



ADM Full Stack Engineering Prep-up

Learning Guide - DotNet



**Why do we need this Full Stack Engineering Prep-up Program?**

Full Stack Prep-up program engages young talents with a comprehensive learning pathway, giving these millennials an opportunity to become a Full Stack Engineer, understand the corporate environment and groom themselves even before they join us.

Cognizant emphasizes on Learner Autonomy where students take charge of their own learning pathway, with the available tools and resources. More focus is given to “learning” than “teaching”. Get ready to embark your own learning adventure!



**Program at a glance**

# Program at a glance

# Program at a glance

# Program at a glance

# Program at a glance

# Program at a glance

# Program at a glance

# Program at a glance

Full Stack Prep-up Internship Program has 4 stages:

* Stage 1 - Core Programming Fundamentals for Full-Stack Web Development
* Stage 2 - Best Practices and Foundations of Backend Development
* Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics
* Stage 4 - Full Stack Development: Front-end Development and Version Control
* Integrated Development Project (IDP)



**Program Highlights**

**Program Highlights**

**Program Highlights**

**Program Highlights**

**Program Highlights**

**Program Highlights**

**Program Highlights**

**Program Highlights**

* 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.
* Learner Autonomy is implemented via Flipped Classroom, where the learning platform offers world class learning resources, and students would not be constrained by tutelage of an instructor.
* Get mentored by Subject Matter Experts, whose motivation and guidance will help you accelerate in the learning journey.
* Higher order framework concepts would be dealt with complete Trainer support in Instructor Led training mode.

x

**Know Your Service Line – ADM (Application Development and Maintenance)**

**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.

**What is Application Development?**

Application development goes through a process of planning, creating, testing, and deploying an information system, also known as the software development lifecycle. Applications are also often developed to automate some type of internal business process or processes, build a product to address common business challenges, or drive innovation.

**What is Application Maintenance?**

Application maintenance is the continuous updating, analyzing, modifying, and re-evaluating of your existing software applications. Application maintenance must be an ongoing task to ensure your applications are always running to the best of their abilities. Due to evolving customer expectations, the fight to survive in an existing market, and technological advancements, modifying and implementing new strategies is critical in maintaining sustainability and staying competitive. Every competitive business needs to constantly enhance and manage the IT solutions that have been developed in order to stay relevant and meet the wavering needs of users. This is where application maintenance and support come into the picture.

Contrary to popular belief, application maintenance is not just about fixing defects, but modifying a software product after delivery to correct faults, as well as to improve performance. Application maintenance and enhancement to existing applications begin with a thorough study of existing applications to identify areas of improvement.

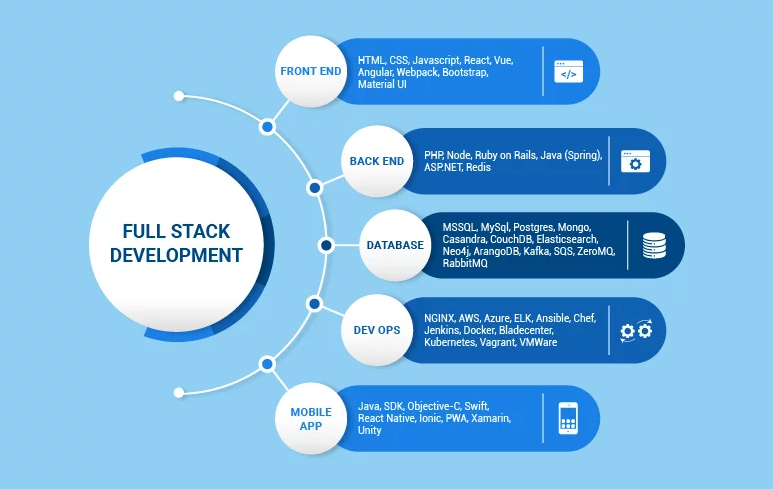
## **Tips for Successfully Carrying Out Application Development and Maintenance**

Great user experience to end customers through the development and maintenance of modern apps is a must-have. Today, applications (web or mobile) are the most cost-effective and powerful ways to reach out to a vast market and generate revenues. With millions of applications being rolled out every day, it’s a good idea to keep in mind a few tips:

* Be as clear as possible as to what your requirements for your application are
* Thoroughly understand the services offered by application development companies and identify the right partner if you’re using a partner
* Evaluate the various development platforms and choose the one that best fits the needs of your business
* Make sure to embed processes that focus on continuous improvements and iterations to add new features and/or fix bugs
* When developing your application, make security your top priority
* Regularly update and test your application to deliver improved and better performance, high security, and a bug-free, seamless user experience

**What is Full Stack Development?**

Full Stack Development (FSD) is a software development process that includes both the front and back end. To that end, a Full Stack Developer may design and create the front end while simultaneously designing, developing, and debugging databases and the software’s backend. There are two significant components to full-stack application development. Development of the Front End and Back End.



**Roles and Responsibilities of a Full Stack Developer**

A full stack developer is responsible for both the front-end and back-end aspects of a web application. The specific roles and responsibilities can vary depending on the size of the development team and the complexity of the project, but some common responsibilities include:

1. Design and develop end-to-end web applications.
2. Implement front-end and back-end components using relevant technologies (e.g. HTML, CSS, JavaScript, React, Node.js, etc.).
3. Write clean, efficient, and well-documented code.
4. Debug and resolve technical issues.
5. Collaborate with the team and other stakeholders to deliver project on time.
6. Stay up-to-date with the latest technologies and industry trends.
7. Write automated tests to ensure code quality and improve application reliability.
8. Develop and maintain databases, servers and application deployment infrastructure.
9. Manage code repositories and version control systems (e.g. Git).
10. Participate in code reviews to ensure high-quality code.
11. Contribute to the architecture and design of applications.
12. Collaborate with designers, product managers, and other stakeholders to understand the requirements and build solutions that meet them.

**Learning Journey thru Flipped Classroom**

The flipped classroom inverts the traditional learning experience. Lectures are shared outside of class time for individual review as homework, and classroom time is reserved for class discussion and interactive projects. The principal goals of flipping are as follows:

* To make the classroom an active learning environment
* To enable students to learn at their own pace, and
* To give the instructor more time to teach each student individually, rather than the class as a whole.

The 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**

Flipped Classroom

Flipped Classroom

Flipped Classroom

Flipped Classroom

Flipped Classroom

Flipped Classroom

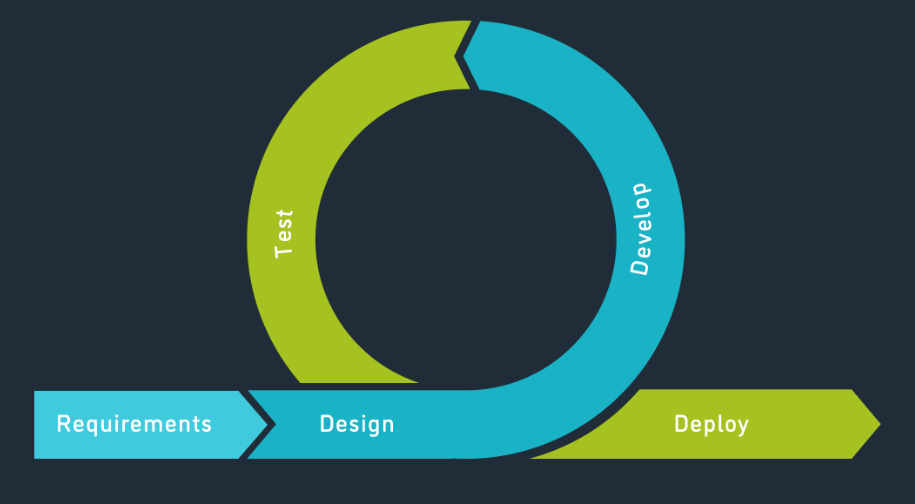
Flipped Classroom

**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.

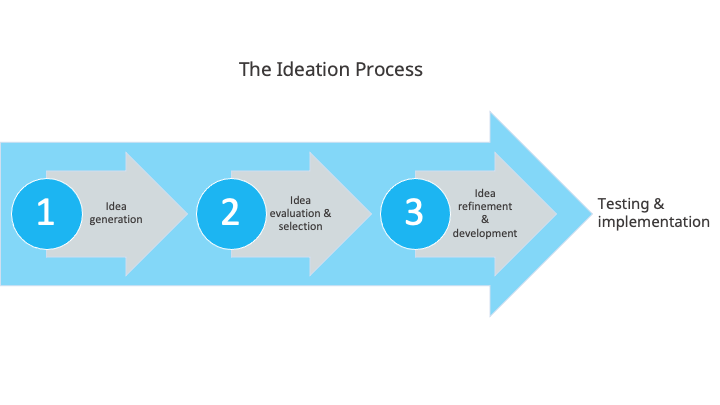


**Stages of IDP**

Following are the four seminal phases of IPD.

