

Generation Cognizant (GenC)

ADM Standard Java Suite – Learning Guide

Learning Journey Guide

Java Track – Full



**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

x

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

**Service Lines**

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

**What is Application Development?**

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

**What is Application Maintenance?**

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

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

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

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

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

**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.

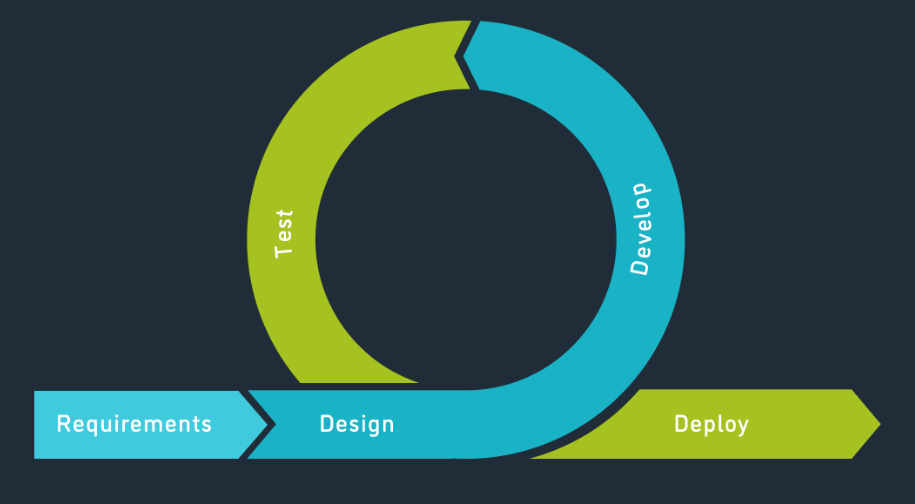
**Flipped Classroom**

**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.

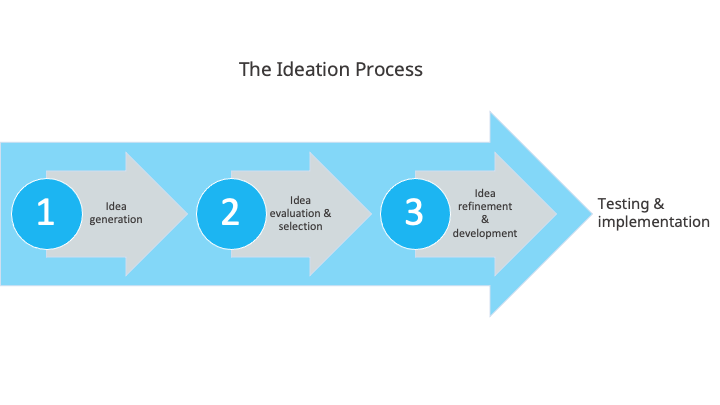


**Stages of IDP**

Following are the four seminal phases of IPD.

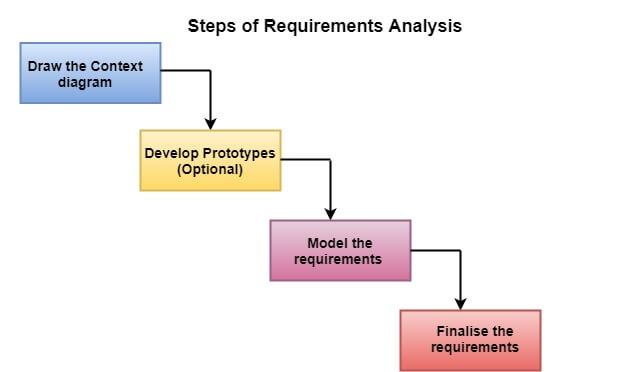
**Phase 1: Ideation**

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.



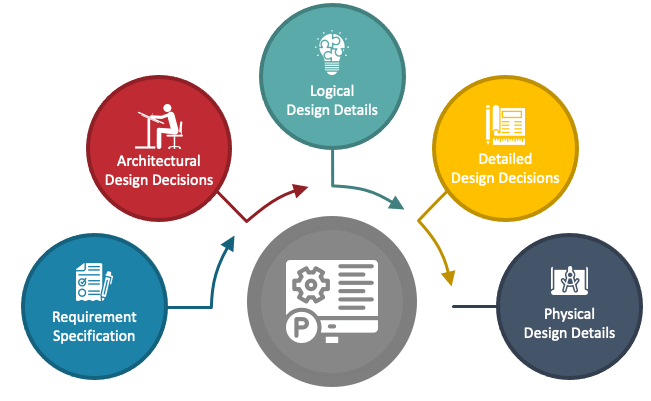
**Phase 2: Requirement Analysis**

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.



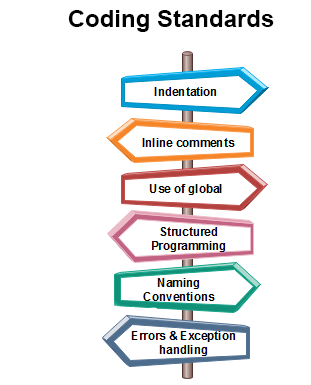
**Phase 3: Project Design**

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 two stages of technical learning and a Project**.**

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

**Stage 1 - Foundation in Software Development**

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

**Integrated Development Project (IDP)**

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

**Self-Learning via Udemy**

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

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

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

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

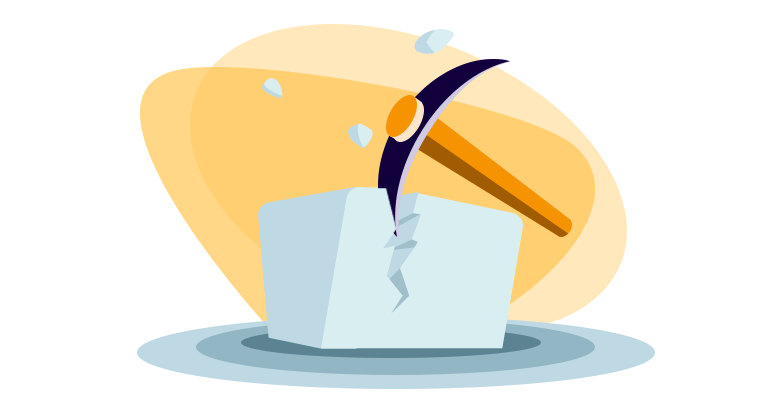
**Interim Evaluation:**

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

**Final Evaluation:**

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

**Icebreaker**



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 Engineering Basics
* UNIX
* GIT
* HTML5, CSS3, JavaScript, Bootstrap and jQuery
* C#
* ANSI SQL using Oracle
* Core Java
* Logging Frameworks: Log4j, SLF4J

**How and From Where to Learn?**

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

**Integrated Development Project (IDP) Roadmap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phases** | **Duration** | **Activities** | **Deliverables** |
| Phase1: Ideation/Planning | 6 Days | 1. Project ideation on an individual basis. 2. Conducting various brainstorming sessions to generate project ideas. 3. Finalizing the project idea and reviewing it with the technical SME. | 1. Project abstract and a title. |

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

**Overview**

Milestone 1 will be focusing on the Software Engineering basics and Agile Methodology.

