

Table of Contents

DAY 1	1
OSI NETWORK LAYERS AND ETHERNET	1
UNDERSTAND THE OSI MODEL AND ITS SEVEN LAYERS.	1
EXPLORE THE DATA LINK LAYER (LAYER 2) AND ETHERNET TECHNOLOGY.....	1
<i>Suggested Tutorials</i>	1
<i>Questions and Answers</i>	1
IP ADDRESSES, SUBNETTING, AND BINARY NUMBERING	1
UNDERSTAND IP ADDRESSING AND THE INTERNET PROTOCOL (LAYER 3).	1
INTRODUCE THE BINARY NUMBERING SYSTEM AND ITS RELEVANCE TO NETWORKING.	1
<i>Suggested Tutorials</i>	1
<i>Questions and Answers</i>	2
DAY 2.....	3
FOCUS ON THE TRANSPORT LAYER	3
UNDERSTAND THE CONCEPT OF ADDRESS RESOLUTION PROTOCOL (ARP):	3
LEARN ABOUT CLASSLESS INTER-DOMAIN ROUTING (CIDR)	3
EXPLORE THE TCP HANDSHAKE PROCESS.....	3
COMPARE TCP AND UDP PROTOCOLS	3
UNDERSTAND DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)	4
<i>Suggested Tutorials</i>	4
<i>Questions and Answers</i>	4
GIT FUNDAMENTALS.....	5
UNDERSTAND THE BASIC CONCEPTS OF VERSION CONTROL AND GIT	5
LEARN HOW TO SET UP GIT ON YOUR LOCAL MACHINE.	5
STATE THE STEPS OF THE FUNDAMENTAL GIT WORKFLOW.	5
BECOME PROFICIENT IN USING ESSENTIAL GIT COMMANDS.	5
<i>Suggested Tutorials</i>	5
<i>Questions and Answers</i>	5
DAY 4.....	7
ADVANCED GIT	7
GIT CONCEPTS AND WORKFLOWS	7
UNDERSTAND GIT BRANCHING STRATEGIES.	7
LEARN ABOUT GIT REBASE AND ITS APPLICATIONS.	7
GAIN FAMILIARITY WITH GIT HOOKS AND OTHER ADVANCED FEATURES.	7
<i>Suggested Tutorials</i>	7
<i>Questions and Answers</i>	7
CRYPTOGRAPHY	8
INTRODUCTION TO CRYPTOGRAPHY	8
SYMMETRIC CRYPTOGRAPHY	8
ASYMMETRIC CRYPTOGRAPHY	8
HASH FUNCTIONS AND MESSAGE AUTHENTICATION CODES (MACs)	8
CRYPTOGRAPHIC PROTOCOLS AND APPLICATIONS.....	9
<i>Suggested Tutorials</i>	9
<i>Questions and Answers</i>	9
DAY 5.....	10
PYTHON BASICS.....	10

Instructional Objectives (reStart Program)

INTRODUCTION TO PYTHON:	10
DATA TYPES AND VARIABLES	10
CONTROL FLOW AND FUNCTIONS	10
WORKING WITH LISTS AND DICTIONARIES.....	10
INTRODUCTION TO MODULES AND PACKAGES	10
INTRODUCTION TO FASTAPI CONCEPTS	10
HANDS-ON EXERCISES	11
<i>Suggested Tutorials</i>	11
<i>Questions and Answers</i>	11

Day 1

OSI Network Layers and Ethernet

Understand the OSI model and its seven layers.

- Learn the purpose of each OSI layer and how they interact.
- Recognize examples of protocols and devices associated with each layer.

Explore the data link layer (Layer 2) and Ethernet technology.

- Learn about Ethernet frames and their structure.
- Understand MAC addresses and their role in Ethernet communication.

Suggested Tutorials

- [OSI Model Explained] https://www.youtube.com/watch?v=vv4y_uOneC0
- [Ethernet Explained] <https://www.youtube.com/watch?v=AYm-4zJhEG4>

Questions and Answers

- What is the purpose of the data link layer?
- How is a MAC address different from an IP address?
- What are the seven layers of the OSI model? Provide an example protocol for each layer.

IP Addresses, Subnetting, and Binary Numbering

Understand IP addressing and the Internet Protocol (Layer 3).

- Learn about IPv4 and IPv6 addressing formats.
- Understand the concept of subnetting and subnet masks.

Introduce the binary numbering system and its relevance to networking.

- Learn how to convert decimal numbers to binary and vice versa.
- Understand how binary is used in subnetting and IP addressing

Suggested Tutorials

- [IP Addressing and Subnetting] <https://www.youtube.com/watch?v=0UVYTZeQsM4>
- Binary Numbers Explained] <https://www.youtube.com/watch?v=8afbTaA-gOQ>

Questions and Answers

- What is the difference between a public IP address and a private IP address?
- How do you calculate the number of available hosts in a subnet?
- Convert the decimal number 187 to binary.
- What is the binary representation of the subnet mask 255.255.255.0?

