

Generation Cognizant (GenC)

ADM Standard DotNet Suite - Learning Guide



**Why do we need this Academy enablement Program?**

Academy enablement program engages young talents with a comprehensive learning pathway, giving these millennials an opportunity to interact with Subject Matter Experts (SME) and understand the corporate environment and groom themselves even before they join us.

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



**Program at a glance**

Learning consisting of 2 Stages and an Integrated Development Project:

* **Stage 1** - Foundation in Software Development
* **Stage 2** - Application Development and Maintenance Practices
* Integrated Development Project (IDP)



**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.
* Get mentored by SMEs from BU, whose motivation and guidance will help you accelerate in the learning journey.
* Learner Autonomy is encouraged via Flipped Classroom, where the learning platform offers world class learning resources, and students would not be constrained by tutelage of an instructor.
* Through Project-Based Learning (PBL), GenCs will go through the entire project phase, embracing Agile practices

**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

**Learning Journey with Flipped Classroom**

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

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

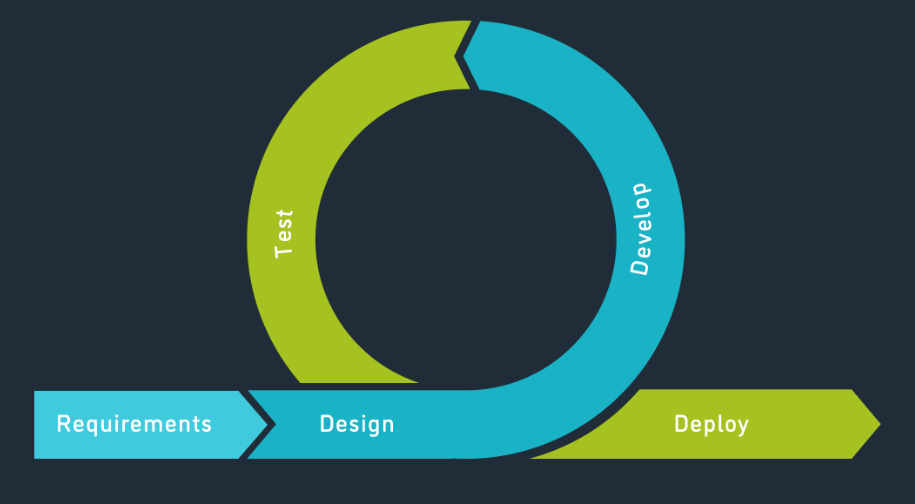
**Flipped Classroom**

**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.

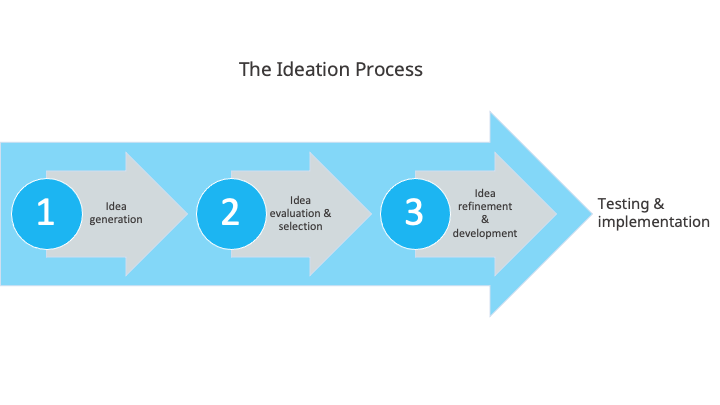


**Stages of IDP**

Following are the four seminal phases of IPD.

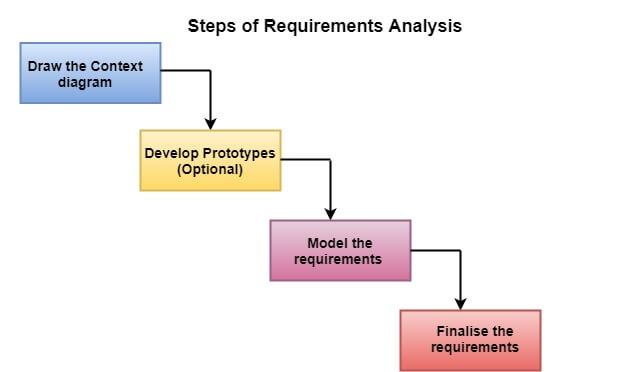
**Phase 1: Ideation**

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.



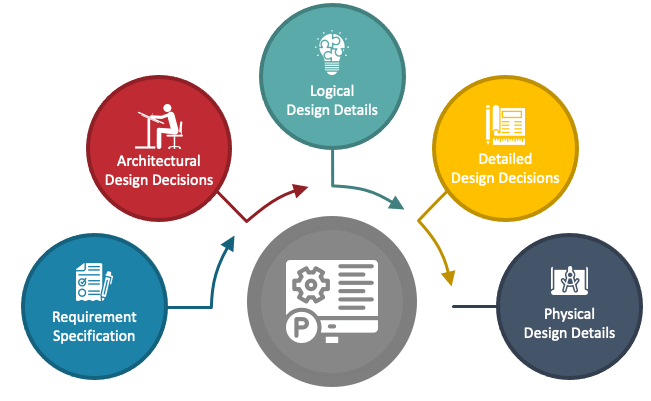
**Phase 2: Requirement Analysis**

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.



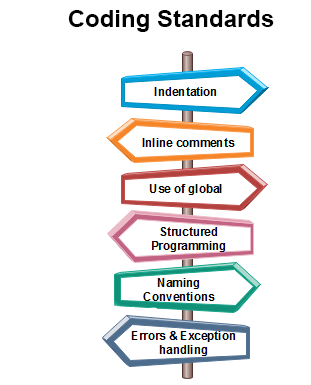
**Phase 3: Project Design**

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 technical learning that contains **2 stages** along with a **Project.**

* Stage 1 - Foundation in Software Development
* Stage 2 - Application Development and Maintenance Practices
* Integrated Development Project (IDP)

**Stage 1 - Foundation in Software Development**

**Stage 2 - Application Development and Maintenance Practices**

**Key Learning and Evaluation Components of the Program**

**Self-Learning via Udemy**

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

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

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

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

**Interim Evaluation:**

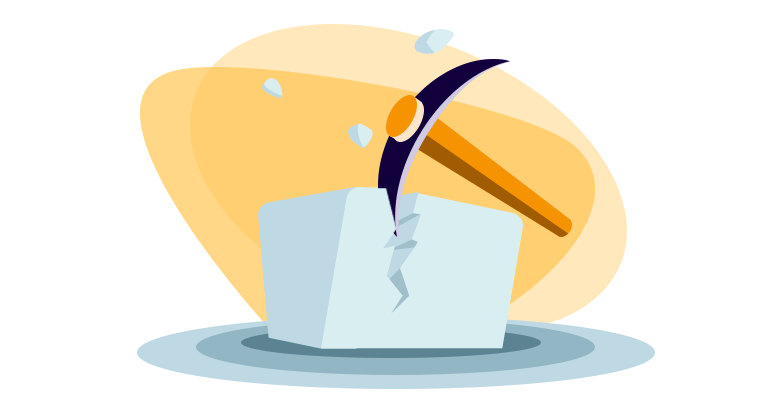
During the interim evaluation, the GenC will undergo a video interview on the learning platform. This interview will be conducted by a Tech SME from the BU. The purpose of this evaluation is to assess the GenC's knowledge and understanding of the skills covered in the training program up to the halfway 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.

**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…

**Stage 1 - Foundation in Software Development**

**Overview**

**Stage 1** deals with foundational technology 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
* Software Support and Maintenance
* Unix Commands & Shell Scripting Basics
* GIT
* HTML5, CSS3 and JavaScript
* Bootstrap
* jQuery
* ANSI SQL using SQL Server
* C#

**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/Brainstorm | 6 Days | 1. Project Ideation by Forming the PoD 2. Conducting various brainstorming sessions and generate project ideas 3. Finalize the project idea | 1. Project abstract and a title |

**Milestone 1 - Software Engineering Basics and Agile**

**Overview**

**Milestone 1** will be focusing on the **Software Support and Maintenance** and **Agile Methodology**.

**Software support and maintenance** are essential for ensuring that software applications remain functional and up-to-date. Support involves assisting users with any issues they encounter while using the software, while maintenance includes activities like fixing bugs, improving performance, and adding new features. Both support and maintenance are critical for keeping software reliable, secure, and aligned with changing user needs.

