li> <li>• Data Mugging using Pandas</li> <li>• Building a Predictive Model</li> </ul>	<p><b><u>SEABORN:</u></b></p> <ul style="list-style-type: none"> <li>• Style functions</li> <li>• Color palettes</li> <li>• Distribution plots</li> <li>• Categorical plots</li> <li>• Regression plots</li> <li>• Axis grid objects</li> </ul> <p><b><u>WEB SCRAPING:</u></b></p> <ul style="list-style-type: none"> <li>• Scraping Webpages</li> <li>• BeautifulSoup package</li> <li>• Real time project</li> </ul> <p><b><u>INTRODUCTION TO ML</u></b></p> <ul style="list-style-type: none"> <li>• What is ML? And Why ML?</li> <li>• Introduction to Supervised ML</li> <li>• Introduction to Unsupervised ML</li> <li>• Mathematical Background for ML- Matrix ops Probability Theory (Bayes' Theorem)</li> <li>• ML Glossary- Variable types, k-fold</li> <li>• CV, AUC, F1 score,</li> <li>• Overfitting / Underfitting</li> <li>• Data split &amp; hyper parameter</li> </ul>	<p><b><u>SCIKIT-LEARN</u></b></p> <p><b>Supervised learning –</b></p> <ul style="list-style-type: none"> <li>• Regression <ul style="list-style-type: none"> <li>➢ Different types of Regression-</li> <li>➢ Linear Regression</li> <li>➢ Logistic Regression</li> <li>➢ Decision tree Algorithms</li> </ul> </li> <li>• Classification <ul style="list-style-type: none"> <li>➢ Naive- Bayes'</li> <li>➢ KNN Classification</li> <li>➢ Curse of Dimensionality-PCA</li> <li>➢ SVM Classification</li> </ul> </li> </ul> <p><b>Unsupervised learning –</b></p> <ul style="list-style-type: none"> <li>• Clustering <ul style="list-style-type: none"> <li>➢ k-means clustering</li> </ul> </li> </ul> <p>Random Forest</p> <p><b><u>INTRODUCTION TO BIG DATA</u></b></p> <ul style="list-style-type: none"> <li>• What is Hadoop?</li> <li>• MapReduce</li> <li>• File handling with Hadoop</li> <li>• Pig</li> <li>• Pyspark</li> </ul> <p><b>Projects:</b></p> <p><u>3. Real time projects</u></p>
--	---	--

## 3,NodeJs

### Overview:

JavaScript's rising popularity has brought with it a lot of changes, and the face of web development today is dramatically different. The things that we can do on the web nowadays with JavaScript running on the server, as well as in the browser, were hard to imagine just several years ago, or were encapsulated within sandboxed environments like Flash or Java Applets.

Node.js is a packaged compilation of Google's V8 JavaScript engine, the libuv platform abstraction layer, and a core library, which is itself primarily written in JavaScript." Beyond that, it's worth noting that Ryan Dahl, the creator of Node.js, was aiming to create real-time websites with push capability, "inspired by applications like Gmail". In Node.js, he gave developers a tool for working in the non-blocking, event-driven I/O paradigm.

After over 20 years of stateless-web based on the stateless request-response paradigm, we finally have web applications with real-time, two-way connections.

### Course Objectives

- ❖ Understand the JavaScript and technical concepts behind Node JS
- ❖ Structure a Node application in modules
- ❖ Understand and use the Event Emitter
- ❖ Understand Buffers, Streams, and Pipes
- ❖ Build a Web Server in Node and understand how it really works
- ❖ Use npm and manage node packages
- ❖ Build a web application and API more easily using Express
- ❖ Connect to a SQL or Mongo database in Node
- ❖ Understand how the MEAN stack works

#### Pre-requisite / Target Audience:

- ❖ Basic Knowledge of JavaScript and OOPS
- ❖ Knowledge in async programming will be added advantage

#### Module 1:- Introduction to Node JS

In this module, you learn What is Node JS and advantages of Node JS and How Node JS Works and the difference between the traditional web server and what are the limitations of the traditional web server modal.

- ❖ Introduction
- ❖ What is Node JS?
- ❖ Advantages of Node JS
- ❖ Traditional Web Server Model
- ❖ Node.js Process Model

#### Module 2:- Setup Dev Environment

In this chapter, you will learn about the tools required and steps to setup development environment to develop a Node.js application and to work with the REPL to work with the Console.

- ❖ Install Node.js on Windows
- ❖ Installing in mac os
- ❖ Working in REPL
- ❖ Node JS Console

#### Module 3:- Node JS Modules

In this chapter, you will know what is a module, functions and examples. And achieving modularity and separation of concern with the NodeJS Modules.

- ❖ Functions
- ❖ Buffer
- ❖ Module
- ❖ Module Types
- ❖ Core Modules
- ❖ Local Modules
- ❖ Module.Exports

## Module 4:- Node Package Manager

This deals with working with npm to install and update packages and Updating your package to the NPM and managing and updating your current Packages.

