Day 1 Date: 03/02/2025

Assets (General Definition)

Definition: Assets are any valuable resources owned or controlled by an individual, organization, or company that provide economic benefits.

IT Assets (Subset of Assets)

Definition: IT (Information Technology) assets are specific assets related to an organization's technology infrastructure.

IT Asset Management (ITAM)

IT Asset Management (ITAM) is the process of tracking, managing, and optimizing an organization's IT assets throughout their lifecycle. It ensures that IT resources are effectively used, maintained, and accounted for to maximize value while minimizing risks and costs.

Vulnerability in Cybersecurity

A vulnerability in cybersecurity is a weakness or flaw in a system, software, network, or hardware that can be exploited by attackers to gain unauthorized access, cause damage, or steal data.

Obsolescence in Cybersecurity

Obsolescence in cybersecurity refers to the state in which hardware, software, or security practices become outdated and no longer receive updates, patches, or support. This creates security risks because obsolete technology is more vulnerable to cyber threats.

Compliance in Cybersecurity

Cybersecurity compliance refers to the process of ensuring that an organization follows security regulations, standards, and best practices to protect sensitive data, IT systems, and networks from cyber threats. Compliance helps organizations meet legal, industry, and contractual security requirements.

Maintenance in Cybersecurity

Cybersecurity maintenance refers to the ongoing process of monitoring, updating, and improving security systems, policies, and infrastructure to protect an organization from cyber threats. It ensures that security controls remain effective, vulnerabilities are addressed, and compliance standards are met.

End of support (EOS)

The vendor no longer provides security patches or technical assistance, but the product may still function.

End of Life (EOL)

The product is completely discontinued and may no longer be available for use.

End of maintenance (EOM)

The vendor stops releasing feature updates and improvements but may still provide security patches.

Asset Hygiene in Cybersecurity

Asset hygiene in cybersecurity refers to the practice of maintaining a clean, secure, and well-managed IT environment by properly tracking, updating, and securing all digital and physical assets. Good asset hygiene reduces security risks, prevents unauthorized access, and ensures compliance with security standards.

Crown Jewels in Cybersecurity

In cybersecurity, crown jewels refer to the most valuable, sensitive, and critical assets within an organization. These are the assets that, if compromised or stolen, would cause the most significant harm to the organization's operations, reputation, financial stability, or compliance status. Protecting these "crown jewels" is a primary focus of cybersecurity efforts.

Inventory in Cybersecurity

In cybersecurity, inventory refers to the comprehensive and up-to-date list of all assets within an organization's IT environment. This includes both hardware (computers, servers, routers, etc.) and software (applications, systems, databases, etc.), as well as network components, data storage, and security tools. Proper asset inventory management is essential for securing the organization's IT infrastructure and ensuring all assets are properly protected.

NVD (National Vulnerability Database) in Cybersecurity

The National Vulnerability Database (NVD) is a publicly accessible database maintained by the National Institute of Standards and Technology (NIST). It serves as a comprehensive resource for tracking and categorizing security vulnerabilities in software and hardware products. The NVD is integral to vulnerability management, providing essential information to organizations seeking to assess and mitigate potential threats.

Patch Management in Cybersecurity

Patch management in cybersecurity is the process of identifying, acquiring, testing, deploying, and managing patches (updates) to software, systems, and applications in order to fix vulnerabilities, improve functionality, or add new features. Effective patch management is crucial for maintaining the security and integrity of IT systems, as unpatched software and hardware can expose organizations to cyber threats and attacks.

SaaS Platform

A SaaS platform is a cloud-based service model that provides access to applications through the internet. SaaS stands for Software as a Service.

Compititors

ApexaiQ

Apexa iQ is a SaaS-based, agentless continuous asset assurance platform in the cybersecurity sector. The company offers a solution for asset data management that includes deduplication, prioritization, enrichment with obsolescence, warranty, vulnerability, and compliance information, all accessible through a single dashboard. Apexa iQ primarily serves sectors such as fintech, healthcare, technology, and retail, providing tools for CISOs, CTOs, CIOs, CFOs, and CEOs to manage and secure their technology assets. It was founded in 2021 and is based in Milford, Massachusetts.