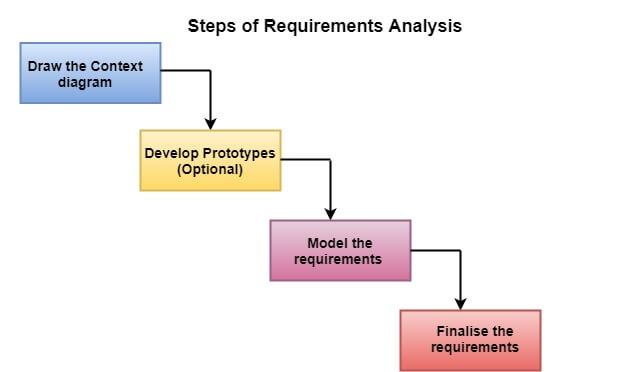
**Phase 1: Ideation**

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.



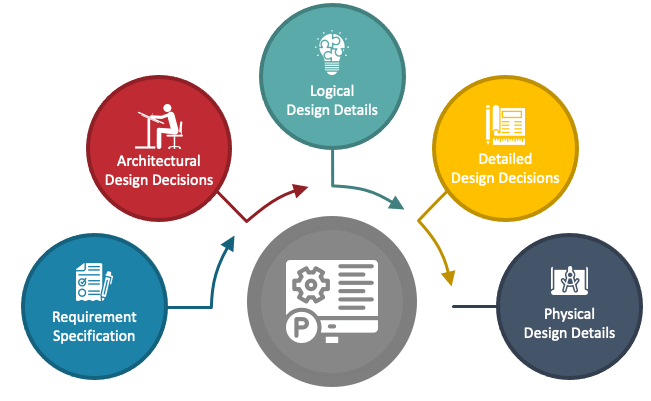
**Phase 2: Requirement Analysis**

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.



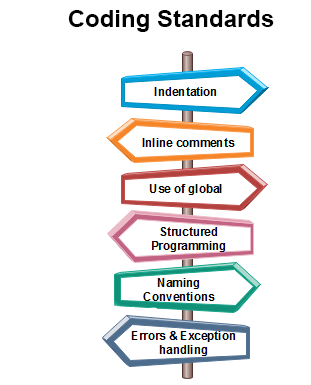
**Phase 3: Project Design**

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 a technical learning that contains **4 stages** and they are the following:

* Stage 1 - Core Programming Fundamentals for Full-Stack Web Development
* Stage 2 - Best Practices and Foundations of Backend Development
* Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics
* Stage 4 - Full Stack Development: Front-end Development and Version Control
* Integrated Development Project (IDP)

**Stage 1 - Core Programming Fundamentals for Full-Stack Web Development**

**Stage 2 - Best Practices and Foundations of Backend Development**

**Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics**

**Stage 4 - Full Stack Development: Front-end Development and Version Control**

All the 4 stages would be executed in the **Flipped classroom model** through Learning paths configured on the **Tekstac** platform.

There will be an integrated project called **IDP** (Integrated Development Project) which will be executed in an incremental fashion and is part of all the 4 stages.

**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.

The program doesn’t ONLY concentrate on the technical skilling, but also on the shaping up of the behavioral skills. **32 hours of Behavioral learning** would be done in ILT mode, with few Self-paced learning modules too.

GenCs will undergo evaluation at the end of Stage 1 of their learning in the form of a skill-based qualifier assessment. At the conclusion of Stage 3, there will be an interim evaluation during which learners will be assessed based on the technical skills covered up to that point and their progress in project deliverables. Towards the end of Stage 4, a final evaluation will take place, which will cover the entire scope of the training.

**Program Completion Criteria**

**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 learning 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.

The above evaluation components will attribute to the **Performance Health Status** **(PHS)** of a GenC. Additional Learning Components like Hands-On, Quizzes, CCs, and ICTs will help you to enhance your expertise level.

**Mandatory Hands-On Exercises Completion**

* Completion of 100% of the hands-on exercises is mandatory to qualify for the stage 1 qualifier, interim, and final evaluations.

**Icebreaker**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

|  |  |
| --- | --- |
| 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… | 5 Icebreaker Activities for a Creative Meeting Introduction - Beekast blog |

**Recommended Day-wise Schedule**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

A recommended day-wise schedule is provided below for the learning, with the learning content for the day, the practice hands-on and extended hands-on to be done for the day or any other activities are listed. Few days might be interleaved to accommodate the extension due to Behavioral Training.

**Stage 1 – Core Programming Fundamentals for Full-Stack Web Development**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

**Overview**

**Stage 1** deals with Core Programming Fundamental skills that help GenCs to get start with their software engineering career. We provide unique learning experience to learners by including diversified learning content and learning methodologies that are based on adult learning principles. At the end of this stage, there will be a **Qualifier Assessment** which determines the direction of the learning journey of a GenC at Cognizant.

As part of Stage 1 of your training, the following skills will be covered.

* Agile Methodology
* HTML5, CSS3 and JavaScript
* Bootstrap 4
* jQuery
* ANSI SQL Using SQL Server
* C#
* ADO.NET

**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.

**Integrated Development Project (IDP) Roadmap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phases** | **Duration** | **Activities** | **Deliverables** |
| Phase1: Ideation/Planning | N/A | 1. Project Ideation 2. Conducting various brainstorming sessions and generate project ideas 3. Finalize the project idea | 1. Project abstract and a title |

**Stage 1: Milestone 1 - Agile Methodology**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

**Overview**

In **Milestone 1 of 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 module, 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.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Agile Methodology | * 8 hrs. | * N/A. |

|  |
| --- |
| **Day 1** |

**Agile Methodology**

**Key Topics:** Agile Basics

**Learn and Practice**

Learn about Agile process from the below Udemy course.

|  |  |
| --- | --- |
| 4FCA95A2 | [Agile Crash Course: Agile Project Management; Agile Delivery](https://cognizant.udemy.com/course/agile-crash-course/learn/lecture/5315576#overview)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **All** Sections * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Stage 1: Milestone 2 - UI and Scripting Technologies**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

**Overview**

**Milestone 2 of Stage 1** focuses on **Web UI technologies, including HTML5, CSS3, JavaScript, Bootstrap4, and jQuery**. These technologies are fundamental to building modern, responsive, and interactive user interfaces for web applications. HTML5 provides the structure for web pages, CSS3 allows for styling and layout customization, and JavaScript adds interactivity and dynamic behavior to the pages. Bootstrap4 is a popular CSS framework that simplifies the process of creating responsive designs, while jQuery is a JavaScript library that simplifies HTML document traversal and manipulation, event handling, and animation. Mastering these technologies is essential for developing engaging and user-friendly web applications.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Define HTML and common terminology related to HTML, recognize correct HTML syntax, and write a brief error-free HTML code.
* Apply style to an existing/new web page as per the requirement using CSS3.
* Write and employ JavaScript code to solve practical web design problems.
* Make responsive, cross-platform and modern websites using Bootstrap4.
* Illustrate animated, interactive web pages using jQuery libraries.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * HTML5, CSS3 and JavaScript | * 12 hrs. | * 8 hrs. |
| * Bootstrap4 | * 8 hrs. | * 8 hrs. |
| * jQuery | * 4 hrs. | - |

|  |
| --- |
| **Day 2** |

**HTML5, CSS3**

**Key Topics:** Introduction, Getting Started, Elements & Attributes, Navigation, Events, Web Forms 2.0, Web Storage, Web SQL Database, Geo location

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Responsive Web Design: HTML5 + CSS3 for Entrepreneurs 2018](https://cognizant.udemy.com/course/html-css-more)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * Lets Learn Some HTML 5 * CSS3 & First Project * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Technical Quiz**

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level in HTML5 and CSS3.

* Quiz 1 - HTML 5 & CSS3

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level in HTML5 and CSS3. You have to secure 70% in order to clear this challenge.

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

* Code Challenge - HTML5 and CSS3

|  |
| --- |
| **Day 3** |

**JavaScript**

**Key Topics:** Getting Started, Fundamentals, Operators, Control Flow, Objects, Arrays

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Javascript basics for beginners](https://cognizant.udemy.com/course/javascript-basics-for-beginners/) |
|  | * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 1:** Getting Started   + **Section 2:** Basics   + **Section 3:** Operators   + **Section 4:** Control flow * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Technical Quiz**

Attempt the following technical quiz in the Learning Path at Tekstac to check your knowledge level in JavaScript.

* Quiz 2 - JavaScript

**Additional Learning**

Go through the Udemy course to understand the usage of Chrome Developer Tools, a comprehensive toolkit for developers built directly into the Chrome browser. These tools allow you to edit web pages in real-time, diagnose problems more quickly, and build better websites faster.

|  |  |
| --- | --- |
| 4FCA95A2 | [Devtools Pro: The Basics of Chrome Developer Tools](https://cognizant.udemy.com/course/devtools-2017-the-basics-of-chrome-developer-tools/) |
|  | * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **All** Sections * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

|  |
| --- |
| **Day 4** |

**JavaScript**

**Key Topics:** JavaScript Deep Dive

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Javascript basics for beginners](https://cognizant.udemy.com/course/javascript-basics-for-beginners/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 5:** Objects * **Section 6:** Arrays * **Section 7:** Functions * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Code Challenge (For Practice Only)**

Attempt the following Code Challenges through the Learning Path at Tekstac for checking your skill level on JavaScript. There will be only 3 attempts and you have to secure 70% in order to clear this challenge.

