الجمهورية اللبنانية
وزارة التربية والتعليم العالي
المديرية العامة للتعليم المهني والتقني

قـرار رقــم ٢٠٢٤/٣٥٥

إعتماد المنهاج الرسمي لإختصاص " أنظمة وشبكات" المعدل
لمستوى شهادة الإمتياز الفني

إن المديرة العامة للتعليم المهني والتقني بالتكليف،
بناء على المذكرة الإدارية رقم ٦٥/م/٢٠١٩ تاريخ ٢٠١٩/٥/٢٤ (قبول طلب المدير العام بالتكليف للتعليم المهني والتقني إعفاء من تكليفه بهذه المديرية العامة وتكليف مدير المعهد الوطني للعناية التمريضية بمهام المدير العام لها)،
بناء على المرسوم رقم ٧٨٨٠ تاريخ ١٩٦٧/٧/٢٥ وتعديلاته (تنظيم حقول ومراحل وشهادات التعليم المهني والتقني)،
بناء على المرسوم رقم ٨٥٩٠ تاريخ ٢٠١٢/٨/٢ (تنظيم حقول ومراحل وشهادات التعليم المهني والتقني)،
إستناداً الى مذكرة التفاهم الموقعة بين جمعية بنات مريم أم المعونة والمديرية العامة للتعليم المهني والتقني بتاريخ ٢٠٢١/٥/١٨ (التعليم التقني والمهني لتحسين الكفاءات وفرص العمل اللائق والإزدهار للشباب المستضعفين في لبنان)
إستناداً الى الدراسة التي انجزت في مراجعة وتعديل منهاج إختصاص "أنظمة وشبكات" بين خبراء في جمعية بنات مريم أم المعونة وفريق من الأساتذة في المديرية العامة للتعليم المهني والتقني،
بناء على إقتراح رئيس المصلحة الفنية بالتكليف،

يقرر ما يأتي:

المادة الاولى: يُعتمد المنهاج الرسمي المعدل لإختصاص " **أنظمة وشبكات**" لمستوى شهادة الإمتياز الفني، في معاهد ومدارس التعليم المهني والتقني الرسمية والخاصة وفقاً للمنهاج الملحق بهذا القرار إعتباراً من العام الدراسي ٢٠٢٤-٢٠٢٥:

المادة الثانية: يُبلغ هذا القرار من يلزم، وينشر على الموقع الإلكتروني للمديرية العامة للتعليم المهني والتقني.

الدكوانه في: ٢٣ أيار ٢٠٢٤
المديرة العامة للتعليم المهني والتقني بالتكليف

د.هنادي بري

صورة طبق الأصل
مروه نابلسي

٢٠٢٤ - ٢٧٣ - ٣٥٥

Technical and Vocational Education to Improve Competences and Opportunities for Decent Work and Prosperity for Young Vulnerable People in Lebanon

AID 012590/02/0

CURRICULUM FOR TECHNICIEN SUPÉRIEUR

SUBJECT "SYSTEMS AND NETWORKS"

28 May 2024

In partnership with

DISCLAIMER

The AICS-funded project EDU TOP - Technical and Vocational Education to Improve Competences and Opportunities for Decent Work and Prosperity for Young Vulnerable People in Lebanon (AID 012590/02/0) has developed this Curriculum in the framework of its overall effort to promote a technical education system of quality, equitable and inclusive, responsive to the needs of the labor market, in favor of the most excluded communities and individuals.

This Curriculum was updated by *COMI ETS - Cooperazione per il Mondo in via di Sviluppo* and its local counterpart *FMA - Filles de Marie Auxiliatrice*, in coordination with the *DGTVET - Directorate General of Technical and Vocational Education and Training* and with the support of *Fondazione CIOFS-FP ER EMILIA ROMAGNA ETS* and *A.E.M.S. S.A.L. Advance Engineering and Management Services*.

This Curriculum was produced with the contribution of AICS - Italian Agency for Development Cooperation. Its contents are exclusive responsibility of COMI ETS and do not necessarily represent the views of the Agency.

SUBJECT	1st YEAR		2nd YEAR	
	course	lab	course	lab
General subjects				
First foreign language	60	-	-	-
second foreign language	60	-	-	-
General Mathematics	60	-	-	-
Right	60	-	60	-
Probability and statistics	60	-	-	-
Total	**300**		**60**	
Graph theory	60	-	-	-
Operating systems I-II	-	180	-	180
Computer architecture	90	-	-	-
Algorithm and data structures	60	-	-	-
Object Oriented Programming I-II	30	90	30	90
Networks I-II	90	60	90	
TP-Networks	-	-	-	90
Network security	-	-	30	90
Client-server architecture	-	-	60	-
Virtualization I-II	30	30	30	30
Cloud computing I-II	-	90	30	90
Scripting and Automation I-II	-	60	30	60
Remote maintainance I-II	-	30	-	30
End of studies' project and Internship	-	-	-	180
Total	**360**	**540**	**300**	**840**
TOTAL	**1200**		**1200**	

FIRST YEAR

SPECIALIZATION SUBJECTS

1. GRAPH THEORY
2. OPERATING SYSTEMS I
3. COMPUTER ARCHITECTURE
4. ALGORITHM AND DATA STRUCTURES
5. OBJECT ORIENTED PROGRAMMING I
6. NETWORKS I
7. VIRTUALIZATION I
8. CLOUD COMPUTING I
9. SCRIPTING AND AUTOMATION I
10. REMOTE MAINTAINANCE I

GRAPH THEORY
(60 PERIODS)

OBJECTIVE

This subject aims to develop foundational concepts and tools in discrete mathematics and graph theory, alongside their algorithmic applications.

Program detrails learns to understand the graph algorithms applicable to computer networks, minimum cost calculation problems, optimal pathfinding, and scheduling methods (e.g., project management).

CHAPTER 1 (8 Periods)

Introduction to Graph Theory

1.1 Definitions and terminology
- 1.1.1 Simple directed graphs: vertices, arcs, following, precedents of a vertex, paths, circuits.
- 1.1.2 Simple undirected graphs
- 1.1.3 Multi-graphs
- 1.1.4 Subgraphs and partial graphs

1.2 Problems in graph theory
- 1.2.1 Routing issues
- 1.2.2 Search for remarkable sets of vertices
- 1.2.3 Tree problems
- 1.2.4 Graph traversal methods

CHAPTER 2 (10 Periods)

Paths in Graphs

2.1 Introduction

2.2 Matrix algorithms

2.3 Best paths from a fixed vertex
- 2.3.1 Dijkstra's algorithm

2.4 - Shortest or longest paths from a fixed vertex (for graphs without circuits)
- 2.4.1 Bellman and Ford algorithm

2.5 Shortest or longest paths from a fixed vertex, in the case of lengths of arbitrary sign for graphs without circuits.

2.6 Connectedness and strong connectedness

CHAPTER 3 (10 Periods)

Scheduling and Flows

3.1 Scheduling issues
- 3.1.1 Organization of tasks, PERT method

3.2 Flows in a graph
- 3.2.1 Sets of arcs entering or leaving a set of vertices
- 3.2.2 Determination of maximum flow value: Ford and Fulkerson algorithm
- 3.2.3 Bounded flows:Assignment problems, Transportation issues

CHAPTER 4 (4 Periods)

Building Cliques

4.1 Step-by-step method

CHAPTER 5 (4 Periods)

Stable Sets of Vertices of a Graph

5.1 Internally and externally stable sets
5.2 Kernels of a graph
5.3 Minimum coverage problem
5.4 Exercises and algorithms

CHAPTER 6 (16 Periods)

Linear Programming

6.1 Notion of a project
6.2 Concept of an activity
6.3 Representation of a project by a PERT network
6.4 Critical Activity
6.5 Critical path and non-critical path
6.6 Evaluation and interpretation of margins
6.7 Improvement of a project at the lowest cost

CHAPTER 7 (8 Periods)

Cartesian Paths

7.1 Gray's Code
- 7.1.1 Definitions and notations
- 7.1.2 Operations on totally ordered sets
- 7.1.3 Gray order, Gray product
- 7.1.4 Generalized Gray product
  - Rank of an element in the generalized product
  - Sequential construction of Gray's order

OPERATING SYSTEM I
(180 PERIODS OF PRACTICAL EXERCISES)

OBJECTIVE
In general, the main objective of the course is to equip students with the skills and knowledge necessary to effectively install, configure, and manage operating systems across a variety of hardware platforms (Windows and Linux), and to address the common challenges associated with such processes.

CHAPTER 1 (20 Periods of practical exercises)

Operating systems installation

1.1 Introduction to Operating Systems
- 1.1.1 Definition of an operating system
- 1.1.2 Types of operating systems (Windows, Linux, macOS)
- 1.1.3 System requirements for installation
- 1.1.4 Partitioning and formatting concepts

1.2 Preparation for Installation
- 1.2.1 Creating installation media (USB, DVD)
- 1.2.2 Downloading Windows and Linux ISO images
- 1.2.3 Verifying ISO file integrity

CHAPTER 2 (60 Periods of practical exercises)

Windows infrastructure.

2.1 Introduction to Microsoft Windows
- 2.1.1 Overview of Windows operating systems
- 2.1.2 Installation and configuration of Windows
- 2.1.3 Basic concepts of the Windows operating system
- 2.1.4 Desktop and Start menu usage

2.2 File and Folder Management
- 2.2.1 File Explorer navigation
- 2.2.2 File and folder management: copy, move, rename, delete
- 2.2.3 Using libraries and favorites
- 2.2.4 File permissions and security management

2.3 Customization and Configuration of Windows
- 2.3.1 Desktop and taskbar customization
- 2.3.2 System settings configuration
- 2.3.3 User account management and privacy settings
- 2.3.4 Notification Center and notification settings usage

2.4 Introduction to Networking in Windows
- 2.4.1 Basic concepts of networks and Internet connections
- 2.4.2 Configuration of wired and wireless network connections
- 2.4.3 Basic network troubleshooting
- 2.4.4 Sharing network resources: files and printers

2.5 Security and Protection in Windows
- 2.5.1 User account and password management
- 2.5.2 Windows firewall configuration
- 2.5.3 Using Windows Defender and other security tools
- 2.5.4 Data backup and restore with Windows Backup

2.6 System Maintenance and Optimization
- 2.6.1 Using system tools: Task Manager, Resource Monitor

2.6.2 Disk cleanup and disk defragmentation
2.6.3 Printer and hardware device management
2.6.4 System and application update and maintenance

2.7 Introduction to PowerShell
2.7.1 Overview of PowerShell and basic concepts
2.7.2 Basic command usage in PowerShell (Get-Command, Get-Help, Get-Process, Stop-Process, Start-Process, Get-Service, Start-Service, Stop-Service, Restart-Service, Get-EventLog, New-EventLog, Clear-EventLog, Set-Location, New-Item,Remove-Item, Copy-Item, Move-Item, Rename-Item, Get-Content, Set-Content, Test-Connection, Start-Sleep….)
2.7.3 Basic scripting with PowerShell
2.7.4 Automating system tasks with PowerShell

CHAPTER 3 (60 Periods of practical exercises)

Linux infrastructure

3.1 UNIX vs. LINUX system
3.1.1 History, Main characteristics, Architecture
3.1.2 Unix/Linux work session: Login, Logout

3.2 What is a process in Unix?

3.3 File system
3.3.1 Concept
3.3.2 Internal structure

3.4 Directories
3.4.1 The / directory
3.4.2 Standard Unix directories (/etc, /bin, /usr, /dev, /home,/root, /lib, /mnt, /sbin, /tmp)
3.4.3 The base directory
3.4.4 The current directory
3.4.5 Paths: absolute, relative

3.5 Files
3.5.1 Names: rules, expansion, multiple dots
3.5.2 Hidden files
3.5.3 Types: ordinary, directory, block, character, link, pipe, socket