- ❖ What is NPM
- ❖ Installing Packages Locally
- ❖ Adding dependency in package.json
- ❖ Installing packages globally
- ❖ Updating packages

## Module 5:- Creating Web server

In this chapter we will learn creating web server, hadling GET, POST,PUT and Delete requests and listening to certain port numbers and handling routing with basic web server.

- ❖ Creating web server
- ❖ Handling http requests
- ❖ Sending requests

## Module 6:- File System

In this module, we will work with files, reading, writing, updating files, and the concept of chunks, buffers, and uploading files synchronously and asynchronously.

- ❖ Fs.readFile
- ❖ Writing a File
- ❖ Writing a file asynchronously
- ❖ Opening a file
- ❖ Deleting a file
- ❖ Other IO Operations

## Module 7:- Debugging Node JS Application

In this chapter you will learn how to debug node js application. Debugging is a process of tracing the bugs and performance issues to optimize your code.

- ❖ Core Node JS debugger
- ❖ Debugging with Visual Studio

## Module 8:- Events

In this chapter you work with the events in Node JS, and the significance of the events, writing your own events because Node is event driven framework.

- ❖ EventEmitter class
- ❖ Returning event emitter
- ❖ Inheriting events

## Module 9:- Express.JS

In this chapter you will learn how to use express framework to create web applications. Express is the most popular framework to build efficient web applications with minimum coding.

- ❖ Configuring routes
- ❖ Working with express

## Module 10:- Serving Static Resources

In this chapter you will learn how to serve static html pages to the browser, and serving other file formats and restricting certain files.

In this you will to serve static resources with built in middle ware.

- ❖ Serving static files
- ❖ Working with middle ware

#### **Module 11:- Database connectivity**

In this chapter you will learn how to connect to SQL Server and perform CRUD operations. It is like ADO.net for MS.NET applications and JDBC for java. We will use different Node JS modules to connect with database.

- ❖ Connection string
- ❖ Configuring
- ❖ Working with select command
- ❖ Updating records
- ❖ Deleting records

#### **Module 12:- Template Engines**

In this chapter you learn how use template engines to perform 2 way databinding and appending dynamic data to the webpage and different view engines and their syntax.

- ❖ Why Template Engine
- ❖ What is Jade
- ❖ What is vash
- ❖ Example

#### **Real-time Project involving most of the above concepts with following will be provided**

- Product Abstract Document
- Requirement Specification Document
- Step-by-Step procedure for building the project from ground up
- Complete Source Code
- Database Script with Sample data
- Instructions to Setup the Project on a Development box
- Instruction to Deploy the project on Production Box / Microsoft Azure

**At the end of the course participants will be able to**

1. develop a web application which involves database operations.

# Database

## 1, Mysql Server

### Basics

- MYSQL Server Basics
- Database models
- ER Model Overview
- Data types
- Understanding Test Database
- Basics Queries
- Removing Duplicates
- Data Filters Using Operators
- Data Sorting

### Query Design & Functions

- Grouping
- Joins
- Arithmetic and String functions
- Advanced Functions
- SET Operators
- Creating Complex Queries
- DML operations - Insert, Update & Delete
- TSQL

### Database Operations

- Database Objects - Create, Alter and Drop Tables
- Views
- Complex Views
- Indexes
- Advanced Index Concepts
- How Data is Stored
- Security - User Management

## Procedures

- PLSQL Basics
- Programming Concepts in PLSQL
- Cursors
- Nested Cursors
- Functions
- Procedures
- Interoperability between Functions & Procedures
- Query Optimization Techniques
- **Live Project**

## 2,Mongodb

MongoDB Document Model

Creating an Atlas Free-Tier Cluster

Installing MongoDB on Windows

Installing MongoDB on OSX

Getting Data Into MongoDB

Using Compass to View Data in an Atlas Cluster

Working with Data in MongoDB

How to Connect to Your Atlas Cluster from Python

Intro to the Aggregation Framework

Your First Aggregation (Using \$group)

Incremental Improvements with \$sort and \$sortByCount

Wowza! You can do that? (\$facet)

Filtering on Scalar Fields (\$match, find(), and Compass)

mongoimport

Creating an Atlas Free-Tier Cluster

Getting Data Into MongoDB

Using Compass to View Data in an Atlas Cluster

Working with Data in MongoDB

How to Connect to Your Atlas Cluster from Python

Connecting to Atlas

Intro to the Aggregation Framework

Your First Aggregation (Using \$group)

Incremental Improvements with \$sort and \$sortByCount

Analyzing Data with Aggregation

## Projecting Queries (Part 1)

Projecting Queries (Part 2)

Projecting Queries (Part 3)

Updating Documents (Part 1)

Updating Documents (Part 2)