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

* Code Challenge - JavaScript

|  |
| --- |
| **Day 5** |

**Bootstrap4**

**Key Topics:** Bootstrap Fundamentals, Grid System, NAV Bar

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Bootstrap 4 Bootcamp](https://cognizant.udemy.com/course/bootstrap-4-bootcamp/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 3:** Getting Started With Bootstrap 4   + **Section 4:** Bootstrap 4 Basics   + **Section 5:** Super Useful Utilities   + **Section 7:** Navbars and Flexbox! * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 6** |

**Bootstrap**

**Key Topics:** Bootstrap Components

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Bootstrap 4 Bootcamp](https://cognizant.udemy.com/course/bootstrap-4-bootcamp/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 8:** The Magical Grid System   + **Section 9:** Cards and List Groups   + **Section 10:** Other Components   + **Section 11:** JavaScript Components * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 7 - Forenoon** |

**Bootstrap4**

**Key Topics:** Bootstrap Styles

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Bootstrap 4 Bootcamp](https://cognizant.udemy.com/course/bootstrap-4-bootcamp/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 12:** Customizing Bootstrap w/ SASS * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 7 - Afternoon** |

**jQuery**

**Key Topics:** Introduction to jQuery and its Features, Basic Components of jQuery, DOM Manipulation and Events in jQuery

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete jQuery Course: From Beginner To Advanced!](https://cognizant.udemy.com/course/jquery-tutorial/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 1:** Introduction   + **Section 3:** Element Selectors   + **Section 4:** Manipulating the DOM I – Inserting, Replacing and Removing Elements   + **Section 5:** Manipulating the DOM II – Changing   Element Data and CSS   * + **Section 6:** Events I ‒ Handling Mouse Events & Keyboard Events   + **Section 7:** Events II ‒ Forms * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Stage 1: Milestone 3 - ANSI SQL**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

**Overview**

In Milestone 3 of Stage 1, you will focus on ANSI SQL programming using SQL Server, a fundamental skill for database management. ANSI SQL is a standardized language used to interact with relational databases, and SQL Server is a widely used relational database management system (RDBMS) developed by Microsoft. Throughout this milestone, you will learn essential SQL concepts such as querying data, modifying data, and creating database objects like tables and indexes.

**Learning Objectives**

* Understand and apply ANSI SQL fundamentals in the context of SQL Server.
* Practice retrieving and manipulating data using the SELECT statement in SQL Server.
* Apply filtering, sorting, and grouping techniques to query data effectively.
* Gain an understanding of join operations and subqueries for combining data from multiple tables.
* Develop skills in modifying data with INSERT, UPDATE, and DELETE statements in SQL Server.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * ANSI SQL Using SQL Server | * 8 hrs. | * 8 hrs. |

|  |
| --- |
| **Day 8** |

**ANSI SQL Using SQL Server**

**Key Topics:** Overview of ANSI SQL, Introduction to Database Design, Data Modelling Process, Database Design, Design Considerations, Data Types, Constraints, Querying data, Sorting data, Filtering data

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Relational Database design](https://cognizant.udemy.com/course/relational-database-design)   * Learn ALL SECTIONS in this Udemy course |
| 4FCA95A2 | [70-461, 761: Querying Microsoft SQL Server with Transact-SQL](https://cognizant.udemy.com/course/70-461-session-2-querying-microsoft-sql-server-2012)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + **Section 2: Session 1 -** Starting SQL Server   + **Section 3: Session 1 -** Creating tables - First pass   + **Section 4: Session 1 -** Number types and functions   + **Section 5: Session 1 -** String data types and functions   + **Section 9: Session 2 -** Creating and Querying part of a table * Implement the examples along with the author. |

Go thru this link for SQL server datatypes

* [SQL Server datatypes](https://www.w3schools.com/sql/sql_datatypes.asp)

**Hands-On**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

**Technical Quiz**

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

* Quiz 1 - Database concepts

|  |
| --- |
| **Day 9, Day 10 - Forenoon** |

**ANSI SQL Using SQL Server**

**Key Topics:** Joining tables, Grouping data, Subquery, Set Operators, Modifying data, Expressions

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [70-461, 761: Querying Microsoft SQL Server with Transact-SQL](https://cognizant.udemy.com/course/70-461-session-2-querying-microsoft-sql-server-2012)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + **Section 10: Session 2 -** Summarising and ordering data   + **Section 11: Session 2 -** Adding a second table   + **Section 12:** **Session 2 -** Find missing data, and delete and update data   + **Section 20: Session 4 - Objective 13:** Combine datasets * 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 solution. Write the query yourself. |

**Technical Quiz**

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

* Quiz 2 - ANSI SQL

**Code Challenges (For Practice Only)**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

**Stage 1: Milestone 4 - Application Programming**

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

Icebreaker

**Overview**

**Milestone 4 of Stage 1** focuses on **application programming using C#**. This milestone is designed to introduce learners to the fundamentals of C# programming language and its application in building software applications. Topics covered include basic syntax, data types, control structures, object-oriented programming concepts, and file handling. Learners will also get hands-on experience in developing simple applications using C#, gaining practical skills that are essential for software development in the IT industry.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand basic C# syntax and structure, and write simple programs.
* Use if, else, switch, while, and for statements to control program flow.
* Understand arrays and how to declare, initialize, and access elements.
* Define and call functions or methods.
* Learn about classes, objects, inheritance, and polymorphism.
* Use collections like List and Dictionary, and understand generics.
* Learn Language Integrated Query concepts for querying collections.
* Understand and use anonymous methods in C#.
* Understand and use lambda expressions in C#.
* Define and use extension methods to extend existing types.
* Understand and use structs in C#.
* Use regular expressions for pattern matching and search operations.
* Handle exceptions using try-catch blocks.
* Perform file input/output operations.
* Create and manage threads for concurrent execution.
* Use async and await keywords for asynchronous programming.
* Understand new features and improvements in C# 10 and .NET 6.
* Learn basics of ADO.NET for database access in C#.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * C#, ADO.NET | * 32 hrs. | * 46 hrs. |

|  |
| --- |
| **Day 10 - Afternoon** |

**C#**

**Key Topics:** Fundamentals, Data Types and Variables, Functions/Methods

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete C# Masterclass](https://cognizant.udemy.com/course/complete-csharp-masterclass)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 1:** Your First C# Programm And Overview Of Visual Studio * **Section 2:** DataTypes and Variables * **Section 3:** Functions/Methods And How To Save Time * Implement the examples along with the author. |

Please go thru this link for [Value type vs reference types](https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/data-types/value-types-and-reference-types)

**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. |

**<TBU>**

|  |
| --- |
| **Day 11** |

**C#**

**Key Topics:** Control Flow Statements, OOP Concepts

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete C# Masterclass](https://cognizant.udemy.com/course/complete-csharp-masterclass)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 4:** Making Decisions * **Section 5:** Loops * **Section 6:** Object-Oriented Programming * **Section 9:** Inheritance and More About OOP * **Section 10:** Polymorphism And Even More on OOP + Text Files * 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. |

**<TBU>**

**Technical Quiz**

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on looping concept in C#.

* Quiz - 1 C# - Looping

|  |
| --- |
| **Day 12** |

**C#**

**Key Topics:** Arrays, Collections and Generics

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete C# Masterclass](https://cognizant.udemy.com/course/complete-csharp-masterclass)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 7:** Collections in C# * 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. |

**<TBU>**

|  |
| --- |
| **Day 13** |

**C#**

**Key Topics:** LINQ

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 25:** LINQ * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 14** |

**C#**

**Key Topics:** Anonymous Methods, Lambda Expressions, Extension Methods

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 18:** Extension Methods and Pattern Matching * **Section 20:** Delegates & Events * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

Practice the example given in the following link in order to understand and get start with Anonymous Methods, Lambda Expressions and Extension Methods

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

* [Anonymous Method](https://www.tutorialsteacher.com/csharp/csharp-anonymous-method)
* [Lambda Expression](https://dotnettutorials.net/lesson/lambda-expression-csharp/)
* [Extension Methods](https://www.tutorialsteacher.com/csharp/csharp-extension-method)

**Technical Quiz**

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

**<TBU>**

|  |
| --- |
| **Day 15** |

**C#**

**Key Topics:** Struct, Enum, DateTime, Math Class, Random Class, Regular Expression

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 13:** Partial & Static Classes, Enumerations * **Section 14:** Structures * **Section 26:** String, DateTime, Math * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

Practice the sample programs and code snippets provided in the following links.

