## PEDAGOGY:

The program content will be presented by experienced faculty members from Great Learning as well as industry professionals. This ensures that the program combines academic rigor with practical industry relevance, providing learners with an exceptional learning experience.

During the sessions, the mentor will serve as the instructor, reinforcing concepts learned through self-paced content, introducing advanced topics, demonstrating concepts through examples, and addressing any doubts or questions to ensure effective learning.

The instructor will actively engage with the learners, encouraging them to share their screens and showcase their code. Additionally, the instructor will ask questions related to the fundamental concepts used in the code to assess and reinforce the learners' understanding.

The goal is to create an interactive and engaging learning environment where learners actively participate, apply their knowledge, and receive guidance from the instructor to enhance their learning outcomes.

## AIM:

Our aim is to foster a dynamic and engaging learning environment through a well-balanced blend of video lectures and online sessions.

We strive to provide a flipped classroom experience, where students have the opportunity to watch the pre-recorded video lectures prior to the live sessions. During the live sessions, the instructor will reinforce the key concepts by working through examples and problem-solving exercises. This approach allows for a deeper understanding of the concepts as the instructor revisits and emphasizes the important points and principles involved.

By actively involving students in problem-solving activities, we aim to enhance their comprehension and retention of the subject matter. Our goal is to create an interactive and collaborative learning experience that facilitates a comprehensive grasp of the material and promotes a strong foundation for further exploration and application.

## Positives:

By revisiting and reinforcing concepts, we aim to ensure a solid understanding of the subject matter. This approach allows learners to strengthen their knowledge and build a strong foundation.

Additionally, our focus on creating a practical class environment enables learners to apply their knowledge in real-world scenarios. Through hands-on exercises and problem-solving activities, learners can gain valuable experience and develop practical skills.

Overall, this combination of concept revision and practical application cultivates a robust learning environment, where learners can enhance their understanding and proficiency in the subject matter.

# DAY WISE BREAKDOWN

|  |
| --- |
| **Day1 Practical Demo(2 hours of live session):**   1. Visual Studio 2019 Installation: Guide learners through the process of installing Visual Studio 2019 on their machines. Provide step-by-step instructions and troubleshoot common installation issues.(Optional) 2. Console Application: Create a console application that prompts the user for their name and greets them. Use the Console.WriteLine and Console.ReadLine methods to interact with the user and display the output in the console. 3. Basic Class Creation: Create a class called "Person" with properties like Name, Age, and Gender. Show how to instantiate objects of the Person class, set the properties, and display the information using Console.WriteLine. |

## Day 2 :

|  |
| --- |
| **Day 2 Practical Demo (2 hours of live session):**   1. Arrays: Create an array of integers and demonstrate how to initialize it, access individual elements, and perform basic operations like finding the sum or maximum value of the array. Use loops to iterate over the array and display the elements. 2. Control Statements: Implement a program that checks if a given number is even or odd using the if statement. Demonstrate the usage of for loops, foreach loops, and switch statements in different scenarios. 3. Access Modifiers: Create a class with private and public methods, properties, and fields. Show how the private members are accessible only within the class, while the public members can be accessed from outside the class. |

## Day 3 (2 hours of live session):

|  |
| --- |
| **Day 3 Practical Demo(2 hours of live session):**   1. Methods: Create a method called CalculateArea that takes the radius of a circle as an argument and returns the area of the circle. Demonstrate how to call the method, pass the radius value, and display the calculated area. 2. Namespaces: Create multiple classes in different namespaces, such as "Geometry" and "Utilities". Demonstrate how to use the using directive to import namespaces and access classes from different namespaces. 3. Unit Testing: Write a basic unit test for the CalculateArea method using a testing framework like NUnit or XUnit. Show how to assert the expected result using assertions like Assert.AreEqual and execute the test to ensure the method is functioning correctly. |