Day 2

Focus on the Transport layer

Assignment 1 discussion review of upcoming topics (pydantic schema classes, dependency injection theory)

Paste image install for vscode

Play with markdown

Install git and gitbash

Have a github account

Understand the concept of Address Resolution Protocol (ARP):

- Define ARP and its role in Layer 2 (Data Link Layer).
- Explain how ARP resolves IP addresses to MAC addresses.
- Discuss ARP cache and its importance in network communication.

Learn about Classless Inter-Domain Routing (CIDR)

- Define CIDR and its significance in IP address allocation.
- Explain CIDR notation and how it represents IP address ranges.
- Discuss the advantages of CIDR over traditional IP address allocation methods.

Explore the TCP handshake process

- Describe the three-way handshake mechanism used by TCP to establish a connection.
- Break down each step of the TCP handshake (SYN, SYN-ACK, ACK).
- Discuss the importance of sequence numbers and acknowledgment numbers in TCP communication.

Compare TCP and UDP protocols

- Differentiate between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).
- Compare their characteristics, including reliability, connection-oriented vs. connectionless, and overhead.
- Discuss scenarios where TCP or UDP is preferred based on application requirements.

Understand Dynamic Host Configuration Protocol (DHCP)

- Define DHCP and its role in IP address allocation and network configuration.
- Explain the DHCP lease process, including DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK.
- Discuss DHCP options and their use in configuring additional network parameters.

Suggested Tutorials

[ARP Protocol Explained] <https://www.geeksforgeeks.org/arp-protocol>

[CIDR Notation Explained] <https://www.ionos.com/digitalguide/server/know-how/cidr-classless-inter-domain-routing/>

Questions and Answers

<https://www.netometer.com/qa/arp.html>

<https://www.netometer.com/qa/cidr.html>

Day 3

Git Fundamentals

Understand the basic concepts of version control and Git

Typical Workflow:

1. Initializing a repository
2. Adding and committing changes
3. Reviewing commit history
4. Branching and merging
5. Resolving merge conflicts
6. Undoing changes
7. Collaborating with remote repositories

Learn how to set up Git on your local machine.

State the steps of the fundamental Git workflow.

Become proficient in using essential Git commands.

Essential Commands:

- ``git init``
- ``git clone``
- ``git add``
- ``git commit``
- ``git status``
- ``git log``
- ``git branch``
- ``git merge``
- ``git pull``
- ``git push``
- ``git diff``
- ``git reset``
- ``git checkout``

Suggested Tutorials

[Atlassian Git Tutorial] <https://www.atlassian.com/git/tutorials>

[Git Documentation] <https://git-scm.com/doc>

[GitHub Learning Lab] <https://github.com/apps/github-learning-lab>

Questions and Answers

1. What is version control, and why is it important?

Instructional Objectives (reStart Program)

- Version control is a system that records changes to files over time, allowing you to recall specific versions later. It's crucial for tracking project history, collaborating with others, and reverting to previous states.

2. How do you initialize a Git repository?

- You use the ``git init`` command in the directory you want to version control.

3. What command is used to stage changes for commit?

- The ``git add`` command is used to stage changes.

4. How do you create a new branch in Git?

- You use the ``git branch <branch_name>`` command to create a new branch.

5. What command is used to merge branches?

- The ``git merge <branch_name>`` command is used to merge branches.

6. How do you undo the last commit?

- You use ``git reset HEAD~1`` to undo the last commit while keeping changes staged.

7. How do you collaborate with remote repositories?

- You use ``git push`` to upload local commits to a remote repository, and ``git pull`` to fetch and merge changes from a remote repository into your local branch.

Day 4

Advanced Git

Git concepts and workflows

Typical Workflow:

1. Feature branching
2. Rebasing
3. Cherry-picking
4. Using Git hooks
5. Submodules and subtrees
6. Advanced conflict resolution

Understand Git branching strategies.

Learn about Git rebase and its applications.

Gain familiarity with Git hooks and other advanced features.

Essential Commands:

- ``git rebase``
- ``git cherry-pick``
- ``git submodule``
- ``git subtree``
- ``git bisect``
- ``git blame``
- ``git stash``
- ``git tag``
- ``git hooks``

Suggested Tutorials

[Git Branching Strategies <https://nvie.com/posts/a-successful-git-branching-model>

[Interactive Git Tutorial] <https://learngitbranching.js.org>

[Advanced Git Tutorials] <https://git-scm.com/book/en/v2>

Questions and Answers

1. What is rebasing in Git, and when would you use it?

- Rebasing is the process of moving or combining a sequence of commits to a new base commit. It's used to maintain a cleaner, linear project history and integrate changes from one branch into another.

2. How do you cherry-pick a commit?

- You use the ``git cherry-pick <commit_hash>`` command to apply a specific commit from one branch to another.

3. What are Git submodules, and how do they differ from subtrees?

- Git submodules are repositories nested within another repository, while subtrees are directories brought into a repository as a subdirectory.