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

* [Struct in C#](https://www.c-sharpcorner.com/article/understanding-structures-in-C-Sharp/)
* [Enum in C#](https://www.c-sharpcorner.com/article/understanding-structures-in-C-Sharp/)
* [DateTime in C#](https://www.tutorialsteacher.com/csharp/csharp-datetime)
* [Math Class in C#](https://www.geeksforgeeks.org/c-sharp-math-class/?ref=ml_lbp)
* [Random Class in C#](https://learn.microsoft.com/en-us/dotnet/api/system.random?view=net-6.0)
* [Regular Expression in C#](https://www.tutorialsteacher.com/regex/regex-in-csharp)

|  |
| --- |
| **Day 16** |

**C#**

**Key Topics:** Exception Handling, File I/O

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 28:** Exception Handling * **Section 27:** IO, Serialization, Encoding * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 17** |

**C#**

**Key Concepts:** Multi-Threading

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 31:** Threading * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

**Code Challenge (For Practice Only)**

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

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

**<TBU>**

|  |
| --- |
| **Day 18** |

**C#**

**Key Concepts:** Asynchronous Programming

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 32:** Tasks * **Section 33:** Asynchronous Programming * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 19** |

**C#**

**Key Concepts:** C# 10 (.NET 6)

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [C# 10 | Ultimate Guide - Beginner to Advanced | Master class](https://cognizant.udemy.com/course/c-sharp-oop-ultimate-guide-project-master-class)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 30:** C# 9 and 10 (.NET 6) - New Features * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 20** |

**C#**

**Key Concepts:** ADO.NET Fundamentals

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete C# Masterclass](https://cognizant.udemy.com/course/complete-csharp-masterclass)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 16:** Using Databases with C# * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 21** |

**ADO.NET**

**Learn and Practice**

* Download and go through the PDF document ADO.Net\_Learning.pdf which is accessible via **ADO.NET Learning** under Learning content section. Refer page numbers 1 to 21, 25 to 28

**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. |

**<TBU>**

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on C# programming and ADO.NET. You have to secure 70% in order to clear this challenge.

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

**<TBU>**

**ICT (Integrated Capability Test) (For Practice Only)**

Take up the following extended integrated practice task in order to check your skill level after completing the Stage 1 of your training. Unlike Code Challenge, the coverage of this practice will be C#, SQL Server and ADO.NET. You have to score a minimum 70% in order to complete this activity successfully.

**<TBU>**

**Mock Qualifier**

* Ensure that you take the mock qualifier before the actual one to become familiar with its pattern. You can attend the mock qualifier multiple times.

**Stage 1 Assessment**

|  |
| --- |
| **Day 22 & 23** |

**Stage 1 Assessment**

* These days will be spent on the Qualifier Assessment and Result Publishing.

|  |
| --- |
| **Day 24** |

* This day is reserved for adjusting the behavioral session.

**Stage 2 - Best Practices and Foundations of Backend Development**

**Overview**

**Stage 2** deals with best practices and foundations of backend development. 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 Patterns and Principles
* Data Structures and Algorithms
* SQL Server
* NUnit & Moq
* Entity Framework 6
* Dapper

**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 reliable sources as they are mentioned in this handbook.

**Integrated Development Project (IDP) Roadmap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phases** | **Duration** | **Activities** | **Deliverables** |
| Design | * N/A | * Design and implement the backend database * Identify and implement the data access methodology * Implement a Data Access Library | * E/R Diagram and DB Script * DAL project |

**Stage 2: Milestone 1 - C# Best Practices**

**Overview**

In Milestone 1 of Stage 2, GenCs will learn about Design Patterns and Principles. Design patterns are reusable solutions to common problems that occur in software design. They provide a way to create flexible, maintainable, and scalable software systems. By understanding and applying design patterns, GenCs will be able to write better quality code that is easier to maintain and understand.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Explain the significance of design principles in software development and how they contribute to creating maintainable and scalable software.
* Recognize and describe common design patterns such as Singleton, Factory Method, Abstract Factory, Builder, Adapter, and Observer.
* Apply design principles to real-world scenarios to improve code quality, maintainability, and scalability.
* Implement design patterns in code to solve specific design problems and improve code quality and maintainability.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Design Patterns and Principles | * 8 hrs. | * 8 hrs. |

|  |
| --- |
| **Day 25** |

**Design Principles and Patterns**

**Key Topics:** Solid Principles

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Design Patterns in C# and .NET](https://cognizant.udemy.com/course/design-patterns-csharp-dotnet)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 0:** undefined   + **Section 1:** The SOLID Design Principles * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 26** |

**Design Principles and Patterns**

**Key Topics:** Commonly used Design Patterns

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Boost Your C# With Structural And Creational Design Patterns](https://cognizant.udemy.com/course/boost-your-csharp-code-with-structural-and-creational-design-patterns)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 1:** Introduction   + **Section 2:** Creational Design patterns   + **Section 3:** Structural Design patterns * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

|  |  |
| --- | --- |
| 4FCA95A2 | [Boost Your C# With Behavioural Design Patterns](https://cognizant.udemy.com/course/boost-your-csharp-code-with-behavioural-design-patterns)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 1:** Introduction   + **Section 2:** Behavioral Design patterns * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Stage 2: Milestone 2 - Data Structures and Algorithms**

**Overview**

In Milestone 2 of Stage 2, GenCs will learn about **Data Structures and Algorithms**. This is a fundamental topic in computer science and programming, essential for understanding how to efficiently store, retrieve, and manipulate data.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand the concept of arrays and their implementation in programming, including operations like insertion, deletion, and traversal.
* Learn about linked lists and their advantages, and how to implement and manipulate them, including insertion, deletion, and traversal.
* Understand the concepts of stacks and queues, and how to implement them using arrays or linked lists, including common operations like push, pop, enqueue, and dequeue.
* Learn about tree data structures, their types, and traversal algorithms.
* Understand the concept of graphs, their representations, and traversal algorithms, and explore common graph algorithms.
* Learn about various sorting algorithms (e.g., bubble sort, selection sort, merge sort) and their time complexity, and understand when to use each algorithm based on the data set.
* Understand different searching algorithms (e.g., linear search, binary search) and their time complexity, and know how to apply them to find elements in a data set.
* Learn the concept of dynamic programming and its application to solve complex problems by breaking them down into simpler subproblems.
* Understand the concept of greedy algorithms and how they make locally optimal choices at each step to find a global optimum.
* Learn the divide and conquer strategy, where a problem is divided into smaller subproblems that are solved independently before combining the solutions.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Data Structures and Algorithms | * 8 hrs. | * 8 hrs. |

|  |
| --- |
| **Day 27** |

**Data Structures and Algorithms**

**Key Topics:** Data Structures

**Learn and Practice**

|  |  |  |
| --- | --- | --- |
| 4FCA95A2 | [Algorithms and Data Structures in C#: Complete Tutorial](https://cognizant.udemy.com/course/algorithms-data-structures-csharp/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 2:** Introduction to Algorithms and Data Structures * **Section 4:** Arrays * **Section 6:** Lists * **Section 7:** Stacks * **Section 8:** Queues * **Section 12:** Graphs * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |  |

**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. |

**<TBU>**

Practice the sample programs and code snippets provided in the following links.

* [Graph](https://www.simplilearn.com/tutorials/c-sharp-tutorial/what-is-graphs-in-c-sharp)
* [Trees](https://www.simplilearn.com/tutorials/c-sharp-tutorial/tree-in-c-sharp)

|  |
| --- |
| **Day 28** |

**Data Structures and Algorithms**

**Key Topics:** Algorithms

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Algorithms and Data Structures in C#: Complete Tutorial](https://cognizant.udemy.com/course/algorithms-data-structures-csharp/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 2:** Introduction to Algorithms and Data Structures   + **Section 5:** Sorting Algorithms   + **Section 9:** Search Algorithms * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

Practice the sample programs and code snippets provided in the following links.

* [Dynamic Programming](https://www.geeksforgeeks.org/dynamic-programming/)
* [Greedy Algorithms](https://www.geeksforgeeks.org/greedy-algorithms/)
* [Divide and Conquer Algorithm](https://www.geeksforgeeks.org/introduction-to-divide-and-conquer-algorithm-data-structure-and-algorithm-tutorials/)

**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. |

**<TBU>**

**Stage 2: Milestone 3 - Advanced SQL**

**Overview**

Milestone 3 of Stage 2 focuses on **Advanced SQL Using SQL Server**, covering various key concepts. Participants will start by understanding the basics and gradually delve into advanced concepts. They will learn how to work with views and indexes for efficient data retrieval and management. The module also covers stored procedures and user-defined functions, which are essential for complex data processing tasks. Additionally, participants will explore triggers and cursors for more granular control over database operations. Exception handling and transaction management, crucial for ensuring data integrity and reliability, will also be covered in this milestone.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Learn how to connect to SQL Server and execute basic SQL queries.
* Learn about SQL Server-specific features like window functions and common table expressions.
* Understand the concept of views and how to create and manage them in SQL Server.
* Learn about indexes and their importance in optimizing query performance.
* Learn how to create and use stored procedures in SQL Server for reusable code execution.
* Understand the concept of user-defined functions and how to create and use them in SQL Server.
* Explore the use of triggers for automatically enforcing constraints or executing actions in response to data modifications.
* Understand the concept of cursors and how to use them to process individual rows returned by a query.
* Learn how to handle exceptions and errors in SQL Server using try-catch blocks.

