

**Program Highlights**

Learning Journey Guide

Java Track

–

Full



**Program at a glance**

Learning consisting of

2

Stages:

•

Stage 2

–

Advanced

Programming

•

Integrated Development Project (IDP)

•

The complete learning journey is

formalized using adult learning principles, where problem solving and

applying the skills gained are given more importance than conceptual learning.

•

Get mentored by SME, whose motivation and guidance will help you accelerate in the learning journey.

•

Lear

ner Autonomy is encouraged via Flipped Classroom, where the learning platform offers world class

learning resources, and students would not be constrained by tutelage of an instructor.

•

Towards the end of this training program, GenCs will be completed the

ir first real

-

time project.



**Why do we need this**

**Academy enablement P**

**rogram?**

Academy enablement program engages young talents with a comprehensive learning pathway, giving these

millennials an opportunity to interact with Subject Matter Experts (SME) and understand the corporate environment

and groom themselves even before they joi

n us.

Cognizant emphasizes on Learner Autonomy where students take charge of their own learning, with the available

tools and resources. More focus is on “learning” than “teaching”. Get ready to embark your own learning

adventure!

Generation Cognizant (GenC)

DX-

Sitecore-v5 (Direct Stage2)

Learning Guide

**Know Your Service Line – Digital Experience - Sitecore**

**Service Lines**

Service lines can simply be defined as a modern organizational structure strategy for resource planning and allocation for any size of business. Typically, traditional organizational structure models are more vertically aligned -- think of an employee who has several bosses in the hierarchical ladder before being directly under the company’s owner or president. Conversely, service lines follow a more horizontal continuum approach, where the company is strategically segmented into more manageable departments. The service line approach tends to focus more on the requirements of customers, which often results in noticeable increases in the customer satisfaction rate.

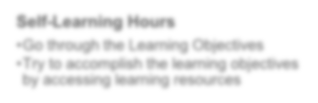
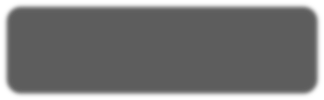
**Learning Journey with Flipped Classroom**

Journey wi Flipped Classroom

This program encourages you to be more autonomous learners during guided self-learning hours, completing the learning objectives on your own pace and style, and get ready for the hands-on practice time.

The complete learning path is set in the [GEN C Learn Platform,](https://cognizant.tekstac.com/login/index.php) which you can login with SSO.

**Flipped Classroom**



**Self**

**-**

**Learning Hours**

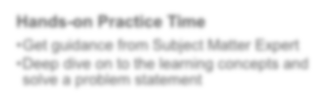
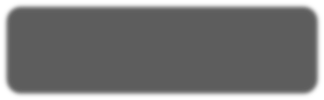
•

Go through the Learning Objectives

•

Try to accomplish the learning objectives

by accessing learning resources



**Hands**

**-**

**on Practice Time**

•

Get guidance from Subject Matter Expert

•

Deep dive on to the learning concepts and

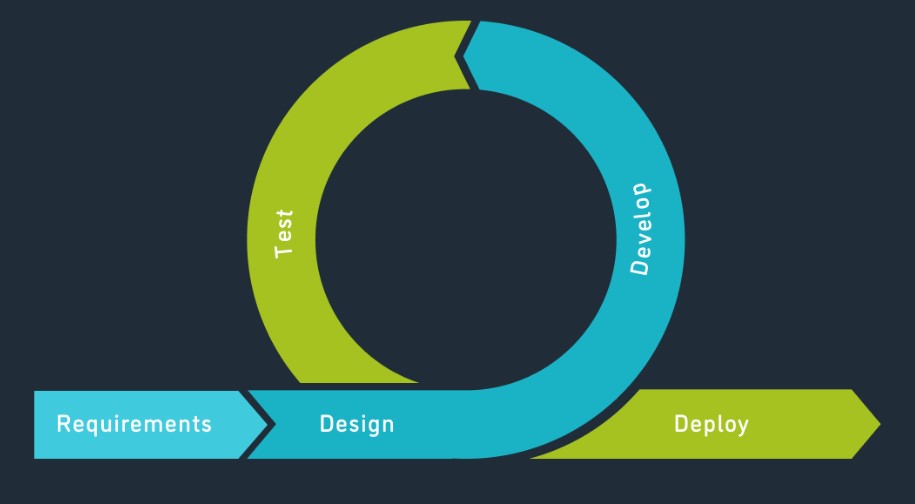
solve a problem statement

**Integrated Development Project (IDP)**

**What is Integrated Development Project (IDP)?**

Integrated Development Project is an approach wherein the learner experiences the entire software development processes in an incremental fashion as part of the GenC Training. The IDP implementation is purely based on **Agile Software Development** methodologies and inspired from **PBL (Project-Based Learning)** which is learning while doing. It gives learners the opportunity to gain a deeper understanding of a topic through problem-solving using real-world examples and challenges.

Following is the Agile Development Methodology at high-level.



# Stages of IDP

Following are the four seminal phases of IPD.