**Software engineering** is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

**Agile methodology** is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

**Learning Objectives**

After completing this module, GenCs will be able to:

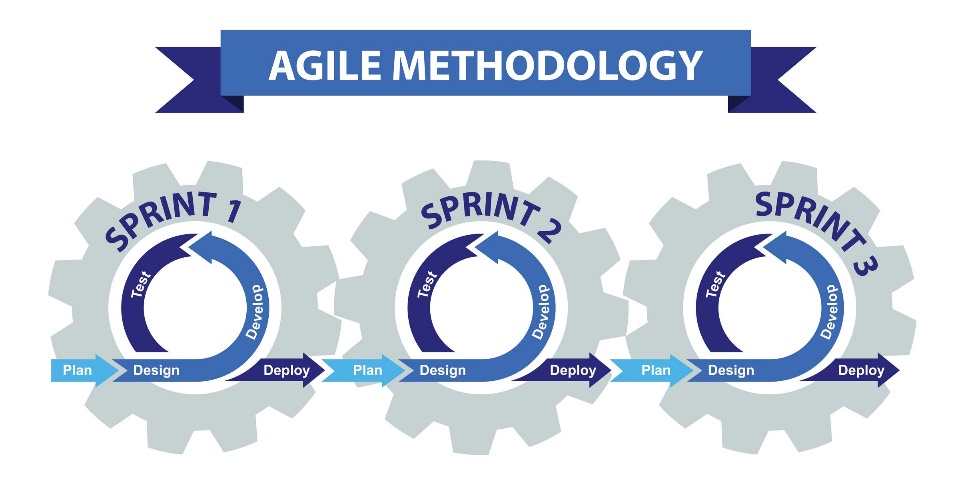
* Apply the software engineering lifecycle by demonstrating competence in communication, planning, analysis, design, construction, and deployment.
* Demonstrate an ability to use the techniques and tools necessary for engineering practice.

|  |
| --- |
| **Do you know?** |
| **Need of Software Engineering** The need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working.   * **Large software -**It is easier to build a wall than to a house or building, likewise, as the size of software become large engineering has to step to give it a scientific process. * **Scalability-**If the software process were not based on scientific and engineering concepts, it would be easier to re-create new software than to scale an existing one. * **Cost-**As hardware industry has shown its skills and huge manufacturing has lower down, the price of computer and electronic hardware. But the cost of software remains high if proper process is not adapted. * **Dynamic Nature-**The always growing and adapting nature of software hugely depends upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role. * **Quality Management-**Better process of software development provides better and quality software product. |

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

**What is Agile?**

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



**Agile Principles**

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

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

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

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

Learn about Agile process from the below Udemy course.

|  |  |
| --- | --- |
| 4FCA95A2 | [Agile Crash Course: Agile Project Management; Agile Delivery](https://cognizant.udemy.com/course/agile-crash-course/learn/lecture/5315576#overview)   * Learn entire sections in this Udemy course. |

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

**Continuous Learning: Technical Enablement**

Learn the basics of Software Engineering from the following Udemy course.

|  |  |
| --- | --- |
| 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 Basic UNIX Commands and its usage in various application development and maintenance scenarios.

**UNIX** is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Identify and use UNIX/Linux utilities to create and manage simple file processing operations.
* Use commands that organize directory structures with appropriate security.
* Explain the importance of filters and their need in UNIX.
* Demonstrate the use of various grep and sed commands.
* Use various process related commands to manage processes in UNIX.
* Use various networking commands to configure and troubleshoot in UNIX.

**UNIX Vs Linux - What’s the difference?**

|  |  |
| --- | --- |
| **Unix** | **Linux** |
| Unix is proprietary. | Linux is open source |
| Unix was created in 1960s by AT&T Bell Labs. | Linux was created in 1991 by Linus Torvalds. |
| Unix default shell is Bourne Shell. | Linux [**default shell is Bash**](https://www.linuxfordevices.com/tutorials/linux/linux-shells). |
| Unix can be installed only on web servers, workstations and PCs. | Linux can be installed on any device from mobile to supercomputers. |
| Various file systems supported by Unix are fs, gpfs, hfs+, hfs, zfs, xfs, ufs. | Various file systems supported by Linux are Ext2, Ext3, Ext4, jfs, NTFS, FAT, FAT32, Btrfs, xfs, vxfs. |
| Its source code isn’t publicly available. | Its source code is publicly available. |
| Unix was developed for servers, workstations and mainframes. | Linux can be used by anyone from a student to a developer. |
| Some popular examples of Unix Operating Systems are Solaris, SunOS, macOS, BSD. | Some popular examples of Linux Operating Systems are Ubuntu, RedHat, Debian GNU, OpenSuse. |

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

**Continuous Learning: Technical Enablement**

Learn the basics of UNIX utilities/commands from the following Udemy course.

|  |  |
| --- | --- |
| 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** |

**Continuous Learning: Technical Enablement**

Learn the basics of Bash Shell Scripting from the following Udemy course.

|  |  |
| --- | --- |
| 4FCA95A2 | [Bash Mastery: The Complete Guide to Bash Shell Scripting](https://cognizant.udemy.com/course/bash-mastery)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: How to Build a Bash Script * Section 2: Variables and Shell Expansions * Section 3: How Bash Processes Command Lines * Section 4: Requesting User Input * Section 5: Logic * Section 6: Processing Options & Reading Files * Section 7: Arrays + For Loops * Section 8: Debugging * 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 Duration: 1 Day**

**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.

**Learning Objectives**

After completing this module, 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.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| GIT | 4 hrs. | 4 hrs. |

|  |
| --- |
| **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** |

**Continuous Learning: Technical Enablement**

Learn the basics of GIT operations from the following Udemy course.

|  |  |
| --- | --- |
| 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 (Offline)**

Complete the following set of hands-on given in the Learning Path at Tekstac. You need **NOT** upload the solutions in the platform.

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

**Learning Objectives**

After completing this module, GenCs will be able to

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

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| HTML5, CSS3 and JavaScript | 12 hrs. | 10 hrs. |
| Bootstrap4 | 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** |

**HTML5, CSS3**

**Key Topics:**

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 technical quiz in the Learning Path at Tekstac to check your knowledge level of HTML5 and CSS3.

* Quiz 1 - HTML 5 & CSS3

**Code Challenge (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac to check your skill level in HTML5 and CSS3. You need to score 70% or higher to clear this challenge.

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

* Code Challenge - HTML5 and CSS3

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

**JavaScript**

**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.   + Getting Started   + Basics   + Operators   + Control flow   + Objects * 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. |

* Placing Order For Cake - String & Math
* Validate Email - Regular Expression & test Function
* Employee Experience Details - Class and Object & Date
* Greetings - DOM

**Technical Quiz**

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

* Quiz 2 - Java Script

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

**JavaScript**

**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. |

* Fixed And Reducing Interest Loan Estimator
* Word Play - Operators, Conditional Control Statements & Loops
* Find Unique Characters - Functions

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

**Bootstrap4**

Learn the basics of Bootstrap4

**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.   + Getting Started With Bootstrap 4   + Bootstrap 4 Basics   + Super Useful Utilities   + Forms * 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

**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 11** |

**Bootstrap 4**

Learn the basics of Bootstrap

**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.   + Navbars and Flexbox!   + The Magical Grid System   + Cards and List Groups * 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 12** |

**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 1: Introduction   + Section 3: Element Selectors   + Section 4: Manipulating the DOM I – Inserting, Replacing and Removing Elements * 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 13 - Forenoon** |

**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 5: Manipulating the DOM II – Changing   + Element Data and CSS   + Section 6: Events I ‒ Handling Mouse Events & Keyboard Events   + Section 7: Events II ‒ Forms * 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 **Database Design** and **SQL**.

**SQL** is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

**RDBMS** stands for **Relational DataBase Management Systems**. It is basically a program that allows us to create, delete, and update a relational database. Relational Database is a database system that stores and retrieves data in a tabular format organized in the form of rows and columns. It is a smaller subset of DBMS which was designed by E.F Codd in the 1970s. The major DBMS like SQL, My-SQL, ORACLE are all based on the principles of relational DBMS.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand the structure and design of relational databases.
* Retrieve and manipulate data in Oracle SQL with SQL queries.
* Retrieve data from multiple tables using Join and Subquery.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * ANSI SQL using Oracle | * 8 hrs. | * 8 hrs. |

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

**ANSI SQL using Oracle 12c**

**Key Topics:** Introduction to Database, Why Oracle Database, What is Relational Database, SQL Basics, Oracle Datatypes

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete Oracle SQL Bootcamp (2023)](https://cognizant.udemy.com/course/oracle-sql-12c-become-an-sql-developer-with-subtitle/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 1: What is Database, Why Oracle Database, RDBMS, What is SQL?   + Section 3: Oracle Datatypes   + Section 15: Data Definition Language (DDL) Statements   + Section 16: Data Manipulation Language (DML) Statements   + Section 17: Constraints * 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 Department table
* Create Student table
* Create Staff table
* Modify the datatype
* Alter table buses
* Add a constraint to Course Table
* Drop Student Table
* Insert Records – Tickets
* Insert Records into Student table
* Update Buses table
* Update Fees for short courses
* Remove Registration Details

**Additional Learning**

* [Introduction to NoSQL](https://www.toptal.com/database/the-definitive-guide-to-nosql-databases)

Try out the samples found in the above web pages.