Encore.io

Encore.io specializes in cybersecurity visibility and operates within the information security sector. The company offers a platform that provides near-real-time visibility, reporting, and management of cybersecurity risks by consolidating data from various security tools and external intelligence feeds. Encore.io primarily serves sectors that require robust cybersecurity measures, such as finance, insurance, and telecommunications. It was founded in 2019 and is based in Maidenhead, England.

Quod orbis

Quod Orbis specializes in continuous controls monitoring (CCM) within the cyber security industry. The company offers a managed platform that automates the monitoring and auditing of security controls, providing visibility and compliance assurance. It was founded in 2018 and is based in London, United Kingdom. In June 2024, Quod Orbis was acquired by Deda Group.

Cloud wize

CloudWize provides cloud security and compliance solutions across various domains. The company offers a platform for threat monitoring, misconfiguration detection, and compliance risk management for cloud environments. CloudWize serves sectors that require cloud security measures, such as the technology and cybersecurity industries. It was founded in 2019 and is based in Netanya, Israel.

DAY 2 Date: 04/02/2025

Python:

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

- web development (server-side),
- software development,
- mathematics,
- system scripting.

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Execution of Python:

Python syntax can be executed by writing directly in the Command Line

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line

Python Indentation:

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Comments:

In Python, you can add comments to your code using the # symbol for single-line comments and triple quotes (''' or """) for multi-line comments or docstrings.

Variables in python:

A variable in Python is used to store data. Unlike some other languages, you don't need to declare the data type explicitly Python automatically assigns it based on the value.

Conditional Statements in Python:

Conditional statements in Python are used to execute certain blocks of code based on specific conditions. These statements help control the flow of a program, making it behave differently in different situations.

- if Statement

Executes a block of code if the condition is True.

- if-else Statement

Executes one block if the condition is True, and another if it's False

- if-elif-else Statement

Checks multiple conditions in sequence.

- Nested if Statements

You can nest if statements inside another if.

Cases used in python:

snake_case ---- Variables & Functions
PascalCase ----- Classes
camelCase ------ (Not commonly used in Python)
UPPER_CASE ----- Constants
_protected ------ Protected variables (internal use)
__private ------- Private variables (name mangling)
__dunder__ ----- Special methods

Exception Handling:

Exception handling in Python allows you to manage and handle errors that may occur during program execution, preventing crashes and improving user experience.

try ----- Defines a block to test for errors
except ----- Handles exceptions that occur in try
else ----- Runs if no exception occurs
finally ---- Executes code whether an exception occurs or not

raise ----- Manually raises an exception

Extension in python:

Python files use different extensions depending on their purpose.

.py --- Python script file

.pyc ---- Compiled Python bytecode

.ipynb ---- Jupyter Notebook file

.pyw ---- Windows Python script (no console)

.whl ---- Python package (wheel format)

Execution of python program:

Python follows this execution process:

1. Loads and reads the script.
2. Compiles it into bytecode (.pyc).
3. Executes the bytecode using the Python Virtual Machine (PVM).
4. Displays the output

OOP:

Object-Oriented Programming (OOP) is a programming paradigm that organizes code using objects and classes. Python supports OOP, allowing for reusable and structured code.

Class ----- A blueprint for creating objects

Object ------ An instance of a class

Attributes ------- Variables associated with an object

Methods ----- Functions defined in a class that operate on objects

Encapsulation ------- Hiding data inside a class to restrict direct access

Inheritance ------ Creating a new class from an existing class

Polymorphism ------ Methods that behave differently based on the object type

Abstraction ------ Hiding complex implementation details from users

List and tuples:

Python provides lists and tuples to store collections of items. Both can hold multiple data types, but they differ in mutability and performance.