Ideation

Requirement

Analysis

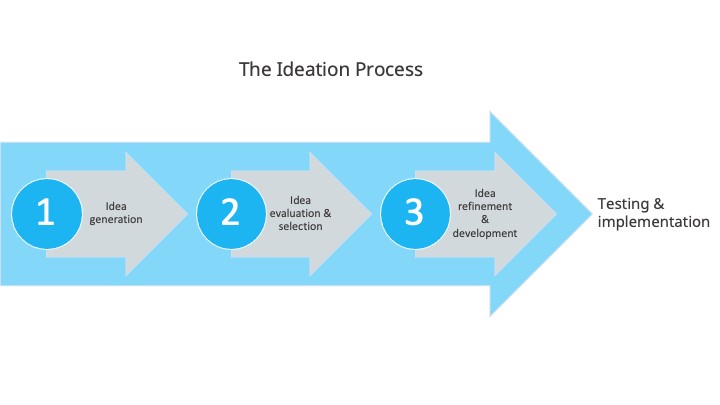
Project Design

Project

Development

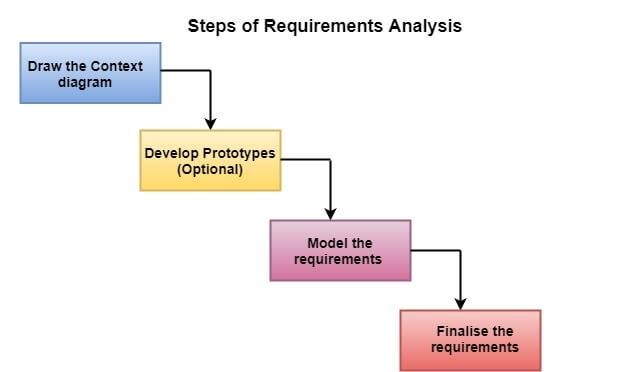
## Phase 1: Ideation

Ideation is the creative process of generating, developing, and communicating ideas. It’s important to note that these ideas don’t have to be completely new. You can ideate to solve specific problems, look into new ways of implementing a solution, or even collect feedback and evaluate ideas.



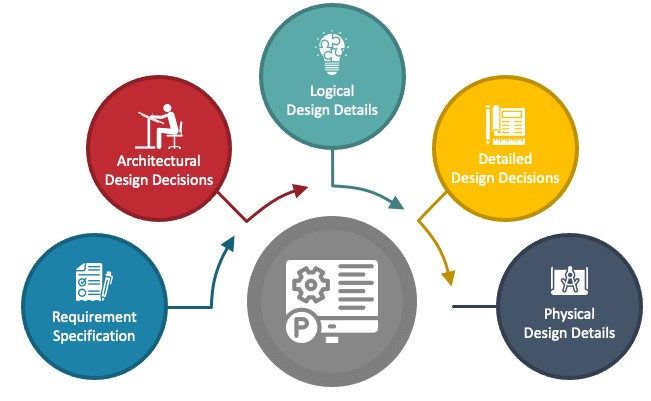
## Phase 2: Requirement Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.



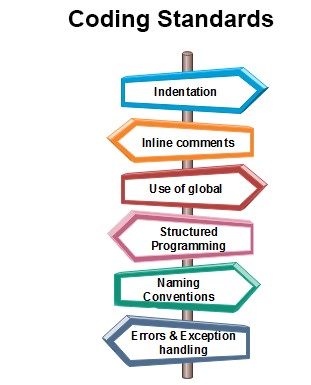
## Phase 3: Project Design

Project design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.



## Phase 4: Project Development

Once the system design phase is over, the next phase is development. In this phase, developers start build the entire system by writing code using the chosen programming language. In this phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.



**Recommended Program Sequence**

The learning journey starts with **5 days of Icebreaker sessions** followed by technical learning that contains **2 stages**, followed by a **Project.**

* Stage 2 – Advanced Programming
* Integrated Development Project (IDP)

Stage 2 – Advanced Programming + IDP

**Interim Evaluation (Project + Technical)**

**Final Evaluation (Project + Technical)**

## Key Learning and Evaluation Components of the Program

**Self-Learning via Udemy**

Cognizant has collaborated with Udemy to provide world class learning videos for the evolving future of work. These Udemy programs are woven into a learning path, empowering you to plan and learn at your style.

The program also connects you with Subject Matter Experts (SMEs) to get the professional guidance on your queries in the learning journey.

**RAG as PHS (Performance Health Status)**

The program continuously evaluates if you are able to apply those self-learnt skills to solve a real-time business problem. Depicted below are the two key evaluation components, which are distributed across the learning journey for the purpose of continuous evaluation.

**Interim Evaluation:**

During the interim evaluation, the GenC will undergo a video interview on the learning platform. This interview will be conducted by a Tech SME from the BU. The purpose of this evaluation is to assess the GenC's knowledge and understanding of the skills covered in the training program up to the halfway point. It also encompasses an evaluation of the GenC's progress in their Integrated Development Project (IDP). The evaluation will involve a technical discussion as well as an assessment of the IDP progression to gauge the GenC's proficiency in the skills learned thus far.

**Final Evaluation:**

For the final evaluation, the GenC will participate in a video interview conducted by a Tech SME from the BU. This evaluation aims to assess the GenC's knowledge and expertise in all the skills covered throughout the entire training program. Similar to the interim evaluation, this assessment will involve a technical discussion via a video interview on the learning platform, along with a project evaluation to assess the GenC's capabilities and their IDP's progress. It serves as a comprehensive evaluation of the GenC's skills and capabilities acquired during the training.

**Icebreaker Sessions**



Icebreaker session will be conducted for a duration of initial **5 days**. During the session, various topics related to Corporate Induction, Talent Management, Cognizant Agenda on Core Values, Leader Talks, Alumni, BU Mentor connects will be covered. Followed by icebreaker, technical training will kick start.