**Agile** is an approach to software development that focuses on flexibility, collaboration, and delivering value quickly. It involves working in short iterations to continuously improve the software based on user feedback. Agile encourages close collaboration between teams and stakeholders, allowing for quick adaptations to changing requirements. This approach is popular for its ability to deliver software that meets user needs efficiently and effectively.

**Performance Outcomes**

After completing this milestone, GenCs will be able to:

* Demonstrate effective problem resolution and timely bug fixing, ensuring software reliability and user satisfaction.
* Adapt to changing needs and efficiently use resources for continuous software improvement and alignment with user requirements.
* Engage in efficient iterative development and collaborative teamwork, delivering value quickly and adapting to changing project requirements.
* Foster a customer-centric mindset and continuous improvement, aiming to enhance the Agile process and deliver better results based on user feedback.

|  |
| --- |
| **Do you know?** |
| **Software Support and Maintenance:**  **Importance of Software Support:**   * + Software support ensures that users can effectively use the software without encountering issues.   + It provides assistance for troubleshooting problems and resolving technical issues.   **Role of Maintenance in Software Development:**   * + Maintenance activities include bug fixing, performance improvements, and security updates.   + Maintenance ensures that the software remains compatible with evolving technologies and user requirements.   **Best Practices in Support and Maintenance:**   * + Following industry best practices ensures that support and maintenance efforts are efficient and effective.   + Practices include regular monitoring, proactive issue identification, and timely resolution.   **Agile Methodology:**  **Key Principles of Agile:**   * + Agile is guided by principles like customer collaboration, responding to change, and delivering value early.   + It aims to satisfy customers by delivering useful software frequently.   **Agile Frameworks:**   * + Common Agile frameworks include Scrum, Kanban, and Extreme Programming (XP).   + These frameworks have different practices for managing work and teams.   **Benefits of Agile:**   * + Agile can lead to faster delivery, better customer satisfaction, and improved team morale.   + It promotes transparency and the ability to adapt to changes.   **Scaling Agile:**   * + Scaling Agile involves applying Agile principles to larger projects or organizations.   + Frameworks like SAFe and LeSS provide guidance for scaling Agile. |

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

**Agile Methodology**

**Key Topics:** Introduction to Agile, Agile Manifesto, Scrum Framework, Agile Estimation and Planning, Agile User Stories, Agile Metrics and Reporting

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Agile Crash Course: Agile Project Management; Agile Delivery](https://cognizant.udemy.com/course/agile-crash-course)   * Learn All Sections in this Udemy course. |

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

**Software Support and Maintenance**

**Key Topics:** SDLC vs Agile, Software maintenance and support - overview

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| C:\Users\125546\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7FDA4829.tmp | [Software Engineering 101: Plan and Execute Better Software](https://cognizant.udemy.com/course/software-engineering-101)   * Learn the sections listed below in this Udemy course and take up the Quizzes in each section in order to check your understanding about the subject.   + Section 2: Software Lifecycle   + Section 3: Requirements and Specifications   + Section 4: Design: Architecture   + Section 5: Design: Modularity   + Section 6: Implementation and Deployment   + Section 7: Testing   + Section 8: Software Development Models |

**Additional Learning**

Learn about Software Maintenance from the following:

* [Overview of Software Maintenance](https://www.geeksforgeeks.org/software-engineering-software-maintenance/)

**Milestone 2 - UNIX**

**Overview**

**Milestone 2** will be focusing on **Unix Commands & Shell Scripting Basics** and their usage in various application development and maintenance scenarios.

**Unix** is a robust operating system known for its stability and versatility. It provides a command-line interface (CLI) where users interact with the system by entering commands. Unix is designed to be multi-user and multitasking, allowing multiple users to work on the system simultaneously. Its file system is hierarchical, organized into directories and subdirectories, making it easy to navigate and manage files. Unix supports various programming languages, making it a popular choice for software development and system administration tasks.

**Shell scripting** involves writing scripts that can be executed by the Unix shell. These scripts are used to automate repetitive tasks, perform system administration functions, and manipulate files and directories. Shell scripts are written in a scripting language compatible with the Unix shell, such as Bash or Bourne shell. They can include commands, control structures like loops and conditional statements, and input/output redirection. Shell scripting is a powerful tool for system administrators and developers, as it allows them to automate complex tasks and manage Unix systems efficiently.

**Performance Outcomes**

After completing this milestone, GenCs will be able to:

* Navigate the Unix file system and manipulate files using commands like cd, ls, cp, and rm.
* Write basic shell scripts to automate tasks and perform system administration functions.

|  |
| --- |
| **Do you know?** |
| **Unix Commands:**   * The "sed" command (stream editor) is a powerful tool for performing text transformations on input streams or files using regular expressions. * The "tar" command (tape archive) is used to create and manipulate archive files, often used for backup purposes. * Unix commands often follow the convention of short, mnemonic names to make them easier to remember and type, such as "ls" for listing directory contents and "mv" for moving files.   **Shell Scripting:**   * Shell scripts can be used to automate repetitive tasks, such as batch processing of files or system maintenance jobs. * The Unix shell provides powerful features for scripting, including variables, flow control structures (like if-else statements and loops), and command substitution. * Shell scripts can be written in a way that allows them to accept command-line arguments, making them flexible and reusable for different scenarios. |

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

**Unix Commands**

**Key Topics:** Introduction, File Management, Directory Management, File Permission / Access Modes, Basic Utilities, Pipes and Filters, Process Management, Network Communication Utilities

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Linux Command Line Bootcamp: Beginner To Power User](https://cognizant.udemy.com/course/the-linux-command-line-bootcamp)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Introduction * Section 3: Command Basics * Section 4: Getting Help * Section 5: Navigation * Section 6: Creating Files & Folders * Section 8: Deleting, Copying, & Moving * Section 10: Working With Files * Section 11: Redirection * Section 12: Piping * Section 14: Finding Things * Section 15: Grep * Section 16: Permissions Basics * Section 17: Altering Permissions * 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. |

* List of Files 2
* List of Files 3
* Copy File - 6
* Copy File 5
* Copy Directory
* Copy Complete Directory
* Move File 1
* Move File 2
* Find string 7
* Find string 8
* Grep Command - 1
* Grep Command - 2

**Additional Learning**

Learn about Process Management and Network Communication Utilities from the following

* [Linux/Unix Process Management](https://www.guru99.com/managing-processes-in-linux.html)
* [Network Communication Utilities](https://www.tutorialspoint.com/unix/unix-communication.htm)

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

**Shell Scripting Basics**

**Key Topics:** Introduction to Shell Scripting, Basic Shell Scripting Concepts, Control Structures, Command-Line Arguments, Functions, Text Processing, Error Handling and Debugging

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Shell Scripting: Discover How to Automate Command Line Tasks](https://cognizant.udemy.com/course/shell-scripting-linux)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Shell Scripting, Succinctly * Section 5: Shell Script Checklist and Template * Section 7: Case Statements * Section 9: While Loops * Section 3: Exit Statuses and Return Codes * Section 4: Functions * Section 8: Logging * Section 10: Debugging * Section 11: Data Manipulation and Text Transformation with Sed * 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>**

**Milestone 3 - Distributed Version Control System**

**Overview**

**Milestone 3** will be focusing on **GIT** and its usage in various application development and maintenance scenarios.

**Git** is a free and open-source version control system, originally created by Linus Torvalds in 2005. Unlike older centralized version control systems such as SVN and CVS, Git is distributed: every developer has the full history of their code repository locally.

**Performance Outcomes**

After completing this milestone, GenCs will be able to:

* Understand common Git workflows
* Create a new Git project and configure it
* Make and track changes to code by using Git
* Explain and demonstrate basic Git operations like git add, git clone, git checkout, git push, git pull
* Perform Branching, Merging and Rebasing

|  |
| --- |
| **Do you know?** |
| **What is Git?** The majority of popular version-control tools rely on [Git](https://git-scm.com/) – a system for distributed version control. The aim is to help teams track changes and improve collaboration among developers. Git’s creation aim is to facilitate cooperation and provide open-source communities with tools. Files tend to get thicker with time as versions get merged and verified. |

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

**GIT**

**Key Topics:** Introduction, Working Locally with Git, Working Remotely with Git, Branching, Merging, and Rebasing with Git

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Git Complete: The definitive, step-by-step guide to Git](https://cognizant.udemy.com/course/git-complete)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: Introduction * Section 2: Installation * Section 3: Git Quick Start * Section 6: Basic Git Commands * Section 8: Comparisons * Section 9: Branching and Merging * 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. |

* Git-T02-HOL\_001
* Git-T02-HOL\_002
* Git-T03-HOL\_001
* Git-T03-HOL\_002
* Git-T03-HOL\_003
* Git-T03-HOL\_005

**Milestone 4 - UI & Scripting Technologies Duration: 1 Day**

**Overview**

**Milestone 4** will be focusing on UI & Scripting Technologies such as HTML5, CSS3, JS, Bootstrap and jQuery that are essential while developing a UI in various web application development and maintenance scenarios.