Understand the importance of error handling in maintaining data integrity and application reliability.

* Understand the concept of transactions and their properties (ACID).
* Learn how to manage transactions in SQL Server to ensure data consistency and recoverability.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Advanced SQL Using SQL Server | * 10 hrs. | * 12 hrs. |

|  |
| --- |
| **Day 29** |

**Advanced SQL Using SQL Server**

**Key Topics:** Overview, Advanced Concepts

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [70-461, 761: Querying Microsoft SQL Server with Transact-SQL](https://cognizant.udemy.com/course/70-461-session-2-querying-microsoft-sql-server-2012/learn/lecture/11725694#overview)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 25:** **Session 5 - Objective 9:** Implement aggregate queries * **Section 26:** **Session 5 - Objective 9d:** Ranking functions * **Section 28:** **Session 5 - Objective 9b:** Grouping sets * **Section 33:** **Session 6 - Objective 7e:** WITH statement * **Section 34:** **Session 6 - Objective 7b:** PIVOTing and UNPIVOTing * **Section 35:** **Session 6 - Objective 7d:** CTE statement * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

|  |
| --- |
| **Day 30** |

**Advanced SQL Using SQL Server**

**Key Topics:** Stored Procedures and User-Defined Functions

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [70-461, 761: Querying Microsoft SQL Server with Transact-SQL](https://cognizant.udemy.com/course/70-461-session-2-querying-microsoft-sql-server-2012/learn/lecture/11725694#overview)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 21: Session 4 - Objective 11 -** Create and alter stored procedure * **Section 36: Session 6 - Objective 14:** Functions * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

|  |
| --- |
| **Day 31, 32 - Forenoon** |

**Advanced SQL Using SQL Server**

**Key Topics:** Views and Indexes, Triggers and Cursors, Exception Handling, Transactions

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [70-461, 761: Querying Microsoft SQL Server with Transact-SQL](https://cognizant.udemy.com/course/70-461-session-2-querying-microsoft-sql-server-2012/learn/lecture/11725694#overview)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 22: Session 4 - Objective 18a -** Implement try/catch/throw * **Section 16: Section 3 - Objectives 2 and 3:** Views * **Section 47: Session 7 - Indexes** * **Section 17: Session 3 - Objective 5:** Create and alter DML triggers * **Section 46: Session 7 -** Manage transactions * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

Go through the following links in order to understand more about Cursors.

* [Cursors](https://www.c-sharpcorner.com/article/cursors-in-sql-server/)

**Hands-On**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

**Technical Quiz**

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

**<TBU>**

**Code Challenge (For Practice Only)**

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

|  |  |
| --- | --- |
|  | Do not copy paste the solution. Write the query yourself. |

**<TBU>**

**Stage 2: Milestone 4: Unit Testing Frameworks**

**Overview**

**Milestone 4 of Stage 2** focuses on unit testing frameworks like **NUnit** and mocking frameworks like **Moq**. Participants will start by understanding the importance of unit testing in software development and how it contributes to building robust and maintainable code. They will then delve into NUnit, a popular unit testing framework for .NET applications, learning how to write and execute unit tests for their code. Additionally, participants will explore Moq, a mocking framework that helps in creating mock objects for testing. They will learn how to use Moq to isolate components for unit testing and simulate interactions between objects. Overall, this milestone aims to equip participants with the skills and knowledge needed to effectively test their code using NUnit and Moq.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand the concept and benefits of automated testing.
* Learn how to set up and use NUnit in Visual Studio for writing and executing unit tests.
* Understand the principles of TDD and how to apply them using NUnit.
* Learn about the characteristics of good unit tests, including readability and maintainability.
* Learn how to set up and tear down test fixtures using NUnit.
* Understand the concepts of dependency injection and mocking using frameworks like Moq.
* Learn best practices for writing trustworthy tests, organizing tests, and achieving code coverage.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * NUnit, Moq | * 5 hrs. | * 7 hrs. |

|  |
| --- |
| **Day 32 - Afternoon** |

**NUnit, Moq**

**Key Topics:** Getting Started, Fundamentals of Unit Testing, Core Unit Testing Techniques

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Getting Started * **Section 2:** Fundamentals of Unit Testing * **Section 3:** Core Unit Testing Techniques * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 33** |

**NUnit, Moq**

**Key Topics:** Breaking External Dependencies

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 5:** Breaking External Dependencies * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

* 05-03-Moq-Handson

**Stage 2: Milestone 5: ORM Framework**

**Overview**

**Milestone 5 of Stage 2** focuses on

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand the concept and benefits of automated testing.
* Learn how to set up and use NUnit in Visual Studio for writing and executing unit tests.
* Understand the principles of TDD and how to apply them using NUnit.
* Learn about the characteristics of good unit tests, including readability and maintainability.
* Learn how to set up and tear down test fixtures using NUnit.
* Understand the concepts of dependency injection and mocking using frameworks like Moq.
* Learn best practices for writing trustworthy tests, organizing tests, and achieving code coverage.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * NUnit, Moq | * 5 hrs. | * 7 hrs. |

|  |
| --- |
| **Day 34** |

**Entity Framework Core 6**

**Key Topics:** Introduction to Entity Framework Core 6, Getting Started with EF Core 6

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Entity Framework Core - A Full Tour](https://cognizant.udemy.com/course/entity-framework-core-a-full-tour)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Introduction * **Section 2:** Environment Setup * **Section 3:** Getting Started with Entity Framework Core * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 35** |

**Entity Framework Core 6**

**Key Topics:** Defining Entities and Relationships, Querying Data with EF Core 6, Working with DbContext

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Entity Framework Core - A Full Tour](https://cognizant.udemy.com/course/entity-framework-core-a-full-tour)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 7:** Interacting With Related Records * **Section 4:** Using Entity Framework Core to Query a Database * **Section 5:** Using Entity Framework Core to Manipulate Data * **Section 6:** Handling Database Changes and Migrations * **Section 7:** Interacting With Related Records * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 36** |

**Entity Framework Core 6**

**Key Topics:** Advanced Topics: - Transactions, Stored procedures, Logging and debugging

**Learn and Practice**

* [Transactions](https://learn.microsoft.com/en-us/ef/core/saving/transactions)
* [RAW SQL Queries](https://www.entityframeworktutorial.net/efcore/raw-sql-queries-in-ef-core.aspx)
* [Stored Procedures](https://www.entityframeworktutorial.net/efcore/working-with-stored-procedure-in-ef-core.aspx)
* [Logging and Debugging](https://learn.microsoft.com/en-us/ef/core/logging-events-diagnostics/simple-logging)

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 37** |

**Dapper**

**Key Topics:** Getting Started, Database Providers with Dapper, Querying Data with Dapper, Executing Non-Query Commands with Dapper

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Dapper - Getting Started](https://cognizant.udemy.com/course/dapper-the-complete-guide)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **All Sections** * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

Learn about database providers from the below.

* [Database Providers with Dapper](https://www.learndapper.com/database-providers)

|  |
| --- |
| **Day 38** |

**Dapper**

**Key Topics:** Using Parameters with Dapper, Using Relationships with Dapper, Bulk Operations with Dapper

**Learn and Practice**

* [Using Parameters with Dapper](https://www.learndapper.com/parameters)
* [Using Relationship with Dapper](https://www.learndapper.com/relationships)
* [Bulk Operations with Dapper](https://www.learndapper.com/bulk-operations)

**Evaluate Yourself!!!**

Now, it’s time to **Refer… Relate… Relish**

You have just finished the Entity Framework learning. How about applying your knowledge in certain **real-time scenarios???**

You need not consider this activity as a mandatory hands-on or your code challenge. Consider this as a Do-it-yourself and complete.

Here is the [link](https://cognizantonline.sharepoint.com/:f:/r/sites/GTP-Solutions/Gencsharepath/Shared%20Documents/Interview_PrepUp/DotNet/Entity%20Framework?csf=1&web=1&e=RlblQz). **PLEASE Go through and keep yourself always updated!!**

**IDP - Demo & Re-work**

|  |
| --- |
| **Day 39** |

**IDP - Internal Demo & Re-work**

* This day will be used for project demo & re-work

|  |  |
| --- | --- |
| At this point, the GenC will demonstrate the project work completed and the BU Mentor/Evaluator will sign off the work on agreeable conditions. | Demo Images – Browse 33,801 Stock Photos, Vectors, and Video | Adobe Stock |

**Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics**

**Overview**

**Stage 3** training is designed to equip participants with advanced skills in modern .NET development and related technologies. The training covers topics such as the latest features in .NET 6 and C# 10, building web applications and APIs with ASP.NET Core 6, implementing microservices architecture, using Docker for containerization, understanding cloud computing and DevOps practices, and mastering application debugging for backend development. Through a combination of theoretical concepts and hands-on exercises, participants will gain a deep understanding of these technologies, enabling them to build scalable, efficient, and maintainable applications in a modern development environment.

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 3** of your training, the following skills will be covered.