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

**ANSI SQL using Oracle**

**Key Topics:** SELECT statement, Function - Scalar & Aggregate in Oracle SQL.

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete Oracle SQL Bootcamp (2023)](https://cognizant.udemy.com/course/oracle-sql-12c-become-an-sql-developer-with-subtitle/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 3: Retrieving Data   Section 4: Restricting Data   * + Section 5: Sorting Data   + Section 7: Single-Row Functions   + Section 8: Conversion Functions   + Section 9: Conditional Expressions   + Section 10: Group Functions * 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. |

**SELECT Statement**

* List Department names
* Department name based on block number
* Display Students Details
* Students Name based on Start and Ending Character
* Display User details
* List Schedule details

**Function - Scalar & Aggregate**

* List Department Name
* Password Generation
* Formatting Date
* Number of departments
* Minimum Mark

Learn about Plan Table from the following link:

* [Oracle Plan Table](https://docs.oracle.com/database/121/REFRN/GUID-0CAFEAD1-8C79-4200-8658-947D04BDFFE2.htm#REFRN29510)

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

**ANSI SQL using Oracle**

**Key Topics:** Joins and Subqueries and Plan table in Oracle SQL.

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete Oracle SQL Bootcamp (2023)](https://cognizant.udemy.com/course/oracle-sql-12c-become-an-sql-developer-with-subtitle/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 12: Joining Multiple Tables   + Section 13: Using Subqueries * 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. |

**Joins & Subquery - Hands-Ons**

* Subject with Staff Details
* Department with Student Count
* Department has least Student Count
* Student with Minimum mark
* Student mark in particular subject
* Maximum mark in Subject with Staff name

**Technical Quiz**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of ANSI SQL Using Oracle.

* Test Your Understanding

**Code Challenge (For Practice Only)**

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

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

* Code Challenge - Oracle SQL

**IDP - High Level Design**

|  |  |
| --- | --- |
| reminder-bot · GitHub Topics · GitHub | **Be ready with the high-level design of the IDP.** |

**Milestone 6 - Application Programming & Logging Duration: 1 Day**

**Overview**

**Milestone 6** will focus on application programming using Java and its practical implementations in various application development and maintenance scenarios. Additionally, this milestone will cover popular Java Logging Frameworks such as Log4j and SLF4J.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Write simple Java programs using basic syntax.
* Demonstrate proficiency in using Java syntax
* Use operators and control flow statements effectively
* Apply OOP concepts to solve real-world problems
* Utilize classes and methods from the java.lang package
* Use collections effectively for storing and manipulating data
* Handle file operations such as reading, writing, and manipulation.
* Use try-catch blocks to handle exceptions gracefully.
* Organize code into packages for better management.
* Use interfaces and abstract classes effectively in program design.
* Apply functional programming concepts using streams and lambdas.
* Work with date and time in Java using the java.time package.
* Create and manage threads in Java.
* Create networked applications using sockets.
* Understand the new features introduced in Java 8, 11, 12, and 17.
* Perform database operations like CRUD (Create, Read, Update, Delete) using JDBC.
* Demonstrate the ability to implement logging in Java applications using standard logging libraries such as Log4j and SLF4J.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Core Java | * 30 hrs. | * 30 hrs. |
| * Logging Frameworks: Log4j, SLF4J | * 8 hrs. | * 4 hrs. |

|  |
| --- |
| **Do You Know?** |
| * **Platform Independence:** Java is known for its "write once, run anywhere" principle, meaning that Java code can be written on one platform and run on any other platform that supports Java without the need for recompilation. * **Object-Oriented:** Java is an object-oriented programming language, which means it is based on the concept of objects that contain both data and methods. * **Automatic Memory Management:** Java uses automatic garbage collection, which means that developers do not need to explicitly deallocate memory, reducing the chances of memory leaks. * **Robust and Secure:** Java is designed to be robust, with features like strong memory management, exception handling, and a strict type system. It also has built-in security features that help protect against malicious code. * **Multi-threaded:** Java has built-in support for multi-threading, allowing programs to execute multiple threads simultaneously, which is useful for developing concurrent applications. * **Popular in Enterprise Applications:** Java is widely used in enterprise environments for developing large-scale, mission-critical applications due to its scalability, reliability, and performance. * **Backward Compatibility:** Java is known for its backward compatibility, which means that newer versions of Java are designed to be compatible with older versions, ensuring that existing code continues to work. * **Community Support:** Java has a large and active community of developers who contribute to its ecosystem by creating libraries, frameworks, and tools, making it a versatile and constantly evolving language. * **Used in a Variety of Applications:** Java is used in a wide range of applications, including web development (Java EE), mobile development (Android), desktop applications (Java SE), and big data processing (Apache Hadoop). * **Maintained by Oracle:** Java is developed and maintained by Oracle Corporation, which provides regular updates and support for the language and its ecosystem. |

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

**Core Java**

**Key Topics:** Introduction to Java, Basic Syntax and Data Types.

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 3: Software Installation and First Java Program   + Section 30: JVM Architecture   + Section 6: Datatypes, Literals, Variables, Type Conversion, Casting & Promotion   + Section 8: Operators and Assignments   + Section 9: Control Flow 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. |

* Display Characters
* Fuel Consumption Calculator
* Highest Placement

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

**Core Java**

**Key Topics:** Java Standard Library (java.lang)

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 22: String Handling   + Section 24: Arrays   + Section 7: Wrapper Classes with Auto boxing and unboxing   + Section 4: Static Members and their execution control flow   + Section 5: Non-Static and their execution control flow * 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. |

* Least offer
* String Concatenation
* Ticket Price Calculation – Static
* Student Details - Constructor

**Technical Quiz**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of Java fundamental concepts.

* Quiz 1- Java Operator, Control flow statement

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

**Core Java**

**Key Topics:** Object-Oriented Programming (OOP)

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 2: Introduction to Java and OOPS * Section 10: Access Modifiers * Section 13: Inheritance * Section 14: Abstraction * Section 15: Polymorphism * Section 16: Encapsulation * 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 Details of Hosteller
* Account Manipulation – Abstract class
* BankAccountDetails
* Employee Loan Eligibility – Polymorphism
* Vehicle-Loan-Insurance - Use Interface

**Technical Quiz**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of OOPS concept in Java.

* Quiz 2- Applying Object Oriented Concepts in java

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

**Core Java**

**Key Topics:** Collections Framework

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604" \l "overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 26: Collections with Generics * 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. |

* Insurance Bazaar
* Number of New Words
* Phone Book Manipulation

**Technical Quiz**

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

* Quiz 3 - Collections Framework

**Code Challenge (For Practice Only)**

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

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

* Code Challenge – Group1

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

**Core Java**

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

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 23: IO Streams (File IO)   + Section 17: Exception Handling and Assertions * 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. |

* Retrieving Data from file
* Array Manipulation - Use try with multi catch
* Register a Candidate - User defined Exception (with throw and throws)

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

**Core Java**

**Key Topics:** Java Packages, Annotations, Interfaces and Abstract Classes

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + **Section 11:** Packages   + **Section 32:** Annotations * Implement the examples along with the author. |

|  |  |
| --- | --- |
| 4FCA95A2 | [Java 17 Masterclass: Start Coding in 2023](https://cognizant.udemy.com/course/java-the-complete-java-developer-course)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + **Section 11:** Abstraction in Java * Implement the examples along with the author. |

**Hands-On**

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

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

**<TBU>**

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

**Java**

**Key Topics:** Java Streams and Lambda Expressions, Java Date and Time

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Java 17 Masterclass: Start Coding in 2023](https://cognizant.udemy.com/course/java-the-complete-java-developer-course)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 14:** Lambda Expressions, Functional Interfaces, and Method References * Implement the examples along with the author. |
| 4FCA95A2 | [Learn JAVA Programming - Beginner to Master](https://cognizant.udemy.com/course/java-se-programming)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 26:** Date and Time API * 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 Loan Eligibility
* Placement Enrollment Count
* Auditing
* Validate Name
* Travel Agency
* Fruit Basket Estimation
* Mall Parking System