**HTML5** is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and final major HTML version that is a World Wide Web Consortium recommendation. The current specification is known as the HTML Living Standard.

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

**JavaScript**, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries.

**Bootstrap** is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

**jQuery** is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of Aug 2022, jQuery is used by 77% of the 10 million most popular websites.

**Performance Outcomes**

After completing this milestone, 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. |
| Bootstrap | 8 hrs. | 8 hrs. |
| jQuery | 8 hrs. | 4 hrs. |

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

**HTML5, CSS3**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Lets Learn Some HTML 5   + CSS3 & First Project * 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. |

* Simple Calculator
* Learning Material Styling

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

**HTML5, CSS3**

**Key Topics:** Introduction, Selectors, Styling, Box Model, Advanced

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + CSS3 & First Project * Implement the examples along with the author. |

Learn about RWD, Media Queries from the following

* [RWD Introduction](https://www.w3schools.com/css/css_rwd_intro.asp)
* [Media Queries](https://www.w3schools.com/css/css_rwd_mediaqueries.asp)
* RWD [Viewport](https://www.w3schools.com/css/css_rwd_viewport.asp)

**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. |

* Feedback Details
* Bill Calculator
* Trainer Feedback Rating Chart

**Additional Learning**

Explore about **Chrome Developer Tools** from the below given Udemy course

|  |  |
| --- | --- |
| 4FCA95A2 | [Devtools Pro: The Basics of Chrome Developer Tools](https://cognizant.udemy.com/course/devtools-2017-the-basics-of-chrome-developer-tools/) |

**Technical Quiz**

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on 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 on HTML5, 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 7 – Afternoon, Day 8, Day 9 - Forenoon** |

**JavaScript**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 1: Getting Started   + Section 2: Basics   + Section 3: Operators   + Section 4: Control Flow   + Section 5: Objects   + Section 6: Arrays   + Section 7: Functions * Implement the examples along with the author. |

**Additional Learning**

Go through **W3Schools** web pages for learning below specific topics

|  |  |
| --- | --- |
| Image result for w3schools logo | [Form Validation](https://www.w3schools.com/js/js_validation.asp)   * JavaScript Form Validation * JavaScript can validate numeric input * Data Validation   [String Methods](https://www.w3schools.com/js/js_string_methods.asp)   * String Length * The substring() Method * String.trim()   [JavaScript HTML DOM](https://www.w3schools.com/js/js_htmldom.asp)   * The HTML DOM (Document Object Model) * What is DOM? * What is the HTML DOM?   [Window alert() Method](https://www.w3schools.com/jsref/met_win_alert.asp)   * Definition and Usage * Example   [Javascript Arrays](https://www.w3schools.com/js/js_arrays.asp)   * All topics except Associative Arrays   [JSON](https://www.w3schools.com/js/js_json.asp)  [Regular Expression](https://www.w3schools.com/js/js_regexp.asp)  [Regular Expression](https://www.w3schools.com/jsref/jsref_regexp_charset_not.asp)  [isNan() function](https://www.w3schools.com/jsref/jsref_isnan.asp)  [indexof function](https://www.w3schools.com/jsref/jsref_indexof.asp) |

Go through web pages for learning below specific topics

* [HTML5 Events](https://www.w3schools.com/tags/ref_eventattributes.asp)
* [HTML5 - Geo location](https://www.w3schools.com/html/html5_geolocation.asp)
* [HTML5 – Geo location](https://www.tutorialspoint.com/html5/html5_geolocation.htm)
* [HTML5 - Web Storage](https://www.w3schools.com/html/html5_webstorage.asp)
* [HTML5-Web SQL Database](https://www.tutorialspoint.com/html5/html5_web_sql.htm)
* [WEB Forms 2.0](https://www.tutorialspoint.com/html5/html5_web_forms2.htm)

Go through **javascript-coder.com** web page for learning form submission

* [JavaScript Form Submit Example](http://javascript-coder.com/files/form-submit/javascript-form-submit-example.html)
* Refer code example in this web page

**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. |

* Placing Order For Cake - String & Math
* Validate Email - Regular Expression & test Function
* Employee Experience Details - Class and Object & Date
* Greetings - DOM
* Fixed And Reducing Interest Loan Estimator
* Word Play - Operators, Conditional Control Statements & Loops
* Find Unique Characters - Functions

**Technical Quiz**

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

* Quiz 2 - Java Script

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac to check your skill level in JavaScript. You need to score 70% or higher to clear this challenge.

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

* Code Challenge – JavaScript

|  |
| --- |
| **Day 9 - Afternoon** |

**Bootstrap 4**

**Key Topics:** Fundamentals, Containers, Grid System

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Bootstrap 4 Bootcamp](https://cognizant.udemy.com/course/bootstrap-4-bootcamp/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 3: Getting Started With Bootstrap 4   + Section 4: Bootstrap 4 Basics   + Section 5: Super Useful Utilities   + Section 6: Forms   + Section 8: The Magical Grid System * 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. |

* Contact US
* Feedback Form

**Additional Learning**

Go through the below Udemy course in order to learn how to use Visual Studio Code as an Editor for UI development.

|  |  |
| --- | --- |
| 4FCA95A2 | [Beginner VS Code](https://cognizant.udemy.com/course/beginner-vs-code/learn/lecture/9969118) |

**Note:** You can use **Visual Studio Code** to practice Bootstrap hands-on on local machine

|  |
| --- |
| **Day 10, Day 11 - Forenoon** |

**Bootstrap 4**

**Key Topics:** NAV Bar, Troubleshooting, Bootstrap Components, Bootstrap Styles

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Bootstrap 4 Bootcamp](https://cognizant.udemy.com/course/bootstrap-4-bootcamp/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 7: Navbars and Flexbox!   + Section 9: Cards and List Groups   + Section 10: Other Components   + Section 11: JavaScript Components * Implement the examples along with the author. |

Go through web pages for learning below specific topics

* [Overriding Bootstrap Styles](https://www.bootstrapdash.com/bootstrap-css-styles/)

**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. |

* Bootstrap’s Navigation Bar
* Page Layout
* Responsive Web Page

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

**jQuery**

**Key Topics:** jQuery and its features, Basic components

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete jQuery Course: From Beginner To Advanced!](https://cognizant.udemy.com/course/jquery-tutorial/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 1: Introduction   + Section 3: Element Selectors * 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. |

* Load jQuery
* Welcome Message

|  |
| --- |
| **Day 12, Day 13 - Forenoon** |

**jQuery**

**Key Topics:** DOM manipulation & events, Basic AJAX with jQuery

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete jQuery Course: From Beginner To Advanced!](https://cognizant.udemy.com/course/jquery-tutorial/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + 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   + Section 8: Ajax with jQuery * Implement the examples along with the author. |

Go through the below mentioned topics on jQuery Ajax

* [Introduction](https://www.w3schools.com/jquery/jquery_ajax_intro.asp)
* [Load()](https://www.w3schools.com/jquery/jquery_ajax_load.asp)
* [Post()](https://www.w3schools.com/jquery/jquery_ajax_get_post.asp)

**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. |

* Three Divisions
* Select the Boxes
* Customer Data
* Vertical Menu
* Get JSON Data
* Error Message
* Login Form
* Alternate Rows - Selectors
* Ice Cream Flavours - Selectors

**Milestone 5 - SQL Duration: 1 Day**

**Overview**

**Milestone 5** will be focusing on ANSI SQL and its querying functionality in various application development and maintenance scenarios.

**ANSI SQL** is a standard way of writing SQL queries that works across different database systems. In SQL Server, ANSI SQL is used for tasks like retrieving data, modifying data, defining database structures, and controlling transactions. It provides a common syntax that makes SQL queries portable across various databases. While SQL Server supports ANSI SQL, it also has its own unique features and optimizations.