* .NET 6 and C# 10 Features
* ASP.NET Core 6
* ASP.NET Core 6 Web API
* Microservices Architecture
* Code Analysis & Logging
* Docker Basics
* Cloud and DevOps Basics
* Application Debugging - Backend

**How and From Where to Learn?**

* You can learn from the sources as they are mentioned in this learning guide.

**Integrated Development Project (IDP) Roadmap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phases** | **Duration** | **Activities** | **Deliverables** |
| * Implementation | * N/A | * Server-side coding & implantation | * Web API Code base |

**Offline Hands On (Additional Practice)**

**We strongly recommend** that you try out the offline hands-on exercises given in the Learning Path, as they will help you to gain a better understanding of the concepts covered in the course and provide you with a more practical and comprehensive learning experience.

**Milestone 1 - .Net Core and Ancillary Topics**

|  |
| --- |
| **Day 40** |

**.NET 6 and C# 10**

**Key Topics:** Introduction to .NET 6, C# 10 Features

**Learn and Practice**

* [Getting Started with .NET 6.0](https://www.c-sharpcorner.com/article/getting-started-with-net-6-0/)
* [C# 10 Features](https://dotnetcrunch.in/csharp10/)

**Offline Hands-On (Additional Practice)**

**Note:** Remember to complete the offline hands-on exercises in addition to the exercises mentioned under the Hands-On section. These exercises will help you improve your skills and gain practical experience. Please note that these hands-on exercises are self-guided practice and will not be evaluated by the trainer.

* Try out the offline hands-on exercises given in the learning path on **C# 10 Features**.

|  |
| --- |
| **Day 41 - Forenoon** |

**Code Quality Standards**

**Key Topics:** Coding Standards, Performance, Code Reviews

**Learn and Practice**

* [C# Coding Standards](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions)
* [Performance](https://dev.to/sineni/performance-in-net-4i0e)
* [Code Reviews](https://www.codeguru.com/csharp/c-sharp-code-quality/)

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **C#** **Code Quality.**

|  |
| --- |
| **Day 41 - Afternoon, 42** |

**ASP.NET Core 6**

**Key Topics:** Introduction to ASP.NET Core MVC and .NET 6, Model-View-Controller Architecture, Razor Pages and Views

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Asp.Net Core 8 (.NET 8) | True Ultimate Guide](https://cognizant.udemy.com/course/asp-net-core-true-ultimate-guide-real-project)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Introduction * **Section 2:** Getting Started * **Section 3:** HTTP * **Section 4:** Middleware * **Section 6:** Controllers & IActionResult * **Section 8:** Razor Views * **Section 9:** Layout Views * **Section 10:** Partial Views * **Section 11:** View Components * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Hands-On**

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

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **ASP.NET Core 6**.

|  |
| --- |
| **Day 43** |

**ASP.NET Core 6**

**Key Topics:** Routing,Data Access with Entity Framework Core

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [ASP.NET Core 6.0 Course - MVC - Blazor - Razor - EF Core](https://cognizant.udemy.com/course/aspnet-6-course)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 5:** Routing * **Section 18:** EntityFrameworkCore * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **ASP.NET Core 6**.

|  |
| --- |
| **Day 44** |

**ASP.NET Core 6**

**Key Topics:** Working with Forms and Validations, Middleware and Filters, Authentication and Authorization

**Learn and Practice**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [ASP.NET Core 6.0 Course - MVC - Blazor - Razor - EF Core](https://cognizant.udemy.com/course/aspnet-6-course)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 7:** Model Binding and Validations * **Section 16:** CRUD Operations * **Section 4:** Middleware * **Section 21:** Filters * **Section 25:** Identity, Authorization, Security * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

|  |
| --- |
| **Day 45 - Forenoon** |

**ASP.NET Core 6**

**Key Topics:** Advanced Topics and Best Practices

**Learn and Practice**

* [Consuming APIs in MVC Applications](https://www.c-sharpcorner.com/article/consume-web-api-by-mvc-client-in-net-core/)
* [Dependency Injection and IoC](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-6.0)
* [Caching in ASP.NET Core](https://learn.microsoft.com/en-us/aspnet/core/performance/caching/overview?view=aspnetcore-6.0)

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on ASP.NET Core. 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 Core

**Milestone 2 - Web API**

**Overview**

This module deals with **ASP.NET Core** which helps in creating**RESTful** services, also known as Web APIs.

|  |
| --- |
| **Day 45 - Afternoon** |

**ASP.NET Core 6 Web API**

**Key Topics:** Introduction to ASP.NET Core 6 Web API, API Creation and Resource Return

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [ASP.NET CORE WEB API | The Complete Guide](https://cognizant.udemy.com/course/the-complete-guide-to-aspnet-web-api)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Before You Get Started * **Section 2:** Introduction to Web API * **Section 3:** Building Your First ASP.NET Core API * **Section 5:** Controller Action Return Types * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **ASP.NET Core 6 Web API**.

|  |
| --- |
| **Day 46** |

**ASP.NET Core 6 Web API**

**Key Topics:** Resource Manipulation and Input Validation

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Build ASP.NET Core Web API - Scratch To Finish (.NET 7 API)](https://cognizant.udemy.com/course/build-rest-apis-with-aspnet-core-web-api-entity-framework)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 2:** Create New ASP.NET Core Web API & Domain Models * **Section 3:** Create New Controller - Regions Controller and CRUD Operations / Action Methods * **Section 4:** Asynchronous Programming, Repository Pattern and Automapper * **Section 5:** Functionality For Walks - Create, Read, Update, Delete For Walks * **Section 6:** Validations in ASP.NET Core Web API * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

|  |
| --- |
| **Day 47** |

**ASP.NET Core 6 Web API**

**Key Topics:** Working with Services and Dependency Injection, Using Entity Framework Core in Controllers

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [ASP.NET CORE WEB API | The Complete Guide](https://cognizant.udemy.com/course/the-complete-guide-to-aspnet-web-api)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 4:** Working with Relational Data using Entity Framework * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

|  |
| --- |
| **Day 48** |

**ASP.NET Core 6 Web API**

**Key Topics:** Search, Filter, and Page Resources

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Build ASP.NET Core Web API - Scratch To Finish (.NET 7 API)](https://cognizant.udemy.com/course/build-rest-apis-with-aspnet-core-web-api-entity-framework)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 7:** Filtering, Sorting, Pagination in ASP.NET Core Web API * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

|  |
| --- |
| **Day 49** |

**ASP.NET Core 6 Web API**

**Key Topics:** Securing Your API, API Versioning, Documenting Your API with Swagger/OpenAPI

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Build ASP.NET Core Web API - Scratch To Finish (.NET 7 API)](https://cognizant.udemy.com/course/build-rest-apis-with-aspnet-core-web-api-entity-framework)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 8:** Securing our ASP.NET Core API - Authentication and Authorization - JWT Tokens * **Section 11:** Versioning in ASP.NET Core Web API * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 8 (.NET 8) | True Ultimate Guide](https://cognizant.udemy.com/course/asp-net-core-true-ultimate-guide-real-project)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 27:** Swagger / Open API * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Milestone 3 - Microservices**

|  |
| --- |
| **Day 50** |

**Microservices Architecture**

**Key Topics:** Introduction to Microservices Architecture

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Creating .Net Core Microservices using Clean Architecture](https://cognizant.udemy.com/course/creating-net-core-microservices-using-clean-architecture)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Introduction * **Section 2:** Developing Catalog Microservice using Clean Architecture * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Additional Learning**

Walkthrough the following resources in order to understand the basics of DDD (Domain-Driven Design) and 12-Factor App methodology.

* [12-factor-app](https://www.redhat.com/architect/12-factor-app)
* [Domain Driven Design](https://medium.com/microtica/the-concept-of-domain-driven-design-explained-3184c0fd7c3f)

**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. |

**<TBU>**

|  |
| --- |
| **Day 51** |

**Microservices Architecture**

**Key Topics:** Microservice Communication and Data Management

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Creating .Net Core Microservices using Clean Architecture](https://cognizant.udemy.com/course/creating-net-core-microservices-using-clean-architecture)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 7:** Async Communication between Rabbit MQ, Basket and Ordering Microservice * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

Practice the sample programs and code snippets provided in the following links.

* [Creating a simple data-driven CRUD microservice](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/data-driven-crud-microservice)

**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. |

**<TBU>**

|  |
| --- |
| **Day 52, 53 - Forenoon** |

**Microservices Architecture**

**Key Topics:** Security, Monitoring, and Deployment

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Creating .Net Core Microservices using Clean Architecture](https://cognizant.udemy.com/course/creating-net-core-microservices-using-clean-architecture)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 9:** Identity Server Introduction * **Section 10:** Securing Catalog and Basket Microservice * **Section 11:** Securing Service to Service Communication * **Section 26:** Deploying Microservices to Kubernetes and AKS * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

Practice the sample programs and code snippets provided in the following links.