**Following sessions will be covered during the 5 days of icebreaker**

* Corporate Induction
* Talent Manager Connect
* Cognizant Agenda Sessions on Core Values
* Leader Talks (Academy) and many more…

## age 1 – Foundational Technology Skills Duration: 5

**How and From Where to Learn?**

* Udemy courses are recommended for learning, and hands-on exercises are organized within a learning path on the **Tekstac** platform for practice. Additionally, you can utilize other sources mentioned in this handbook for learning.

**Stage 1 -> Course -> Agile Methodology**

**Course Overview**

In the **Course 1** of the **Stage 1**, learners will be introduced to the basics of **Agile methodology**. Agile is an approach to project management and software development that emphasizes flexibility, collaboration, and customer satisfaction. It involves adaptive planning, iterative development, early delivery, and continuous improvement. Agile methodologies, like Scrum and Kanban, focus on delivering value to the customer and responding to change effectively.

**Learning Objectives**

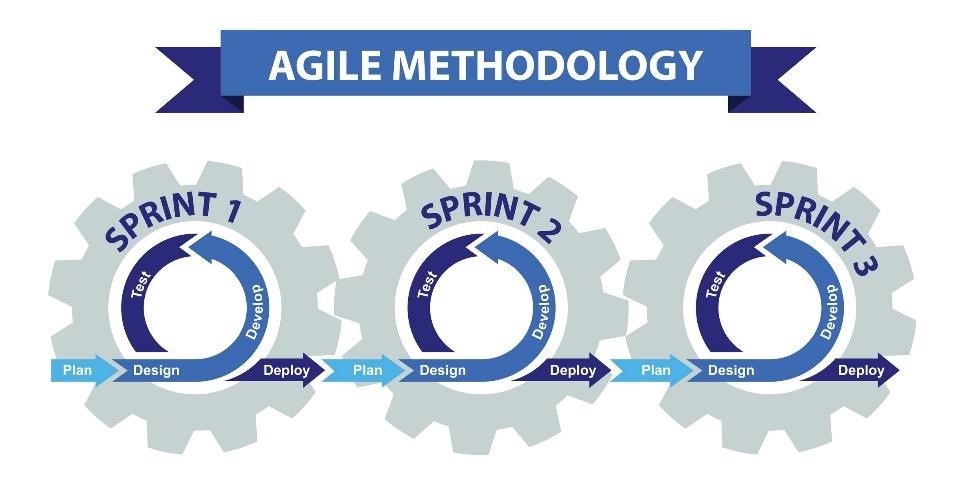
After completing this course, GenCs will be able to:

* Understand the principles and values of Agile methodology.
* Describe the benefits of using Agile in software development.
* Explain the differences between Agile and traditional project management approaches.
* Identify the key roles and responsibilities in Agile teams.
* Describe the iterative and incremental nature of Agile development.
* Explain the importance of customer collaboration and feedback in Agile.
* Describe common Agile practices, such as user stories, sprints, and retrospectives.
* Identify common Agile frameworks, such as Scrum, Kanban, and Extreme Programming (XP).
* Explain how Agile principles can be applied in different project environments.

|  |
| --- |
| Day 1 |

**What is Agile?**

Agile is a set of principles that are used to improve the process of project management and software development. To put in simple terms, Agile helps teams in delivering value to customers quickly and effortlessly.



### Agile Principles

Here is a sneak peek into some of the principles that help make the Agile Process what it is:

1. **Customer satisfaction:** Customers need to be satisfied with the quick delivery of the product at the earliest.
2. **Welcome change:** Even if the change is late in the development process, it needs to be addressed and handled as soon as possible.
3. **Deliver frequently:** The focus must be on the continuous delivery of software in a shorter timescale.
4. **Work together:** Business units and developers need to work in tandem throughout the project lifespan.
5. **Motivated team:** The projects need to have motivated team members. They must also be trusted to get the work done.
6. **Face-to-face:** Conversations that take place face-to-face have maximum efficiency and effectiveness.
7. **Working software:** The primary measure of progress is evaluated based on the working software created.
8. **Constant pace:** The agile process is greatly beneficial when it comes to sustainable development.
9. **Good design:** Focusing on technological excellence and good design can significantly affect agility.
10. **Simplicity:** The amount of work not being done needs to be reduced via simpler processes.
11. **Self-organized:** Self-organized teams end up providing the best architectures, designs, and requirements.
12. **Reflection and adjustment:** The effectiveness can be significantly improved by regular reflection on it, by the team.

**What Are the Advantages of the Agile Process?**

* Thanks to agile, there will be plenty of interaction between the project team and the clients.
* The clients can have greater insight into every phase of the project, due to improved transparency.
* The outputs are easily predicted, and can sometimes be delivered faster than expected.
* Most projects follow a rigid schedule and can incur predictable costs.
* Agile enables changes that can empower the product catalog to be refined and reprioritized.
* The maximum project value can be ensured since the client can decide the priorities of the features.
* By understanding the needs of the customer, the team can provide more value effortlessly.
* Since the project is broken down into smaller units, development, testing, and collaboration will be of the highest quality.

Learn about Agile process from the below Udemy course.