**Performance Outcomes**

After completing this milestone, GenCs will be able to:

* Write SQL queries to retrieve specific data from one or multiple tables using SELECT statements.
* Apply various clauses such as WHERE, ORDER BY, and GROUP BY for filtering, sorting, and grouping data effectively.
* Demonstrate proficiency in using joins (INNER JOIN, LEFT JOIN, RIGHT JOIN) and subqueries to retrieve complex data sets.
* Create and manage views to provide a simplified and abstracted view of complex data structures.
* Write stored procedures to encapsulate a set of SQL statements for reuse.
* Create user-defined functions (UDFs) to encapsulate reusable logic that can be used within SQL statements.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * ANSI SQL using SQL Server | * 14 hrs. | * 12 hrs. |

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

**ANSI SQL using SQL Server**

**Key Topics:** Basics, Data Types, Constraints, Querying data, Sorting data, Filtering data

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 4: Session 1 - Number types and functions   + Section 5: Session 1 - String data types and functions   + Section 6: Session 1 - Date data types functions   + Section 9: Session 2 - Creating and querying part of a table   + Section 10: Session 2 - Summarising and ordering data   + Section 15: Session 3 - Objective 4 - Create and modify constraints (simple statements) * 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. |

* Create table with Foreign Key constraint
* Alter Supplier table with Check Constraint

**Additional Learning**

Go through the below topic in order to learn how to filter data using T-SQL.

* [Sort and filter results in T-SQL](https://learn.microsoft.com/en-us/training/modules/sort-filter-queries/)
* [Summarize data with GROUP BY](https://learn.microsoft.com/en-us/training/modules/use-built-functions-transact-sql/6-summarize-group-by)
* [Filter groups with HAVING](https://learn.microsoft.com/en-us/training/modules/use-built-functions-transact-sql/7-filter-groups)

Try out the samples found in the above web pages.

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

**ANSI SQL Using SQL Server**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 11: Session 2 - Adding a second table   + Section 12: Session 2 - Find missing data, and delete and update data   + Section 25: Session 5 - Objective9: Implement aggregate queries   + Section 28: Section 5 - Objective 9b: Grouping sets   + Section 32: Session 6 - Objective 7: Sub-queries   + 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 code. Write the code yourself. |

* Employees working in New York
* Employee with reporting manger
* Worker and Admin Departments

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

**ANSI SQL Using SQL Server**

**Key Topics:** Stored Procedures, User-Defined Functions, Triggers and Cursors, Exception Handling

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 21: Session 4 - Objective 11 - Create and alter stored procedures (simple statements   + Section 22: Session 4 - Objective 18a - Implement try/catch/throw   + Section 36: Session 6 - Objective 14: Functions   + Section 17: Session 3 - Objective 5: Create and alter DML triggers   + Section 50: Session 7 - Evaluate the use of row-based operations vs. set-based operations * Implement the examples along with the author. |

**Hands-On**

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

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

* Employee Count per Department
* Procedure to display the Employees of a specific Department
* Procedure to display all the Departments
* Create a Procedure delete\_status
* Trigger - claims
* Department records using cursors

**Additional Hands-On**

Try out the following additional exercises to learn more about stored procedure, trigger and cursor related scenarios.

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

* Retrieve the Employee Details
* Procedure to retrieve the Player data
* Insert\_records- Triggers-af\_insert
* Update Records -Trigger
* Employee records using cursors

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

**ANSI SQL Using SQL Server**

**Key Topics:** Server Performance & Activity Monitoring, Transactions, Concurrency, Locks in SQL Server

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 46: Session 7 - Manage transactions * Implement the examples along with the author. |

**Additional Learning**

Go through the below web page content in order to learn about Server Performance & Activity Monitoring in SQL Server

* [Server Performance and Activity Monitoring](https://learn.microsoft.com/en-us/sql/relational-databases/performance/server-performance-and-activity-monitoring?view=sql-server-2016)

Learn about database concurrency and locks in SQL Server from the following links

* [Manage database concurrency](https://www.microsoftpressstore.com/articles/article.aspx?p=2756486)
* [Locks in SQL Server](https://www.javatpoint.com/locks-in-sql-server)

**Technical Quiz**

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

* Quiz 1 - SQL Server

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac to check your skill level in SQL Server. You need to score 70% or higher to clear this challenge.

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

* Code Challenge - SQL Server

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

**IDP - High Level Design**

|  |  |
| --- | --- |
| reminder-bot · GitHub Topics · GitHub | **Submit your Design Document and Prototype** |

**Milestone 6: Application Programming**

**Overview**

Milestone 6 will be focusing on **Application Programming using C#** and its practical implementations in various application development and maintenance scenarios.

C# (pronounced "C sharp") is a modern, general-purpose programming language developed by Microsoft as part of its .NET initiative. It is designed for building a wide range of applications, from web and mobile apps to desktop software and enterprise systems. C# is known for its simplicity, type-safety, and strong support for object-oriented programming concepts like classes, inheritance, and interfaces. It also includes features like asynchronous programming, LINQ (Language Integrated Query), and extensive standard libraries, making it a versatile language for various development tasks. C# is widely used in the industry and is often chosen for its balance of power and ease of use.

**Performance Outcomes**

After completing this module, GenCs will be able to:

* Read, write, execute, and debug C# applications
* Understand variables and data types
* Code decision and control structures (if, if/else, switch, while, do/while, for) and use
* primitive data types
* Write user-defined methods
* Write and manipulate arrays
* Write programs using object-oriented programming techniques including classes, objects,
* inheritance, and polymorphism

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * C# | * 40 hrs. | * 34 hrs. |

|  |
| --- |
| **Do You Know?** |
| **Interesting Facts about C#**   * The name of the C Sharp language is stimulated by the musical notation. Here sharp, represent the written note which should be made a semitone higher in pitch. * Microsoft first time uses the name C# in 1988. * The syntax of C# language is similar to the C-style family such as Java, C, C++. * C# language is suitable for writing applications for embedded systems. * C# language is good for developing games. It is also used by Unity (the majority leader in commercial game engines) to develop games. * C# language contains the highest class that supports Generics and Templates. * C# supports internationalization. * C# language is used to developing web pages, android applications, etc. * C# and XAML are the main languages used to develop Windows Store Apps. * C# language has native garbage-collection. |

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

**C#**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 * Section 4: Making Decisions * Section 5: Loops * 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. |

* Find Square and Cube
* Registration Form
* BooleanResult

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

**C#**

**Key Topics:** Arrays, OOP Concept

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 6: Object-Oriented Programming * Section 7: Collections in C# * Topics (93 – 109) * 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. |

**Additional Learning**

Go thru the below tutorial site in order to learn more about OOAD.

* [Object Oriented Analysis & Design](https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm)

**Technical Quiz**

Attempt the following Code Challenge through the Learning Path at Tekstac to check your skill level in Looping concept of C#. You need to score 70% or higher to clear this challenge.

* Quiz - 1 C# - Looping

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

**C#**

**Key Topics:** Collections and Generics

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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. |

**Code Challenge (For Practice Only)**

Attempt the following Code Challenges through the Learning Path at Tekstac for checking your skill level in Collections and OOP with C# programming. You need to score 70% or higher to clear this challenge.

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

* Code Challenge - C# - Group 1

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

**C#**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 18: Linq * Implement the examples along with the author. |

**Additional Learning**

Go thru the below website in order to understand Encapsulation concept in OOP

* [Anonymous Methods](https://dotnettutorials.net/lesson/anonymous-method-c-sharp/#google_vignette)
* [Lambda Expressions](https://dotnettutorials.net/lesson/lambda-expression-csharp/)
* [Extension Methods](https://dotnettutorials.net/lesson/extension-methods-csharp/)

**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. |

* Vehicles Released in Certain Years – Linq

**Additional Hands-On**

Try out the examples found in the following website in order to understand more about LINQ and its usage in C#.

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

* [LINQ Samples](https://linqsamples.com/)

**Technical Quiz**

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

* Quiz - 2 C# - Lambda Expressions

|  |
| --- |
| **Day 22** |

**C#**

**Key Topics:** Strut, Enum, DataTime, Math Class, Random Class, Regular Expression

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 11: Advanced C# Topics * Implement the examples along with the author. |

**Hands-On**

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

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

|  |
| --- |
| **Day 23** |

**C#**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 given in the Learning Path at Tekstac.

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

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

**C#**

**Key Topics:** Multi-Threading,

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 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 given in the Learning Path at Tekstac.