* [Make secure .NET Microservices and Web Applications](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/secure-net-microservices-web-applications/)

**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. |

**<TBU>**

**Milestone 4 - Static Code Analysis**

**Overview**

Static code analysis in .NET involves automated examination of source code to identify potential vulnerabilities, bugs, and code quality issues without executing the code. Tools like Roslyn analyzers, ReSharper, and SonarQube scan the codebase for patterns that could lead to runtime errors, security vulnerabilities, or performance issues. This analysis helps improve code quality by identifying and fixing issues related to coding standards, maintainability, readability, security vulnerabilities, and performance bottlenecks.

|  |
| --- |
| **Day 53 - Afternoon, 54** |

**Code Analysis**

**Key Topics:** Overview of Code Analysis, Enabling and Configuring Code Analysis, Code Metrics, Code Analyzers and Diagnostics, Common Code Analysis Rules, Code Analysis in Code Reviews, Introduction to Logging in .NET 6, Configuring Logging in .NET 6, Logging Levels and Categories, Logging in Controllers and Services, Log Filtering and Configuration, Introduction to log4net, Basic Logging with log4net

Walk through the following articles and try out the examples given in them.

**Code Analysis**

* [Overview of .NET source code analysis](https://learn.microsoft.com/en-us/dotnet/fundamentals/code-analysis/overview?tabs=net-6)
* [Configure live code analysis for .NET](https://learn.microsoft.com/en-us/visualstudio/code-quality/configure-live-code-analysis-scope-managed-code?view=vs-2022)
* [Code metrics - Cyclomatic complexity](https://learn.microsoft.com/en-us/visualstudio/code-quality/code-metrics-cyclomatic-complexity?view=vs-2022)
* [Code quality rules](https://learn.microsoft.com/en-us/dotnet/fundamentals/code-analysis/quality-rules/)
* [About .NET source code analysis in Visual Studio with Roslyn analyzers](https://learn.microsoft.com/en-us/visualstudio/code-quality/roslyn-analyzers-overview?view=vs-2022)

**Logging**

* [Logging in .NET Core and ASP.NET Core](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-6.0)
* [How to Implement Log4Net in ASP.NET Core Application](https://aspdotnethelp.com/implement-log4net-in-asp-net-core-application/)

**Milestone 5 - Containerization**

**Overview**

This module introduces learners to Docker, a popular containerization platform used in modern application development. Docker allows developers to package applications and their dependencies into containers, which can run consistently across different environments. The module covers key concepts such as Docker containers, images, volumes, and networking. Learners will understand the benefits of Docker, including improved portability, scalability, and efficiency in resource utilization. They will also learn how to create, manage, and deploy Docker containers, making it easier to build, ship, and run applications in a consistent and isolated environment.

|  |
| --- |
| **Day 55** |

**Docker Basics**

**Key Topics:** Introduction, Docker Commands. Docker Run, Docker Images, Docker Compose

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Docker for the Absolute Beginner - Hands On - DevOps](https://cognizant.udemy.com/course/learn-docker)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Introduction * **Section 2:** Docker Commands * **Section 3:** Docker Run * **Section 4:** Docker Images * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 56** |

**Docker Basics**

**Key Topics:** Docker Registry, Docker Engine, Docker Storage, Docker Networking, Container Orchestration

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Docker for the Absolute Beginner - Hands On - DevOps](https://cognizant.udemy.com/course/learn-docker)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 6:** Docker Registry * **Section 7:** Docker Engine, Storage and Networking * **Section 9:** Container Orchestration - Docker Swarm & Kubernetes * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Milestone 6 - Cloud and DevOps**

**Overview**

Cloud and DevOps using Azure involve leveraging Microsoft Azure cloud services and DevOps practices to build, deploy, and manage applications efficiently. Azure offers a wide range of cloud services, including computing, storage, databases, networking, and more, enabling organizations to scale their applications as needed.

DevOps practices focus on collaboration between development and operations teams, emphasizing automation, continuous integration/continuous deployment (CI/CD), and monitoring. With Azure DevOps services, teams can automate build and release pipelines, manage code repositories, and track work items, facilitating faster delivery of high-quality software.

|  |
| --- |
| **Day 57** |

**Cloud and DevOps Basics**

**Key Topics:** Introduction to the Cloud, Introduction to Azure, Azure Basic Concepts, Azure Compute

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Microsoft Azure: From Zero to Hero - The Complete Guide](https://cognizant.udemy.com/course/microsoft-azure-from-zero-to-hero-the-complete-guide)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 2:** Introduction to the Cloud * **Section 3:** Introduction to Azure * **Section 4:** First Look at Azure * **Section 5:** Azure Basic Concepts * **Section 7:** Azure Compute * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 58** |

**Cloud and DevOps Basics**

**Key Topics:** Azure Networking, Data in Azure, Messaging in Azure, Azure Active Directory

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Microsoft Azure: From Zero to Hero - The Complete Guide](https://cognizant.udemy.com/course/microsoft-azure-from-zero-to-hero-the-complete-guide)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 8:** Azure Networking * **Section 9:** Data in Azure * **Section 10:** Messaging in Azure * **Section 11:** Identity Management with Azure AD (Entra) * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

|  |
| --- |
| **Day 59** |

**Cloud and DevOps Basics**

**Key Topics:** Introduction to Azure DevOps, Azure Repos, Azure Boards, Azure Pipelines Basics, Advanced Azure Pipelines, Azure Release Pipelines, Azure Test Plans

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Azure Devops Crash Course -Build CI/CD release pipelines](https://cognizant.udemy.com/course/azure-devops-fundamental)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Introduction to Azure Devops * **Section 2:** Deployment Basics/What are CI/CD Pipelines * **Section 3:** Setting up Build CI/CD Pipelines for Projects using Azure * **Section 4:** Setting up Release CI/CD Pipelines for Projects using Azure * **Section 5:** Introduction to Azure Repositories and Release components * **Section 6:** Adding Test Automation Jobs to the Release Pipelines * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Milestone 7 - Application Debugging**

**Overview**

Backend application debugging, especially in ASP.NET Core API and MVC applications, involves identifying and fixing issues in the server-side code of web applications. It includes troubleshooting errors, performance bottlenecks, and unexpected behavior that may occur in the backend logic. Developers use tools like Visual Studio debugger, logging frameworks (e.g., Serilog, NLog), and profiling tools to diagnose and resolve issues efficiently. Debugging techniques such as breakpoints, exception handling, and logging are commonly used to understand the flow of execution and pinpoint the source of problems.

|  |
| --- |
| **Day 60** |

**Application Debugging - Backend**

**Key Topics:** Overview of Web API Debugging, Debugging in Visual Studio, Handling Exceptions, Using Postman for API Testing, Debugging API Calls with Postman, Advanced Debugging Techniques

**Learn and Practice**

Walk through the following articles and try out the examples given in them.

* [Debug ASP.NET Core with the Visual Studio debugger](https://learn.microsoft.com/en-us/visualstudio/debugger/quickstart-debug-aspnet?view=vs-2022)
* [Handle errors in ASP.NET Core web APIs](https://learn.microsoft.com/en-us/aspnet/core/web-api/handle-errors?view=aspnetcore-6.0)
* [How To Use Postman With ASP.NET Core Web API Testing](https://www.c-sharpcorner.com/article/how-to-use-postman-with-asp-net-core-web-api-testing/)
* [Remote Debugging ASP.NET Core Application on a Remote IIS](https://www.c-sharpcorner.com/article/remote-debugging-asp-net-core-application-on-a-remote-iis/)

**Learning References**

Watch the following demo videos provided in the platform for getting a complete picture about how to debug a Web API using Visual Studio

**Note:** The below given cognizant kpoint videos can be accessed even after your Tekstac platform access is revoked.

* [Application Debugging - Deep Dive](https://cognizant.kpoint.com/app/video/gcc-c04f0933-9a0b-4b39-ac87-85ed5c541c27)
* [Application Debugging - How to debug a Web API in Visual Studio using Postman](https://cognizant.kpoint.com/app/video/gcc-8561b0c7-7c64-47e0-84bb-fab11d7ca6dd)

**Interim Evaluation**

|  |
| --- |
| **Day 60 - Afternoon, 61** |

**Interim Evaluation (Project + Technical)**

* Interim evaluation will be conducted on this day, and the mode will be a video interview on the Tekstac platform.

**Stage 4 - Full Stack Development: Front-end Development and Version Control**

**Overview**

**Stage 4** deals with the Niche skills such as Angular or React, GIT which 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 4** of your training, the following skills will be covered.

* GIT
* Angular OR React
* Application Debugging - Front-end

**How and From Where to Learn?**

* You can learn from the sources as they are mentioned in this learning guide.