3.6 Orders
3.6.1 General syntax
3.6.2 Execute a command
3.6.3 Executing multiple commands on the same line (;)

3.7 The two modes
3.7.1 The command mode
3.7.2 Entry mode

3.8 File navigation: 0, $, b, w, [n]G, h, j, k, l

3.9 Input mode commands, mainly:
3.9.1 Insert: I, i, a, A, o, O
3.9.2 Delete: [n]x, X, dd, d0, d$, db, dw
3.9.3 Copy and Paste: n[yy], p, P
3.9.4 Replacement: r, R, cc, cb, cw
3.9.5 Undo changes made: u, U
3.9.6 Repetition of the last executed command: .
3.9.7 Show status line: Ctrl+G

3.10 Command mode commands, mainly:
3.10.1 Save changes: :w, :w!
3.10.2 Output with or without backup: :q, :q!, :wq, :x

3.10.3 Inserting a file: :r
3.10.4 Search string: /, ?, n, N
3.10.5 Substitute string in text: :[n]s, :%s, g (global)
3.10.6 Execute a shell command: :!
3.11 Set editor with (:set): all, number, ignorecase, nonu, noic
3.12 Configure the editor with the file (.exrc)
3.13 Essential commands
3.13.1 Show date (date):-s, +%d, m, Y, I, M, S, x, X
3.13.2 List (ls): -a, -d, -F, -h, -i, -l, -r, -R, -s, -S, -t, -u
3.13.3 Clear screen (clear)
3.13.4 Concatenate and Display files (cat): -b, -n, -s, -T, -E, >, >>
3.13.5 Show logged in users (who): -b, -H, -q, -T, -u, ami, ami i
3.13.6 Show current directory (pwd)
3.13.7 Create a directory (mkdir): -p, -v
3.13.8 Change directory (cd)
3.13.9 Copy file/directory (cp): -b, -i, -f, -r, -u, -v
3.13.10 Move file/directory (mv): -b, -i, -f, -u, -v
3.13.11 Delete file/directory (rm): -i, -f, -r, -v
3.13.12 Delete directory (rmdir): -p, -v
3.13.13 Display free disk space (df): -h, -i, -k, -m
3.13.14 Show disk space used (from): -a, -c, -h, -k, -m, -s
3.13.15 Display the difference between 2 files (diff):-B, -b,-i,-q, -s, -w
3.13.16 Display the difference between 2 files (cmp):-s
3.13.17 Search for a string in multiple files (grep): -vs, -I, -not, -v, -R, -h, -H, -L, -l, -w, -x, ^, $
3.13.18 Search for a file in different directories (find): -name, -iname, -empty, -size, -type, -user, -uid, -group, -gid, -mtime, -atime, -ctime, -mmin, -amin, -cmin, -newer, -a, -o, -print
3.13.19 Sort file contents (sort): -b, -f, -o, -r, -u
3.13.20 Link logically and physically (ln): -b, -i, -s, -v
3.13.21 Run a command periodically (crontab): -e, -i,-l, -r, -u
3.13.22 Show process statuses (ps): -a, -A, -t, -u
3.13.23 Destroy a process (kill): -9
3.13.24 Copy to media (tar): -c, -t, -x, -f, -v, -z, -C
3.13.25 init
3.13.26 Runlevels
3.13.27 The /etc/inittab file
3.13.28 Start the system (reboot)
3.13.29 Shutdown: -c, -h, -k, -r, -t, -f, -F
3.13.30 touch
3.13.31 head: -n, -c, -q, -v
3.13.32 tail: -n, -c, -q, -v
3.13.33 more: space, enter, b, q
3.13.34 less: space, enter, b, q
3.13.35 wc: -l, -c, -w, -L
3.13.36 cal: indicated month, year
3.13.37 aliases
3.13.38 unalias: -a
3.14 Online help (man)
3.15 Metacharacters: ?, *, […], [!...]
3.16 Redirection (standard output, standard input, standard error): >, >>, <, 2>
3.17 The tubes (pipes): |
3.18 Concept of a user account

3.18.1 The predefined super user
3.18.2 Create a user (useradd): -d, -g, -G, -m, -c, -p, -s, -u, -o,-D
3.18.3 Password: passwd
3.18.4 Delete a user (userdel): -r
3.18.5 Modifying a user (usermod): -d, -g, -G, -c, -p, -s, -u, -e,-l, -L, -U
3.18.6 The home directory ($HOME)
3.18.7 The /etc/passwd file
3.18.8 The /etc/shadow file
3.18.9 The /etc/profile file
3.18.10 Files: .profile, .logout, .history
3.18.11 The command: su
3.18.12 Commands: id, groups
3.19 Notion of user group
3.19.1 Predefined groups
3.19.2 Create a group (groupadd): -g, -o, -r
3.19.3 Delete a group (groupdel)
3.19.4 Modifying a group (groupmod): -n, -g, -o
3.19.5 The /etc/group file
3.20 Access rights to a file or directory
3.20.1 User classes
3.20.2 Signs: rwx
3.20.3 chmod: -R, -v, (Methods: numeric, symbolic)
3.20.4 File access rights
3.20.5 Directory permissions
3.20.6 Change owner (chown): -R, -v
3.20.7 Change owning group (chgrp): -R, -v
3.20.8 Set default rights (umask): -S (Methods: numeric, symbolic)
3.21 Print a file (lpr): -P, -#, -T
3.22 Show print queue (lpq): -P, -a
3.23 Cancel printing a file (lprm): -P, -
3.24 Get printing statistics (lpstat): -a, -l, -u

CHAPTER 4 (40 Periods of practical exercises)

Post-Installation Configuration and troubleshooting

4.1 Networks and connections
4.2 Installing additional software
4.3 Configuring user accounts and passwords
4.4 Optimizing system settings
4.5 Troubleshooting and Q&A
4.5.1 Common installation issues and solutions
4.5.2 Diagnostic tools and problem-solving techniques
4.5.3 Student questions and answers
4.5.4 Tips for long-term operating system maintenance

COMPUTER ARCHITECTURE
(90 PERIODS)

OBJECTIVE

This course represents an introduction to the hardware architecture of computer systems. Students study the fundamentals of architecture and the Pipeline of Architecture.

CHAPTER 1 (4 Periods)

Structure of Computers

1.1 Concept of hardware and software
1.2 Processors
- 1.2.1 Organization of central processing units.
- 1.2.2 The execution steps of an instruction.

1.3 Memories
- 1.3.1 Bits.
- 1.3.2 Memory addresses.
- 1.3.3 Byte ordering.
- 1.3.4 Error correcting codes
- 1.3.5 Principle and use of cache memories.
- 1.3.6 Memory type
- 1.3.7 Hierarchy of memories.
- 1.3.8 Secondary memories
  - 1.3.8.1 Hard Disk Drive (HDD)
  - 1.3.8.2 Solid State Drive (SSD)
  - 1.3.8.3 Solid State Hybrid Drive (SSHD)
  - 1.3.8.4 USB Flash Drives
  - 1.3.8.5 Memory Cards

1.4 Inputs / outputs

CHAPTER 2 (16 Periods)

Number Representation

2.1 Finite precision numbers.
2.2 Conversion from one base to another (base 2, base 16)
2.3 Negative numbers
2.4 Binary arithmetic
2.5 Floating point number.

CHAPTER 3 (16 Periods)

Logic circuits

3.1 Boolean algebra.
3.2 Logic gates.
3.3 Realization of boolean functions.
3.4 Equivalence relation of circuits.
3.5 Combinatorial circuits.
- 3.5.1 Multiplexer
- 3.5.2 decoder
- 3.5.3 Comparator.

3.6 Arithmetic circuits
- 3.6.1 Adder

3.6.2 Arithmetic and logic unit
3.7 clocks
3.8 Sequential Circuits
3.8.1 The scales
3.8.2 The registers
3.8.3 Counters and down counters.
3.8.4 Internal organization of a memory
3.8.5 RAMs and ROMs (characteristics and types)

CHAPTER 4 (5 Periods)

Microprocessors and buses

4.1 The typical microprocessor
4.2 The buses
4.3 Example bus
4.4 Interface technology

CHAPTER 5 (5 Periods)

Advanced Architecture

5.1 RISC machines (evolution)
5.1.1 From CISC to RISC
5.2 Design principle of RISC machines
5.3 Use of the register
5.4 Instruction execution in parallel
5.4.1 The Pipeline
5.5 Vector processor
5.6 RISC architecture example
5.6.1 Data type.
5.6.2 Formatinstruction.

CHAPTER 6 (20 Periods)

Pipeline Architecture

6.1 Definition of a pipeline
6.2 Execution cycles of an instruction in a machine without a pipeline
6.3 Execution in a pipeline machine
6.3.1 Concept of pipeline stages
6.3.2 The pipeline registers
6.3.3 The basic performance of the pipeline.
6.3.4 Pipeline Performance Reducing Effects
6.3.5 Data hazards
6.3.6 Control hazards
6.4 Exception handling in a pipeline machine
6.4.1 Overflow, programmer interrupt, breakpoint, page fault)
6.5 Floating pipeline
6.5.1 Floating adder, divider and multiplier

CHAPTER 7 (10 Periods)

Exploiting the pipeline architecture

7.1 Parallelism of a sequence of instructions
7.2 Simple pipeline scheduling and loop unwinding
7.3 dependencies
- 7.3.1 Data dependency
- 7.3.2 Name dependency
- 7.3.3 Control dependency
- 7.3.4 Examples and exercises

CHAPTER 8 (14 Periods)

Multiprocessors

8.1 Introduction
8.2 A shared memory multiprocessor
8.3 A distributed memory multiprocessor
8.4 Cache consistency problem for multiprocessors
8.5 Cache consistency issues
- 8.5.1 Write-invalidate protocol
- 8.5.2 Protocol for updating writes or broadcasting writes

ALGORITHM AND DATA STRUCTURES
(60 PERIODS)

OBJECTIVE

Description of the material

An algorithm is a sequence of instructions, which once executed correctly, leads to a given result.

Learning algorithms means learning to handle the logical structure of a computer program. This dimension is present regardless of the programming language; but when you program in a language (in Java, Python, etc.) you also have to grapple with problems of syntax, or types of instructions, specific to this language. Learning algorithms separately means sorting out the difficulties in order to better overcome them.

Objective of the subject

At the end of this subject, the student should be able to:

- Know how to break down a problem to better solve it.
- To be able to go from a problem to its algorithmic solution.
- Gradually become aware of the principles of research, use and complexity of algorithms;
- Acquire the fundamental principles that allow to analyze and program concrete problems;
- Learn the structured design process, learn how to write a program in algorithmic language in order to translate it into a programming language;
- Adapt an algorithm to different data structures.

CHAPTER 1 (2 Periods)

General introduction to algorithms and programming

1.1 Definition and basic concepts.
1.2 The concept of coding and instruction.
1.3 The notion of variable (mutable objects and basic affection).
1.4 From algorithm to program (specification, program).
1.5 The basic types.
1.6 Operators (logical, arithmetic, relationship, etc.).
1.7 Expressions (logical, arithmetic, etc.).
1.8 Parameters.
1.9 Examples and applications.

CHAPTER 2 (3 Periods)

Structure of the algorithm

2.1 Sequences.
2.2 Alternativity (if … then … otherwise)
2.3 Selection and choice (cases)
2.4 Iterativity (while, for, repeat, recursion, …)
2.5 Examples and applied exercises.

CHAPTER 3 (2 Periods)

Nested structures

3.1 Nested alternatives (if…then…if…then…else…else,).
3.2 Nested loops.
3.3 Complex structures:
  3.3.1 decomposition,
  3.3.2 call.
3.4 Examples and applied exercises.