[Agile Fundamentals: Including Scrum & Kanban](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcognizant.udemy.com%2Fcourse%2Fagile-fundamentals-scrum-kanban-scrumban%2F&data=05%7C02%7CKanimozhi.Selvakumar%40cognizant.com%7C834f7a4b280146a48be908dd36fcda21%7Cde08c40719b9427d9fe8edf254300ca7%7C0%7C0%7C638727183047825952%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=G1PDUCvgrXwIO5p7Cu1k%2FmfoNd9PsaenWOvFiGN1qII%3D&reserved=0)

* Go through the given sections and implement examples along with the author of this course.

## Software Development Life Cycle- Overview

**What is SDLC?**

SDLC is a process that defines the various stages involved in the development of software for delivering a high-quality product. SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.

Adhering to the SDLC process leads to the development of the software in a systematic and disciplined manner.

**Purpose:**

Purpose of SDLC is to deliver a high-quality product which is as per the customer’s requirement.

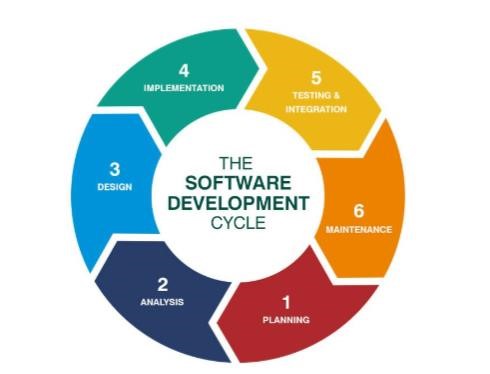
SDLC has defined its phases as, Requirement gathering, Designing, Coding, Testing, and Maintenance. It is important to adhere to the phases to provide the Product in a systematic manner.

**For Example,** A software has to be developed and a team is divided to work on a feature of the product and is allowed to work as they want. One of the developers decides to design first whereas the other decides to code first and the other on the documentation part.

This will lead to project failure because of which it is necessary to have a good knowledge and understanding among the team members to deliver an expected product.

### SDLC Cycle

SDLC Cycle represents the process of developing software.



### SDLC Phases

**Given below are the various phases:**

* Requirement gathering and analysis
* Design
* Implementation or coding
* Testing
* Deployment
* Maintenance

**#1) Requirement Gathering and Analysis**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

**For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

**#2) Design**

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

**#3) Implementation or Coding**

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

**#4) Testing**

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

**#5) Deployment**

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

**#6) Maintenance**

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers

|  |
| --- |
| Day 2,3 |

During these days revise the stage 1 courses

* UI & Scripting Technology - HTML5, CSS3, JavaScript
* Programming & Database Technology - ANSI SQL Using SQL Server, C#

**Stage 2 – Advanced Programming**

##### **Overview**

**Stage 2** deals with application frameworks and tools that are seminal while developing/maintaining various software applications. We provide unique learning experience to learners by including diversified learning content and learning methodologies that are based on adult learning principles.

As part of Stage 2 of your training, the following skills will be covered.

* Design Principles and Patterns
* Entity Framework
* ASP.NET MVC
* NUnit & Moq
* Azure DevOps, Jira
* Sitecore

**How and From Where to Learn?**

* Udemy learnings are recommended in the Platform to understand the fundamental concepts. In addition to this, you can also learn from any other sources as they are mentioned in this handbook.

**Course 1 - Design Principles and Patterns**

##### **Overview**

Milestone 1 will be focusing on **Design Principles and Patterns** and its practical implementations in various application development and maintenance scenarios.

**Design Principles** provide high level guidelines to design better software applications. They do not provide implementation guidelines and are not bound to any programming language. The SOLID (SRP, OCP, LSP, ISP, DIP) principles are one of the most popular sets of design principles.

**Design Pattern** provides low-level solutions related to implementation, of commonly occurring objectoriented problems. In other words, design pattern suggests a specific implementation for the specific object-oriented programming problem.

Design patterns are tested by others and are safe to follow, e.g. Gang of Four patterns: Abstract Factory, Factory, Singleton, Command, etc.

##### **Performance Outcomes**

After completing this milestone, GenCs will be able to

* Should be able to use SOLID principles in Object-oriented software development
* Should be able to understand the concept of pattern-based analysis and design
* Should be able to understand the importance of design patterns in software development
* Should be able to explain various design patterns that are common in software applications
* Should be able to refactor existing designs to use design patterns

|  |
| --- |
| **Do You Know?** |
| **Uses of Design Patterns**    Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.  Often, people only understand how to apply certain software design techniques to certain problems. These techniques are difficult to apply to a broader range of problems. Design  patterns provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.  In addition, patterns allow developers to communicate using well-known, well understood names for software interactions. Common design patterns can be improved over time, making them more robust than ad-hoc designs. |

|  |
| --- |
| Day 4,5 |

Different Types of Software Design Principles, GoF Design Patterns - Creational Design Patterns, Structural Design Patterns, Behavioral Design Patterns

##### **Continuous Learning: Technical Enablement**

 [Design Patterns in C# and .NET](https://cognizant.udemy.com/course/design-patterns-csharp-dotnet/)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
  + Section 1: The SOLID Design Principles
* Implement the examples along with the author.

 [Design Microservices Architecture with Patterns & Principles](https://cognizant.udemy.com/course/design-microservices-architecture-with-patterns-principles)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

o Section 2: Monolithic Architecture

o Section 4: Layered (N-Layer) Architecture

o Section 5: Service-Oriented Architecture (SOA)

o Section 6: Microsrevices Architecture

* Implement the examples along with the author.