**Integrated Development Project (IDP) Roadmap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phases** | **Duration** | **Activities** | **Deliverables** |
| * Integration | * 8 hrs. | * Integrating UI with the REST API | * Front-End Application |
| * Evaluation | * 16 hrs. | * Final Project Demo * Evaluation & Re-work (if any) | * Final Project Code base * Sprint Backlog * PPT |

**Milestone 1 - Version Control System**

**Overview**

**Module 1** of the **Stage 4** training focuses on **Git**, a distributed version control system widely used in software development. In this module, learners will explore the fundamentals of Git, including version control concepts, repository management, branching strategies, and collaboration workflows. They will learn how to use Git commands for tracking changes, resolving conflicts, and managing project history. Understanding Git is essential for modern application development as it enables teams to efficiently collaborate on code, track changes, revert to previous versions, and maintain a well-organized codebase.

|  |
| --- |
| **Day 62** |

**GIT**

**Key Topics:** Introduction to Version Control, Understanding Git, Setting Up Git, Basic Git Commands, Branching and Merging, Remote Repositories, Collaborating with Git

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Git & Github Bootcamp](https://cognizant.udemy.com/course/git-and-github-bootcamp)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course   + **Section 2:** Introducing…Git!   + **Section 3:** Installation and Setup   + **Section 4:** The very Basics of Git: Adding & Committing   + **Section 6:** Working With Branches   + **Section 7:** Merging Branches, Oh Boy!   + **Section 8:** Comparing Changes with Git Diff   + **Section 12:** Fetching & Pulling   + **Section 14:** Git Collaboration Workflows * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **GIT**.

**Milestone 2 - Front-End JS Framework/Library**

**Overview**

Milestone 2 of the Stage 4 training focuses on **Angular/React**; two popular front-end JavaScript frameworks used for building modern web applications. In this milestone, learners will delve into advanced concepts and best practices for developing robust and scalable applications using Angular /React. By the end of this module, learners will gain the skills needed to design and develop complex front-end applications with Angular and React, enhancing their proficiency in modern web development.

|  |
| --- |
| **Day 63** |

**Angular**

**Key Topics:** Introduction to Angular

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course   + **Section 1:** Getting started * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**React**

**Key Topics:** Introduction to React

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 1:** Getting Started * **Section 2:** JavaScript Refresher * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 64** |

**Angular**

**Key Topics:** Angular Basics

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 2:** The Basics   + **Section 3:** Course Project - The Basics * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**<TBU>**

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** React Components

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 3:** React Essentials - Components, JSX, Props, State & More * **Section 14:** An Alternative Way Of Building Components: Class-based Components * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 65** |

**Angular**

**Key Topics:** Components and Templates

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 5:** Components & Data Binding Deep Dive   + **Section 6:** Course Project - Components & Data Binding   + **Section 7:** Directives Deep Dive   + **Section 8:** Course Project – Directives   + **Section 17:** Using Pipes to Transform Output * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** State Management

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 4:** React Essentials - Deep Dive * **Section 5:** React Essentials - Practice Project * **Section 8:** Working with Refs and Portals * **Section 9:** Practice Project: Project Management App (with Components, State, Refs & More) * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 66** |

**Angular**

**Key Topics:** Services and Dependency Injection

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course   + **Section 9:** Using Services & Dependency Injection * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** React Hooks

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 11:** Handling Side Effects & Working with the useEffect() Hook * **Section 12:** Practice Project: Building a Quiz App * **Section 16:** Building Custom React Hooks * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 67** |

**Angular**

**Key Topics:** Routing and Navigation

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 11:** Changing Pages with Routing * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**Code Challenge (For Practice Only)**

Take on the following Code Challenge in the Tekstac Learning Path to assess your skill level in Angular. You will have 3 attempts, and you must score 70% or higher to pass this challenge.

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

* Code Challenge: Angular

**React**

**Key Topics:** React Router

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 21:** Building a Multi-Page SPA with React Router * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

**Code Challenge (For Practice Only)**

Take on the following Code Challenge in the Tekstac Learning Path to assess your skill level in React. You will have 3 attempts, and you must score 70% or higher to pass this challenge.

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

* Code Challenge: React

|  |
| --- |
| **Day 68** |

**Angular**

**Key Topics:** Forms and Validations

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 15:** Handling Forms in Angular Apps   + **Section 16:** Course Project - Forms * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** Forms and Controlled Components

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 17:** Working with Forms & User Input * **Section 18:** Practice Project: Building a Food Order App * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 69** |

**Angular**

**Key Topics:** Http and Observables

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular – The complete guide](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 18:** Making Http Requests   + **Section 19:** Course Project - Http   + **Section 13:** Understanding Observables   + **Section 14:** Course Project - Observables * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** HTTP Requests and Data Fetching

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Modern React with Redux [2024 Update]](https://cognizant.udemy.com/course/react-redux)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 5:** Using an API with React * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 70** |

**Angular**

**Key Topics:** Angular Material

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete Angular 11 - Ultimate Guide - with Real World App](https://cognizant.udemy.com/course/complete-angular-indepth-easy)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 20:** Angular Material (New) * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** State Management with Context API

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 10:** React's Context API & useReducer - Advanced State Management * **Section 19:** Diving into Redux (An Alternative To The Context API) * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 71** |

**Angular**

**Key Topics:** Testing in Angular

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular - The Complete Guide (2024 Edition)](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 30:** A Basic Introduction to Unit Testing in Angular Apps * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** Testing in React

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 29:** Testing React Apps (Unit Tests) * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

|  |
| --- |
| **Day 72** |

**Angular**

**Key Topics:** Deployment and Best Practices

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Angular - The Complete Guide (2024 Edition)](https://cognizant.udemy.com/course/the-complete-guide-to-angular-2)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.   + **Section 22:** Angular Modules & Optimizing Angular Apps   + **Section 23:** Deploying an Angular Apps * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **Angular**.

**React**

**Key Topics:** Deployment and Best Practices

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [React - The Complete Guide 2024 (incl. React Router & Redux)](https://cognizant.udemy.com/course/react-the-complete-guide-incl-redux/)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 23:** Deploying React Apps * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**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. |

**Offline Hands-On (Additional Practice)**

* Try out the offline hands-on exercises given in the learning path on **React**.

**Stage 4: Milestone 3 - Application Debugging**

**Overview**

**Module 3** of the **Stage 4** training focuses on front-end application debugging in Angular or React. In this module, learners will gain essential skills in identifying and resolving issues in their front-end applications. They will learn debugging techniques specific to Angular or React, including using developer tools, logging, and debugging libraries. The module will cover common debugging scenarios such as troubleshooting errors, optimizing performance, and understanding application behavior.

|  |
| --- |
| **Day 73** |

**Application Debugging - Front-end**

**Key Topics:** Debugging Basics, Debugging Angular Applications, Debugging React Applications, Advanced Debugging Techniques

**Learn and Practice**

|  |  |
| --- | --- |
| 4FCA95A2 | [Devtools Pro: The Basics of Chrome Developer Tools](https://cognizant.udemy.com/course/devtools-2017-the-basics-of-chrome-developer-tools)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **All** Sections * Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section. |

**Demo Videos**

Watch the following demo videos provided in the platform for getting a complete picture about how to debug an/a Angular/React application using Chrome’s DevTools and VS Code Debugger.

**Note:** The below given Cognizant Kpoint videos can be accessed even after your Tekstac platform access is revoked.

* [Debugging Angular Part 1](https://cognizant.kpoint.com/app/video/gcc-2a51d737-459b-491e-9f2e-fc1497affa81)
* [Debugging Angular Part 2](https://cognizant.kpoint.com/app/video/gcc-36239c0b-e7e6-4d21-81b8-1c3ebc4505e9)

**IDP- Project Activities**

|  |
| --- |
| **Day 74** |

**IDP - Sprint 2 Development (Integration of front-end with backend)**

* These days will be spent on integrating various aspects of the project such as front-end and back-end

**Final Evaluation**

|  |
| --- |
| **Day 75, 76** |

**Final Evaluation (Project + Technical)**

* Final Evaluation will be conducted on these days, and the mode will be a video interview on the Tekstac platform.

**Final Assessment**

|  |
| --- |
| **Day 77** |

**HackerRank Prep & Assessment**

* Learners will have HackerRank assessment on this day.

**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**

|  |  |
| --- | --- |
| man staring at white sky taken at daytime | 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**

|  |  |
| --- | --- |
| woman covering her face with white book | 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**

|  |  |
| --- | --- |
| person holding pile of books near face | 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 |

**FAQs**

FAQs

FAQs

FAQs

FAQs

FAQs

FAQs

FAQs

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?**

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

1. **What is Integrated Capability Test (ICT)?**

A 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. **How many attempts are provided for the Coding challenge and ICTs?**

**Is it open all the time for practice?**

The Coding challenges and ICTs are open and there are 3 attempts to take them up.

1. **What are the entry criteria for qualifier?**

 A 100% hands-on completion and attempt in CC & ICT are the eligibility criteria for qualification.

1. **What skills are covered in the qualifier?**

The skills of Stage 1 are covered in the qualifier. Only ONE attempt is provided to clear with a minimum score of 70%

1. **What if I fail in the Interim evaluation?**

Your coach will notify your performance in the Interim evaluation. However, you can continue with the learning.

1. **How many chances will I get in the Final evaluation?**

You’ll get 2 chances in the Final evaluation which covers ALL the skills in the learning journey.

1. **Will we be provided with Projects to work on?**

No, you will have to ideate, design and develop the project which will be reviewed and assessed by the project mentor.

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

Coach is your point of contact.