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

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

**C#**

**Key Topics:** Secure Coding, Debugging and Tracing

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete C# Masterclass](https://cognizant.udemy.com/course/complete-csharp-masterclass)   * Learn the Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 8:** Debugging * 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. |

Please find the hyperlink to an article on secure coding and tracing on the internet below and ensure you study its contents thoroughly.

* [Secure Coding in C#](https://www.c-sharpcorner.com/article/secure-coding-in-c-sharp-protecting-applications-and-user-data/)
* [Debugging and Tracing in C#](https://learn.microsoft.com/en-us/troubleshoot/developer/visualstudio/csharp/language-compilers/trace-and-debug)

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

**C#**

**Key Topics:** C# 10 Features

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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. |

|  |
| --- |
| **Day 27, 28 - Forenoon** |

**ADO.NET**

|  |
| --- |
| **About ADO.NET (System software)** |
| **ADO.NET** is a data access technology from the Microsoft .NET Framework that provides communication between relational and non-relational systems through a common set of components. ADO.NET is a set of computer software components that programmers can use to access data and data services from a database.  Introduction to ADO.NET - [Click Here](https://dotnettutorials.net/lesson/what-is-ado-net/#google_vignette)  ADO.NET Code Examples - [Click Here](https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-code-examples) |

**Key Topics:** ADO.NET Fundamentals

**Continuous Learning: Technical Enablement**

* Go through the PDF document **ADO.Net\_Learning.pdf.** 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. |

* ADO.NET\_WorkshopEnrollmentApp\_HandsOn1
* Seminar Ticket Booking

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac to check your skill level in C# programming and ADO.NET. You need to score 70% or higher to clear this challenge.

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

* Code Challenge - C# - Group 2

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

Attempt the Integrated Capability Test (ICT) available on the Learning Path at Tekstac to assess your skill level in Stage 1, particularly focusing on database and application programming skills. You must achieve a score of 70% or higher to pass this practice test.

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

* Dotnet Integrated Capability Test (ICT)

**Stage 1 - Qualifier Assessment**

|  |
| --- |
| **Day 28 - Afternoon, 29, 30** |

**Qualifier Assessment**

* These days will be dedicated for the Qualifier Assessment and Result Publishing

**IDP - Project Activities**

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

**IDP – Project Design**

* These two days will be spent on IDP - Project Design work

**Stage 2 - Application Development and Maintenance Practices**

**Overview**

**Stage 2** focuses on Application Development and Maintenance Practices essential for the development and maintenance of various software applications. We offer a unique learning experience to learners by providing diverse learning content and methodologies based on adult learning principles.

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

* Design Patterns and Principles, UML Basics
* Entity Framework Core 6.0
* Entity Framework
* ASP.NET Core MVC (.NET 6)
* Microservices Basics
* Application Debugging using Visual Studio Debugger
* NUnit & Moq
* ITIL
* Jira, ServiceNow, Azure Boards
* Windows Service
* Python 3
* Cloud Computing Basics

**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 |
| Development | 48 hrs. | Sprint 1 Development & Review | DAO |
| Sprint 2 Development & Review | Completed project’s code base |

**Milestone 1 - Design Patterns and Principles, UML Basics**

**Overview**

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

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

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

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

**UML**, or Unified Modeling Language, is a standardized way of visually representing a system's design in software engineering. It uses diagrams to show different parts of a system, like its structure or how it behaves. These diagrams help understand and communicate about the system's design throughout the software development process. UML includes various types of diagrams, each showing a different aspect of the system.

**Performance Outcomes**

After completing this module, GenCs will be able to:

* Understand and apply the Gang of Four (GoF) Design Patterns to solve common design problems in software development.
* Apply various software design principles, such as SOLID and DRY to enhance the quality and maintainability of software systems.
* Explain the basics of Unified Modeling Language (UML) and its importance in software development.
* Create and interpret different types of UML diagrams, including class diagrams, sequence diagrams, and activity diagrams.
* Identify and utilize the essential elements of UML diagrams, such as classes, objects, relationships, and behaviors, to model software systems effectively.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Design Principles and Patterns, UML Basics | * 16 hrs. | * 8 hrs. |

|  |
| --- |
| **Do You Know?** |
| 1. **Design Patterns**:    * The concept of design patterns was popularized by the book "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, often referred to as the "Gang of Four" (GoF).    * The GoF book identifies 23 classic design patterns, categorized into three groups: creational, structural, and behavioral patterns.    * Design patterns are not specific to a particular programming language but are rather general solutions to common software design problems. 2. **Software Design Principles**:    * SOLID is a set of five design principles introduced by Robert C. Martin (Uncle Bob) which, when followed, can lead to more maintainable and scalable software.    * The SOLID principles are:      + Single Responsibility Principle (SRP)      + Open/Closed Principle (OCP)      + Liskov Substitution Principle (LSP)      + Interface Segregation Principle (ISP)      + Dependency Inversion Principle (DIP)    * These principles aim to make software designs more understandable, flexible, and maintainable over time. 3. **Unified Modeling Language (UML)**:    * UML was developed by Grady Booch, Ivar Jacobson, and James Rumbaugh at Rational Software in the 1990s and has since been standardized by the Object Management Group (OMG).    * UML is not a programming language but a modeling language used to visualize, specify, construct, and document the artifacts of a software system.    * UML diagrams can represent various aspects of a system, including its structure, behavior, and interactions, making it a valuable tool for software developers, architects, and designers. |

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

**Design Patterns and Principles**

**Key Topics:** GoF Design Patterns

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Design Patterns in C# and .NET](https://cognizant.udemy.com/course/design-patterns-csharp-dotnet/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Builder * Section 3: Factories * Section 5: Singleton * Section 6: Adapter * Section 7: Bridge * Section 10: Façade * Section 13: Chain of Responsibilities * Section 14: Command * Section 20: Observer * Implement the examples along with the author. |

|  |  |
| --- | --- |
| 4FCA95A2 | [Design Microservices Architecture with Patterns & Principles](https://cognizant.udemy.com/course/design-microservices-architecture-with-patterns-principles)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Monolithic Architecture * Section 4: Layered (N-Layer) Architecture * Section 5: Service-Oriented Architecture (SOA) * Section 6: Microservices Architecture * 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. |

* 06-01-DP-Handson
* 06-02-DP-Handson
* 06-03-DP-Handson
* 06-04-DP-Handson
* 06-05-DP-Handson
* 06-06-DP-Handson

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

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

**Design Patterns and Principles**

**Key Topics:** Different Types of Software Design Principles

|  |  |
| --- | --- |
| 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 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. |

Please find the hyperlink to an article on DRY principles in C# on the internet below and ensure you study its contents thoroughly.

* [DRY Principles in C#](https://amarozka.dev/dry-principle-with-csharp-examples/)

|  |
| --- |
| **Day 35,** **36 - Forenoon** |

**UML Basics**

**Key Topics:** UML Diagram Types, UML Diagram Elements

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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. * 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. |

Please find the hyperlink to an article on DRY principles in C# on the internet below and ensure you study its contents thoroughly.

* [UML Diagram Elements](https://www.guru99.com/uml-notation-symbol.html)

**IDP – Project Activities**

|  |
| --- |
| **Day 36 - Afternoon to Day 39** |

**IDP - Development & Review**

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

**Interim Evaluation**

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

**Interim Evaluation (Project + Technical)**

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

**Milestone 2 - ORM Framework**

**Overview**

**Milestone 2** will be focusing on **Entity Framework Core 6** which is an Object Relational Mapping (ORM) framework that gives GenCs an automated way to store and access databases.

Entity Framework (EF) Core 6 is a lightweight, extensible, and cross-platform Object-Relational Mapping (ORM) framework for .NET applications. It is an open-source framework maintained by Microsoft and is designed to simplify data access and database interactions in .NET applications. EF Core 6 introduces several new features and improvements over previous versions, including better support for NoSQL databases, improved performance, and enhanced query capabilities. It also provides seamless integration with ASP.NET Core, making it a popular choice for building modern, scalable, and maintainable data-driven applications across different platforms.

