**Current Course objectives:**

This course will enable students to

Learn .NET Framework and OOPS concept in C#

Implement the concepts of Delegates, Events, AJAX and ADO .NET

Develop window application using C# .NET

Implement Web application using ASP .NET and AJAX

**Recommended Course objectives:**

This course will enable students to Qualify for in-demand job titles: Back-End Developer, Website Developer, Software Engineer

**Module 1** : Introduction to

* Libraries , Frameworks , Platforms ,
* Implement Version Control systems & Manage code revisions & CI / CD Pipelines using Jenkins.
* API development & Principles of REST architecture
* Build a basic API using Spring boot with a pipeline

**Module 2** : OOPS & OOAD concepts  ( as is also include OOAD basics)

**Module 3** : Introduction to the MERN stack Web Development, Front-End Development/React,

and Back-End Development.

* Event driven concepts in Node JS

Each module builds on the next, and is designed to prepare you to enter the job market as an entry-level full-stack developer, or to specialize in one of these areas with further skill development.

**Module 4:** Web Development

You will develop foundational skills in web development using JavaScript, HTML, and CSS. You’ll also create your own personal portfolio in GitHub, which you will add to as you progress.

**Module 5**: Front-End Development and React

You will dive deep into front-end development using one of the most popular frameworks, React. You will write clean, concise code with JavaScript ES6 and use these skills to work with web components in React. You will learn how to build React applications and how to test and deploy them.

**Module 6**: Back-End Development

You will work with MongoDB, ExpressJS, and NodeJS to become confident in the MERN stack.

Object-Oriented Analysis and Design (OOAD) is a software engineering methodology that involves using object-oriented concepts to design and implement software systems. OOAD involves a number of techniques and practices, including object-oriented programming, design patterns, UML diagrams, and use cases. Here are some important aspects of OOAD:

Object-Oriented Programming: Object-oriented programming involves modeling real-world objects as software objects, with properties and methods that represent the behavior of those objects. OOAD uses this approach to design and implement software systems.

Design Patterns: Design patterns are reusable solutions to common problems in software design. OOAD uses design patterns to help developers create more maintainable and efficient software systems.

UML Diagrams: Unified Modeling Language (UML) is a standardized notation for creating diagrams that represent different aspects of a software system. OOAD uses UML diagrams to represent the different components and interactions of a software system.

Use Cases: Use cases are a way of describing the different ways in which users interact with a software system. OOAD uses use cases to help developers understand the requirements of a system and to design software systems that meet those requirements.