**Technical Quiz**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of advanced concepts in Java.

* Quiz 4- Advanced Java Concepts

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

**Core Java**

**Key Topics:** Java Threads and Concurrency, Networking and Sockets

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Learn JAVA Programming - Beginner to Master](https://cognizant.udemy.com/course/java-se-programming)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 19:** Multithreading * Implement the examples along with the author. |

|  |  |
| --- | --- |
| 4FCA95A2 | [Java 17 Masterclass: Start Coding in 2023](https://cognizant.udemy.com/course/java-the-complete-java-developer-course)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * **Section 26:** Java Networking Programming * 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. |

* Employee Promotion

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

**Core Java**

**Key Topics:** JDBC (Java Database Connectivity), Java 11, 12 & 17 Features

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Java Database Connection: JDBC and MySQL](https://cognizant.udemy.com/course/how-to-connect-java-jdbc-to-mysql/)   * Go through entire course. * 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. |

* Zaro Flight System
* Train Management
* Selection Management System

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Core Java Made Easy (Covers the latest Java 18)](https://cognizant.udemy.com/course/corejavamadeeasy/learn/lecture/5607604#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 40: Java 11 Features   + Section 41: Java 12 Features   + Section 46: Java 17 * 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. |

* EB Connection - Switch Enhancements & Compact Number Formatting
* Holiday Homework - String Methods (Java 11)
* Association Details - String Methods (Java 12)
* School Bus allocation - File operations
* Minimum And Maximum Marks - Teeing Collector

**Code Challenges (For Practice Only)**

Attempt the following Code Challenge through the Learning Path at Tekstac to assess your skill level in Java and JDBC concepts. You need to score 70% or higher to clear this challenge. You will have a maximum of three attempts to complete the practice test.

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

* Code Challenge - Group2

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

**Core Java**

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

Attempt the following Integrated Capability Test (ICT) through the Learning Path at Tekstac to assess your skill level in Stage 1, focusing on database and application programming skills. You must achieve a minimum score of 70% to pass this practice test. You will have a maximum of three attempts to complete the practice test.

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

* Java: Integrated Capability Test

|  |
| --- |
| **Day 25 - Afternoon, 26** |

**Logging Frameworks: Log4j, SLF4J**

**Key Topics:** Introduction to Logging, Java Logging Frameworks

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Software Architecture and Clean Code Design in OOP](https://cognizant.udemy.com/course/software-architecture-learnit)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 12: Logging in Java * Implement the examples along with the author. |

**Qualifier Assessment**

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

**Stage 1 Qualifier Assessment**

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

**IDP - Project Activities**

|  |
| --- |
| **Day 29, 30** |

**IDP - Project Design**

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

|  |
| --- |
| **Day 31** |

* This day has been added to adjust the behavioral training hours throughout Stage 1.

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

**Overview**

**Stage 2** focuses on Application Development and Maintenance Practices essential for developing and maintaining diverse software applications. We offer a unique learning experience by providing diversified 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
* TDD, JUnit and Mockito, Code Quality Tools (PMD, Checkstyle, SONAR, JaCoCo, FindBugs)
* Oracle PL/SQL
* Spring Core Basics, Maven
* Spring Data JPA
* Spring MVC with Spring Boot
* Microservices Basics
* Data Structures and Algorithms
* Application Debugging Using Eclipse
* ITIL
* Jira & ServiceNow
* 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 & Generic Modelling Language**

**Overview**

**Milestone 1** will focus on **Design Principles and Patterns, as well as UML**, 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.

**Unified Modeling Language (UML)** is a standardized modeling language used in software engineering to visually represent a system's design. It provides a set of graphical notations to depict the structure and behavior of a system, allowing software developers to communicate and understand the system's architecture, design, and functionality. UML diagrams, such as class diagrams, use case diagrams, sequence diagrams, and activity diagrams, aid in visualizing different aspects of a system, facilitating the analysis, design, and documentation of software projects. UML is widely adopted in the industry due to its versatility and ability to capture complex system designs in a clear and understandable manner.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Design and construct simple components by applying design principles and patterns.
* Employ design principles in object-oriented software development.
* Refactor existing designs to use design patterns.
* Demonstrate proficiency in creating and interpreting UML diagrams for designing and communicating software system architectures.

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

|  |
| --- |
| **Do You Know?** |
| **Types of Design Patterns**  Design patterns are categorized into three main types: creational, structural, and behavioral patterns. Creational patterns focus on object creation mechanisms, structural patterns deal with object composition, and behavioral patterns describe how objects interact.    **Importance in Software Development**  Design patterns promote reusability, flexibility, and maintainability in software development. They provide a common language for developers to communicate solutions to design problems and improve the overall quality of software systems.  **Types of UML Diagrams**  UML offers various types of diagrams, including structural diagrams (e.g., class diagrams, component diagrams), behavioral diagrams (e.g., sequence diagrams, state diagrams), and interaction diagrams (e.g., use case diagrams, activity diagrams), each serving a specific purpose in system modeling.  Flowchart depicting the different types of UML diagrams  **Application in Software Development**  UML diagrams are widely used to model software systems, enabling developers to visualize system architecture, design classes and relationships, specify behavior, and communicate design decisions effectively throughout the software development lifecycle. |

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

**Design Patterns and Principles, UML Basics**

GoF Design Patterns

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Complete Java Design Patterns masterclass 2023](https://cognizant.udemy.com/course/javadesignpatterns)   * Learn entire sections of this Udemy course and implement the examples along with the author. * All Sections |

**Hands-On**

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

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

**<TBU>**

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

**Design Patterns and Principles, UML Basics**

Different Types of Software Design Principles

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [SOLID Principles: Introducing Software Architecture & Design](https://cognizant.udemy.com/course/solid-design)   * Learn entire sections of this Udemy course and implement the examples along with the author. * All Sections |

**Hands-On**

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

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

**<TBU>**

|  |
| --- |
| **Day 33 - Afternoon, 34** |

**Design Patterns and Principles, UML Basics**

Introduction to UML, UML Diagram Types, UML Diagram Elements

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete UML Course (2024): Learn to Design UML Diagrams](https://cognizant.udemy.com/course/unified-modeling-language-uml-course-uml-diagram-software-enginnering)   * Learn entire sections of this Udemy course and implement the examples along with the author. * All Sections |

**Milestone 2 - Unit Testing**

**Overview**

Milestone 2 will focus on **Unit Testing tools and frameworks like JUnit and Mockito, as well as Code Quality Tools (PMD, Checkstyle, SONAR, JaCoCo, FindBugs)** and their implementation in 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.

**JUnit** is a unit testing framework for Java programming language. JUnit has been important in the development of test-driven development. TDD is one of those practices which contribute to better code quality and decreases bugs.

**Mockito** is a Java framework that creates and configures mock objects for unit testing. It makes it easier to simulate dependencies, define behaviors, and verify method calls. Mockito is commonly used with JUnit to write effective and focused unit tests, improving the quality of Java applications.

**Code Quality** - The code quality is important, as it impacts the overall software quality. And quality impacts how safe, secure, and reliable your codebase is. Developers can improve the Java code quality by simply applying best practices and using the right analysis tools.

**PMD** is a popular tool for analyzing Java code. It checks for common coding mistakes, adherence to coding standards, and potential bugs. By using PMD, developers can improve the quality of their code and identify issues early in the development process.

**Checkstyle** is a tool that checks code for consistency with coding standards. It helps maintain code readability and quality by identifying issues like naming conventions and formatting. By using Checkstyle, teams can ensure that their code follows best practices, leading to better software quality.

**SonarQube** is an open-source platform used for continuously checking code quality. It identifies bugs, code smells, and security vulnerabilities in code written in various languages like Java and JavaScript. SonarQube integrates with build tools and continuous integration servers to automatically analyze new code. It helps teams maintain code quality standards and improve overall software maintainability.

**JaCoCo** is a popular Java code coverage library used to measure how much of a Java program's source code is tested by a particular test suite. It generates reports showing the percentage of code lines, branches, and methods covered by tests, helping developers identify areas that need more testing. JaCoCo is valuable for improving the quality and reliability of Java applications by ensuring comprehensive test coverage.