Bulk Updates

Data Types in MongoDB

Filtering on Array Fields

How MFlix Works with MongoDB

Sort, Skip, and Limit

Query movies using operators

The \$elemMatch Operator

Querying on tomatoes subdocument

Inserting Comments in MFlix

Updating comments

Deleting Data in MFlix

Setting Up MFlix

Data Types in MongoDB

Cleansing Data with Updates

Sort, Skip, and Limit

Querying movies using operators

The \$elemMatch Operator

Querying for Documents on an Array Field

Querying on subdocuments

Inserting Comments in MFlix

Updating Documents

Deleting Data in MFlix

## Indexes in Movies (Part 1)

Indexes in Movies (Part 2)

Geospatial Queries

Graphing with MongoDB

MongoDB Indexes

Improve Query Performance

Geospatial Queries

Finding Things Nearby

Graphing with MongoDB

Making Plots with Matplotlib & MongoDB

# Version Control

## Git

### DevOps and Git in a Nutshell

Git Overview

(Sourcetree) Installation and Getting Started

(Command Line) Installation and Getting Started

Git Locations

(Sourcetree) Create a Local Repository

(Command Line) Create a Local Repository

(Sourcetree) Commit to a Local Repository

(Command Line) Commit to a Local Repository

Create a Remote Repository

(Sourcetree) Push to a Remote Repository

(Command Line) Push to a Remote Repository

(Lab- Sourcetree) Installation and Getting Started

(Lab- Command Line) Installation and Getting Started

(Lab- Sourcetree) Create a Local Repository

(Lab- Command Line) Create a Local Repository

(Lab- Sourcetree) Commit to a Local Repository

(Lab- Command Line) Commit to a Local Repository

(Lab) - Create a Remote Repository

(Lab- Sourcetree) Push to a Remote Repository

(Lab- Command Line) Push to a Remote Repository

## Branching and Merging I

This module discusses two of the key concepts involved with Git: branching and merging. It starts with foundational knowledge on Git's commit graph model, Git IDs and references. Branching and merging are then covered, including the difference between fast-forward merges and using merge commits.

### Git's Graph Model

(Sourcetree) Git IDs

(Command Line) Git IDs

(Sourcetree) Git References

(Command Line) Git References

(Sourcetree) Branches

(Command Line) Branches

(Sourcetree) Merging

(Command Line) Merging

(Lab- Sourcetree) Git References

(Lab- Command Line) Git References

(Lab- Sourcetree) Branches

(Lab- Command Line) Branches

(Lab- Sourcetree) Merging

(Lab- Command Line) Merging

## Branching and Merging II

This module continues the discussion of branching and merging. It starts with resolving a merge conflict. Then synchronizing with remote repositories is covered, including the fetch, pull and push commands. The other major topic of this module is rewriting history. The commit history is an important part of a project. A clean commit history makes it easier to understand the project, and then contribute to it by adding new features or fixing problems. If you have not shared your commits, you can rewrite them in a way that makes a cleaner project history. Rewriting history includes amending commits and rebasing.

### (Sourcetree) Resolving Merge Conflicts

(Command Line) Resolving Merge Conflicts

(Sourcetree) Tracking Branches

(Command Line) Tracking Branches

(Sourcetree) Fetch, Pull and Push

(Command Line) Fetch, Pull and Push

(Sourcetree) Rebasing

(Command Line) Rebasing

(Sourcetree) Rewriting History

(Command Line) Rewriting History

(Lab- Sourcetree) Resolving Merge Conflicts

(Lab- Command Line) Resolving Merge Conflicts

(Lab- Sourcetree) Tracking Branches

(Lab- Command Line) Tracking Branches

(Lab- Sourcetree) Fetch, Pull and Push

(Lab- Command Line) Fetch, Pull and Push

(Lab- Sourcetree) Rebasing

(Lab- Command Line) Rebasing

(Lab- Sourcetree) Rewriting History

## (Lab- Command Line) Rewriting History

### Git Workflows

This module covers topics on how teams work with Git. Git is flexible enough to work with many types of projects and teams. The better you understand Git's capabilities, the more effectively you can use it in your projects. Topics include pull requests, centralized workflows, feature branch workflows, forking workflows and Gitflow workflows. This module concludes with a project in which you use Git with a Gitflow workflow.

#### [\(Sourcetree\) Pull Requests](#)

#### [\(Command Line\) Pull Requests](#)

#### [Pull Requests](#)

#### [Git Workflows](#)

#### [\(Lab- Sourcetree\) Pull Requests](#)

#### [\(Lab- Command Line\) Pull Requests](#)

#### [\(Lab- Sourcetree\) Pull Requests](#)

#### [\(Lab- Command Line\) Pull Requests](#)

#### [Final Project](#)

# Cloud service providers Certifications

1, Aws Certified Solutions Architect

2,Microsoft Azure Administrator

3,GCP Associated Cloud Engineer

## Containers

Docker

Kubernetes

## Continuous Integration and continuous development

Jenkins

Servers

Nginix

Apache

Weblogic

## Project methodology

Agile

Scrum

Microservices