##### **Hands-On**

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* UA Highschool – Singleton Pattern
* GadgetHub – Factory Pattern

|  |
| --- |
| Day 6 |

 [Design Patterns in C# and .NET](https://cognizant.udemy.com/course/design-patterns-csharp-dotnet/)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. o Section 2: Builder o Section 3: Factories o Section 5: Singleton o Section 6: Adapter o Section 7: Bridge o Section 10: Façade
  + Section 13: Chain of Responsibilities
  + Section 14: Command
  + Section 20: Observer
* Implement the examples along with the author.

##### **Hands-On**

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* SecureDocs – Proxy Pattern
* FoodiesDelight – Command Pattern

**Note:** Download the dependencies from the **Reference** section inside the milestone box from the LP.

**AI Learnings**

This AI course is designed to equip learners with:

[**Fundamentals of Generative AI [101-Basics]**](https://cognizantlearning.sumtotal.host/rcore/c/pillarRedirect?isDeepLink=1&relyingParty=LM&url=https%3A%2F%2FCOGNIZANTLEARNING.sumtotal.host%2Flearning%2Fcore%2Factivitydetails%2FViewActivityDetails%3FUserMode%3D0%26ActivityId%3D2048054%26ClassUnderStruct%3DFalse%26CallerUrl%3D%2Flearning%2Flearner%2FHome%2FGoToPortal%3Fkey%3D0%26SearchCallerURL%3Dhttps%253A%252F%252FCOGNIZANTLEARNING.sumtotal.host%252Fcore%252FsearchRedirect%253FViewType%253DList%2526SearchText%253DELRNG01863%2526startRow%253D0%26SearchCallerID%3D2) **(Activity Code: ELRNG01863)**

* The foundational knowledge and skills required to harness the power of Generative AI.
* The ability to identify opportunities for innovation and implementation of AI within their organizations.
* The skills to drive organizations toward a future of enhanced creativity and competitive advantage using AI techniques.

**Course 2 – ORM Framework**

##### **Overview**

Milestone 2 will be focusing on **Entity Framework** which is an Object Relational Mapping (ORM) framework that gives GenCs an automated way to store and access databases.

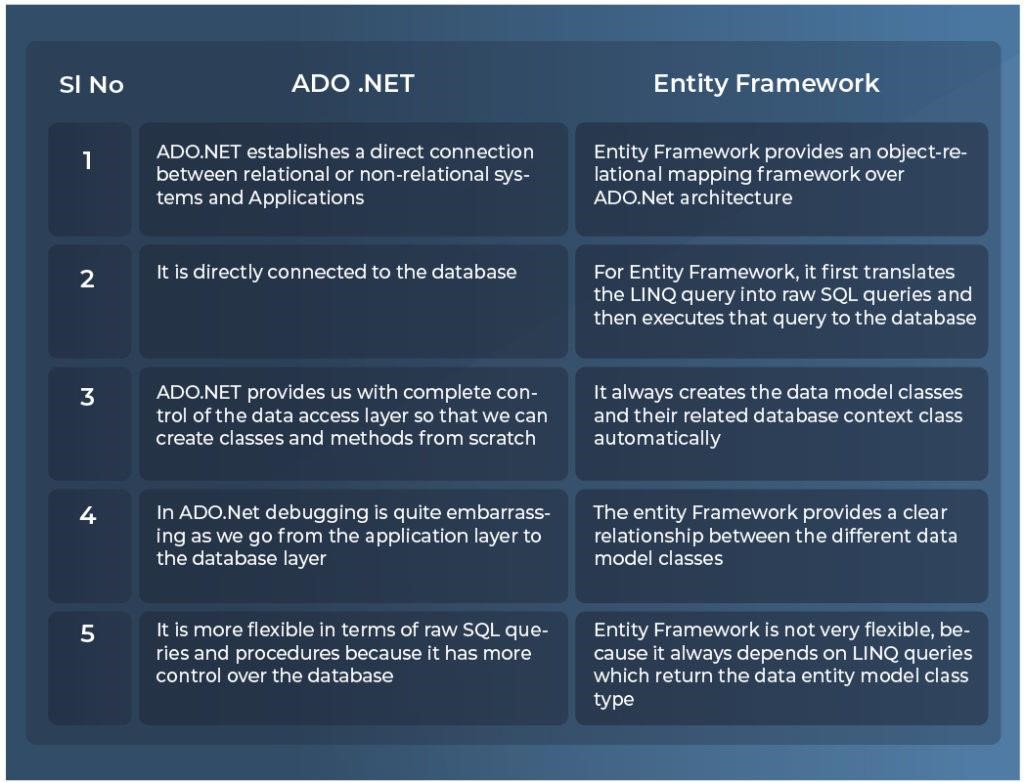
Entity Framework is an open-source object–relational mapping framework for ADO.NET. It was originally shipped as an integral part of .NET Framework. Starting with Entity Framework version 6, it has been delivered separately from the .NET Framework.

##### **Performance Outcomes**

After completing this milestone, GenCs will be able to

* Understand the differences between database-first and code-first workflows
* Build a domain model using database-first workflow
* Build a domain model using code-first workflow (with an existing or a new database)
* Override code-first conventions (using data annotations and fluent API)
* Apply best practices with Entity Framework
* Use code-first migrations to upgrade or downgrade your database
* Query data using LINQ (using query syntax and extension methods)
* Understand what the repository pattern is, why you need it, and how to implement it

##### **ADO.NET vs Entity Framework**



|  |
| --- |
| Day 7 |

**Entity Framework**

Getting Stated, building a Model using Database-First Workflow

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Entity Framework in Depth: The Complete Guide](https://cognizant.udemy.com/course/entity-framework-tutorial)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
  + Section 2: Getting Started
  + Section 3: Building a Model using Database-First Workflow
* Implement the examples along with the author.

