

Digital Nurture 3.0

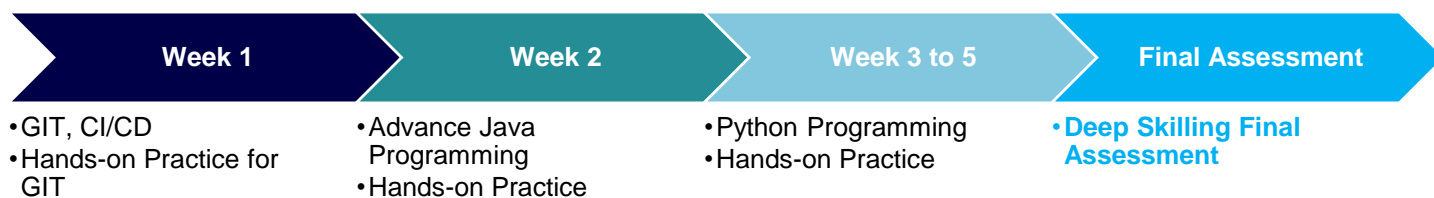
Deep Skilling Handbook – Cybersecurity

Program Highlights

- The Deep Skilling learning program runs for a period of 5 weeks. Go through the recommended self-learning resources and practice the exercises to excel in the recommended Cybersecurity modules.
- **Deep Skilling covers the below mentioned skills.**
 - GIT, CI/CD | Advanced - Collection Framework, Multithreading, Database Connectivity (JDBC), Java I/O | Python 3 (Basics and Advanced)

Recommended Program Sequence

The learning journey contains below modules, followed by Deep skilling final assessment.



Program Completion Criteria

To successfully complete this learning program, candidates must meet the following criteria:

Weekly Hands-on Exercises:

- Candidates must complete the hands-on exercises assigned for each week. These exercises are designed to reinforce the learning objectives and ensure practical understanding of the material.

Final Assessment:



- Candidates must pass the final Assessment (Python – Coding based, Other Skills - MCQ). This assessment will evaluate their comprehensive understanding and application of the concepts covered throughout the program.

Learning Approach

DN 3.0 Deep Skilling Program adopts a comprehensive and blended learning approach to ensure an engaging and effective educational experience.

The program comprises two essential learning components:

- **Self-paced learning** through open-source learning reference links.
- **Weekly SME connect** sessions conducted by experts.

 Self-Paced Learning using Open-source Reference Links	<ul style="list-style-type: none">➤ Please refer to the learning reference links provided to learn and understand the recommended concepts.➤ We expect you to dedicate 8-10 hours of focused attention weekly to your learning modules.
 SME Connect Session	<ul style="list-style-type: none">➤ We have scheduled sessions with Subject Matter Experts (SMEs) that are designed to deepen your understanding of complex topics.➤ 100% attendance is mandatory for SME Connect sessions.➤ Engage actively in doubt clarification sessions, asking questions and seeking clarification on challenging topics. Benefit from the experts' experience and insights to gain a deeper understanding of the subject matter.
Disclaimer: Cognizant does not claim ownership or responsibility for the content or any issues with the links provided, as they are merely references available on the internet. Candidates are free to leverage additional sources beyond what has been provided to enhance their skill capabilities for Deep Skilling.	

Effective Learning Strategies

Create a Study Schedule	Set Clear Goals	Stay Organized	Active Learning
<ul style="list-style-type: none">▪ Dedicate specific times each week for learning.▪ Aim for 8-10 hours of study per week.	<ul style="list-style-type: none">▪ Define what you want to achieve each week.▪ Break down tasks into manageable chunks.	<ul style="list-style-type: none">▪ Keep track of your progress and deadlines.▪ Use tools like calendars, to-do lists, and reminders.	<ul style="list-style-type: none">▪ Engage with the material through hands-on exercises.▪ Practice coding and solve problems regularly.

Duration Recommendation

Weekly Learning: 8-10 hours	Total Program Duration: 5 weeks
<ul style="list-style-type: none">▪ Allocate 1-2 hours daily, or schedule longer sessions on weekends.▪ Balance your learning with regular academic commitments.	<ul style="list-style-type: none">▪ Follow the weekly recommendations consistently.▪ Ensure to complete all exercises and review sessions.

Where to Practice?

Online Code Editors	Installed Software Tools
<ul style="list-style-type: none">▪ Use platforms below for Java and Python practice. https://www.onlinegdb.com/online_java_compiler https://www.online-java.com/ https://www.jdoodle.com/online-java-compiler https://www.onlinegdb.com/online_python_compiler	<ul style="list-style-type: none">▪ Set up Integrated Development Environments (IDEs) like IntelliJ IDEA or Eclipse, PyCharm▪ Install necessary tools and libraries as per course requirements.