**FindBugs** is a tool that helps identify potential bugs in Java code by analyzing the bytecode. It detects various issues, such as null pointer errors and bad practices, which can cause runtime problems. FindBugs generates detailed reports to help developers fix these issues early, improving the overall quality and reliability of Java applications.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Demonstrate proficiency in writing unit tests using JUnit and Mockito.
* Apply mocking techniques using Mockito to isolate components for unit testing.
* Apply PMD to identify and address potential code issues and maintain code quality.
* Use Checkstyle to enforce coding standards and best practices in the codebase.
* Analyze code quality metrics and reports generated by SONAR to improve code maintainability and reliability.
* Utilize JaCoCo for code coverage analysis to ensure comprehensive test coverage.
* Identify and fix potential bugs and code issues using FindBugs to enhance code quality and stability.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * TDD, JUnit and Mockito, Code Quality Tools (PMD, Checkstyle, SONAR, JaCoCo, FindBugs) | * 12 hrs. | * 12 hrs. |

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

**TDD, JUnit and Mockito**

Test-Driven Development (TDD), Unit Testing Basics, Introduction to JUnit, Introduction to Mockito

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Instant Test Driven Development with Java, JUnit and Mockito](https://cognizant.udemy.com/course/instant-test-driven-development-with-junit)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 1: Introduction   + Section 2: The Basics   + Section 3: Basic Testing with Mocks using Mockito * 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. |

* LMS Refactoring
* Electricity Bill
* Product Login Test suite
* Testing Using Assertion
* Parameterized

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

**TDD, JUnit and Mockito**

JUnit Test Structure, Stubbing and Verification, Mockito Best Practices

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Instant Test Driven Development with Java, JUnit and Mockito](https://cognizant.udemy.com/course/instant-test-driven-development-with-junit)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 4: JUnit in more Depth   + Section 5: More Assertions   + Section 6: Parameterized Tests   + Section 7: Mockito in more Depth * 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. |

* Testing Using Assertion
* Parameterized

**Code Reference Slice**

This module gives an insight on how the code base will look like in a client project. The code reference will help to understand how the code is structured in actual project, also the quality standards that need to be followed during the project implementation phase.

**Hands-On**

Download the code reference slice from the Learning Path at Tekstac. Import the code reference in eclipse and explore the classes in the package structure. Test the APIs to understand the various functionalities and best practices followed in the module.

|  |
| --- |
| **Day 37, 38 - Forenoon** |

**Code Quality Tools (PMD, Checkstyle, SONAR, JaCoCo, SpotBugs)**

PMD, Checkstyle, SonarQube, JaCoCo, SpotBugs

|  |  |
| --- | --- |
| **Tool Name** | **Topics** |
| PMD | [Introduction to PMD](https://www.baeldung.com/pmd) |
| [PMD Rules](https://pmd.github.io/pmd/pmd_rules_java.html) |
| [PMD Reports](https://docs.pmd-code.org/latest/pmd_userdocs_report_formats.html) |
| [Rule Customization and Suppression](https://pmd.github.io/pmd/pmd_userdocs_suppressing_warnings.html) |
| Checkstyle | [Introduction to Checkstyle](https://www.geeksforgeeks.org/introduction-to-checkstyle-plugin-for-checking-java-code-quality/) |
| [Checkstyle Rules](https://github.com/ngeor/checkstyle-rules) |
| [Checkstyle Reports](https://checkstyle.org/report_issue.html) |
| [IDE Integrations](https://checkstyle.org/eclipse-cs/#!/) |
| SonarQube | [Introduction to SonarQube](https://www.javatpoint.com/sonarqube-in-java) |
| [SonarQube Metrics and Rules](https://docs.sonarsource.com/sonarqube/latest/user-guide/metric-definitions/) |
| [SonarQube Reports](https://www.opcito.com/blogs/elevate-code-quality-with-sonarqube-analysis-reporting-details) |
| [IDE Integrations](https://docs.sonarsource.com/sonarlint/eclipse/getting-started/installation/) |
| SpotBugs | [Introduction to SpotBugs](https://spotbugs.readthedocs.io/en/stable/introduction.html) |
| [SpotBugs Analysis](https://spotbugs.readthedocs.io/en/stable/analysisprops.html) |
| [SpotBugs Reports](https://spotbugs.github.io/spotbugs-maven-plugin/usage.html) |
| [IDE Integrations](https://spotbugs.readthedocs.io/en/stable/links.html#ide-integration) |
| JaCoCo | [Introduction to JaCoCo](https://www.baeldung.com/jacoco) |
| [JaCoCo Reports](https://www.lambdatest.com/blog/reporting-code-coverage-using-maven-and-jacoco-plugin/) |

**Hands-On**

* Try out the example codes and steps covered in the above links.

**Milestone 3 - Advanced SQL**

**Overview**

Milestone 3 will be focusing on **Advanced SQL** in various application development and maintenance scenarios.

**PL/SQL** stands for **Procedural Language/Structured Query Language**. It is a combination of SQL with procedural features of programming language. It stored and compiled in the database, runs within the Oracle executable and inherits the security, robustness, and portability of the Oracle Database.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Demonstrate a database solution for a business using PL/SQL.
* Create and manage Stored Procedures and Functions.
* Write and manage database triggers, cursors and Index.

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

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

**Oracle PL/SQL**

**Key Topics:** Introduction to PL/SQL

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete PL/SQL Bootcamp : "Beginner to Advanced PL/SQL"](https://cognizant.udemy.com/course/plsql-beginner-to-advanced-become-a-perfect-plsql-developer)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 1: What is PL/SQL?   + Section 2: Software Installation   + Section 3: Let’s Start Coding * 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. |

* Insert Record using Anonymous Block

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

**Oracle PL/SQL**

**Key Topics:** PL/SQL Variables and Data Types, Control Structures in PL/SQL.

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete PL/SQL Bootcamp : "Beginner to Advanced PL/SQL"](https://cognizant.udemy.com/course/plsql-beginner-to-advanced-become-a-perfect-plsql-developer)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 4: PL/SQL Variables   + Section 5: Control Structures   + Section 6: Using SQL in PL/SQL   + Section 7: Composite Data Types * 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. |

* Update Location
* Area of a Circle

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

**Oracle PL/SQL**

**Key Topics:** Exception Handling, Cursors, Procedures and Functions

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete PL/SQL Bootcamp : "Beginner to Advanced PL/SQL"](https://cognizant.udemy.com/course/plsql-beginner-to-advanced-become-a-perfect-plsql-developer)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 9: PL/SQL Exceptions   + Section 8: PL/SQL Cursors   + Section 10: PL/SQL Functions & Procedures * 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. |

* Insert credit - Procedure
* Select city – Procedure
* Procedure with Exception Handling
* Display department names using Cursors

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

**Oracle PL/SQL**

**Key Topics:** Triggers, Packages, Dynamic SQL in PL/SQL, Error Handling and Logging, Bulk Processing

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [The Complete PL/SQL Bootcamp : "Beginner to Advanced PL/SQL"](https://cognizant.udemy.com/course/plsql-beginner-to-advanced-become-a-perfect-plsql-developer)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 11: PL/SQL Packages   + Section 12: PL/SQL Triggers   + Section 13: PL/SQL Debugging: Debugging Your Codes & Subprograms   + Section 14: Using Dynamic SQL&PL/SQL in PL/SQL! * 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. |

* Package with a Procedure to update salary
* Insert a Record – Triggers

**Code Challenge (For Practice Only)**

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

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

* Code Challenge - PL/SQL

**IDP - Project Activities**

.

|  |
| --- |
| **Day 42, 43, 44** |

**Sprint 1 Development & Demo**

* These three days will be spent on Sprint 1 development & rework

**Interim Evaluation**

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

**Interim Project Evaluation & Interim Technical Evaluation**

* Interim evaluation will be conducted on this day

**Milestone 4 - IoC and Dependency Injection Framework**

**Overview**

**Milestone 4** will be focusing on **IoC and Dependency Injection Framework**.

In software design, **Inversion of Control (IoC)** shifts the usual control flow from application code to a framework or container. Traditionally, code manages its own components, but IoC lets a container handle the creation, lifecycle, and connections between these components. This promotes a more modular and loosely coupled system, as the application code no longer dictates how components are instantiated and used.

**Dependency Injection (DI)** is a practical application of IoC, focusing on managing class dependencies. Instead of classes creating their dependencies, DI involves injecting them from external sources. The Spring Framework, a popular Java framework, excels in implementing DI. Its IoC container, the ApplicationContext, takes care of managing and injecting these dependencies, promoting modular, maintainable, and testable code.