##### **Hands-On**

##### 

Try out the example given below in order to understand about Database-First workflow while working with EF6.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Car Repository - Insert

|  |
| --- |
| Day 8 |

**Entity Framework**

Building a Model using Code-First Workflow

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Entity Framework in Depth: The Complete Guide](https://cognizant.udemy.com/course/entity-framework-tutorial)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

o Section 4: Building a Model using Code-First Workflow

* Implement the examples along with the author.

##### **Hands-On**

Complete the following hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Car Repository – Eager Loading

Try out the examples given below in order to understand about Querying database using LINQ while working with EF6.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* [Querying Data using LINQ Examples - Part 1](https://dotnettutorials.net/lesson/querying-in-entity-framework/)
* [Querying Data using LINQ Examples - Part 2](https://dotnettutorials.net/lesson/linq-to-entities-queries-in-entity-framework/)

|  |
| --- |
| Day 9 |

**Entity Framework**

Code First Data Annotations, Fluent API, Querying Data using LINQ, Loading Related Objects, Updating Data, Repository Pattern

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Entity Framework in Depth: The Complete Guide](https://cognizant.udemy.com/course/entity-framework-tutorial)

* + Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
    - * Section 5: Overriding Code-First Conventions
      * Section 6: Querying Data using LINQ
      * Section 7: Loading Related Objects
      * Section 8: Updating Data
      * Section 9: Using Entity Framework in Your Applications

### Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Car Repository – Lazy Loading

Try out the examples given below in order to understand about Querying database using LINQ while working with EF6.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* [Eager Loading Example](https://dotnettutorials.net/lesson/eager-loading-in-entity-framework/)
* [Lazy Loading Example](https://dotnettutorials.net/lesson/lazy-loading-in-entity-framework/)
* [Explicit Loading Example](https://dotnettutorials.net/lesson/explicit-loading-in-entity-framework/)

### Technical Quiz

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on Entity Framework.

• Quiz 1 - Entity Framework

|  |
| --- |
| Day 10 |

**Entity Framework**

Code First Data Annotations, Fluent API, Querying Data using LINQ, Loading Related Objects, Updating Data, Repository Pattern

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Entity Framework in Depth: The Complete Guide](https://cognizant.udemy.com/course/entity-framework-tutorial)

* + Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

### Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on Entity Framework. You have to secure 70% in order to clear this challenge.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Assess-Type-1: Code Challenge - Entity Framework

**Course 3 – Web Application Framework**

#### Overview

Course 3 will be focusing on **ASP.NET MVC5** which is a web framework based on Model-ViewController (MVC) architecture. GenCs can build dynamic web applications using ASP.NET MVC framework that enables a clean separation of concerns, fast development, and TDD friendly.

ASP.NET MVC is a web application framework developed by Microsoft that implements the model– view–controller pattern. It is no longer in active development. It is open-source software, apart from the ASP.NET Web Forms component, which is proprietary

#### Performance Outcomes

After completing this milestone, GenCs will be able to

* Should be able to understand the MVC pattern
* Should be able to model an application in a domain-driven way
* Should be able to understand MVC request processing
* Should be able to define functionality with actions
* Should be able to implement different types of Filters in ASP.NET MVC applications
* Should be able to use Entity Framework object model in order to retrieve and manipulate data from an ASP.NET MVC application
* Should be able to create forms and implement validations
* Should be able to implement Routing in an ASP.NET MVC application
* Should be able to use various Exception Handling techniques in an ASP.NET MVC application
* Should be able to implement Security features in an ASP.NET MVC application
* Should be able to deploy an ASP.NET MVC application

|  |
| --- |
| **Do You Know?** |
| **Features of ASP.NET MVC**   * The framework is based on existing .NET Framework concepts and web standards. * It separates the code of UI from logic through the use of controllers and views. * Supports unit testing through dependency injection, unobtrusive scripting, and action result return types. * Supports AJAX and Cross-browser compatible jQuery libraries. * Supports XML and JSON encoding for clean, RESTful services. * MVC provides routing capability that allows URLs to map to actions in controllers. |

|  |
| --- |
| Day 11 |

**ASP.NET MVC Fundamentals**

Introduction to ASP.NET MVC Fundamentals

#### Continuous Learning: Technical Enablement

 [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Section 1: Getting Started
* Implement the examples along with the author.

#### Hands-On

Try out the example given in the following web page to get start your ASP.NET MVC application development.

* Quiz Board: Controller – ActionMethod
* Quiz Board: Model and View

**Learning Reference**

• [Create Your First MVC App](https://www.tutorialsteacher.com/mvc/create-first-asp.net-mvc-application)

|  |
| --- |
| Day 12 |

**ASP.NET MVC Fundamentals**

Data Management using ViewBag and Model binding, Building Forms & Validations

#### Continuous Learning: Technical Enablement

 [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

o Section 2: ASP.NET MVC Fundamentals

* Implement the examples along with the author.

#### Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | Do not copy paste the code. Write the code yourself. | |
|  |  |

* Quiz Board: ViewBag
* Quiz Board: TempData
* Quiz Board: ViewData

#### Learning Reference

* [Action Methods](https://www.tutorialsteacher.com/mvc/action-method-in-mvc)

|  |
| --- |
| Day 13 |

**ASP.NET MVC Fundamentals**

Data Management using ViewBag and Model binding, Building Forms & Validations

#### Continuous Learning: Technical Enablement

A red letter on a white background

Description automatically generated [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Implement the examples along with the author.

#### Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |
|  |  |

* Quiz Board: Partial View
* Quiz Board: Data Annotation

#### Learning Reference

* [Action Selectors](https://www.tutorialsteacher.com/mvc/action-selectores-in-mvc)
* [ActionVerbs](https://www.tutorialsteacher.com/mvc/actionverbs-in-mvc)
* [Filters](https://www.tutorialsteacher.com/mvc/filters-in-asp.net-mvc)

|  |
| --- |
| Day 14 |

**ASP.NET MVC Fundamentals**

MVC With Database, Razor Pages

#### Continuous Learning: Technical Enablement

 [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Section 3: Working with Data
* Section 4: Building Forms
* Section 5: Implementing Validation
* Implement the examples along with the author.

#### Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Quiz Board: Data Seed
* Quiz Board: List - Razor Page
* Quiz Board: Create - Razor Page

#### Learning Reference

Try out the example found in the following

* [ASP.NET MVC Form](https://www.educba.com/asp-net-mvc-form/?source=leftnav)
* [Exception Handling in ASP.NET MVC](https://www.tutorialsteacher.com/mvc/exception-handling-in-mvc)
* [Security in ASP.NET MVC](https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_security.htm)

|  |
| --- |
| Day 15 |

**ASP.NET MVC Fundamentals**

Deployment

#### 

#### Continuous Learning: Technical Enablement

 [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

o Section 11: Deployment

* Implement the examples along with the author.

#### Learning Reference

Learn everything about Routing in ASP.NET MVC from the following

* [Routing in ASP.NET MVC - Part 1](https://dotnettutorials.net/lesson/asp-dot-net-mvc-routing/)
* [Routing in ASP.NET MVC - Part 2](https://geeksarray.com/blog/explain-asp-net-mvc-routing-with-example)
* [ASP.NET MVC 5 - Deployment On Windows IIS Server](https://www.c-sharpcorner.com/article/asp-net-mvc5-deployment-on-windows-iis-server/)

**AI Knowledge**

[**GENERATIVE AI QUICK ASSESSMENT FOR ELEARNING QUIZ [101-BASICS]**](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcognizantlearning.sumtotal.host%2FCore%2FpillarRedirect%3FrelyingParty%3DLM%26url%3Dcore%252Factivitydetails%252FViewActivityDetails%253FActivityId%253D2068119%2526UserMode%253D0&data=05%7C02%7CKanimozhi.Selvakumar%40cognizant.com%7C70b4a7bce6d94970372b08dc6a8152b9%7Cde08c40719b9427d9fe8edf254300ca7%7C0%7C0%7C638502350875679043%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=8Fs5H%2BZ5bmwxnJFqxagnWcOVvxJ1WtNR1NFuWbJaaSY%3D&reserved=0)

* This assessment is to assess the knowledge of associates on Generative AI tools and concepts at a Beginner proficiency

|  |
| --- |
| Day 16,17 |

**ASP.NET MVC Fundamentals**

Deployment

#### 

#### Continuous Learning: Technical Enablement

A red letter on a white background

Description automatically generated [The Complete ASP.NET MVC 5 Course](https://cognizant.udemy.com/course/the-complete-aspnet-mvc-5-course)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Implement the examples along with the author.

#### Technical Quiz

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on ASP.NET MVC framework.

* Quiz 1 - ASP.Net MVC

#### Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on ASP.NET MVC framework. You have to secure 70% in order to clear this challenge.

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Code Challenge - ASP.Net MVC with Entity Framework

**IDP - Project Activities**

|  |
| --- |
| Day 18-20 |

**IDP – Sprint1 Development & Review**

* These two days will be spent on project development activities and review

**Course 4- Unit Testing Fundamentals**

##### **Overview**

Milestone 4 will be focusing on **Unit Testing and Mocking Frameworks such as NUnit, Moq** and their implementation during Test Driven Development.

**Unit Testing** is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use.

**NUnit** is an open-source unit testing framework for the .NET Framework and Mono. It serves the same purpose as JUnit does in the Java world, and is one of many programs in the xUnit family.

**Moq** is a mocking framework built to facilitate the testing of components with dependencies.

##### **Performance Outcomes**

After completing this milestone, GenCs will be able to

* Understand the fundamental concepts of test-driven development.
* Perform Unit Testing using the NUnit Framework, Moq, and Visual Studio.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Know the Difference** |  | |  | |
| **NUnit 3.x** | **MSTest v2.x.** | **xUnit.net 2.x** |  | **Comments** |
| [Test] | [TestMethod] | [Fact] | | Marks a test method. |
| [TestFixture] | [TestClass] | n/a | | Marks a test class. |
| [SetUp] | [TestInitialize] | Constructor | | Triggered before every test case. |
| [TearDown] | [TestCleanup] | IDisposable.Dispose | | Triggered after every test case. |
| [OneTimeSetUp] | [ClassInitialize] | IClassFixture<T> | | One-time triggered method before test cases start. |
| [OneTimeTearDown] | [ClassCleanup] | IClassFixture<T> | | One-time triggered method after test cases end. |
| [Ignore("reason")] | [Ignore] | [Fact(Skip="reason")] | | Ignores a test case. |
| [Property] | [TestProperty] | [Trait] | | Sets arbitrary metadata on a test. |
| [Theory] | [DataRow] | [Theory] | | Configures a data-driven test. |
| [Category("")] | [TestCategory("")] | [Trait("Category", "")] | | Categorizes the test cases or classes. |