CHAPTER 4 (10 Periods)

Tables

4.1 One-dimensional arrays
  4.1.1 Presentation in memory and vocabulary.
  4.1.2 Browse and search.
  4.1.3 Insertion and deletion.
  4.1.4 Rotation and Sorting.
  4.1.5 Examples and applied exercises.
4.2 Two-dimensional arrays
  4.2.1 Presentation in memory.
  4.2.2 Browse and search.
  4.2.3 Insertion and deletion.
  4.2.4 Transfer, Interchange and Triage.
  4.2.5 Examples and applied exercises.

CHAPTER 5 (3 Periods)

Procedures and functions

5.1 Concepts and introduction.
5.2 Principles and methods:
  5.2.1 mechanism,
  5.2.2 operation,
  5.2.3 setting.
5.3 Iterative and recursive methods.
5.4 Application exercises.

CHAPTER 6 (5 Periods)

Character strings

6.1 Notions: chains.
6.2 Handling exercises: Sorting a sequence of names, conversion, selection, search, concatenation, insertion, deletion, etc.
6.3 Using arrays with strings.
6.4 String functions.

CHAPTER 7(5 Periods)

Introduction to sequential files

7.1 Notions of file and access primitives.
7.2 Exercises dealing with a single file: manipulations, search, access, sorting, etc.
7.3 Exercises dealing with several files: copy, concatenation, splitting, merging, etc.

CHAPTER 8 (5 Periods)

Introduction to Lists

8.1 Notions: cell and pointer, dynamic memory management, free lists.

8.2 Linked lists: application exercises: manipulation, insertion, deletion, search, copy, sort, merge, concatenation, splitting, etc. (iterative and recursive methods).

8.3 Queues (FIFO): application exercises: contiguous representation (vector), chained representation (manipulation);

8.4 Stacks (LIFO): application exercises: contiguous representation (vector), chained representation, processing of expressions

8.5 Circular Lists and Bidirectional lists: insertion, deletion, search, sorting, merging, concatenation

CHAPTER 9 (5 Periods)

Introduction to trees

9.1 Notions: definition, representations (binary, n-ary, degenerate, B+, etc.).

9.2 Representation of arithmetic expressions.

9.3 Ordered binary tree: application exercises: search, insertion, deletion, sorting of a vector.

9.4 Contiguous representation (vector) of a binary tree

OBJECT ORIENTED PROGRAMMING I
(120 PERIODS OF WHICH 90 ARE PRACTICAL EXERCISES)

CHAPTER 1 (6 Periods of which 4 are practical exercises)

Introduction to Java

1.1 Introduction to Java and its role in modern development.
1.2 Overview of JDK, Javac, Java, and IDEs.
1.3 Basic structure of a Java program
1.4 Introduction to Java applications:
  1.4.1 Variables
  1.4.2 Blocks, statements and Expressions
  1.4.3 The Scanner class
  1.4.4 Comments
1.5 Suggested exercises:
  1.5.1 Setting up Java and IDE.
  1.5.2 Writing basic Java programs focusing on input/output and simple calculations through the command line.

CHAPTER 2 (16 Periods of which 12 are practical exercises)

Java Fundamentals

2.1 Java primitive types.
2.2 Operators: binary, unary, relational, logical, conditional, assignment and rules of precedence.
2.3 Control flow statements (if, switch, loops).
2.4 The Math class and its methods (incl. pow, sqrt, abs, min, max, round, random).
2.5 Introduction to error handling.
2.6 Suggested exercises:
  2.6.1 Exercises on using different types, operators, and control statements.
  2.6.2 Building applications that utilize Math class methods for calculations.

CHAPTER 3 (24 Periods of which 18 are practical exercises)

Object-Oriented Programming in Java

3.1 Classes, objects, methods, and constructors.
3.2 Object-oriented principles: encapsulation, inheritance, and polymorphism.
3.3 Using the this keyword.
3.4 Access modifiers and method overloading.
3.5 Suggested exercises:
  3.5.1 Designing and implementing classes.
  3.5.2 Working with objects, using this, and method overloading.
  3.5.3 Simple projects to model real-world scenarios using OOP principles.

CHAPTER 4 (18 Periods of which 14 are practical exercises)

Data Structures in Java (Arrays & Collections)

4.1 Arrays and array manipulation.
4.2 Introduction to Java Collections Framework, focusing on ArrayList and Vector.
4.3 Practice:
4.4 Array exercises: initialization, copying, and multidimensional arrays.
4.5 Collections exercises: using ArrayList and Vector for dynamic data handling.

CHAPTER 5 (12 Periods of which 9 are practical exercises)

Strings and Regular Expressions

5.1 Working with the String class and its methods.
5.2 Introduction to regular expressions in Java.
5.3 Suggested exercises:
- 5.3.1 String manipulation exercises.
- 5.3.2 Practical applications of regular expressions for pattern matching.

CHAPTER 6 (16 Periods of which 12 are practical exercises)

OOP and Design Patterns

6.1 Abstract classes, interfaces.
6.2 Introduction to design patterns and their application in Java:
- 6.2.1 Creational: Singleton, Factory
- 6.2.2 Structural: Adapter, Decorator
- 6.2.3 Behavioral: Observer, Strategy

6.3 Suggested exercises:
- 6.3.1 Implementing interfaces and abstract classes.
- 6.3.2 Simple design pattern implementations in Java.

CHAPTER 7 (10 Periods of which 7 are practical exercises)

Java Input/Output (I/O) Networking

7.1 Basics of Java I/O, including streams and file handling.
7.2 Introduction to Java I/O Streams
7.3 Introduction to the java.io package
7.4 File Handling:
- 7.4.1 The File class
- 7.4.2 Using FileInputStream, FileOutputStream, FileReader, and FileWriter

7.5 Buffered Streams (BufferedReader, BufferedWriter, BufferedInputStream, BufferedOutputStream).
7.6 Serialization: ObjectInputStream and ObjectOutputStream.
7.7 Suggested exercises:
- 7.7.1 Exercises on file I/O operations.

CHAPTER 8 (10 Periods of which 7 are practical exercises)

Networking

8.1 Basics of Networking in Java: the java.net package and InetAddress class.
8.2 Using Sockets and ServerSockets
8.3 Introduction to UDP protocol and its application in Java.
8.4 URLs and HTTP: accessing and retrieving web resources.
8.5 Suggested exercises:
- 8.5.1 Creating a basic TCP/IP Client-Server Application sending and receiving small pieces of data.
- 8.5.2 Writing a program that retrieves data from a specific URL and processes it, such as downloading a web page or consuming a RESTful web service.

CHAPTER 9 (8 Periods of practical exercises)

Final project

Students will work individually or in small teams to develop a final project that demonstrates their understanding of Java fundamentals.

Possible Simple Applications students could develop include:

- **Personal Finance Manager**: A desktop or web application that allows users to track their expenses, incomes, and financial goals. This project can utilize arrays and collections to manage financial records, I/O to save data persistently, and the GUI for user interaction.
- **Educational Quiz Application**: A quiz application that supports multiple-choice questions on various topics. It can feature a GUI for desktop or web interfaces, utilize collections to manage questions and answers, and apply I/O operations for reading quiz content from files.
- **Weather Application**: A simple application that fetches weather data from a public API and displays it to the user. This project can introduce API interaction over the network, JSON data parsing, and GUI development for displaying weather information.

NETWORKS I

(150 PERIODS OF WHICH 60 ARE PRACTICAL EXERCISES)

CHAPTER 1 (10 Periods)

Introduction to Networking

1.1 Summary of computer networks.
1.2 Identify Terms: Servers, Clients, Hosts, etc...
1.3 Differentiate network types: LAN, WAN, MAN.
1.4 Recognize different connectors model (RJ45, Fiber Optic).

CHAPTER 2 (16 Periods)

Network Models and Organisms

2.1 Definition and Role
    2.1.1 Different types of architectural models (OSI, DSA, SNA, etc.)
2.2 Recognize 7 layers of OSI model
    2.2.1 Definition
    2.2.2 Diagrams
    2.2.3 Roles

CHAPTER 3 (14 Periods of which 5 are practical exercises)

Transmission System

3.1 Representing Information:
    3.1.1 Defining information and data.
    3.1.2 The significance of encoding and representation.
    3.1.3 Concepts of Analog/Digital signals, ASCII, and Unicode.
3.2 Classification of Data Transmission elements:
    3.2.1 Exchange control modes.
    3.2.2 Link modes and control (e.g., point-to-point, multipoint).
    3.2.3 Physical parameters (Parallel, serial, synchronous, asynchronous).
3.3 Transmission Media and Techniques:
    3.3.1 Characteristics of transmission media.
    3.3.2 Guided (e.g., coaxial cable, fiber optic) and unguided media (e.g., microwave links).
    3.3.3 Baseband vs. Broadband transmission.
    3.3.4 DTE/DCE interfaces.

CHAPTER 4 (5 Periods of which 5 are practical exercises)

Safety Considerations, equipment and Tools. Equipment handling, personal protective gear.

4.1 Lab Materials:
- 4.1.1 Cables (Types and categories).
- 4.1.2 Cabling tools (RJ45, Sleeving, Combs etc.…).
- 4.1.3 Cabinets.
- 4.1.4 Patch Panels.
- 4.1.5 Ethernet Hubs.
- 4.1.6 LAN / WAN / WLAN routers.

4.2 Measuring equipment:
- 4.2.1 Reflectometer to test fiber cables.
- 4.2.2 Category 7 cable scanner (bidirectional tests).
- 4.2.3 Digital Oscilloscope with the FFT function.
- 4.2.4 LAN / WAN Protocol Analyzer.
- 4.2.5 Proxy server for internet Access.
- 4.2.6 Network Printer.
- 4.2.7 Internet connection: WIFI, and Cable.
- 4.2.8 ISDN adapter Card (Integrated Services Digital Network).

4.3 Software Tools
- 4.3.1 Operating systems
- 4.3.2 Network antivirus software
- 4.3.3 File compression software
- 4.3.4 Multiboot boot software

4.4 Network specific software
- 4.4.1 Remote maintenance and control tool: "Remote desktop, VNC, etc."
- 4.4.2 Network element monitoring tools: "Openview, WhatsUp Gold, Ciscoview, etc.)
- 4.4.3 Network Plan Software (Microsoft Visio or else)
- 4.4.4 Internet software (FTP clients & server, email client & server, videoco ferencing, etc.).

CHAPTER 5 (30 Periods of which 8 are practical exercises)

Local Network Characteristics and Architectures

5.1 Definition and Architectures
- 5.5.1 Flow of information conveyed
- 5.5.2 Network Resource sharing
- 5.5.3 Multiplexing (TDMA, FDM, WDM)

5.2 Advantages and Disadvantages

5.3 Network Topologies and Criteria (Tree, Bus, Ring, Star, etc…)

CHAPTER 6 (16 Periods of which 6 are practical exercises)

Networking Protocols

6.1 Definition and Usage TCP/IP

6.2 IP and TCP (Layers)

6.3 Explain and Differentiate IPv4 and IPv6.

6.4 Differentiate Private and Public IP.

6.5 Classes of IP addresses (Class A, Class B, etc…).

6.6 Understanding Domain Name System (DNS).

6.7 Dynamic Host Configuration Protocol (DHCP).

CHAPTER 7 (12 Periods of which 6 are practical exercises)

Wide Network Characteristics and Architectures

7.1 Definition and Architectures
- 7.1.1 Data Transmission Protocol (ATM, Frame Relay, etc…)
- 7.1.2 Bond Types (ISDN Link, Satellite Link)

7.2 Advantages and Disadvantages

7.3 Network Topologies and Criteria (Point to Point, Star, etc…)

CHAPTER 8 (15 Periods of which 10 are practical exercises)

Network Components and Devices

8.1 Definition, Differences, Usage, Types and Characteristics (Routers, Switches, Access Points, Hubs, Repeaters, etc.…)

8.2 Using a mix of devices and protocols in practical Lab (Servers, Computers, Routers, Switches, Printers, DHCP, TCP/IP, etc…)

CHAPTER 9 (8 Periods of which 4 are practical exercises)

Fundamentals Network Concepts for server

9.1 Physical Servers vs. Virtual servers

9.2 Workgroup vs. Domain.

9.3 Importance of Security in Windows Server
- 9.3.1 Hardware firewall.
- 9.3.2 Proxy Servers.
- 9.3.3 De-militarized zones (DMZs).
- 9.3.4 Virtual LANs (VLANs).
- 9.3.5 IPS and IDS.