**Performance Outcomes**

After completing this milestone, GenCs will be able to:

* Understand the core concepts and architecture of Entity Framework Core 6.0.
* Set up and configure Entity Framework Core for database interaction.
* Perform CRUD operations using DbContext and DbSet.
* Write and execute LINQ queries to retrieve and manipulate data.
* Manage database schema changes, relationships, and optimize performance using EF Core features.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Entity Framework Core 6.0 | * 18 hrs. | * 14 hrs. |

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

**Entity Framework Core 6.0**

**Key Topics:** Introduction to Entity Framework Core 6.0, Setting Up Entity Framework Core, Entity Data Model, DbContext and DbSet, Migrations and Database Schema

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [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 * **Section 6:** Handling Database Changes and Migrations * 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 42** |

**Entity Framework Core 6.0**

**Key Topics:** CRUD Operations, LINQ Queries

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [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 4:** Using Entity Framework Core to Query a Database * **Section 5:** Using Entity Framework Core to Manipulate Data * **Section 8:** Working With Raw SQL, Views and Stored Procedures * 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 43** |

**Entity Framework Core 6.0**

**Key Topics:** Relationships and Navigation Properties, Performance Optimization

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [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 * 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 44** |

**Entity Framework Core 6.0**

**Key Topics:** Transactions and Concurrency, Inheritance and Complex Types, Advanced Topics

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [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 8:** Working With Raw SQL, Views and Stored Procedures * 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. |

Please find the hyperlink to an article on Inheritance and Complex Types on the internet below and ensure you study its contents thoroughly.

* [Inheritance and Complex Types](https://learn.microsoft.com/en-us/ef/core/modeling/inheritance)

**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>**

**Technical Quiz**

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level in Entity Framework Core 6.

**<TBU>**

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level in Entity Framework Core 6. You need to score 70% or higher to clear this challenge.

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

**<TBU>**

**Milestone 4 - Web Application Framework**

**Overview**

**Milestone 4** will be focusing on **ASP.NET Core 6 MVC.**

ASP.NET Core 6 MVC is a web application framework that implements the Model-View-Controller architectural pattern, dividing an application into three main components: Model (data and logic), View (user interface), and Controller (handles user input and application flow). It offers improved performance, cross-platform support, and integration with modern client-side frameworks. ASP.NET Core 6 MVC is designed for building scalable, high-performance web applications with enhanced tooling and development experience, making it a popular choice for creating modern web solutions.

**Learning Objectives**

By the end of this module, you will be able to:

* Create basic MVC applications in ASP.NET Core, including controllers, views, and models, to understand the MVC pattern and its role in web development.
* Define and use routes in ASP.NET Core MVC to handle incoming requests and generate URLs that are user-friendly and SEO-friendly.
* Use HTML Helpers to generate HTML elements with server-side code, reducing manual HTML coding and improving code maintainability.
* Apply data annotations for model validation to ensure data integrity and handle invalid data effectively.
* Manage user state using cookies, sessions, and client-side storage in ASP.NET Core MVC to enhance user experience and application performance.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * ASP.NET Core MVC (.NET 6) | * 28 hrs. | * 28 hrs. |

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

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Introduction to ASP.NET Core, ASP.NET Core Basics

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 * 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 46** |

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** ASP.NET Core MVC Basics

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 6:** Controllers & IActionResult * 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 47** |

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Data Passing Techniques, Routing

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 5:** Routing * **Section 16:** CRUD Operations * 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. |

Please find the hyperlink to an article on Data Passing Techniques on the internet below and ensure you study its contents thoroughly.

* [Data Passing Techniques in ASP.NET Core](https://www.dotnettricks.com/learn/aspnetcore/viewdata-vs-viewbag-vs-tempdata-vs-session-vs-cookies)

**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 48** |

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Views

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 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**

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

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

**<TBU>**

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

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Action Results, HTML Helpers, Tag Helpers

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 6:** Controllers & IActionResult * **Section 17:** Tag Helpers * 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. |

Please find the hyperlink to an article on HTML Tag Helpers on the internet below and ensure you study its contents thoroughly.

* [HTML Helpers in ASP.NET Core MVC](https://dotnettutorials.net/lesson/html-helpers-in-asp-net-core-mvc/)

**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 50** |

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Model Binding, Data Annotation Validation, ASP.NET Core MVC Using EF Core

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 7:** Model Binding and Validations * **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 hands-on given in the Learning Path at Tekstac.

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

**<TBU>**

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

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** Filters,Logging and Error Handling

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 21:** Filters * **Section 20:** Logging and Serilog * **Section 22:** Error Handling * 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 52- Forenoon** |

**ASP.NET Core MVC (.NET 6)**

**Key Topics:** ASP.NET Core security

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Asp.Net Core 7 (.NET 7) | 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 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. |

**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>**

**Milestone 4 - Software Architecture**

**Overview**

Milestone 4 will be focusing on the fundamental aspects of **Microservices** and their core features**.**

Software architecture using microservices is an architectural style where an application is composed of small, independent services that are built around specific business capabilities and communicate with each other through well-defined APIs. Each microservice is designed to be modular, highly maintainable, and independently deployable, allowing for flexibility and scalability in large and complex applications.

**Learning Objectives**

After completing this milestone, GenCs will be able to:

* Understand the fundamental concept of microservices and their advantages over monolithic architectures, including scalability, flexibility, and resilience, through real-world examples.
* Identify and describe the key building blocks of microservices, such as service boundaries, APIs, and communication protocols, and how they contribute to the overall architecture of a microservices-based application.
* Explore the challenges and best practices of data management in a microservices architecture, including database per service, eventual consistency, and the use of polyglot persistence to handle different types of data.
* Understand the principles of deploying and managing microservices in a production environment, including infrastructure as code (IaC), continuous integration and deployment (CI/CD), and deployment patterns like blue-green deployments and canary releases.

|  |  |  |
| --- | --- | --- |
| Courses/Skills | Learning Duration | Practice Duration |
| * Microservices Basics | * 4 hrs. | * N/A |

|  |
| --- |
| **Day 52 - Afternoon** |

**Microservices Basics**

**Key Topics:** Introduction to Microservices, Building Blocks, Data Management, Deployment and Infrastructure

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Microservices Architecture - The Complete Guide](https://cognizant.udemy.com/course/microservices-architecture-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. |

**Milestone 5 - Application Debugging**

**Overview**

Milestone 5 will be focusing on **Application Debugging using Visual Studio Debugger Tool** which can help you navigate through code to inspect the state of an app and show its execution flow.

**Debugging** is the process of detecting and removing of existing and potential errors (also called as ‘bugs’) in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects. When various subsystems or modules are tightly coupled, debugging becomes harder as any change in one module may cause more bugs to appear in another. Sometimes it takes more time to debug a program than to code it.

**Learning Objectives**

After completing this milestone, GenCs will be able to

* Explain what is Debugging and why do we need it
* Use Visual Studio Debugger that helps in navigating through code to inspect the state of an app and show its execution flow.
* Employ various debugging techniques during application development and maintenance.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Application Debugging using Visual Studio Debugger | * 8 hrs. | * 8 hrs. |

|  |
| --- |
| **Do You Know?** |
| **Advantages of Debugging**  **Saves Time:** Performing debugging at the initial stage saves the time of software developers as they can avoid the use of complex codes in software development. It not only saves the time of software developers but also saves their energy.  **Reports Errors**: It gives error report immediately as soon as they occur. This allows the detection of error at an early stage and makes the software development process a stress free.  **Easy Interpretations:** It provides easy interpretations by providing more information about data structures  **Release bug-free software:** By finding errors in software, it allows developers to fix them before releasing them and provides bug-free software to the customers. |

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

**Navigate through code by using the Visual Studio debugger**

**Key Topics:** Introduction to Debugging, Visual Studio Debugger, Navigate through code by using the Visual Studio debugger

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Visual Studio Mastery with C# - Double Your Productivity](https://cognizant.udemy.com/course/visual-studio-mastery-productive-and-fast-coding)   * Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. * **Section 6:** Debugging Tools in Visual Studio * 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 Application Debugging from the following

* [Debugging in Visual Studio](https://www.youtube.com/watch?v=u-HdLtqEOog)
* [Debugging in C# - Finding and Fixing Problems in Your Application](https://www.youtube.com/watch?v=d6IYH8Ro9aI)

Additionally, watch the videos under Demo section in the LP on Tekstac in order to see the Debugging in action.

* Application debugging - Complete application
* Application debugging - Basic application and concepts

**Hands-On**

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

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

* Debugging\_HOL\_001

**Milestone 6 - Unit Testing**

**Overview**

Milestone 6 will be focusing on **Unit Testing and Mocking Frameworks such as NUnit, Moq** and their implementation during Test Driven Development.