The **Spring Framework** is a versatile Java framework that embraces IoC and DI principles. Spring's IoC container, the ApplicationContext, oversees the creation and management of Java objects, ensuring smooth IoC implementation. Dependency Injection in Spring is facilitated through approaches like constructor and setter injection, offering flexibility for developers. Annotations like @Autowired simplify dependency injection, making Spring an effective tool for building scalable, modular, and maintainable Java applications.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Create and manage a simple application using Maven.
* Apply Dependency Injection Scenario using Spring Core.
* Use Spring JDBC and Transactions for a simple scenario.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Spring Core Basics, Maven | * 12 hrs. | * 12 hrs. |

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

**Spring Core Basics**

**Key Topics:** Introduction to Spring Framework, Dependency Injection

|  |  |
| --- | --- |
| 4FCA95A2 | [Spring Framework in Easy Steps](https://cognizant.udemy.com/course/springframeworkineasysteps/)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: Introduction * Section 2: Software Setup * Section 3: Spring Core Concepts * Section 4: Setter Injection * Section 5: Life Cycle Methods * Section 6: Dependency Check, Inner beans and scopes * Section 7: Constructor Injection * Section 9: Autowiring * Section 10: Standalone Collections * Section 11: Stereotype Annotations * Section 13: Injecting Interfaces * 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. |

* Constructor Injection
* Autowiring
* CurrencyConverter-Collections
* Customer-Address-Scope
* DBConfig-SetterBasedInjection
* Customer-Address Inner Bean
* EZEE Transport
* EBanking

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

**Spring Core Basics**

**Key Topics:** Reading values from Property Files, Spring JDBC and Transactions

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[Spring Framework in Easy Steps](https://cognizant.udemy.com/course/springframeworkineasysteps/)](https://cognizant.udemy.com/course/entity-framework-tutorial)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 8: Using Properties   + Section 14: Spring JDBC * 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. |

* Billing Software Application
* EBill

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

**Maven**

**Key Topics:** Maven Basics, Build Lifecycle, Plugins, Running Maven Builds

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Maven Crash Course](https://cognizant.udemy.com/course/mavencrashcourse/learn/lecture/6031666" \l "overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 1: Introduction   + Section 2: Simple Software Setup   + Section 3: Maven Project Creation and Key Concepts   + Section 4: Maven in Eclipse   + Section 5: Maven Web Application   + Section 6: Multi Module Project Creation   + Section 7: Organizing the multi module project   + Section 8: Scopes   + Section 9: Dependency Management   + Section 10: Profiles   + Section 15: Plugins   + Section 16: Properties * Implement the examples along with the author. |

**Additional Learning**

* [Spring Message resource bundle](https://mkyong.com/spring/spring-resource-bundle-with-resourcebundlemessagesource-example/)

**Hands-On**

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

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

* Engine Analysis
* Passport Service

**Code Challenge (For Practice Only)**

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

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

* Code Challenge - Spring Framework

**Milestone 5 - Java Persistence API**

**Overview**

**Milestone 5** will be focusing on **Spring Data JPA**.

**Spring** **Data** **JPA**is a Java specification for managing **relational** data in Java applications. It allows us to access and persist data between Java object/ class and relational database. JPA follows **Object-Relation Mapping**(ORM).

**Learning Objectives**

After completing this module, GenCs will be able to:

* Write DAO layer with ORM concepts using Spring Data JPA.
* Perform basic CRUD operations using Spring Data JPA.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Spring Data JPA | * 8 hrs. | * 8 hrs. |

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

**Spring Data JPA**

**Key Topics:** ORM Basics

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Spring Data JPA using Hibernate](https://cognizant.udemy.com/course/spring-data-jpa-using-hibernate/learn/lecture/8483682#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 4: ORM Basics * Section 5: Simple CRUD operations * Implement the examples along with the author. |

**Additional Learning**

Go through the below topic to learn Spring Data JPA features and Repository.

* [Spring Data JPA Introduction](https://www.javatpoint.com/spring-boot-starter-data-jpa)

**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. |

* SpringData\_JPA\_Oracle\_001
* SpringData\_JPA\_Oracle\_002
* SpringData\_JPA\_Oracle\_003

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

**Spring Data JPA**

**Key Topics:** Mapping, Persisting objects into database, Detached Objects

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Spring Data JPA using Hibernate](https://cognizant.udemy.com/course/spring-data-jpa-using-hibernate/learn/lecture/8483682#overview)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 6: Generators * Section 7: Spring Data Finder Methods * Section 8: Paging and Sorting * Section 9: JPQL * Implement the examples along with the author. |

**Additional Learning**

Go through the below topic to learn Spring Data JPA Relationships.

* [Spring Data JPA Relationships](https://medium.com/huawei-developers/database-relationships-in-spring-data-jpa-8d7181f50f60)

**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. |

* SpringData\_JPA\_Oracle\_004
* SpringData\_JPA\_Oracle\_005

**Milestone 6 - MVC Framework**

**Overview**

**Milestone 6** will be focusing on **Spring MVC with Spring Boot**.

**Spring MVC with Spring Boot** is widely used framework to create scalable web applications. With Spring Boot, you’ll be able to develop Spring applications with more agility and be able to focus on addressing your application’s functionality needs with minimal. You can easily create a self-contained HTTP server using embedded Tomcat, Jetty, or Undertow. Most web applications will use the spring-boot-starter-web module to get up and running quickly. Spring Boot provides auto-configuration for Spring MVC.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Initialize a Spring Boot Project and work with an existing project.
* Able to use simple MVC components for coding the presentation layer of a module using Spring MVC.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Spring MVC with Spring Boot | * 16 hrs. | * 12 hrs. |

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

**Spring MVC with Spring Boot**

**Key Topics:** Setting Up Development Environment, Spring Boot Fundamentals

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[NEW] Spring Boot 3, Spring 6 & Hibernate for Beginners](https://cognizant.udemy.com/course/spring-hibernate-tutorial)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: NEW - Spring Boot 3 Quick Start * 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. |

* Age Calculator
* BodyMassIndex

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

**Spring MVC with Spring Boot**

**Key Topics:** Spring MVC Basics, Spring Boot and Spring MVC Integration, Handling Form Submissions

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[NEW] Spring Boot 3, Spring 6 & Hibernate for Beginners](https://cognizant.udemy.com/course/spring-hibernate-tutorial)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + **Section 6:** NEW - Spring MVC * 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. |

* Zee Zee Login
* Bakingo Cake Service

|  |
| --- |
| **Day 53** |

**Spring MVC with Spring Boot**

**Key Topics:** Data Access with Spring Boot

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[NEW] Spring Boot 3, Spring 6 & Hibernate for Beginners](https://cognizant.udemy.com/course/spring-hibernate-tutorial)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 7: NEW - Spring MVC CRUD   + Section 3: NEW - Hibernate/JPA CRUD * Implement the examples along with the author. |

**Additional Learning**

Spring MVC Internationalization (i18n) - implement internationalization using the Spring MVC framework. Learn the additional topics as mentioned below.

Refer this [document](https://www.concretepage.com/spring-4/spring-mvc-internationalization-localization) and implement the example.

* [Introductionto Tomcat](https://www.baeldung.com/tomcat)
* [Tomcat servlet interactions](https://www.mulesoft.com/tcat/tomcat-servlet)
* [Application Server Vs Web Server](https://www.tutorialspoint.com/difference-between-web-server-and-application-server)
* [MVC Frameworks Overview](https://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm)
* [MVC Architecture](https://dzone.com/tutorials/java/struts/struts-tutorial/struts-mvc-architecture-tutorial.html)

**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. |

* HolidayParty-Validations
* Front End-Internationalization

|  |
| --- |
| **Day 54, 55 - Forenoon** |

**Spring MVC with Spring Boot**

**Key Topics:** Spring Security

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[NEW] Spring Boot 3, Spring 6 & Hibernate for Beginners](https://cognizant.udemy.com/course/spring-hibernate-tutorial)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 8: NEW - Spring MVC Security * Implement the examples along with the author. |

**Technical Quiz**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of Spring MVC with Spring Boot.

* Quiz 1 - Spring MVC and Spring Boot

**Code Challenge (For Practice Only)**

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

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

* Code Challenge - Spring MVC with Spring Boot

**Milestone 7 - Software Architecture**

**Overview**

In this milestone, learners will focus on the fundamentals of Microservices Architecture, a contemporary approach to software design that emphasizes building applications as a collection of loosely coupled services. The curriculum will cover key concepts such as the principles of microservices, including single responsibility, autonomy, and independent scalability.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Explain the differences between microservices and monolithic architecture.
* Identify best practices for designing and implementing microservices architecture.