A list is an ordered, mutable (changeable) collection that allows duplicates.

A tuple is an ordered, immutable (unchangeable) collection that allows duplicates.

RegEx (Regular Expressions):

Regular Expressions (RegEx) in Python are used for pattern matching and string manipulation. The re module provides functions to work with RegEx.

re.search()	Searches for a match anywhere in the string
re.match()	Matches only at the beginning of the string
re.findall()	Returns all matches as a list
re.finditer()	Returns an iterator of match objects
re.sub()	Replaces a matched pattern with another string
re.split()	Splits a string based on a pattern
re.compile()	Pre-compiles a regex pattern for efficiency

Multithreading:

Multithreading allows yo u to run multiple threads (smaller units of a process) concurrently, making your program more efficient, especially when there are tasks that are I/O-bound (e.g., reading from a file or network requests). Python's threading module provides tools for working with threads.

File Handling:

File handling in Python allows you to read, write, and manipulate files stored on your system. Python provides built-in functions and modules like open() and os to work with files.

'r' --- Read (default mode, file must exist)

'w' ----- Write (creates a new file or truncates an existing file)

'a' ----- Append (adds to the end of the file)

'b' ----- Binary mode (e.g., 'rb' or 'wb')

'x' ----- Exclusive creation (fails if the file exists)

't' ----- Text mode (default, e.g., 'rt' or 'wt')

'r+' ----- Read and write (file must exist)

'w+' Write and read (creates a new file or truncates an existing file)

Project Management:

Project management in the context of Agile and Scrum focuses on delivering value to the customer through iterative and incremental development. Unlike traditional project management methods (like Waterfall), which rely on detailed upfront planning and a linear approach, Agile and Scrum embrace flexibility, collaboration, and rapid delivery.

Agile is a mindset and a set of principles for software development (and beyond) that prioritizes:

Planning, Req Analysis, designing, implementation, testing, deployment, maintenance.

Scrum is a specific Agile framework used to manage and execute projects, especially in software development. It defines roles, ceremonies, and artifacts to organize and manage work. Scrum is widely adopted due to its simplicity and flexibility.

Testing Unit:

Testing is the process of evaluating a system or its components to ensure that it behaves as expected and meets the required specifications. In software development, testing involves running a program or application to find bugs, errors, or unexpected behavior before it is deployed or released to users.

Unit testing is the process of testing individual units or components of a software application in isolation to ensure they function correctly. A "unit" in software development typically refers to the smallest testable part of an application, such as a function, method, or class. The goal of unit testing is to validate that each part of the software performs as expected in isolation before integrating it with other components.

Coding Standards:

Coding standards are a set of guidelines, rules, and best practices that software developers follow when writing code. These standards are put in place to ensure consistency, readability, and maintainability of code across teams and projects. By adhering to these standards, developers can create code that is easier to understand, debug, and extend, as well as ensure collaboration runs smoothly between different team members.

PEP 8 stands for Python Enhancement Proposal 8, and it is the style guide for writing Python code. It provides conventions for writing clean, readable, and consistent Python code, making it

easier to collaborate with others and maintain code in large projects. PEP 8 covers various aspects of Python code, such as naming conventions, indentation, line length, spacing, and more.

PEP 8 is widely regarded as the *official* guide for Python code style and is followed by many Python developers. It is not a rule but a *recommendation*, though it is widely adopted in the Python community.

Wrapper Functions:

A wrapper function in Python is a function that is designed to "wrap" or "enclose" another function or block of code. It essentially takes a function as an argument, enhances or modifies its behavior, and then calls the original function. Wrapper functions are commonly used in Python for a variety of purposes,

Risk Management:

Risk management is the process of identifying, assessing, and prioritizing risks followed by the coordinated application of resources to minimize, monitor, and control the probability or impact of unfortunate events. It’s a proactive approach to preventing or managing risks that could potentially harm an organization or project. In business and project management, risk management is crucial for ensuring that any uncertainties or potential threats are adequately addressed to achieve desired outcomes.