Exercise Instructions

In this learning program, you will be required to complete recommended weekly exercises designed to reinforce the concepts learned during the week. These exercises are hosted on a public GitHub repository and must be downloaded and solved on a weekly basis. Follow the instructions below to ensure a smooth and productive exercise workflow:

1. Accessing the Exercises:

- Each week, exercises will be made available in our public GitHub repository.
- The repository URL is <https://github.com/trinity2040/Digital-Nurture-3.0>
- Navigate to the repository and locate the folder named **Cybersecurity**. Inside it, you will find a set of exercises for each week.

2. Downloading the Exercises:

- Download the files for the week's exercises by clicking the download button on the GitHub repository page.
- Alternatively, you can clone the repository using Git. To clone, use the command:

```
git clone https://github.com/trinity2040/Digital-Nurture-3.0.git
```

3. Solving the Exercises:

- Solve the problem statements provided in the downloaded files.
- Ensure that you understand the problem requirements and apply the concepts learned during the week.
- Take your time to think through the solutions and code them accurately.

4. Self-Evaluation:

- After completing the exercises, evaluate your solutions based on the problem criteria.
- Compare your approach with any provided hints or solutions if available.
- Reflect on any mistakes or areas where you can improve.

5. Submitting Solutions:

- Firstly, organize your solutions by week and keep them in a folder.
- Create a **public repository** in your personal GitHub account, upload your solution folder, and share the URL with the POC upon request. This will be evaluated by our SMEs.

6. Additional Support:

- If you encounter difficulties or have questions about the exercises, seek help from peers.
- Utilize the resources and links provided in this handbook for further assistance.

Module 1 - GIT, CI/CD

Week #: 1

Overview:

This module provides learners with a comprehensive understanding of essential tools and practices and focusing on version control with Git, and Continuous Integration/Continuous Deployment (CI/CD) pipelines.

Learning Objectives:

After completing this module, learners will be able to:

- Explain the importance of version control and distinguish between different types of version control systems.
- Master Git fundamentals including repository setup, branching strategies, and remote repository management.
- Implement Git workflows such as feature branching and Git Flow for collaborative development.
- Define DevOps and its principles, and recognize the benefits of adopting DevOps practices.
- Differentiate between Continuous Integration (CI) and Continuous Deployment/Delivery (CD), and articulate their roles in software development.
- Identify and evaluate popular CI/CD tools to streamline development, testing, and deployment processes effectively.

Self-Learning (Open-source links):

Key Topics	Sub-topics	Learning Reference Links
Introduction to Version Control	Definition and Purpose, Benefits of Version Control, Types of Version Control Systems	https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control
Understanding Git	Introduction to Git, Git as a Distributed Version Control System (DVCS), Working Directory, Staging Area, Repository	https://www.simplilearn.com/tutorials/git-tutorial/what-is-git
Setting Up Git	Download and Installation Steps, Configuring Basic Settings (username, email), Initializing a New Repository, Cloning an Existing Repository	https://www.simplilearn.com/tutorials/git-tutorial/git-installation-on-windows
Basic Git Commands	Initializing a New Repository, Cloning an Existing Repository; Staging Changes, Using Wildcards; Committing Changes, Adding Commit Messages, Checking the Status of Your Repository, Viewing Commit History, Options for Customizing Log Output	https://www.simplilearn.com/tutorials/git-tutorial/git-commands
Branching and Merging	Creating and Managing Branches, Switching Between Branches, Merging Branches, Handling Merge Conflicts, Feature Branching, Release Branching, Git Flow Workflow	https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging
Remote Repositories	Linking Local and Remote Repositories, Multiple Remotes, Pulling Changes from a Remote Repository, Pushing Changes to a Remote Repository, Tracking and Creating Remote Branches	https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes
Introduction to DevOps	What is DevOps?, Goals and Benefits of DevOps, Key DevOps Practices	https://www.nimblework.com/blog/introduction-to-devops/
Understanding CI/CD	What is Continuous Integration (CI)?, What is Continuous Deployment/Delivery (CD)?, Differences between CI and CD, Benefits of CI/CD	https://www.synopsys.com/glossary/what-is-cicd.html
CI/CD Tools and Platforms	Overview of Popular CI/CD Tools (e.g., Jenkins, GitHub)	https://www.synopsys.com/glossary/what-is-cicd-tools.html

Hands-On:

- Complete the respective Week 1 hands-on exercises to reinforce your learning. Refer to the [Exercise Instructions](#) section above to access the practice content.