CHAPTER 10 (14 Periods of which 8 are practical exercises)

Cabling, Medium and Connectors

10.1 Roles and usage:
- 10.1.1 Planning a network with a specific network planning software
- 10.1.2 Cabling (Twisted pair, Coaxial, Fiber-Optic).
- 10.1.3 Medium (Wireless, ethernet, Fiber-Optic (using reflectometer)).
- 10.1.4 Connectors (RJ45, BNC, SC, LT, SC, MTP/MTO, etc.).

CHAPTER 11 (10 Periods of which 8 are practical exercises)

Cabinet, Wiring and Troubleshooting

11.1 Cabinet Importance and Management
- 11.1.1 Ensuring efficiency.
- 11.1.2 Organizing wires
- 11.1.3 Managing the Cabinet (Power Unit, Cooling Fans, Racks, Patch Panel, Grounding,
- 11.1.4 Labeling, etc.).

11.2 Troubleshooting.
- 11.2.1 Testing Tools and Utilities:
- 11.2.2 Ping, Traceroute/Tracert, Netstat/Nbtstat, Ipconfig/Ifconfig, NSLookup/dig, telnet.

VIRTUALIZATION I

(60 PERIODS OF WHICH 30 ARE PRACTICAL EXERCISES)

CHAPTER 1 (14 Periods of which 7 are practical exercises)

Introduction to Virtual Computing:

1.1 Overview of Virtual Machines (Definition and Benefits)
1.2 Virtualization Software Categories and Products
    1.2.1 (Hypervisors, Platforms, Cloud…)
1.3 Virtualization Features
    1.3.1 (Hypervisors, Networking, Storage, Security)
1.4 Comparing and Downloading Virtualization Software
    1.4.1 Type of virtualization (Server, Desktop, etc.…)
    1.4.2 Virtualization products (VMware, Microsoft Hyper-V, VirtualBox, etc.)

CHAPTER 2 (15 Periods of which 10 are practical exercises)

Working with VMware Workstation:

2.1 Getting started with VMware
    2.1.1 Check Hardware compatibility
    2.1.2 Install and configure (Vcenter server)
    2.1.3 Create Virtual Machines
    2.1.4 Configure network and storage
2.2 Using administrative console menus.
2.3 Working with virtual machines

CHAPTER 3 (12 Periods of which 6 are practical exercises)

Working with VMware Vcenter Server:

3.1 Implementation and Shared storage
3.2 Working with Vcenter Server.
3.3 Implementation and Control Security

CHAPTER 4 (14 Periods of which 7 are practical exercises)

Working with Microsoft Hyper-V:

4.1 Installing Hyper-V
4.2 Working with Hyper-V manager.
4.3 Working with virtual machines in Hyper-V
4.4 Using Checkpoints.

CHAPTER 5 (5 Periods)

Introduction to Cloud Computing:

5.1 Introduction to Cloud Services.
5.2 VMware Cloud Services.
5.3 Microsoft Cloud Services

CLOUD COMPUTING I
(90 PERIODS OF PRACTICAL EXERCISES)

CHAPTER 1 (30 Periods of practical exercises)

Microsoft Azure

1.1 Introduction to Microsoft Azure
- 1.1.1 Overview of Microsoft Azure: services offered, deployment models (Azure Resource Manager vs. classic deployment).
- 1.1.2 Creation of an Azure account and introduction to the management console.
- 1.1.3 Suggested exercises: Setting up an Azure Account and Exploring the Dashboard

1.2 Fundamentals of MS Azure
- 1.2.1 Key concepts of MS Azure: resources, resource groups, regions, and availability zones.
- 1.2.2 Storage services in MS Azure: Blob Storage, Table Storage, File Storage.
- 1.2.3 Suggested exercises: Configuring Storage Accounts and Uploading Data to Blob Storage

1.3 Azure Compute Services
- 1.3.1 Virtual Machines
- 1.3.2 Configurations, Images, and Deployment.
- 1.3.3 Azure Kubernetes Service (AKS)
- 1.3.4 Introduction to Containers and Kubernetes on Azure.
- 1.3.5 App Services
- 1.3.6 Web Apps, API Apps, and Mobile Apps.
- 1.3.7 Suggested exercises:
  - 1.3.7.1 Deploying a Web App Using Azure App Service
  - 1.3.7.2 Setting Up and Deploying a Containerized Application with AKS

1.4 Web Services and Apps in MS Azure
- 1.4.1 Implementation of Web Services and Apps
- 1.4.2 Utilization of Azure App Service
- 1.4.3 Suggested exercises: Implementing a RESTful API and Deploying to Azure

CHAPTER 2 (30 Periods of practical exercises)

Google Cloud Platform

2.1 Introduction to Google Cloud Platform
- 2.1.1 Overview of Google Cloud Platform
  - 2.1.1.1 Services offered and deployment models.
- 2.1.2 Creating a Google Cloud Platform Account

2.2 Introduction to the management console.
- 2.2.1 Suggested exercises: Setting Up a GCP Account and Exploring the Console

2.3 Fundamentals of Google Cloud Platform
- 2.3.1 Key Concepts of GCP
  - 2.3.1.1 Resources, projects, regions, and zones.
- 2.3.2 Exploration of Storage Services
  - 2.3.2.1 Cloud Storage, Cloud SQL.
- 2.3.3 Suggested exercises: Configuring Cloud Storage and Setting Up a Cloud SQL Instance

2.4 Google Cloud Compute Services
- 2.4.1 Compute Engine
  - 2.4.1.1 Virtual Machines, Machine Types.
- 2.4.2 Google Kubernetes Engine (GKE)

2.4.2.1 Kubernetes clusters, container deployment.

2.4.3 App Engine

2.4.3.1 Deployment of scalable web applications.

2.4.4 Suggested exercises: Deploying an Application Using Google App Engine

2.4.5 Suggested exercises: Setting Up a Kubernetes Cluster with GKE

2.5 Web Services and Apps

2.5.1 Implementing Web Services

2.5.2 Using Google App Engine

2.5.3 Suggested exercises: Developing and Deploying a Simple Web Service

2.6 Google Cloud Databases

2.6.1 Cloud Firestore

2.6.1.1 NoSQL databases, document and collection structures.

2.6.2 Cloud Spanner

2.6.2.1 Fully managed relational database with unlimited scale.

2.6.3 Suggested exercises:

2.6.3.1 Implementing a NoSQL Database with Cloud Firestore

2.6.3.2 Suggested exercises: Configuring and Using Cloud Spanner

CHAPTER 3 (30 Periods of practical exercises)

Amazon Web Services (AWS)

3.1 Introduction to Amazon Web Services (AWS)

3.1.1 Overview of AWS

3.1.1.1 Services offered and deployment models.

3.1.2 Creating an AWS Account

3.1.2.1 Introduction to the AWS Management Console.

3.1.3 AWS Global Infrastructure

3.1.3.1 Regions, Availability Zones, and Edge Locations.

3.1.4 Suggested exercises: Setting Up an AWS Account and Exploring the AWS Console

3.2 Fundamentals of AWS

3.2.1 AWS Core Services Overview

3.2.1.1 EC2, S3, RDS, Lambda.

3.2.2 AWS Security Basics

3.2.2.1 Overview of IAM, Security Groups, and Network ACLs.

3.2.3 AWS Networking Essentials

3.2.3.1 VPC, Subnets, Internet Gateways.

3.2.4 AWS Storage Options

3.2.4.1 S3, EBS, Glacier, Storage Gateway.

3.2.5 Suggested exercises:

3.2.5.1 Launching an EC2 Instance and Configuring IAM Roles

3.2.5.2 Setting Up an S3 Bucket and Uploading Objects

3.3 AWS Compute Services

3.3.1 EC2 Deep Dive (1 Period)

3.3.1.1 Instance Types, AMIs, EBS.

3.3.2 Elastic Load Balancing and Auto Scaling

3.3.2.1 Configuring ELB and Auto Scaling Groups.

3.3.3 AWS Lambda

3.3.3.1 Serverless Compute with Lambda Functions.

3.3.4 AWS Elastic Beanstalk

3.3.4.1 Deploying Applications with Elastic Beanstalk.

3.3.5 Suggested exercises:

3.3.5.1 Deploying a Serverless Application with AWS Lambda

3.3.5.2 Creating a Highly Available Environment with ELB and Auto Scaling

3.4 Databases in AWS

3.4.1 Amazon RDS

3.4.1.1 Setting up relational databases.

3.4.2 Amazon DynamoDB

3.4.2.1 NoSQL database service.

3.4.3 AWS Database Migration Service

3.4.3.1 Migrating databases to AWS.

3.4.4 Suggested exercises:

3.4.4.1 Configuring an RDS Instance and Connecting to a Database

3.4.4.2 Working with DynamoDB

SCRIPTING AND AUTOMATION I
(60 PERIODS OF PRACTICAL EXERCISES)

CHAPTER 1 (30 Periods)

Bash Fundamentals

1.1 Overview of scripting and automation in modern IT environments.
1.2 Introduction to Bash: roles, environments, and when to use each.
1.3 Basic structure of scripts in Bash.
1.4 Suggested exercises:
  1.4.1 Setting up environments for Bash
  1.4.2 Writing and executing simple scripts for basic tasks with Bash.
1.5 Bash syntax and command-line fundamentals.
1.6 Variables, operators, and control flow statements (if, loops).
1.7 Script debugging and basic error handling.
1.8 Functions, arrays, and dictionaries
1.9 Suggested exercises:
  1.9.1 Writing Bash scripts for file management, text processing, and system administration tasks.
  1.9.2 Practice with control flow structures to automate routine tasks.
  1.9.3 Developing reusable scripts and functions for common administrative tasks.

CHAPTER 2 (30 Periods)

PowerShell Fundamentals

2.1 Introduction to PowerShell: roles, environments, and when to use each.
2.2 Basic structure of scripts in PowerShell.
2.3 Suggested exercises:
  2.3.1 Setting up environments for PowerShell.
2.4 Writing and executing simple scripts for basic tasks with PowerShell.
2.5 PowerShell syntax, cmdlets, and the PowerShell pipeline.
2.6 Variables, operators, and control flow in PowerShell.
2.7 Error handling and debugging in PowerShell scripts.
2.8 Custom objects, modules, and remote scripting
2.9 Suggested exercises:
  2.9.1 Automating Windows system administration tasks with PowerShell.
  2.9.2 Utilizing PowerShell to manage files, processes, and Windows Registry.
  2.9.3 Creating scripts that leverage cmdlets and custom functions.
  2.9.4 Developing reusable scripts and functions for common administrative tasks.