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

|  |
| --- |
| **Do You Know?** |
| 1. **Evolution of Architecture:** Microservices architecture is an evolution of traditional monolithic architecture, aiming to address the limitations of monolithic systems by breaking down applications into smaller, independently deployable services. 2. **Modularity and Scalability:** Microservices promote modularity, allowing teams to work on different services independently, which can lead to faster development cycles. It also enables scalability, as each service can be scaled independently based on its specific needs. 3. **Technology Diversity:** Microservices architecture allows for the use of different technologies and programming languages within a single application. This approach, known as polyglot programming, enables developers to choose the best tool for each specific task. 4. **Fault Isolation:** In a microservices architecture, if one service fails, it does not bring down the entire system. This fault isolation ensures that failures are contained within the affected service, minimizing the impact on other parts of the application. 5. **Resilience to Change:** Microservices are designed to be resilient to change, making it easier to adapt to evolving business requirements. This flexibility is particularly valuable in fast-paced, dynamic environments. 6. **Communication via APIs:** Communication between microservices is typically done through well-defined APIs, which allows for loose coupling between services. This loose coupling enables services to evolve independently without disrupting the entire system. 7. **Deployment Flexibility:** Microservices can be deployed independently of each other, allowing for continuous delivery and deployment. This flexibility is essential for maintaining a fast pace of development and responding quickly to market demands. 8. **Industry Adoption:** Microservices architecture has gained significant adoption in industries such as e-commerce, finance, and technology, where scalability, flexibility, and resilience are critical for success. 9. **Challenges:** While microservices offer many benefits, they also come with challenges such as managing distributed systems, ensuring data consistency, and implementing effective monitoring and management solutions. |

**Microservices**

**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)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below.   + Section 2: History of Microservices   + Section 3: Problems with Monolith & SOA   + Section 4: Microservices Architecture   + Section 5: Problems Solved by Microservices   + Section 6: Designing Microservices Architecture   + Section 7: Deploying Microservices * Implement the examples along with the author. |

**Milestone 8 - Problem Solving Using Data Structures and Algorithms**

**Overview**

**Milestone 8** will be focusing on **Data Structures and Algorithms**.

Data Structures and Algorithms that are an integral part of a computer program.

**Data Structures** determine memory organization principles of data, which help in efficient storage of data in storage device

**Algorithm** is a step-wise representation of a solution to a given problem, which makes it easy to understand and is not dependent on any programming language.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Apply data structures and algorithms in real-world software development scenarios, optimizing code by selecting the most efficient data structures and algorithms for a given problem.
* Solve algorithmic problems using techniques like dynamic programming, greedy algorithms, and backtracking, identifying problem-solving patterns and applying them to different scenarios.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Data Structures and Algorithms | * 6 hrs. | * 6 hrs. |

|  |
| --- |
| **Do You Know?** |
| Data structures and algorithms form the backbone of computer science and are essential for writing efficient and optimized code. Here are some key points to know about data structures and algorithms:   1. **Foundation of Computer Science:** Data structures and algorithms are foundational concepts in computer science that enable programmers to store, retrieve, and manipulate data efficiently. 2. **Problem-Solving Tools:** They provide tools and techniques for solving complex computational problems by organizing and managing data effectively. 3. **Efficiency Matters:** Understanding data structures and algorithms is crucial for writing code that runs efficiently, as it impacts the performance and scalability of software applications. 4. **Common Data Structures:** Examples of common data structures include arrays, linked lists, stacks, queues, trees, and graphs, each with its own strengths and use cases. 5. **Algorithm Design:** Algorithms are step-by-step procedures or formulas for solving problems, and their design involves careful consideration of time and space complexities. 6. **Real-World Applications:** Data structures and algorithms are used extensively in real-world applications such as search engines, databases, operating systems, and more. 7. **Continuous Learning:** As technology evolves, new data structures and algorithms emerge, making it important for programmers to continuously learn and adapt to stay competitive. 8. **Problem-Solving Skills:** Proficiency in data structures and algorithms is often a key requirement for technical interviews and coding assessments in software engineering roles. 9. **Optimization and Performance:** Knowledge of data structures and algorithms allows developers to optimize code performance, leading to faster and more scalable applications. 10. **Collaborative Field:** Data structures and algorithms are studied and researched by computer scientists and software engineers worldwide, contributing to ongoing advancements in the field of computing. |

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

**Data Structures and Algorithms**

**Key Topics:** Introduction to Data Structures and Algorithms, Analysis of Algorithms, Data Structures

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Data Structures and Algorithms: Deep Dive Using Java](https://cognizant.udemy.com/course/data-structures-and-algorithms-deep-dive-using-java)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 1: Introduction * Section 2: Arrays and Big-O Notation * Section 4: Lists * Section 5: Stacks * Section 6: Queues * Section 7: Hashtables * Section 9: Trees * 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. |

* Array-DS
* 2D Array – DS
* left-rotation
* Sparse-Arrays
* Array Manipulation

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

**Data Structures and Algorithms**

**Key Topics:** Algorithms

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Data Structures and Algorithms: Deep Dive Using Java](https://cognizant.udemy.com/course/data-structures-and-algorithms-deep-dive-using-java)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 3: Sort Algorithms * Section 8: Searching Algorithms * 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. |

* Min-Max Sum
* Time Conversion
* Divisible Sum Pairs

**Milestone 9 - Application Debugging**

**Overview**

Milestone 8 will be focusing on **Application Debugging using Eclipse** 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 module, GenCs will be able to:

* Should be able to explain what is Debugging and why do we need it.
* Should be able to use Eclipse Debugger that helps in navigating through code to inspect the state of an app and show its execution flow.
* Should be able to employ various debugging techniques during application development and maintenance.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Application Debugging Using Eclipse | * 6 hrs. | * 6 hrs. |

|  |
| --- |
| **Do You Know?** |
| 1. **What is Application Debugging?**    * Application debugging is the process of identifying and fixing errors, bugs, or unexpected behavior in software applications. It involves tracing and analyzing the program's execution to locate the source of issues and resolve them. 2. **Tools for Application Debugging:**    * Various tools are available for application debugging, including integrated development environments (IDEs) with built-in debuggers (e.g., Visual Studio, IntelliJ IDEA), standalone debuggers (e.g., gdb for C/C++), and browser developer tools (e.g., Chrome DevTools for web applications). 3. **Common Techniques Used in Application Debugging:**    * Setting breakpoints: Pausing the program's execution at specific points to inspect its state.    * Stepping through code: Executing the program line by line to observe its behavior.    * Inspecting variables: Viewing the values of variables at different points in the program.    * Logging: Adding log messages to track the flow of the program and the values of variables.    * Using watch expressions: Monitoring specific variables or expressions during debugging. 4. **Types of Errors Addressed in Application Debugging:**    * Syntax errors: Errors in the code structure that prevent it from running.    * Logical errors: Flaws in the program's logic that lead to incorrect behavior.    * Runtime errors: Issues that occur while the program is running, such as null pointer exceptions or array out-of-bounds errors. 5. **Challenges in Application Debugging:**    * Reproducing bugs: Some bugs may be hard to reproduce consistently, making them difficult to debug.    * Debugging in production: Debugging live systems can be challenging due to the potential impact on users.    * Complex environments: Debugging in complex environments with multiple components or dependencies can be challenging. 6. **Best Practices for Application Debugging:**    * Understand the problem: Reproduce the issue and understand its symptoms before starting debugging.    * Use version control: Work on a copy of the code to avoid making irreversible changes.    * Divide and conquer: Narrow down the scope of the problem by isolating sections of code.    * Document findings: Keep track of the debugging process, including steps taken and observations made. 7. **Debugging in Different Environments:**    * Debugging may differ based on the environment, such as debugging a web application in a browser versus debugging a backend service on a server. 8. **Importance of Application Debugging:**    * Application debugging is crucial for delivering high-quality software by identifying and fixing issues that could impact functionality, performance, or security.   Debugging is an essential skill for developers, as it enables them to maintain and improve the quality of their software products. |

|  |
| --- |
| **Day 56-Afternoon, 57- Forenoon** |

**Application Debugging**

**Key Topics:** Introduction to Debugging, Eclipse Debugger, navigate through code by using the Eclipse debugger

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Eclipse Debugging Techniques and Tricks](https://cognizant.udemy.com/course/eclipse-debugging-techniques-and-tricks/learn/lecture/2386012#overview)   * Learn All Sections in the following Udemy course. * Implement the examples along with the author. |