Efficient Code:

Efficient code refers to code that performs its intended task in the most optimal manner, balancing speed, memory usage, and clarity. Efficient code aims to solve problems quickly, using minimal resources, and remaining easy to read and maintain.

Efficient coding is important because it can lead to better performance, lower costs, and smoother user experiences, especially in large-scale applications or systems with limited resources.

Ruff:

Ruff is a fast, feature-rich, and highly extensible Python linter designed to provide static code analysis, helping developers catch errors and maintain high-quality code. It is known for its speed, as it is implemented in Rust, a systems programming language that is known for its performance.

Ruff works similarly to other linters like Flake8 or Pylint, but it focuses on being faster and more efficient, while also offering more advanced features and better customization options. It's

designed to integrate well into various development workflows, from individual use to continuous integration/continuous deployment (CI/CD) pipelines.

Data Frame Operations:

In Python, a DataFrame is a 2-dimensional, size-mutable, and potentially heterogeneous tabular data structure. It is commonly used in pandas, a popular data analysis library, and is one of the primary data structures for working with structured data.

DataFrame operations refer to the various methods or functions that allow you to manipulate, analyze, and process the data stored in a DataFrame. These operations can include data filtering, sorting, aggregation, merging, and transforming data, among others.

Virtual Environment:

A virtual environment in Python is an isolated environment where you can install and manage Python packages without interfering with the system-wide Python installation. It allows you to create a self-contained workspace for a Python project, enabling the use of specific versions of libraries and dependencies for that project, independent of other projects or the global environment.

This isolation is particularly useful when working on multiple projects that may require different package versions or dependencies, helping you avoid conflicts between them.

SDLC Cycle:

SDLC stands for Software Development Life Cycle, which is a structured approach to software development that outlines the stages and processes involved in creating high-quality software. The SDLC cycle is essential for managing the development process systematically and ensuring the software is delivered on time, within budget, and meets user requirements.

The SDLC cycle consists of a series of phases that guide the development team through the entire process, from initial concept through to deployment and maintenance.

Version Control:

Version control is a system that allows developers to manage changes to code or documents over time. It keeps track of modifications, additions, or deletions made to files, and allows you to revert to previous versions if needed. It is essential for collaborative projects, as it helps prevent conflicts, ensures consistency, and maintains the history of changes made by each contributor.

Version control systems (VCS) help developers keep track of their work, share changes with others, and collaborate efficiently on a project. They are widely used in software development but can also be applied to any type of document or project that evolves over time.

Error Handling

Error handling is the process of responding to and managing errors that occur in a program. In any software application, errors can happen for various reasons, such as invalid input, hardware failures, or unexpected situations. Effective error handling is crucial to ensure that the program doesn't crash and provides meaningful feedback to the user.

In Python, error handling is done using the try, except, else, and finally blocks, which allow developers to catch and handle exceptions (errors) in a controlled way.

DAY 3 Date: 05/02/2025

Web Scrapping

DAY 4 Date: 06/02/2025

Rest API:

What is a REST API?
A REST API (Representational State Transfer Application Programming Interface) is a set of rules and conventions for building and interacting with web services. It allows different software applications to communicate over the internet using standard HTTP methods.

1. Client-Server Architecture
    - The client (e.g., web browser, mobile app) makes requests to the server.
    - The server processes the requests and sends back responses.
2. Statelessness
    - Each request from the client to the server must contain all the necessary information.
    - The server does not store the client's state between requests.
3. Resource-Based
    - Everything in a REST API is treated as a resource (e.g., users, products, orders).
    - Each resource is identified by a unique URL (Uniform Resource Locator).
4. Standard HTTP Methods (CRUD Operations)
    - GET → Retrieve data from the server (Read)
    - POST → Create new data on the server (Create)
    - PUT/PATCH → Update existing data (Update)

    - DELETE → Remove data from the server (Delete)
5. Use of JSON or XML
    - REST APIs commonly use JSON (JavaScript Object Notation) for data exchange because it is lightweight and easy to parse.
    - XML can also be used but is less common.
6. Stateless Communication
    - Each API request is independent, meaning that the server does not retain any session information.