Module 2 – Advance Java Programming**Week #: 2****Overview:**

This course covers advanced topics in core Java, including the Collection Framework, Multithreading, Database Connectivity (JDBC), Java 8 Features, and Java I/O. You will learn about sets, lists, and maps in the Collection Framework, creating and managing threads in Multithreading, connecting to databases and implementing CRUD operations in JDBC, and file handling using the java.io package in Java I/O.

Learning Objectives:

After completing this module, learners will be able to:

- Understand the different data structures and how to use them effectively in Java programming (Set, HashSet, TreeSet, Linked HashSet, List, ArrayList, Linked List, Map, HashMap, TreeMap, Linked HashMap, Hash Table).
- Learn about multithreading in Java, including creating threads, thread states, thread synchronization, inter-thread communication, and enhanced multithreading and concurrency features.
- Gain knowledge of JDBC (Java Database Connectivity) to connect and interact with databases using Java programming, including CRUD operations and data retrieval.
- Master Java I/O for file handling using the java.io package, including reading and writing files.

Key Topics	Sub-Topics	Learning Reference Links
Collection Framework	Set and HashSet, TreeSet, Linked HashSet, List interface and ArrayList, Linked List, Map and HashMap, Tree Map, Linked HashMap, Hash Table	https://www.javatpoint.com/collections-in-java https://www.geeksforgeeks.org/iterator-vs-foreach-in-java/
Multithreading	Thread and states of thread, Creating Thread, main thread, Multithreading, sleep, yield and join, Synchronization, Inter thread communication, Enhanced Multithreading, and concurrency features	https://www.geeksforgeeks.org/java-multithreading-tutorial/?ref=shm

Database Connectivity (JDBC)	Connect and maintain database. Database access using Java programming. JDBC Architecture, JDBC API and Driver Connection, JDBC API Data Retrieval using SELECT Query JDBC API CRUD Implementation and Demonstration, JDBC APPLICATION TO ACCESS DATA FROM TABLE	https://www.javatpoint.com/java-jdbc
Java I/O	File handling using java.io package. Reading and writing files.	https://www.geeksforgeeks.org/file-handling-in-java/ http://www.btechsmartclass.com/java/java-File-reading-and-writing.html

Additional Learning:

The following Udemy course is available free of charge. You can deepen your understanding of the concepts mentioned above.

Java Programming (Arrays):basics for beginners

<https://www.udemy.com/course/java-arrays/>

Java Database Connection: JDBC and MySQL

<https://www.udemy.com/course/how-to-connect-java-jdbc-to-mysql/>

Java Multithreading

<https://www.udemy.com/course/java-multithreading/>

Hands-On:

- Complete the respective Week 2 hands-on exercises to reinforce your learning. Refer to the [Exercise Instructions](#) section above to access the practice content.

Module 3 – Python Programming

Week #: 3 to 5

Overview:

This comprehensive module on Python programming can be highly beneficial for you to enter cybersecurity roles. As you are aware, Python is a popular and versatile programming language that is commonly used in cybersecurity for tasks such as data analysis, automation, and network security. By learning Python, you can develop the skills needed to write scripts and programs that can help identify vulnerabilities, analyze security data, and automate routine tasks. Python's simplicity and readability make it a great language for you to start with, allowing you to quickly grasp key concepts and begin making valuable contributions to cybersecurity projects. Overall, learning Python programming can provide you with a strong foundation in cybersecurity and help advance in your careers in the IT Cybersecurity field.

Learning Objectives:

After completing this module, learners will be able to:

- Understand the basics of Python programming language.
- Learn how to install Python on Windows, Mac, and Linux.
- Explore the various ways to run a Python program on different operating systems.
- Identify and use different data types such as strings, lists, tuples, dictionaries
- Understand the concepts of variables and their usage in Python programming.
- Perform mathematical operations using Python.
- Capture user input and control the flow of programs using various programming constructs.
- Learn the importance of white space in Python programming.
- Organize Python programs effectively.
- Define and use functions in Python.
- Utilize important built-in Python functions.
- Read from and write to files in Python.
- Understand the difference between binary and text files.
- Explore to handle errors effectively in Python programming.
- Explore various ways of getting help and finding Python documentation.
- Explore custom classes and utilize object-oriented programming concepts in Python.
- Gain proficiency in Python programming for cybersecurity roles.