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

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

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

**Learning Objectives**

After completing this milestone, GenCs will be able to

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

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * NUnit & Moq | * 8 hrs. | * 8 hrs. |

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

**NUnit & Moq**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: Getting Started * Section 2: Fundamentals of Unit Testing * Section 3: Core Unit Testing Techniques * Implement the examples along with the author. |

**Hands-On**

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

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

* Test Case - Check Equality
* Test Case - Check Object
* Test Case - Conditional Statement
* Test Case - String Compare
* Test Case - Collections
* Purchase Asset

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

**NUnit & Moq**

**Key Topics:** Moq-Breaking External Dependencies

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Unit Testing for C# Developers](https://cognizant.udemy.com/course/unit-testing-csharp)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 5: Breaking External Dependencies * Implement the examples along with the author. |

**Hands-On**

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

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

* 05-03-Moq-Handson

**IDP - Project Activities**

|  |
| --- |
| **Day 57, 58** |

**Sprint 2 Development**

* These two days will be spent on Sprint 2 Development

**Milestone 7 - ITSM Framework**

**Overview**

Milestone 7 will be focusing on **ITIL (Information Technology Infrastructure Library)** which is a widely accepted set of best practices that is designed to support an organization in gaining optimal value from IT by aligning IT services with business strategy.

**ITIL** was originally created by the British government in the 1980s. At the time of its inception, they were looking for a set of standards to improve IT performance. Over the years, ITIL has grown in popularity and evolved as new versions have been released. ITIL is now owned by Axelos, a joint venture between the British Government Cabinet Office and Capita. In 2019, they released the latest version of ITIL, ITIL 4, which takes a more holistic and adaptable approach to ITSM.

**Learning Objectives**

After completing this milestone, GenCs will be able to:

* Define ITIL and its importance in the context of IT service management (ITSM), including its role in improving service delivery, customer satisfaction, and overall business value.
* Describe the purpose and scope of ITIL Service Operation, emphasizing its role in maintaining stability and continuity of services.
* Define key ITIL processes such as Event Management, Incident Management, Problem Management, Request Fulfilment, and Access Management, and their roles in ensuring efficient service delivery and support.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * ITIL | * 6 hrs. | * 1hr |

Also, this milestone will cover **SCM Tools** like **ServiceNow** which is a tool that provides workflows designed for centralized IT service management **and JIRA Service Management** which supports diverse teams and bring development and IT operations teams together on the same platform developers already work in.

**ServiceNow** is a cloud-based software platform for **IT Service Management (ITSM)** which helps to automate IT Business Management. It is designed based on ITIL guidelines to provide service-orientation for tasks, activities, and processes. It uses machine learning to leverage data and workflows to help businesses become faster and scalable.

**JIRA** is a tool developed by Australian Company Atlassian. This software is used for bug tracking, issue tracking, and project management. The JIRA full form is actually inherited from the Japanese word “Gojira” which means “Godzilla”. The basic use of this tool is to track issue and bugs related to your software and Mobile apps. It is also used for project management. The JIRA dashboard consists of many useful functions and features which make handling of issues easy.

**Performance Outcomes**

After completing this milestone, GenCs will be able to

* Should be able to use the ITIL terminology
* Should be able to understand the ITIL processes
* Should be able to explain various functions of ITIL Service Operation
* Should be able to explain various processes under Service Operation
* Should be able to develop procedures for Incident resolving
* Should be able to describe the ServiceNow user interface
* Should be able to explain how to track and handle issues, set up boards and report on them with JIRA

|  |  |  |
| --- | --- | --- |
| Courses/Skills | Learning Duration | Practice Duration |
| * ITIL | * 6 hrs. | * 1hr |
| * JIRA | * 6 hrs. | * N/A |
| * ServiceNow | * 6 hrs. | * N/A |

|  |
| --- |
| **Do You Know?** |
| **ITIL vs ITSM: What’s the difference?**  To get at the difference between ITIL and ITSM, let’s first start by defining ITSM. ITSM, or IT service management, is how IT teams manage the end-to-end delivery of IT services to customers. This includes all the processes and activities to design, create, deliver, and support IT services. ITSM is service-centric; its core concept is the belief that IT should be delivered as a service.  So while ITSM is a kind of methodology for delivering IT to the business, ITIL is a commonly-used set of practices that outlines how to implement ITSM in a business. For those of you familiar with Agile methodologies, the difference between ITSM and ITIL is akin to the one between Agile and Scrum. While ITSM (or Agile) is a methodology, ITIL (or Scrum) is a framework for implementing that methodology.  Obviously, the connection between the two is strong; ITIL was created with ITSM in mind. But the distinction between the two can be boiled down to one idea: ITIL is a framework or a set of guidelines to assist in implementing the activities involved in ITSM. |

|  |
| --- |
| **Do You Know?** |
| **What is JIRA used for?**  Jira software can be used for the following purposes:   * Requirements and Test case management * In Agile Methodology * Project Management * Software Development * Product Management * Task Management * Bug Tracking   **Why use Service Now?**  Here are the prime reasons for using ServiceNow software:   * All stakeholders including employee and customer make changes to the same platform which streamlines operations and provides a single version of the truth * Allows your employee to perform better, and the service levels will eventually improve * Helps to reduce ITSM costs up to 60% * Helps you to replace unstructured work patterns/business processes with intelligent workflows * It offers many ways to get help including forms, questionnaires, chat, email, etc. * Web services and email actions handle events from various monitoring tools and external sources. * ServiceNow technology will help you work very quickly which makes your work process smarter and faster. * Being SaaS, you do not need to worry about configuration, deployment, updates, and maintenance. * You can offer a customer friendly self-service portal with your branding. |

|  |
| --- |
| **Day 54** |

Jira-Basic Concepts, Using Team-Managed Projects, Using Company-Managed Projects, Administration Basics, Introduction to ServiceNow, ServiceNow Fundamentals, Core Applications

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Jira for Beginners - Detailed Course to Get Started in Jira](https://cognizant.udemy.com/course/jira-for-beginners-detailed-course-to-get-started-in-jira-online)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Concepts You Need to Know in Jira * Section 3: Using Team-Managed Projects * Section 4: Using Company-Managed Projects * Section 5: Jira’s Administrative Back End * Implement the examples along with the author. |

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete ServiceNow System Administrator Course](https://cognizant.udemy.com/course/the-complete-servicenow-system-administrator-course)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: ServiceNow Overview * Section 3: Working With Lists & Forms * Section 5: Tables & Fields * Section 6: User Administration * Section 7: Core Applications * Implement the examples along with the author. |

**Milestone 7 - Job Schedulers**

**Overview**

Milestone 7 will be focusing on **Windows Service** which is a core component of the Microsoft Windows operating system and enable the creation and management of long-running processes

Unlike regular software that is launched by the end user and only runs when the user is logged on, Windows Services can start without user intervention and may continue to run long after the user has logged off. The services run in the background and will usually kick in when the machine is booted. Developers can create Services by creating applications that are [installed as a Service](https://msdn.microsoft.com/en-us/library/d56de412(v=vs.110).aspx), an option ideal for use on servers when long-running functionality is needed without interference with other users on the same system.

The services manage a wide variety of functions including network connections, speaker sound, data backup, user credentials and display colors. Windows Services perform a similar function as UNIX daemons.