Difference of JSON and XML:

JSON (JavaScript Object Notation) and XML (Extensible Markup Language) are both formats used for data exchange, but they have key differences in structure, readability, and performance.

- JSON is more compact and easier to read.
- XML is more complex with nested tags and attributes.
- JSON is lightweight and faster because it does not require closing tags.
- XML is heavier due to its tag-based structure, making it slower to parse.

Operations of REST API:

Method	Usage (CRUD Operation)	Description
GET	Read (Retrieve)	Fetches data from the server
POST	Create (Insert)	Adds a new resource to the server
PUT	Update (Modify)	Updates an entire existing resource
PATCH	Partial Update	Updates specific fields of a resource
DELETE	Remove (Delete)	Deletes a resource from the server

Status codes:

HTTP status codes are 3-digit numbers returned by a server in response to a client's request. They indicate whether the request was successful, encountered an error, or requires further action.

Code	Meaning	Description
200	OK	The requested action was successful.
201	Created	A new resource was created.
202	Accepted	The request was received, but no modification has been made yet.
204	No Content	The request was successful, but the response has no content.
400	Bad Request	The request was malformed.
401	Unauthorized	The client is not authorized to perform the requested action.
404	Not Found	The requested resource was not found.
415	Unsupported Media Type	The request data format is not supported by the server.

Code	Meaning	Description
422	Unprocessable Entity	The request data was properly formatted but contained invalid or missing data.
500	Internal Server Error	The server threw an error when processing the request.

Why 300 Status code is not used:

The 300 Multiple Choices status code is part of the 3xx redirection category and is rarely used in REST APIs.

Day 5 07/02/2025

Technical debt management:

Technical debt management is the process of identifying and reducing the technical issues that arise from poor software development practices.

Deliberate Debt – Conscious trade-offs made for faster delivery, with a plan for future refactoring.

Accidental/Unintentional Debt – Arises due to lack of knowledge, poor initial design, or evolving requirements.

Bit Rot Debt – Accumulates due to poorly maintained or outdated systems.

Environmental Debt – Results from using outdated tools, frameworks, or dependencies.

Code optimization, code quality and maintenance:

Code optimization is the process of improving the performance, efficiency, and resource utilization of a program while maintaining its functionality.

Goals of Code Optimization:

- Reduce execution time (faster performance).
- Minimize memory and CPU usage.
- Improve readability and maintainability.
- Enhance scalability.

Code quality refers to the maintainability, readability, reliability, and efficiency of software code. High-quality code is easy to understand, modify, and extend while minimizing bugs.

Key Aspects of Code Quality:

- Readability – Follows clear formatting and naming conventions.
- Maintainability – Easy to update or modify with minimal effort.
- Scalability – Can handle increased load or complexity.
- Security – Free from vulnerabilities or exploits.
- Reliability – Functions correctly under different conditions without crashing.
- Testability – Supports unit testing and integration testing.

Code maintenance refers to the process of modifying, improving, or fixing software code after its initial deployment to ensure its long-term efficiency and functionality.

Types of Code Maintenance:

1. Corrective Maintenance – Fixing bugs, errors, and security vulnerabilities.
2. Adaptive Maintenance – Updating the code to work with new environments, platforms, or APIs.
3. Perfective Maintenance – Enhancing performance, readability, or efficiency without changing functionality.
4. Preventive Maintenance – Refactoring, restructuring, and optimizing code to prevent future issues.

CI/CD development:

CI/CD stands for Continuous Integration (CI) and Continuous Deployment (CD) or Continuous Delivery (CD). It is a software development practice that automates code integration, testing, and deployment, enabling faster and more reliable software releases.