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

* Eclipse\_Debugging

**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

**IDP – Project Activities**

|  |
| --- |
| **Day 57- Afternoon, 58, 59 - Forenoon** |

**Sprint 2 Development**

* These two days will be spent on Sprint 2 Development

**Milestone 10 - ITSM Framework**

**Overview**

**Milestone 10** 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 |

|  |
| --- |
| **Do You Know?** |
| ITIL 4, the latest version of the Information Technology Infrastructure Library (ITIL), is a framework for IT service management (ITSM) that emphasizes the integration of IT with business needs and outcomes. It provides a holistic approach to managing IT services, focusing on collaboration, flexibility, and continuous improvement.  **Key Concepts:**   1. **Service Value System (SVS):** ITIL 4 introduces the Service Value System, which describes how all the components and activities of an organization work together to enable value creation. It includes the guiding principles, governance, service value chain, practices, and continual improvement. 2. **Service Value Chain (SVC):** The Service Value Chain is a core concept in ITIL 4, representing a flexible set of interconnected activities that can be combined in various ways to create value. It consists of six activities: Plan, Improve, Engage, Design & Transition, Obtain/Build, and Deliver & Support. 3. **Four Dimensions of Service Management:** ITIL 4 introduces four dimensions—organizations and people, information and technology, partners and suppliers, and value streams and processes—that must be considered in the design and management of IT services. 4. **Guiding Principles:** ITIL 4 defines seven guiding principles that guide an organization in adopting and adapting the framework. These principles include focusing on value, starting where you are, progressing iteratively with feedback, collaborating and promoting visibility, thinking and working holistically, keeping it simple and practical, and optimizing and automating. 5. **ITIL Practices**: ITIL 4 includes 34 practices that provide organizations with a set of practical and flexible guidelines for delivering business value. These practices are organized into three categories: general management practices, service management practices, and technical management practices. 6. **ITIL Certification**: ITIL 4 offers a streamlined certification path with four certification levels: Foundation, Managing Professional, Strategic Leader, and Master. Each level is designed to provide candidates with the skills and knowledge required to implement and manage IT services effectively.   **Recent Developments:** ITIL 4 was released in 2019, marking a significant evolution from previous versions of the framework. Since its release, ITIL 4 has continued to evolve, with updates and additions to the guidance and certification scheme. The framework has gained traction globally, with organizations increasingly adopting ITIL 4 to improve their ITSM practices and align IT with business objectives.  ITIL 4 has also embraced modern practices such as Agile, DevOps, and Lean, recognizing the need for flexibility and responsiveness in today's rapidly changing business environment. This integration allows organizations to combine ITIL practices with other methodologies to create a more holistic and efficient approach to ITSM.  Overall, ITIL 4 represents a shift towards a more holistic and adaptable approach to ITSM, emphasizing the importance of collaboration, flexibility, and continuous improvement in delivering value to the business. |

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

**ITIL**

**Key Topics:** Introduction to ITIL 4, Service Management, Four Dimensions of Service Management, ITIL 4 Service Value System (SVS), ITIL 4 Service Value Chain (SVC), ITIL 4 Practices, ITIL 4 Guiding Principles, ITIL 4 Continual Improvement

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [Introduction to Service Management with ITIL 4](https://cognizant.udemy.com/course/service-management-itil-4)   * Go through all sections of this course to understand ITIL 4. * Take up the Practice Exam given as part of this course to check your understanding about ITIL 4. |

**Milestone 11 - SCM Tools**

**Overview**

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.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Understand the use of JIRA for agile project management and software development.
* Explain how ServiceNow can be used for incident management, change management, and service catalog management.

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

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * JIRA | * 4 hrs. | * N/A |
| * ServiceNow | * 4 hrs. | * N/A |

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

**ITIL, JIRA**

**Key Topics:** ITIL Quiz, Introduction to JIRA, Getting Started with JIRA, Working with Issues in JIRA, JIRA Workflows, JIRA Administration

**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. |

**Quiz - Mandatory**

Attempt the technical quiz in the Learning Path at Tekstac to check your knowledge level of ITIL.

* ITIL Quiz

|  |
| --- |
| **Day 61- Forenoon** |

**Service Now**

**Key Topics:** Introduction to ServiceNow, ServiceNow Architecture, ServiceNow Modules and Applications, ServiceNow Administration, ServiceNow IT Service Management (ITSM)

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 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 12 - Job Schedulers**

**Overview**

**Milestone 12** 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.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Explain what is a Windows Service, why we need it and what are the differences between Windows Services and Regular Applications.
* Develop and manage a Windows Service.

|  |  |  |
| --- | --- | --- |
| **Courses/Skills** | **Learning Duration** | **Practice Duration** |
| * Windows Service | * 4 hrs. | * 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 need 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 61 - Afternoon** |

**Windows Service**

**Key Topics:** Introduction to Windows Service and Job Scheduling, Java Integration with Windows Service, Scheduling Jobs in Java, Communication Between Windows Service and Java Application, Error Handling and Logging

**Continuous Learning: Technical Enablement**

Learn about Windows Service and Job Scheduling from the following:

* [Watching Directory](https://docs.oracle.com/javase/tutorial/essential/io/notification.html)
* [Watch folder/Directory in Java](https://fullstackdeveloper.guru/2020/12/23/how-to-watch-a-folder-directory-or-changes-using-java/)
* [Java Application as Windows Service](https://coretechnologies.com/products/AlwaysUp/Apps/RunJavaApplicationAsAService.html)
* [Java Mail Scheduler](https://www.youtube.com/watch?v=O0mv8NjbfT4)
* [Run Automated Task – fixed interval](https://chillyfacts.com/run-task-periodically-java-without-java-ide/)
* [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
* JOBS\_HOL\_004
* JOBS\_HOL\_005

**IDP – Project Activities**

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

**Sprint 2 Review & Demo**

* This day will be spent for Sprint 2 Review and Demo.

**Milestone 13 - Scripting & Automation**

**Overview**

**Milestone 13** will be focusing on **Python 3** 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.

**Learning Objectives**

After completing this module, 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 3 | * 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 63** |

**Python 3**

**Key Topics:** 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 64** |

**Python 3**

**Key Topics:** 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 65** |

**Python 3**

**Key Topics:** 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 to check your skill level in Python 3. You need to score 70% or higher to clear this challenge.

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

* Code Challenge - Python

**Milestone 14 - Cloud**

**Overview**

**Milestone 14** 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.

**Learning Objectives**

After completing this module, GenCs will be able to:

* Explain what is Cloud computing and its characteristics.
* Explain types of Cloud.
* Explain various Cloud Service Models and Cloud Service Providers.

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

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

**Cloud Computing Basics**

**Key Topics:** Introduction to Cloud Computing, Types of Cloud, Cloud Service Models, Cloud Service Providers

**Continuous Learning: Technical Enablement**

|  |  |
| --- | --- |
| 4FCA95A2 | [[Introduction to Cloud Computing on AWS for Beginners [2023]](https://cognizant.udemy.com/course/azure-fundamentals-lab)](https://cognizant.udemy.com/course/introduction-to-cloud-computing-on-amazon-aws-for-beginners)   * Learn the sections listed below in this Udemy course and complete the corresponding hands-on coding given below. * Section 3: IT Fundamentals * Section 4: Cloud Computing Concepts |

**Final Evaluation**

|  |
| --- |
| **Day 67, 68, 69** |

**Final Project Evaluation & Final Technical Evaluation**

* Final evaluation will be conducted on these days

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

* This day has been added to accommodate the cohort mentoring sessions.

**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 is 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.