REMOTE MAINTAINANCE I
(30 PERIODS OF PRACTICAL EXERCISES)

CHAPTER 1 (5 Periods of practical exercises)

Introduction to Remote Maintenance

1.1 Overview and Benefits.
1.2 Importance in (Time and Cost savings, etc...)
1.3 Problems and Solutions (Connectivity, Security, etc.…)

CHAPTER 2 (15 Periods of practical exercises)

Remote Access Setup and Technologies

2.1 Introduction and Configuring Remote Desktop Protocol (RDP).
2.2 Setup Secure Shell (SSH) end-to-end Connection.
2.3 Authentication and Tunneling enabled by SSH.
2.4 TeamViewer Tools:
- 2.4.1 Introduction and Benefits.
- 2.4.2 Comparing TeamViewer with other Remote Tools (Features, Advantages and Disadvantages).
- 2.4.3 Install and Configure TeamViewer.
- 2.4.4 Practical Exercises for RDP, SSH and TeamViewer.
- 2.4.5 Troubleshooting and Q&A.

2.5 Configuring Remote Access Policies.

CHAPTER 3 (10 Periods of practical exercises)

Remote Maintenance Security

3.1 Security Overview and Importance
- 3.1.1 User Authentication.
- 3.1.2 Device Security.

3.2 Data Protection and Encryption.
- 3.2.1 Secure Data (Storage, Transmission and Implementation).
- 3.2.2 Encryption Overview and Role.

3.3 Two Factor Authentication.
- 3.3.1 Importance, Overview and Types.
- 3.3.2 Configuring and Implementing.

SECOND YEAR

SPECIALIZATION SUBJECTS

1. OPERATING SYSTEMS II
2. OBJECT ORIENTED PROGRAMMING II
3. NETWORKS II
4. TP-NETWORKS
5. NETWORKS AND SECURITY
6. CLIENT SERVER ARCHITECTURE
7. VIRTUALIZATION II
8. CLOUD COMPUTING II
9. SCRIPTING AND AUTOMATION II
10. REMOTE MAINTAINANCE II
11. END OF STUDIES' PROJECT AND INTERNSHIP

OPERATING SYSTEM II
(180 PERIODS OF PRACTICAL EXERCISES)

Objective of the course

At the end of this course, the student should be able to:

- Install and configure a Windows and Linux server, manage user accounts (create users, create user groups, ensure user groups, guarantee and revoke access privileges on various files and resources for various users and user groups).
- Manage domains and domain users.
- Manage the Security and Workgroup Management

CHAPTER 1 (90 Periods of pratical exercises)

Windows Server Structure

1.3 Basic concepts of Windows (Windows rewiev TS1) and introduction to Windows Server
- 1.3.1 Windows architecture
- 1.3.2 Introduction to Windows Server and its history.
- 1.3.3 Basic concepts of the Windows Server operating system.
- 1.3.4 Introduction to management through graphical interface and command line.
- 1.3.5 Installation and initial configuration of Windows Server.
- 1.3.6 PowerShell Commands for Server Management:
  - 1.3.6.1 `GetHelp`: PowerShell help.
  - 1.3.6.2 `GetService`, `StartService`, `StopService`: Service management.
  - 1.3.6.3 `GetProcess`, `StopProcess`: Process management.
  - 1.3.6.4 `GetWindowsFeature`, `InstallWindowsFeature`, `UninstallWindowsFeature`: Windows Server feature management.

1.4 Active Directory Domain Services
- 1.4.1 Implementation of an Active Directory domain controller.
- 1.4.2 Configuration of domain services such as DNS and DHCP.
- 1.4.3 Management of Active Directory objects.
- 1.4.4 Backup and restore of Active Directory.
- 1.4.5 PowerShell Commands for Active Directory Management:
  - 1.4.5.1 `InstallWindowsFeature Name ADDomainServices`: Active Directory role installation.
  - 1.4.5.2 `AddWindowsFeature Name DNS`: DNS role installation.
  - 1.4.5.3 `AddWindowsFeature Name DHCP`: DHCP role installation.
  - 1.4.5.4 `NewADForest`, `NewADDomain`, `SetADForest`, `SetADDomain`: Active Directory domain configuration.
  - 1.4.5.5 `BackupADDSForest`, `RestoreADDSForest`: Active Directory backup and restore.

1.5 User and Group Management
- 1.5.1 Creation and management of user accounts (Local and Domain).
- 1.5.2 Configuration of account security policies.
- 1.5.3 Implementation of groups and roles.
- 1.5.4 Access rights management.
- 1.5.5 PowerShell Commands for User and Group Management:
  - 1.5.5.1 `NewADUser`, `GetADUser`, `SetADUser`, `RemoveADUser`: User management.
  - 1.5.5.2 `NewADGroup`, `GetADGroup`, `SetADGroup`, `RemoveADGroup`: Group management.
  - 1.5.5.3 `AddADGroupMember`, `RemoveADGroupMember`: Group member management.

1.6 File System and Permissions Management
- 1.6.1 Creation and configuration of network shares.
- 1.6.2 Assignment of permissions on files and folders.
- 1.6.3 Implementation of disk quotas.
- 1.6.4 Monitoring of storage resources.
- 1.6.5 PowerShell Commands for File System and Permissions Management:
  - 1.6.5.1 `NewSMBShare`, `GetSMBShare`, `SetSMBShare`, `RemoveSMBShare`: SMB share management.
  - 1.6.5.2 `NewItem`, `CopyItem`, `MoveItem`, `RemoveItem`: File and folder management.
  - 1.6.5.3 `GetAcl`, `SetAcl`: Permissions management.

1.7 Security and Workgroup Management
- 1.7.1 Implementation of group policies.
- 1.7.2 Configuration of Windows firewalls.
- 1.7.3 Monitoring and auditing of security activities.
- 1.7.4 Management of workgroups and nondomain servers.
- 1.7.5 PowerShell Commands for Security and Workgroup Management:
  - 1.7.5.1 `NewGPO`, `GetGPO`, `SetGPO`, `RemoveGPO`: Group Policy management.
  - 1.7.5.2 `EnableNetFirewallRule`, `DisableNetFirewallRule`: Firewall rule management.
  - 1.7.5.3 `GetWinEvent`, `GetEventLog`: Security event monitoring.
  - 1.7.5.4 `SetNetConnectionProfile`: Network profile configuration.

1.8 Troubleshooting network communications
- 1.8.1 Overview
- 1.8.2 Troubleshooting network communication with (Ping and PowerShell)
- 1.8.3 Testing connectivity to specific ports using (Command prompt and PowerShell)

1.9 Servers and DMZ
- 1.9.1 Importance and benefits of multi servers in computer network.
- 1.9.2 Servers Types (Web, Mail, DHCP, DNS, File, Print, and Database etc.)
- 1.9.3 Servers Roles and Features
- 1.9.4 Differences and Similarities between some servers
- 1.9.5 Definition and purpose of DMZ network
- 1.9.6 Importance and benefits of DMZ
- 1.9.7 DMZ Architecture and Design (Single and Dual firewall)
- 1.9.8 DMZ environment examples.

1.10 Install and Configure Web Server IIS
- 1.10.1 Importance and benefits of Web Server.
- 1.10.2 Installing, adding server roles and features
- 1.10.3 Configure IIS using server manager
- 1.10.4 Creating and uploading Website files, Virtual directories
- 1.10.5 Configuring Certificates, logging and monitoring

1.11 Managing print services
- 1.11.1 Install and Configure network printer driver using (Management Console and PowerShell)
- 1.11.2 Manage printer permissions and Queues
- 1.11.3 Publish a printer in Active directory using (Management Console and PowerShell)

PRACTICAL PROJECT

Implementation of a practical project involving the installation, configuration, and management of a Windows server. (Students can work on individual or group projects, with the support of the teacher)

Tasks:

1. Server installation
2. Deploy and configure various servers (Web, File, Mail, etc.)
3. User and Group Management
4. Monitoring
5. Logging in
6. Troubleshooting
7. Backing up
8. Documentation
9. Presentation

CHAPTER 2 (90 Periods of pratical exercises)

Linux Server Structure

2.1 Basic concepts of the command line (linux rewiev TS1) and introduction to Linux server structure.

- 2.1.1 File and directory management.
  - 2.1.1.1 `ls`, `cd`, `pwd`, `man`: Basic shell commands.
  - 2.1.1.2 `cat`, `less`, `head`, `tail`: Viewing file contents.
  - 2.1.1.3 `mkdir`, `rmdir`, `touch`: Managing directories and files.
  - 2.1.1.4 `cp`, `mv`, `rm`: Copying, moving, removing files and directories.
  - 2.1.1.5 `chmod`, `chown`: Modifying file permissions and ownership.
- 2.1.2 Text editor usage (vi, nano, etc.).
- 2.1.3 Fundamentals of shell scripting.

2.2 Installation and System Configuration

- 2.2.1 Installation of a Linux distribution.
- 2.2.2 Initial system configuration.
- 2.2.3 Package management and system updates.
  - 2.2.3.1 `apt`, `aptget`, `dpkg`: Package management on Debian/Ubuntu systems.
  - 2.2.3.2 `yum`, `dnf`: Package management on Red Hat/Fedora/CentOS systems.
- 2.2.4 User and group management.
- 2.2.5 Basic system security: firewall, SELinux/AppArmor, SSH.
  - 2.2.5.1 `hostnamectl`, `ifconfig`, `ip`: Network management and system configuration.
  - 2.2.5.2 `systemctl`: Control of system services (systemd).

2.3 System Administration

- 2.3.1 Process and service management.
  - 2.3.1.1 `ps`, `top`, `htop`: Process viewing and resource monitoring.
  - 2.3.1.2 `kill`, `killall`, `pkill`: Process termination.
- 2.3.2 System resource monitoring.
- 2.3.3 Task scheduling (cron, at).
- 2.3.4 Disk and partition management.
  - 2.3.4.1 `cron`, `at`: Task scheduling.
  - 2.3.4.2 `df`, `du`: Disk space and partition management.

2.4 Networking and Network Services

- 2.4.1 Network configuration: IP addressing, routing, DNS, DHCP.

2.4.1.1 `ping`, `traceroute`, `netstat`: Network connectivity testing and connection viewing.
2.4.2 Configuration of the Apache server for website management.
2.4.3 Configuration of the FTP (File Transfer Protocol) server.
2.4.4 Configuration of the SSH server for secure remote access.
2.4.4.1 `iptables`, `ufw`: Firewall configuration.
2.4.4.2 `sshd`: SSH server configuration.
2.4.4.3 `httpd`, `nginx`: Web server configuration.
2.5 Database Service Management
2.5.1 Introduction to databases: MySQL/MariaDB, PostgreSQL.
2.5.2 Installation and configuration of a MySQL/MariaDB server.
2.5.2.1 `mysql`, `mysqladmin`: MySQL database management.
2.5.2.2 Creation and management of databases and users.
2.5.2.3 `psql`: Interaction with the PostgreSQL database.
2.5.2.4 Database backup and restore.
2.5.2.5 `mysqldump`, `pg_dump`: Database backup.
2.6 Virtualization and Containers
2.6.1 Basics of virtualization.
2.6.2 Installation and configuration of VirtualBox.
2.6.2.1 `VirtualBox`: Virtual machine management.
2.6.3 Creation and management of virtual machines.
2.6.4 Introduction to Docker and containerization.
2.6.4.1 `docker`: Docker container management.
2.7 Advanced Automation and Scripting
2.7.1 Automation of system administration tasks.
2.7.2 Advanced shell scripting.
2.7.2.1 Bash scripting: Advanced usage of loops, conditions, functions, variables.
2.7.3 Introduction to Ansible for configuration management.
2.7.3.1 `grep`, `sed`, `awk`: Filtering and manipulation of text data.
2.8 Advanced Security and Troubleshooting
2.8.1 Advanced Linux system security.
2.8.2 Analysis and resolution of common system problems.
2.8.2.1 `sudo`, `su`: Management of privileged access.
2.8.3 Introduction to best practices for server security.
2.8.4 Advanced management of file and directory permissions.
2.9 Practical Project
2.9.1 Implementation of a practical project involving the installation, configuration, and management of a Linux server. (Students can work on individual or group projects, with the support of the teacher).
Students can complete this project with the collaboration of the teacher of network and network security
2.10 Linux System Administration Concepts
2.10.1 Administrative Tasks
2.10.1.1 Local Server Settings Configuration (`/etc/`, `systemctl`)
2.10.1.2 Adding roles and features (`apt`, `yum`, `dnf`, or `pacman`)
2.10.1.3 Managing other servers (ssh or `rsync`, `scp`, and `sshfs`)
2.10.1.4 Creating server groups (Ansible, Puppet)
2.10.2 Overview of Server Roles and Features
2.10.2.1 Adding and Removing Server Roles and Features using Command Line
2.11 Troubleshooting network communications
2.11.1 Overview

2.11.2 Troubleshooting network communication with (`ping`, `ping -c`, and `ping -s`)
2.11.3 Testing connectivity to specific ports using Command line (`telnet`, `nc` (netcat), or `ncat`)

2.12 Servers and DMZ
2.12.1 Importance and benefits of multi servers in computer network.
2.12.2 Servers Types (Web, Mail, DHCP ISC, DNS BIND, File Samba, Print CUPS, and Database etc.)
2.12.3 Servers Roles and Features
2.12.4 Differences and Similarities between some servers
2.12.5 Definition and purpose of DMZ network
2.12.6 Importance and benefits of DMZ
2.12.7 DMZ Architecture and Design (Single and Dual firewall)
2.12.8 DMZ environment examples.