Continuous Integration means building and testing of code. Continuous Deployment means release and deploy of code.

Detect Bugs Early: Automated tests run on each code commit, catching errors before they reach production.

Improve Code Quality: Frequent integration ensures consistency and reduces merge conflicts.

Enhance Team Collaboration: Developers work on smaller, manageable code changes instead of massive updates

Continuous Delivery (CD) ensures that the software is always in a deployable state. After CI passes, the code is automatically staged in a pre-production or staging environment, where it can be manually or automatically released.

- Ensure software is always ready for deployment with minimal manual effort.
- Reduce release risks by automating the testing and staging process.
- Enable faster software updates by making deployments predictable and reliable.

Continuous Deployment takes Continuous Delivery a step further by automating the deployment process to production without manual intervention**.**

Data privacy and compliance:

Data privacy refers to the protection of personal and sensitive information from unauthorized access, use, or disclosure. It ensures that individuals have control over how their data is collected, stored, and shared.

Key Principles of Data Privacy

1. Transparency – Organizations must inform users about data collection and usage.
2. Consent – Users must give explicit permission for their data to be processed.
3. Data Minimization – Only necessary data should be collected and stored.
4. Security – Strong measures must be in place to protect data from breaches.
5. User Rights – Individuals should have access to their data and the ability to request its deletion or modification.

Data compliance refers to the process of ensuring that organizations follow laws, regulations, and industry standards for handling personal data.

Methodologies and best practices in software development:

Software development methodologies and best practices help teams build high-quality software efficiently while minimizing risks. Below is an overview of popular methodologies and industry best practices.

Different methodologies provide structured approaches to planning, designing, developing, testing, and deploying software.

- Agile Development

  Overview: Agile is an iterative and flexible approach that emphasizes customer collaboration, rapid releases, and adaptive planning.

- Waterfall Model

  Overview: A linear, sequential approach where each phase (Requirement → Design → Development → Testing → Deployment) must be completed before moving to the next.

- DevOps

  Overview: A culture and methodology that integrates development (Dev) and operations (Ops) to enable faster and more reliable software releases.

- Lean Software Development

  Overview: Inspired by lean manufacturing, it focuses on minimizing waste and maximizing efficiency.

- Spiral Model

  Overview: A risk-driven hybrid model combining iterative and Waterfall approaches.

Best practices:

- Regardless of the methodology used, the following best practices help ensure high-quality software.
- Requirement Analysis & Planning
- Clean Code and Code Quality
- Version Control & Collaboration
- Testing and Test Automation
- Continuous Integration & Continuous Deployment (CI/CD)

Networking ports and protocols:

Network Ports:

Network ports and protocols are essential for communication between devices in a network. Ports help direct data to the correct services, while protocols define rules for data transmission.

A network port is a logical endpoint in networking used to identify specific processes or services on a device.

- Ports range from 0 to 65535 and are managed by IANA (Internet Assigned Numbers Authority).
- Ports are used with TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

Network Protocol:

A **protocol** is a set of rules that define how data is transmitted over a network. Protocols operate at different layers of the **OSI (Open Systems Interconnection) model**.

Postman API:

Postman is a popular tool used for developing, testing, and managing APIs. It provides a user-friendly interface to send API requests, inspect responses, automate tests, and collaborate with teams.

What is Postman?

Postman is an API platform that allows developers to:
Send HTTP requests (GET, POST, PUT, DELETE, etc.).
Test APIs with automated assertions.
Manage API collections for different environments.
Mock APIs for frontend/backend development.
Generate API documentation automatically.

Why Use Postman?

No coding required for basic API testing.
Supports REST, SOAP, GraphQL, and WebSocket APIs.
Allows API automation with JavaScript (Postman tests).

Integrates with CI/CD pipelines (Jenkins, GitHub Actions).
Provides built-in API security and performance testing.