|  |
| --- |
| Day 21 |

**NUnit & Moq**

Getting Started, Fundamentals of Unit Testing, Core Unit Testing Techniques

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Section 1: Getting Started
* Section 2: Fundamentals of Unit Testing
* Section 3: Core Unit Testing Techniques
* Implement the examples along with the author.

##### **Hands-On**

Complete the following set of hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Test Case – Validations On String Concatenation
* Test Case – Validations On Vowel Checker

|  |
| --- |
| Day 22,23 |

**NUnit & Moq**

Moq-Breaking External Dependencies

##### **Continuous Learning: Technical Enablement**

A red letter on a white background

Description automatically generated [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.

o Section 5: Breaking External Dependencies

* Implement the examples along with the author.

##### **Hands-On**

Complete the following hands-on given in the Learning Path at Tekstac.

|  |  |  |
| --- | --- | --- |
| A red and white sign  Description automatically generated | |  | | --- | | Do not copy paste the code. Write the code yourself. | |

* Test Case – Validations On Array Search
* Test Case – Validations On Array Multiplication
* Test Case – Validations On List

**Interim Evaluation**

|  |
| --- |
| Day 24,25 |

##### **Interim Project Evaluation & Interim Technical Evaluation**

• Interim evaluation will be conducted on this day

**Azure Devops, Jira**

|  |
| --- |
| Day 26,27 |

**Continuous Learning: Technical Enablement**

 [Jira for Beginners – Detailed Course to Get Started in Jira](https://cognizant.udemy.com/course/jira-for-beginners-detailed-course-to-get-started-in-jira-online/)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Implement the examples along with the author.

[Azure DevOps Fundamentals for Beginners | Udemy Business](https://cognizant.udemy.com/course/azure-devops-for-beginners/learn/lecture/16451864#overview)



* + Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
  + Implement the examples along with the author

[AWS Essentials.](https://cognizant.udemy.com/course/aws-essentials-z/)

* Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.
* Implement the examples along with the author.

**Sitecore**

|  |
| --- |
| Day 28-39 |

These days there will be Sitecore training. Last week of the training there will be a project case study which will be evaluated by SME.

**IDP - Project Activities**

|  |
| --- |
| Day 40-44 |

### IDP – Development & Review

• These days will be spent on project development activities on SharePoint.

**Final Evaluation**

|  |
| --- |
| Day 45-47 |

**Final Evaluation (Project + Technical)**

* Final evaluation will be conducted on these days

**How to learn each day?**

Each day has a set of learning objectives. These learning objectives can be met by going through the Udemy courses and by completing the hands-on exercises mentioned in the daily plan.

The below strategies will help you decide the learning approach.

**Learning Strategy & Approach**

Find below few imaginary profiles. For each of these profiles we have defined a recommended learning approach. This is not an exhaustive list. The approaches below might help invent a new way of learning.

**Profile #1**

|  |
| --- |
| Harry Reacher  **Engineering Discipline:** Electronics  **Skills:** Python, Ruby on Rails, nginx  **Project:** Mining Crime Data to get Route Cause Insights  **Learning Approach to Programming Languages:** I do not want to waste my time learning. I am more practice oriented. I want to work on the problem immediately  What will work for me?   * Directly complete hands on exercises * Refer Internet or Udemy Courses * If hands on are implemented early, clarify your friends questions and troubleshoot their issues |

**Profile #2**

Olivia Richards

**Engineering Discipline:**

Computer Science

**Skills:**

Java, C, C++

**Project:**

Library Management System

**Learning Approach to Programming Languages:**

I have interest, but I

don’t know where to start.

What will work for me?



|  |  |
| --- | --- |
| • | Go through the recommended Udemy Course |
| • | Try completing the hands on exercises |
| • | Get your clarifications solved with help from Tech SME |
| • | Get help from other learners in your batch whom had already completed |

**Profile #3**

|  |
| --- |
| Greg Anderson  **Engineering Discipline:** Civil  **Skills:** C  **Project:** Fiber reinforced concrete  **Learning Approach to Programming Languages:** I am scared of programming languages. I haven’t got my hands dirty with coding What will work for me?     * Go through the recommended Udemy Course * Implement the coding along with the author of the Udemy Course * Try completing the hands on exercises * Clarify queries with SME * Troubleshoot programming issues with help from SME or learner from your classroom whom had already completed |

**FAQ s**

1. Who can participate in this program?

Students who have enrolled for Full Internship Program (or) the Cognizant on-boarded GEN Cs can participate in this program.

1. Is there any pre-learning I should do?

No. This program is open to all students from any academic discipline.

1. What is Code Challenge?

* 1. problem statement will be provided to you and you need to solve it using a single skill.

1. What is Integrated Capability Test (ICT)?

* 1. case study problem statement will be provided to you, that you may need solve using the combination of Skills learnt in the given stage.

1. Whom do I reach out in case of any queries?

Coach is your point of contact.