2.13 Install and Configure Apache Web Server
2.13.1 Importance and benefits of Apache Web Server.
2.13.2 Updating packages, Installing, enabling and disabling services (`sudo`)
2.13.3 Configure firewall and secure Apache
2.13.4 Creating and uploading Website files (`/var/www/html/`)
2.13.5 Logging, Testing and monitoring (`curl`, `tail`, etc.)

2.14 Managing print services
2.14.1 Install and Configure network printer driver using CUPS or Web interface (`lpadmin`, and `lpoptions`)
2.14.2 Manage printer permissions and Queues
2.14.3 18.3 Publish and manage printer sharing using Samba ((`smb.conf`, `smbpasswd`)

PRACTICAL PROJECT

Implementation of a practical project involving the installation, configuration, and management of a Linux server. (Students can work on individual or group projects, with the support of the teacher)

Tasks:

1. Server installation
2. Deploy and configure various servers (Web, File, Mail, etc.)
3. User and Group Management
4. Monitoring
5. Logging in
6. Troubleshooting
7. Backing up
8. Documentation
9. Presentation

OBJECT ORIENTED PROGRAMMING II
(120 PERIODS OF WHICH 90 ARE PRACTICAL EXERCISES)

CHAPTER 1: (25 Periods of which 15 are practical exercises)

Advanced Object-Oriented Programming in Java

1.6 Review of OOP basics
1.7 Advanced OOP features: interfaces, abstract classes, nested classes.
1.8 Design patterns and their application in Java: Factory Method, Singleton, Observer, Strategy, and Decorator.
1.9 Reflection and annotations.
1.10 Suggested practice:
  1.10.1 Implementing various design patterns in Java.
  1.10.2 Projects that require the use of advanced OOP concepts and design patterns.

CHAPTER 2: (25 Periods of which 15 are practical exercises)

Java File I/O and NIO

2.1 Java I/O
2.2 Review of Java I/O: FileInputStream, FileOutputStream, FileReader, and FileWriter, the File class
  2.2.1 Review of Streams, Buffered Streams and serialization
2.3 Java New I/O (NIO) Introduction to the NIO package: Channels, Buffers, Selectors, and Path.
  2.3.1 The advantages of NIO over traditional I/O, including non-blocking I/O operations and improved performance with large files.
  2.3.2 Working with Files in NIO
  2.3.3 Path, Files, and Channels API for file operations.
  2.3.4 Exploring file attributes, directory operations, and file system monitoring.
  2.3.5 Understanding non-blocking I/O with selectors and sockets for scalable network applications.
  2.3.6 Introduction to asynchronous I/O with CompletableFuture and AsynchronousFileChannel.
  2.3.7 Suggested practice: developing a file search utility using NIO features.

CHAPTER 3: (20 Periods of which 15 are practical exercises)

Concurrency in Java

3.1 Introduction to Concurrency:
  3.1.1 Understanding the need for concurrency and its impact on software development.
  3.1.2 Basics of processes and threads: the difference between them, how they work in Java.
3.2 Threads in Java:
  3.2.1 Creating threads using the Thread class and the Runnable interface.
  3.2.2 Thread lifecycle, states, and transitions.
3.3 Synchronization and Thread Safety:
  3.3.1 The concept of thread safety and methods to achieve it in Java.
3.4 Synchronized methods and blocks, the volatile keyword.
3.5 Overview of the java.util.concurrent package.
  3.5.1 Executors, ExecutorService, and Thread Pools for managing a collection of threads.
  3.5.2 Callables and Futures for result-bearing tasks.

3.6 Locks and Synchronization Utilities:
  3.6.1 Lock objects for controlling access to a resource by multiple threads.
  3.6.2 Condition objects for inter-thread communication.
3.7 Introduction to thread-safe collections in Java.
  3.7.1 ConcurrentHashMap, CopyOnWriteArrayList, BlockingQueue, and their use cases.
3.8 Parallel Streams and Fork/Join Framework:
  3.8.1 Leveraging parallel streams for simplifying parallelism in Java applications.
  3.8.2 Understanding the Fork/Join framework for recursive task parallelism.
3.9 Asynchronous Programming in Java:
  3.9.1 CompletableFuture for asynchronous programming and its advantages over Future.
  3.9.2 Using the CompletableFuture API for composing asynchronous logic.
3.10 Suggested practice:
  3.10.1 Exercises on creating, running, and controlling threads.
  3.10.2 Utilizing the Fork/Join framework for tasks that can be broken down into smaller parts.

CHAPTER 4: (30 Periods of which 25 are practical exercises)

Java Networking and Web Programming

4.1 Review of networking basics
4.2 Overview of web application components: Servlets, JSP, and the MVC architecture.
4.3 Exploring Java EE technologies: JSF, JPA for persistence, EJB for business logic.
4.4 Introduction to Spring Framework: Core, MVC, Security, and Boot.
4.5 Planning, developing, testing, and deploying web applications in Java.
4.6 Understanding web servers (Tomcat, Jetty) and application servers (WildFly, GlassFish).
4.7 Introduction to microservices architecture with Spring Boot.
4.8 Basic concepts of web security.
4.9 Implementing authentication and authorization in Java web applications.

CHAPTER 5: (20 Periods of practical exercises)

Final project

Students will work individually or in small teams to develop a final project that demonstrates their understanding of Java concepts.
Possible simple applications students could develop include:

- **E-commerce Platform**: A web-based e-commerce application where users can browse products and add them to a cart.
  User authentication and session management should be implemented.
- **Chat Application**: A real-time chat application using Java networking (sockets) that supports multiple clients connected to a server. It could include features like private and group chats, file transfers, and user statuses (online, offline).

NETWORKS II

(90 PERIODS)

Description of the material

This course is a continuation of the network course given in the 1st year of TS Systems and Networks. It is divided into two parts. The first part deals with interconnection and redundancy in networks. The second deals in depth and in detail with the TCP/IP protocol

Objective of the subject

At the end of the course, the student will be able to:

- Identify the elements of a local or wide area network.
- Design and schematically represent local networks.
- Design and schematically represent VLANs
- Configure some equipment.
- Use network monitoring and control tools.
- Check the TCP/IP protocol.

Skills and Abilities (Learning Outcomes)

At the end of this course, the student should be able to analyze, design, identify the elements of a network and configure a local area network and be familiar with the advanced notions of the network concept.

CHAPTER 1 (6 Periods)

Network Recap

1.1 Open System Interconnection (ISO) Reference Model
1.2 Packages: definition, creation (header)
1.3 The protocols of the different layers of ISO.
1.4 The architecture of the Internet
1.5 Point to point
1.6 ethernet
1.7 IP addressing (classes A, B,…)
1.8 Reserved IPs
1.9 Masking.
1.10 Others….
1.11 Application exercises

CHAPTER 2 (8 Periods)

Network Interconnections

2.1 Introduction.
2.2 LAN-LAN
  2.2.1 Definition
  2.2.2 Examples and application exercises (number of server stations, number of client stations, topology, cabling (type and length), protocol, addresses, interconnection equipment, etc.)
2.3 LAN-WAN.
  2.3.1 Definition
  2.3.2 Routing
    2.3.2.1 Direct and Indirect routing
    2.3.2.2 The routing process and table
    2.3.2.3 Routing protocols (interior v/s exterior)
      2.3.2.3.1 Troubleshooting Routing Information Protocol (RIP)
      2.3.2.3.2 Troubleshooting Open Shortest Path First (OSPF)
      2.3.2.3.3 Troubleshooting Enhanced Interior Gateway Routing Protocol (EIGRP)
  2.3.3 Examples

2.4 WAN - WAN.
- 2.4.1 Definition
- 2.4.2 Exterior Gateway Protocol
  - 2.4.2.1 BGP
- 2.4.3 Examples

2.5 Interconnection functions
- 2.5.1 Issue of different protocols: list the cases.
- 2.5.2 Error handling.

2.6 Application
- 2.6.1 The commands (ipconfig, ipconfig /all, traceroute, ping, road, netconfig, and netstat)

CHAPTER 3 (8 Periods)

Redundancy in networks

3.1 Definition and necessity of redundancy.

3.2 Redundancy architecture at equipment and link level.
- 3.2.1 Standard RAID Levels
  - 3.2.1.1 RAID 0, RAID 1, RAID 5, RAID 6, RAID 10, and RAID 0+1
  - 3.2.1.2 Common Technologies

3.3 New storage solutions
- 3.3.1 Introduction
- 3.3.2 The Network Attached Storage NAS solution
  - 3.3.2.1 Features
  - 3.3.2.2 Example of a NAS server
- 3.3.3 The Storage Area Network SAN Solution
  - 3.3.3.1 Features
  - 3.3.3.2 Benefits
- 3.3.4 SAN vs. NAS

3.4 Recovery
- 3.4.1 Fault detection.
- 3.4.2 Switch to the fallback solution.
- 3.4.3 Back to normal environment.

CHAPTER 4 (30 Periods)

TCP/IP networks

4.1 How TCP/IP Networks Work
- 4.1.1 Synchronization issues
- 4.1.2 IPv4 and IPv6 addressing
- 4.1.3 ARP/RARP: Address resolution protocols
- 4.1.4 The IP architecture (the header)
- 4.1.5 The TCP architecture (the header)
- 4.1.6 Applications that use TCP/IP (Telnet, FTP, SMTP)
- 4.1.7 IP addressing exercises (VLSM and CIDR)

4.2 IP: Internet Protocol
- 4.2.1 IPv4
- 4.2.2 IPv6
- 4.2.3 Conversion from IPv4 to IPv6

4.3 UDP: User Datagram Protocol unreliable transport service
- 4.3.1 Definition
- 4.3.2 On your mind
- 4.3.3 Applications that use UDP (NFS, SNMP, DNS)

4.4 TCP and UDP ports
- 4.4.1 TCP: a reliable transport service (+ Exercises)
- 4.4.2 IPs mobile
- 4.4.3 The allocation of TCP/IP addresses and parameters
  - 4.4.3.1 BOOT strap protocol
  - 4.4.3.2 DHCP (Dynamic Host Configuration Protocol)
  - 4.4.3.3 Functioning
  - 4.4.3.4 implementation
  - 4.4.3.5 on your mind