4. How do you use Git hooks?

- Git hooks are scripts triggered by specific Git events. They reside in the ``.git/hooks`` directory and can be used for tasks like pre-commit validation or post-receive notifications.

5. What is Git bisect used for?

- Git bisect is a binary search tool used to find the commit that introduced a bug by systematically narrowing down the range of commits to search through.

Cryptography

Introduction to Cryptography

- Define cryptography and its importance in securing communication and data.
- Explain the basic principles of cryptography, including encryption, decryption, and key management.
- Understand the difference between symmetric and asymmetric encryption algorithms.

Symmetric Cryptography

- Learn about symmetric encryption algorithms such as DES, AES, and Blowfish.
- Understand the concepts of block ciphers and stream ciphers.
- Practice encrypting and decrypting data using symmetric encryption algorithms.

Asymmetric Cryptography

- Explore asymmetric encryption algorithms such as RSA and ECC.
- Understand the concepts of public and private keys.
- Learn about digital signatures and how they are used for authentication and non-repudiation.

Hash Functions and Message Authentication Codes (MACs)

- Define hash functions and their role in cryptography.

- Understand how hash functions are used to generate message digests and ensure data integrity.
- Learn about MACs and their use in verifying the authenticity of messages.

Cryptographic Protocols and Applications

- Explore common cryptographic protocols such as SSL/TLS, SSH, and IPsec.
- Understand how cryptographic techniques are used to secure network communication, email, and digital transactions.
- Discuss real-world applications of cryptography in areas such as online banking, e-commerce, and cybersecurity.

Suggested Tutorials

[Cryptography Tutorial] <https://www.codingdrills.com/tutorials/cyber-security/cryptography-tutorial>

[Symmetric Encryption Tutorial] <https://www.codingdrills.com/tutorial/cryptography-tutorial/symmetric-key>

[Asymmetric Encryption Tutorial] <https://www.codingdrills.com/tutorial/cryptography-tutorial/symmetric-vs-asymmetric-encryption>

[Hash Functions Tutorial] <https://www.codingdrills.com/tutorial/introduction-to-searching-algorithms/hash-functions>

[SSL/TLS Tutorial] <https://www.codingdrills.com/tutorial/network-security-tutorial/ssl-tls-configuration>

Questions and Answers

[Cryptography Basics Quiz]

<https://www.techtarget.com/searchsecurity/definition/Quiz-Cryptography>

[Hash Functions Quiz] <https://www.geeksforgeeks.org/quizzes/data-structure-gq/top-mcqs-on-hash-data-structure-with-answers/>

[Symmetric Encryption Quiz] <https://www.proprofs.com/quiz-school/story.php?title=symmetric-key-encryption>

[Asymmetric Encryption Quiz] <https://quizlet.com/gb/868926400/asymmetric-encryption-flash-cards/>

Day 5

Install python
Jupyter extension in VsCode
Learning python basics
Discussion of compiling vs interpretation

Python Basics

Introduction to Python:

- Understand the basic syntax and structure of Python.
- Learn how to use Python as an interpreted language.
- Explore Python's interactive mode and script mode.
- Familiarize with Python's indentation-based block structure.
- Managing project requirements (pip, pipenv, poetry)

Data Types and Variables:

- Learn about different data types in Python: integers, floats, strings, lists, tuples, dictionaries, etc.
- Understand variable assignment and naming conventions.
- Practice creating and manipulating variables of different data types.

Control Flow and Functions:

- Introduce conditional statements (if, elif, else) and loops (for, while).
- Explore how to define and call functions in Python.
- Discuss the importance of code organization and reusability.

Working with Lists and Dictionaries:

- Understand the concept of data structures.
- Learn how to create, access, and manipulate lists and dictionaries.
- Practice common operations on lists and dictionaries such as adding, removing, and updating elements.

Introduction to Modules and Packages:

- Understand the concept of modules and how to import them.
- Explore built-in modules and how to use them.
- Discuss the importance of code modularity and packaging for large projects.

Introduction to FastAPI Concepts:

- Briefly introduce FastAPI and its features.
- Discuss the importance of APIs and their role in web development.

Instructional Objectives (reStart Program)

- Explore basic concepts of API endpoints, request methods (GET, POST, PUT, DELETE), and request/response handling.
- Provide an overview of asynchronous programming and its benefits in building APIs with FastAPI.

Hands-On Exercises:

- Work on simple programming exercises to reinforce the concepts learned.
- Practice writing Python code to solve problems and implement basic algorithms.
- Engage with the following suggested tutorials.

Suggested Tutorials

[Python Packaging User Guide]

<https://packaging.python.org/en/latest/tutorials/managing-dependencies>

[The Python Tutorial] <https://docs.python.org/3/tutorial/index.html>

[FastAPI Tutorial] <https://www.youtube.com/watch?v=GN6lCac3OXY>

Questions and Answers

[Python Practice Exercises] <https://learnpython.com/blog/python-practice-for-beginners>