Self-Learning (Open-source links):

Key Topics	Subtopics	Learning Reference Links
Python Basics	<ul style="list-style-type: none">• Installing Python• Running Python Programs• Text Editors and IDEs	https://realpython.com/installing-python/ https://realpython.com/run-python-scripts/ https://realpython.com/python-ides-code-editors-guide/
Python Fundamentals	<ul style="list-style-type: none">• Data Types• Variables• Mathematical Operations• User Input• Control Flow• Whitespace Importance• Organizing Programs	https://www.edureka.co/blog/variables-and-data-types-in-python/ https://www.geeksforgeeks.org/taking-input-in-python/ https://guicommits.com/organize-python-code-like-a-pro/
Advanced Python	<ul style="list-style-type: none">• Modules• Functions• Built-in Functions• File I/O• Error Handling	https://www.geeksforgeeks.org/python-modules/ https://www.geeksforgeeks.org/python-functions/ https://www.w3schools.com/python/python_ref_functions.asp

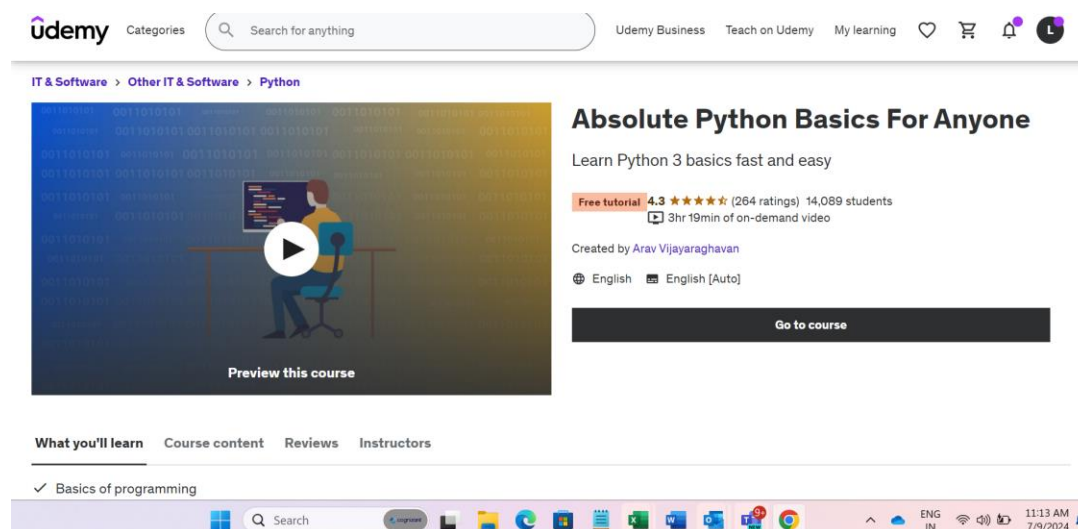
		https://www.geeksforgeeks.org/file-handling-python/ https://www.geeksforgeeks.org/python-exception-handling/
Python Collections	<ul style="list-style-type: none"> • Lists, Dictionaries, Tuples, and Sets 	https://www.pythonforall.com/python-collections https://www.tutorialsteacher.com/python/collections-module
Object-Oriented Python	<ul style="list-style-type: none"> • Classes • Custom Classes • Inheritance and Polymorphism 	https://www.geeksforgeeks.org/python-classes-and-objects/ https://learnpython.com/blog/custom-class-python/ https://nsworldinfo.medium.com/understanding-inheritance-and-polymorphism-in-python-mastering-object-oriented-programming-part-b2b33168e963

Additional Self-Learning (Optional):

The following Udemy course is available free of charge. You can deepen your understanding of the concepts mentioned above.

Absolute Python Basics For Anyone

<https://www.udemy.com/course/absolute-python-basics-for-anyone/>



The screenshot shows the Udemy website interface. At the top, there's a search bar and navigation links. Below, the course 'Absolute Python Basics For Anyone' is featured. The course description says 'Learn Python 3 basics fast and easy'. It has a 'Free tutorial' badge, a 4.3 star rating from 264 ratings, and 14,089 students. The instructor is Arav Vijayaraghavan. There are buttons for 'Go to course' and 'Preview this course'. The 'What you'll learn' section lists 'Basics of programming'.

Hands-On:

- Complete the respective Week 3, 4 & 5 hands-on exercises to reinforce your learning. Refer to the [Exercise Instructions](#) section above to access the practice content.

What's Next?

Congratulations on successfully completing the 5-week DN 3.0 Deep Skilling learning program!

As you have now finished this important phase of your learning, you will be taking Assessment (Python – Coding based, Other Skills - MCQ) to certify your skills. **This assessment will cover all the skills and topics you have learned during the past five weeks**, ensuring you have a comprehensive understanding of the material.

We wish you the best of luck on your assessment and look forward to seeing you apply your newly acquired knowledge and skills. Good luck!