4.5 ICMP: Internet Control Message Protocol
- 4.5.1 error messages in IPv4

4.6 IP telephony

4.7 DNS
- 4.7.1 definition
- 4.7.2 hierarchy
- 4.7.3 components
  - 4.7.3.1 nameserver
  - 4.7.3.2 nameresolver
- 4.7.4 DNS server types
  - 4.7.4.1 rootserver
  - 4.7.4.2 primaryormasterserver
  - 4.7.4.3 secondaryorslaveserver
- 4.7.5 DNS records
  - 4.7.5.1 Internet(IN)
  - 4.7.5.2 NameServer(NS)
  - 4.7.5.3 StartofAuthority(SOA)
  - 4.7.5.4 Address(A, YYYY)
  - 4.7.5.5 CanonicalName(CNAME)
  - 4.7.5.6 MailExchanger(MX)
  - 4.7.5.7 Point(PTR)
  - 4.7.5.8 TextTXT
- 4.7.6 Application exercises

CHAPTER 5 (20 Periods)

Local Virtual Networks - VLAN

5.1 Introduction

5.2 Interest of VLANs

5.3 Reminder on the operation of the switches

5.4 Algorithm of spanning tree (SpanningTree – STP)

5.5 VLAN Example

5.6 VLAN definitions

5.7 VLANs using switch ports
- 5.7.1 Definition
- 5.7.2 Setup example

5.8 VLANs by MAC addresses
- 5.8.1 Definition
- 5.8.2 Setup example

5.9 VLANs by IP addresses
- 5.9.1 Definition
- 5.9.2 Setup example

5.10 Frame labeling (tagged)
- 5.10.1 Definition

CHAPTER 6 (16 Periods)

Internet/Intranet networks

- 6.1 Internet 1
  - 6.1.1 Historical
  - 6.1.2 Standardization status
- 6.2 Internet 2
- 6.3 Areas of use
- 6.4 Topology
- 6.5 ISP
- 6.6 Internet applications
  - 6.6.1 Classic applications
    - 6.6.1.1 SMTP (Simple Mail Transfer Protocol)
    - 6.6.1.2 FTP (File Transfer Protocol)
    - 6.6.1.3 net
    - 6.6.1.4 NFS (Network File System)
    - 6.6.1.5 WWW
  - 6.6.2 Multipoint applications
- 6.7 Intranet
  - 6.7.1 Definition
  - 6.7.2 Intranet network utilities.
  - 6.7.3 Areas of use.
  - 6.7.4 Network management.
  - 6.7.5 Topology
  - 6.7.6 Examples.
- 6.8 Extranet
  - 6.8.1 Definition.
  - 6.8.2 Extranet network utilities.
  - 6.8.3 Areas of use.
  - 6.8.4 Network management.
  - 6.8.5 Topology.
  - 6.8.6 Examples.
- 6.9 Infranet.
  - 6.9.1 Definition.
  - 6.9.2 Infranet network utilities.
  - 6.9.3 Areas of use.
  - 6.9.4 Network management.
  - 6.9.5 Topology.
  - 6.9.6 Examples.
- 6.10 Future market trends : Internet 3, …

CHAPTER 7 (4 Periods)

Access networks

- 7.1 Hertzian access
  - 7.1.1 Wireless local loop
  - 7.1.2 All generations of mobile networks (First, Second, …)
  - 7.1.3 Generations of wireless communications networks
  - 7.1.4 Broadband wireless access
  - 7.1.5 Satellite access
- 7.2 Mobile-IP
  - 7.2.1 Nodes (Mobile, Home, Foreign, Tunneling, and Security)

TP-NETWORK

(90 PERIODS OF PRACTICAL EXERCISES)

Description of the material

The Computer Networks Lab is specially set up for labs and network projects and includes Ethernet or Token Ring networks. The environment is not necessarily homogeneous both in terms of platforms (Windows, Netware, Linux) or network equipment (hubs, switches, bridges, routers). Studies are performed using test tools, simulation software and protocol analyzers.

Objective of the subject

At the end of the project, the student will be able to demonstrate their ability to analyze and develop:

- Local networks with Ethernet equipment, TokenRing, etc.
- TCP/IP protocols and protocol developments;
- Connection to other laboratories via multi-protocol routers;
- Studies using test tools, simulation software and protocol analyzers;
- Administration techniques for heterogeneous networks (IP, IPv6, etc.);
- Network audits: architectures, protocols;
- Network management;
- Securing networks;
- Suggest to students to seek innovative solutions following problems created by the professor allowing them to anticipate strategic developments,

Work to do :

A- Step 1 – Preliminary, carried out by the laboratory manager (6 Periods):

1. Presentation of the different elements of a LAN network (server, client, router, concentrator, etc.)
2. Demonstration of a network analyzer on an existing functional network (congestion, statistics, connection, packet circulation, failure detection, reconfiguration, return to normal, etc.)

B- Step 2 – Analysis of simple cases, carried out by group (6 Periods):

1. Identify the needs of each given case study and represent them on a diagram (equipment, cabling, topology, operating system, etc.).
2. Installation of units (server, stations, network card or modem, cabling, concentrator, etc.).
3. Verification of network operation

C- Step 3 – Practical work covering the following themes (carried out by group) (60 Periods)

1. Orientation of the topology and creation of a small network
2. Using Wireshark to display protocol data units, parsing IP packets
3. Examine an ICMP packet
4. Supervision of the operation of a network (traceroute, ping, IP scanning, port scanning, etc.)
5. Configuring Basic Router Settings
6. Configuring and Troubleshooting a Static Route
7. Configuring and troubleshooting RIPv1 and RIPv2 (VLSM and CIDR)
8. EIGRP configuration and troubleshooting
9. OSPF configuration and troubleshooting
10. VLAN configuration and troubleshooting
11. NAT and DHCP configuration and troubleshooting

D- Step 4 – Security, carried out by group (18 hours):

D.1: Access List

1. Configuration and troubleshooting Access List (standard, extended, named)
2. Hardware-level implementation of a router.

3. Implementing security on a network. (Allow and/or deny access via IP on server, hub and router)

Materials:

Please find below the minimum material required for each group which is composed of a maximum of three students.

For each group

1. Server (containing 2 network cards)
2. 2 client stations (1 network card)
3. Switch (SWITCH)
4. Cabling (50 and 75 ohm coaxial cable, twisted pair (STP, UTP, 4 and 8 lines), Optical fiber (Laser, Infrared, LED)
5. Connectors (RJ45, Terminators,….)
6. Network cable tester.
7. Network oscilloscope (to validate the signal traffic on the cable)
8. Modem
9. Print Server
10. Connection clamp for RJ45 and RJ11

For the lab:

1. Network analyzer (Network Analyzer ex: Fluke)
2. Network monitoring software (What's up gold, SMTP, etc.)
3. Switch
4. Router
5. Barrier protection (FIREWALL)
6. 2 main servers
7. Smart-UPS (RJ45)
8. Printer
9. Modem
10. Internet connection

Simulator: (Choose one or more)

- Packet Tracer
- Boson NetSim
- GNS
- Others …

NETWORK SECURITY
(120 PERIODS OF WHICH 90 ARE PRACTICAL EXERCISES)

CHAPTER 1 (20 Periods of which 10 are practical exercises)

Network Security Recap

1.1 Review of Concepts:
- 1.1.1 Networking terms, types, connectors, and models.
- 1.1.2 Protocols, parameters, devices, and topologies.
- 1.1.3 IP addresses, classes, DNS, and DHCP.

1.2 Transmission:
- 1.2.1 Elements, Media and Techniques.

1.3 Cybersecurity:
- 1.3.1 Importance.
- 1.3.2 Threats: (Malware, Phishing, DoS, MITM…).
- 1.3.3 Security Vulnerability: (Hardware, Software, and Updates).
- 1.3.4 Privacy Protection: (Encryption, Multi factor…).
- 1.3.5 Roles and Technologies: (IDS, IPS, VPN…).

1.4 Security Types, Benefits, and Challenges:
- 1.4.1 Types: (Antivirus, Antimalware, Cloud Security, Data Loss, Email, Firewall…).
- 1.4.2 Benefits: (Functionality, Confidentiality, Integrity and Availability. Intellectual property protection).
- 1.4.3 Challenges: (Remote access, User adherence…).

CHAPTER 2 (35 Periods of which 30 are practical exercises)

Attacks, Security Concepts and Techniques

2.1 Analyzing a Cyber Attack
- 2.1.1 Types of Malware
- 2.1.2 Symptoms of Malware

2.2 Methods of Infiltration
- 2.2.1 Social Engineering
- 2.2.2 Denial-of-Service
- 2.2.3 Distributed DoS
- 2.2.4 Botnet
- 2.2.5 On-Path Attacks
- 2.2.6 SEO Poisoning
- 2.2.7 Wi-Fi Password Cracking
- 2.2.8 Password Attacks
- 2.2.9 Cracking Times
- 2.2.10 Advanced Persistent Threats

2.3 Security Vulnerability and Exploits
- 2.3.1 Hardware Vulnerabilities
- 2.3.2 Software Vulnerabilities
- 2.3.3 Categorizing Software Vulnerabilities
- 2.3.4 Software Updates

2.4 The Cybersecurity Landscape
- 2.4.1 Cryptocurrency
- 2.4.2 Cryptojacking

CHAPTER 3 (35 Periods of which 30 are practical exercises)

Protecting Data and Privacy

3.1 Protecting Your Devices and Network
- 3.1.1 Protecting Your Computing Devices
- 3.1.2 Wireless Network Security at Home
- 3.1.3 Public Wi-Fi Risks
- 3.1.4 Password Security
- 3.1.5 A Strong Password
- 3.1.6 Using a Passphrase
- 3.1.7 Password Guidelines
- 3.1.8 Password Check

3.2 Data Maintenance
- 3.2.1 What Is Encryption?
- 3.2.2 How Do You Encrypt Your Data?
- 3.2.3 Back Up Your Data
- 3.2.4 Are They Really Gone?
- 3.2.5 How Do You Delete Your Data Permanently?

3.3 Who Owns Your Data?
- 3.3.1 Terms of Service
- 3.3.2 Understand the Terms
- 3.3.3 What Are You Agreeing To?
- 3.3.4 The Data Use Policy
- 3.3.5 Privacy Settings
- 3.3.6 Before You Sign Up
- 3.3.7 Protect Your Data

3.4 Safeguarding Your Online Privacy
- 3.4.1 Two Factor Authentication
- 3.4.2 Open Authorization
- 3.4.3 Social Sharing
- 3.4.4 Don't Get Spoofed
- 3.4.5 Email and Web Browser Privacy

CHAPTER 4 (30 Periods of which 20 are practical exercises)

Protecting the Organization

4.1 Cybersecurity Devices and Technologies
- 4.1.1 Security Appliances
- 4.1.2 Firewalls
- 4.1.3 Port Scanning
- 4.1.4 What Does It Mean?
- 4.1.5 Intrusion Detection and Prevention Systems
- 4.1.6 Real-Time Detection
- 4.1.7 Protecting Against Malware
- 4.1.8 Security Best Practices

4.2 Behavior Approach to Cybersecurity
- 4.2.1 Behavior-Based Security
- 4.2.2 NetFlow
- 4.2.3 Penetration Testing
- 4.2.4 Impact Reduction
- 4.2.5 What Is Risk Management?

CLIENT SERVER ARCHITECTURE
(60 PERIODS)

Objective of the Topic

- Design and develop the C/S Components and models.
- Administration of C/S applications.
- Understand mediators and remote procedural calls.

CHAPTER 1 (10 Periods)

Introduction to C/S Architecture:

1.5 Overview of C/S
  1.5.1 Brief History
  1.5.2 Definition
1.6 Network Basics and Protocols.
1.7 Security and Authentications.
1.8 Difference between client-side and server-side architecture.