**Performance Outcomes**

After completing this milestone, GenCs will be able to

* Should be able to explain what is a Windows Service, why we need it and what are the differences between Windows Services and Regular Applications
* Should be able to develop and manage a Windows Service

|  |  |  |
| --- | --- | --- |
| Courses/Skills | Learning Duration | Practice Duration |
| * Windows Service | * 4hrs | * N/A |

|  |
| --- |
| **Do You Know?** |
| **Windows Scheduled Task vs Windows Service**  From time to time you have a need to implement some maintenance operation for your application server. Whether it is just a simple temporary file deleting or something more complicated.  Two most common ways to do this are either using   * Windows Scheduled Task * Windows Service   Both can do most of the work you need, but both also have some advantages and disadvantages and based on your needs you should go with one or another.  The following are things you should take into concern **The frequency of the operation to be executed** If you have to execute some operation not so often, it does not make sense to have piece of memory used most of the time. Let's say you send some emails once a week or daily. This is pretty much a scenario where you should go with scheduled task.  Otherwise, if you are running task very often, scheduled task is not so suitable as process may time more time than schedule pause and at some point **Communication** If your scheduled operations needs to provide some data to other process, you should keep it as a service. Let's say you want to send push notifications to mobile devices. You are not sure about the frequency, which might be even one a day or once in few days, but you need to ensure that some other application can invoke this operation any time.  Scheduled task starts and ends, it does not reside in the memory. This scenario should be implemented as a service which resides in memory and can provide certain operation at any time. **The complexity of implementation** Scheduled operation can be pretty much any executable on the Windows system. It can be simple console application, or it can have some UI which informs user that something is going on in a nice fancy dialog window, it can also be DOS batch (\*.bat) or command (\*.com) file.  Anyhow, you have more options how are you going to perform your operation which gives you certain range of flexibility to do it.  With services you do not have that freedom. Services are not so easy to debug (at least not easy as Windows Forms or console applications).  If you are not so good with code and you do not have some really complex operation, you should go with one of the options for scheduled task **Triggering mechanism** While scheduled task can be only invoked after some time span expires, service have more options to initiate some processing.  For example, if you need to perform some action when some file changes, you would have to use windows service which resides in memory and monitors specific file for a change. |

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

Introduction, Developing and Managing the Windows Service

**Continuous Learning: Technical Enablement**

Learn about Windows Service and Job Scheduling from the following

* [Windows Service](https://learn.microsoft.com/en-us/dotnet/framework/windows-services/walkthrough-creating-a-windows-service-application-in-the-component-designer)
* [TaskScheduler Class](https://learn.microsoft.com/en-us/dotnet/api/system.threading.tasks.taskscheduler?view=net-5.0)
* [Quartz API](https://www.quartz-scheduler.net/documentation/quartz-2.x/tutorial/jobs-and-triggers.html#the-quartz-api)

**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. |

* JOBS\_HOL\_001
* JOBS\_HOL\_002
* JOBS\_HOL\_003

**IDP – Project Activities**

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

**Sprint 2 Review & Rework**

* These day will be spent on Sprint 2 Review & Rework

**Milestone 8 - Scripting & Automation**

**Overview**

Milestone 8 will be focusing on **Python** which is used to build websites and software, automate tasks, and conduct data analysis.

**Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python supports multiple programming paradigms, including Procedural, Object Oriented and Functional programming language. Python design philosophy emphasizes code readability with the use of significant indentation.

**Performance Outcomes**

After completing this milestone, GenCs will be able to

* Design and program Python applications
* Use lists, tuples, and dictionaries in Python programs
* Demonstrate the Object Orientated Programming Concepts using Python
* Use class inheritance in Python for reusability
* Use exception handling in Python applications for error handling
* Develop Python Modules to Create Re-Usable Code

|  |  |  |
| --- | --- | --- |
| Courses/Skills | Learning Duration | Practice Duration |
| * Python | * 12 hrs. | * 12 hrs. |

|  |
| --- |
| **Do You Know?** |
| Here are some interesting facts about python  1. **Origin of python** Python was developed a hobby project by Guido Van Rossum in December 1989, while he was looking for a hobby project to keep him occupied in the week around Christmas. 2. **Logic behind its name** The language’s name isn’t about snakes, but about the popular British comedy troupe Monty Python. Its creator named so because he was a big fan of Monty Python’s Flying Circus. 3. **The Zen of Python** Tim Peters, a major contributor to the Python community, wrote this poem to highlight the philosophies of Python which can be read by just writing import this in the interpreter. 4. **Python does not require a compiler** As a high-level and interpreted language, Python does not need a compiler. This is unlike Java and C++ which have to be compiled first before being interpreted. For Python, it relies on the application interpreter. The Python byte code is stored in the form of a .pyc file which is then executed by an appropriate virtual machine. This machine acts as a run-time engine of Python. 5. **C and Java variants of python** Despite being an independent programming language, Python has variants for C and Java programming languages. The C variant is known as CPython and is designed to give Python the advantages of C. One of these characteristics is in terms of performance. The variant can act both as an interpreter and at the same time as a compiler. The Java variant of Python is known as Jython. It brings some key aspects of Java such as productivity and enables them to run on a virtual machine. 6. **Does not require braces** Unlike Java and C++, Python does not use braces to delimit code. Indentation is mandatory with Python, which keep tracks of the code. 7. **It is an open-source language** Despite the massive popularity that enjoys, Python is an open-source language, it does not have a proprietary license that controls who uses it. As an open-source language, members of the Python community are allowed to make their contributions to the Python ecosystem. 8. **It supports multiple assignments in one statement** Python allows assigning the same value to multiple variables in one statement. It will also let you assign values to multiple variables at once. 9. **Influence of python in JavaScript** Python is one of the 9 languages that influenced the design of JavaScript. Others include AWK, C, HyperTalk, Java, Lua, Perl, Scheme, and Self. 10. **Big Companies Using Python** Some of the big companies and institutions using python are: NASA, Facebook, IBM, Google, Nokia, Netflix, Yahoo! Maps, Quora, Hike, Amazon, Youtube, Mozilla, Uber, Dropbox, Expedia |

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

Basics, Variables and Types, Program Flow, Functions

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete Python Programming Masterclass Beginner to Advanced](https://cognizant.udemy.com/course/complete-python-programming-masterclass-beginner-to-advanced)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Getting Setup with Python * Section 3: Variables and Types * Section 8: Python Program Flow * Section 13: Python Functions * Implement the examples along with the author. |

**Hands-On**

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

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

* Alien's Visit
* Income Tax
* News Report Generation
* Palindrome

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

Python Operators, Collections, Object Oriented Programming (OOP)

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete Python Programming Masterclass Beginner to Advanced](https://cognizant.udemy.com/course/complete-python-programming-masterclass-beginner-to-advanced)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 4: Python Operators * Section 13: Python Collections * Section 14: Python Object Oriented Programming (OOP) * Implement the examples along with the author. |

**Hands-On**

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

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

* Search Student Data
* Password Protection
* Pass or Fail
* AEIMA’s Online Courses
* Arrange Names

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

File I/O, Exception Handling, Modules

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete Python Programming Masterclass Beginner to Advanced](https://cognizant.udemy.com/course/complete-python-programming-masterclass-beginner-to-advanced)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 10: Working With Files * Section 15: Handling Errors in Python * Section 7: Python Modules * Implement the examples along with the author. |

**Hands-On**

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

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

* Copy the File
* Store Student Data
* Farewell
* Rhythm Composer
* Time Table Planning

**Code Challenge (For Practice Only)**

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

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

* Code Challenge - Python

**Milestone 9 - Cloud**

**Overview**

Milestone 9 will be focusing on **Cloud Computing Basics** which is the delivery of different services through the Internet, including data storage, servers, databases, networking, and software.

**Cloud computing** is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each of which is a data center.

**Performance Outcomes**

After completing this milestone, GenCs will be able to

* Should be able to explain what is Cloud computing and its characteristics
* Should be able to explain types of Cloud
* Should be able to explain various Cloud Service Models and Cloud Service Providers

|  |  |  |
| --- | --- | --- |
| Courses/Skills | Learning Duration | Practice Duration |
| * Cloud Computing Basics | * 8 hrs. | * N/A |

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

Introduction to Cloud Computing, Types of Cloud, Cloud Service Models, Cloud Service Providers

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [AZ-900 Microsoft Azure Fundamentals + FULL Practice Exam!](https://cognizant.udemy.com/course/azure-fundamentals-lab)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Laying the Groundwork: Describe Cloud Concepts |

**Final Evaluation**

|  |
| --- |
| **Day 61, 62, 63** |

**Final Evaluation (Project + Technical)**

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

**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**

1. **Who can participate in this program?**

**Ans:** Students who have enrolled for Full Internship Program (or) the Cognizant on-boarded GenCs can participate in this program.

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

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

1. **What is Code Challenge?**

**Ans:** 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)?**

**Ans:** A case study problem statement will be provided to you that you may need to solve using the combination of skills learned in the given stage.

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

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

**Ans:** The coding challenges and ICTs are open from day 1, and a maximum of 3 attempts will be provided.

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

**Ans:** The eligibility criterion for the qualifier is 100% hands-on completion and attempt in the CC & ICT.

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

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

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

**Ans:** 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?**

**Ans:** 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?**

**Ans:** 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?**

**Ans:** Coach is your point of contact.