**Typescript**

**Duration: 3 Days**

**Day 1**

**1. Introduction to TypeScript**

**• History and Importance of JavaScript**

**• JavaScript Strengths, Weaknesses**

**• Limitations of JavaScript**

**• Advantages of TypeScript**

**• TypeScript Design Goals**

**• Installing TypeScript**

**• Getting Started**

**2. TypeScript Language Basics**

**• Declaring Variables**

**• Basic Types**

**• Type Annotations**

**• Non-Nullable Types**

**• Basic Data Structures**

**• Operators**

**• Flow Control Statements**

**3. Using Visual Studio Code for TypeScript Development**

**• Tooling Options**

**• VS Code Features**

**• Limitations**

**• Project System**

**• Navigation Basics**

**• Productivity**

**• JavaScript Intellisense**

**• Configuring TypeScript Projects**

**• Compiling TypeScript**

**• Debugging**

**Day 2**

**4. Task Automation, Unit Testing, and Continuous Integration**

**• Need for Task Automation**

**• Task Runners**

**• Introduction to Gulp**

**• Using Gulp**

**• Configuring Build Tasks**

**• Debugging Gulp Tasks**

**• Testing Approaches**

**• Testing Frameworks**

**• BDD with Jasmine**

**• Running Tests in a Browser**

**• Writing Tests in TypeScript**

**• Test Runners**

**• Continuous Integration**

**5. TypeScript Type System**

**• Type Compatibility**

**• Type Inference**

**• Access Modifiers**

**• Union Types**

**• Type Guards**

**• Intersection Types**

**• Type Aliases**

**• String Literal Types**

**• Polymorphic “this” Types**

**6. Functional Programming**

**• Functional Programming**

**• Named Functions**

**• Arrow Functions**

**• Function Types**

**• Optional and Default Parameters**

**• Rest Parameters**

**• Function Overloading**

**7. Asynchronous Programming**

**• Importance of Async**

**• Callbacks**

**• Callback Hell**

**• Promises**

**• Generators**

**• Async and Await**

**Day 3**

**8. Object-Oriented Programming**

**• Introduction to Object-Oriented Programming**

**• SOLID Principles**

**• Interfaces**

**• Classes**

**• Inheritance**

**• Mixins**

**• Abstract Classes**

**• Class Expressions**

**9. Generics**

**• Algorithm Reuse**

**• Generic constructs**

**• Type Parameters**

**• Constraints**

**• Generic Functions**

**• Generic Interfaces and Classes**

**10. Namespaces and Modules**

**• Name Collisions**

**• Namespaces**

**• Defining Namespaces**

**• Limitations of HTML Script Tags**

**• Module Loaders**

**• ECMAScript 2015 Module Support**

**• Module Organization**

**• Exporting from Modules**

**• Importing from Modules**

**• Module Guidelines**

**11. Practical TypeScript with Express and Angular**

**• SPA Architecture**

**• Introduction to REST**

**• Express Framework**

**• Routing Basics**

**• MV\* Architecture**

**• Web Client Frameworks**

**• Angular 2 Concepts**

**• Angular Components**

**• Angular Templates**

**• Angular Metadata**

**• TypeScript Decorators**

**• Services and Dependency Injection**

**• HTTP Requests**