CHAPTER 2 (20 Periods)

C/S Architecture Examples & Components:

2.4 C/S Architecture (***WEB and Network***) Examples
  2.4.1 Mail Servers, File Servers, and WEB Servers.
  2.4.2 Banking Application
  2.4.3 Healthcare, Banking, and Airport Application
2.5 Components of C/S Architecture
  2.5.1 Workstations (Client-side)
  2.5.2 Servers (Mail, File, DNS, and WEB, etc.)
  2.5.3 Networking Devices (Hubs, Switches, Repeaters, and Bridges, etc.)

CHAPTER 3 (20 Periods)

Working Steps and Types for a C/S Architecture:

3.4 How does C/S Architecture work?
3.5 C/S Architecture types and Subtypes
  3.5.1 One-tier Architecture
  3.5.2 Two-tier Architecture
  3.5.3 Three-tier Architecture
  3.5.4 N-tier Architecture
  3.5.5 Thin-Client Architecture
  3.5.6 Fat-Client Architecture
3.6 Difference Between Peer-to-Peer Network and C/S Architecture.
3.7 Advantages and Disadvantages of C/S Architecture.
3.8 Use Cases of C/S Architecture.

CHAPTER 4 (10 Periods)

Mediators:

4.5 Definition and Objectives
4.6 Basic and Architecture of Mediators.
4.7 Main features of the Mediation Service
- 4.7.1 Login procedure
- 4.7.2 Preparation of requests
- 4.7.3 Executing queries
- 4.7.4 Retrieval of results
- 4.7.5 Results cache management
- 4.7.6 Query cache management

4.8 Families of mediators
- 4.8.1 RPC middleware
- 4.8.2 Object Middleware
- 4.8.3 Message Oriented Middleware
- 4.8.4 Component Middleware
- 4.8.5 Middleware Service
- 4.8.6 Database middleware
- 4.8.7 Persistence middleware
- 4.8.8 transaction middleware

VIRTUALIZATION II

(60 PERIODS OF WHICH 30 ARE PRACTICAL EXERCISES)

CHAPTER 1 (12 Periods of which 6 are practical exercises)

Recap of Virtualization

1.1 Importance of Virtual Computing.
- 1.1.1 Types (Hardware, Software, etc.)

1.2 Software categories and features.
- 1.2.1 Overview of Hypervisors Types.
- 1.2.2 Installing and configuring products (VMware, Hyper-V, etc.)

CHAPTER 2 (14 Periods of which 7 are practical exercises)

Virtual environment and Networking:

2.1 Concept and Importance of Virtual network.
2.2 Functions and roles of Virtual Switches.
2.3 Explain and Explore network modes (NAT, Bridged, etc.)

CHAPTER 3 (10 Periods of which 5 are practical exercises)

Windows and Linux Virtual environment:

3.9 Difference between VM in both OS
- 3.9.1 Types, Management Tools, Cost, Licensing and Open source or No
- 3.9.2 Examples and Practice about above points

CHAPTER 4 (12 Periods of which 6 are practical exercises)

Microsoft Hyper-V Advanced concepts (Windows):

4.1 Advanced Hyper-V components and features
4.2 Hyper-V Replica between hosts.
4.3 Configure and Implement Network (Redundancy, Load Balancing, and Isolation etc.)
4.4 Configure Remote FX settings, and Advanced Security features
4.5 Practice on Configuration, Securing and Troubleshooting.

CHAPTER 5 (12 Periods of which 6 are practical exercises)

VMware Advanced concepts (Linux):

5.1 Advanced VMware components and features
5.2 Configure and Implement Network (Teaming, Load Balancing, and Segmentation)
5.3 Configure Storage technologies (MFS “Machine file system” and NFS “Network FS”)
5.4 Implement Backup and Security Encryption
5.5 Practice on Configuration, Securing and Troubleshooting.

CLOUD COMPUTING II
(120 PERIODS OF WHICH 90 ARE PRACTICAL EXERCISES)

CHAPTER 1 (30 Periods)

Understanding the Cloud

1.1 Core Cloud Concepts
- 1.1.1 Virtualization: Understand the technology that enables cloud computing, including hypervisors and virtual machines.
- 1.1.2 Storage in the Cloud: Introduction to cloud storage options, object storage, block storage, and file storage.
- 1.1.3 Networking in the Cloud: Basics of cloud networking, including virtual networks, subnets, and internet gateways.

1.2 Security and Compliance in the Cloud
- 1.2.1 Cloud Security Best Practices: The shared responsibility model, encryption, and IAM.
- 1.2.2 Compliance and Regulations: Overview of common compliance standards (e.g., GDPR, HIPAA) and how cloud providers help meet these requirements.

1.3 Cloud Monitoring and Management
- 1.3.1 Monitoring Tools: Introduction to monitoring cloud resources for performance, availability, and cost.
- 1.3.2 Alerts and Notifications: Setting up alerts for monitoring cloud resources.
- 1.3.3 Cloud Management Practices: Overview of managing cloud resources, including automation and configuration management.

1.4 Cloud Networking
- 1.4.1 Introduction to Cloud Networking: Basics of setting up and managing networks in the cloud.
- 1.4.2 Load Balancers and Traffic Management: Understand how load balancing works for distributing traffic and ensuring high availability.
- 1.4.3 Network Security: Basics of network security groups (NSGs), firewalls, and other security measures.

1.5 Database Services in the Cloud
- 1.5.1 Overview of Cloud Databases: Introduction to types of database services offered in the cloud (SQL, NoSQL).
- 1.5.2 Choosing the right Database service: Factors to consider when selecting a database service for your application.

1.6 DevOps in the Cloud
- 1.6.1 Understanding DevOps: Explore the DevOps philosophy, benefits, and its importance in cloud computing.
- 1.6.2 Continuous Integration and Continuous Deployment (CI/CD): Basics of automating the software delivery process.
- 1.6.3 Infrastructure as Code (IaC): Introduction to managing infrastructure using code for consistent and efficient deployment.

1.7 Microservices and Containers
- 1.7.1 Introduction to Microservices: Understand the microservices architecture and its benefits.
- 1.7.2 Containers and Orchestration: Learn about containerization technologies (e.g., Docker) and orchestration tools (e.g., Kubernetes).

1.8 Cloud Development Tools
- 1.8.1 CI/CD Tools: Overview of continuous integration and continuous deployment tools available in the cloud.
- 1.8.2 Infrastructure as Code Tools: Introduction to tools like Terraform and AWS CloudFormation.

1.8.3 Version Control Systems: Importance of using version control systems (e.g., Git) in cloud development.

CHAPTER 2 (30 Periods of practical exercises)

Microsoft Azure

2.1 Security and Compliance
- 2.1.1 Security Principles in MS Azure
- 2.1.2 Roles and Access Control
- 2.1.3 Compliance and Regulations
- 2.1.4 Suggested practice: configuring Azure Active Directory and Role-Based Access Control (RBAC)

2.2 Monitoring and Diagnostics
- 2.2.1 Monitoring Tools in MS Azure
- 2.2.2 Configuration of Alerts and Notifications
- 2.2.3 Suggested practice: Setting Up Monitoring and Alerts for a Web Application

2.3 Azure Networking
- 2.3.1 Virtual Networks : Configuration and Connectivity.
- 2.3.2 Load Balancers and Traffic Management
- 2.3.3 Network Security Groups (NSGs) and Application Gateway
- 2.3.4 Suggested practice:
  - 2.3.4.1 Creating and Configuring a Virtual Network and NSG
  - 2.3.4.2 Configuring Load Balancers for High Availability

2.4 Database Services in Azure
- 2.4.1 Azure SQL Database
- 2.4.2 Cosmos DB
- 2.4.3 Azure Database for MySQL and PostgreSQL
- 2.4.4 Suggested practice:
  - 2.4.4.1 Deploying and Managing an Azure SQL Database
  - 2.4.4.2 Implementing a NoSQL Database with Cosmos DB

2.5 Azure Development Tools
- 2.5.1 Azure DevOps Services: Boards, Repos, Pipelines, Artifacts.
- 2.5.2 GitHub Actions for Azure
- 2.5.3 Visual Studio Code Azure Extensions
- 2.5.4 Suggested practice:
  - 2.5.4.1 Setting Up a CI/CD Pipeline with Azure DevOps
  - 2.5.4.2 Deploying Code from GitHub to Azure using GitHub Actions

2.6 Final Project
- 2.6.1 Design, Implement, and Deploy an Application Solution using Azure Services

CHAPTER 3 (30 Periods of practical exercises)

Google Cloud Platform

3.1 Security and Compliance in GCP
- 3.1.1 Security Principles
- 3.1.2 Roles and Access Control
- 3.1.3 Compliance and Regulations
- 3.1.4 Suggested practice: managing IAM Policies and Roles

3.2 Monitoring and Management Tools
- 3.2.1 Stackdriver Monitoring: Monitoring cloud resources and applications.
- 3.2.2 Stackdriver Logging: Log management and analysis.
- 3.2.3 Stackdriver Error Reporting: Identifying and understanding errors in your applications.
- 3.2.4 Suggested practice:
  - 3.2.4.1 Configuring Monitoring and Alerts with Stackdriver
  - 3.2.4.2 Implementing Logging and Error Reporting

3.3 Networking in Google Cloud
- 3.3.1 Virtual Private Cloud (VPC): Creating isolated networks within GCP.
- 3.3.2 Cloud Load Balancing: Distributing user traffic across multiple instances.
- 3.3.3 Cloud VPN and Interconnect: Connecting on-premises networks to GCP.
- 3.3.4 Suggested practice:
- 3.3.5 Setting Up and Configuring a VPC Network
- 3.3.6 Implementing Load Balancing for High Availability

3.4 DevOps in Google Cloud
- 3.4.1 Cloud Build: Continuous integration and delivery platform.
- 3.4.2 Cloud Deployment Manager: Infrastructure as code for GCP resources.
- 3.4.3 Cloud Source Repositories: Private Git repositories hosted on GCP.
- 3.4.4 Cloud Functions: Event-driven serverless computing.
- 3.4.5 Suggested practice:
  - 3.4.5.1 Automating Deployment with Cloud Deployment Manager
  - 3.4.5.2 Building and Deploying Applications with Cloud Build
- 3.4.6 Final Project
  - 3.4.6.1 Designing, Implementing, and Deploying a Solution Using GCP Services

CHAPTER 4 (30 Periods of practical exercises)

Amazon Web Services (AWS)

4.1 AWS Security and Compliance
- 4.1.1 Advanced IAM: Policies, Groups, and Best Practices.
  Compliance on AWS: Understanding AWS Compliance Programs.
  Data Protection: Encryption, KMS, and Certificate Manager.
- 4.1.2 Suggested practice: Implementing Security Best Practices with IAM

4.2 Monitoring and Diagnostics
- 4.2.1 Amazon CloudWatch: Monitoring AWS resources and applications.
- 4.2.2 AWS CloudTrail: Governance, compliance, operational auditing, and risk auditing of your AWS account.
- 4.2.3 AWS X-Ray: Analyzing and debugging distributed applications.
- 4.2.4 Suggested practice:
  - 4.2.4.1 Setting Up Monitoring with Amazon CloudWatch
  - 4.2.4.2 Tracing Application Requests with AWS X-Ray

4.3 Developing and Deploying Applications
- 4.3.1 CI/CD on AWS: AWS CodeCommit, CodeBuild, CodeDeploy, and CodePipeline.

4.3.2 AWS Developer Tools: Overview of the AWS Developer Tools Suite.
4.3.3 Microservices on AWS: Deploying microservices-based applications on AWS.
4.3.4 Container Services: Amazon ECS and EKS.
4.3.5 Suggested practice: Building and Deploying a Web Application Using AWS Developer Tools

4.4 Advanced Networking on AWS
4.4.1 Amazon Route 53: Scalable Domain Name System.
4.4.2 Amazon CloudFront: Content Delivery Network services.
4.4.3 AWS Direct Connect: Dedicated network connection to AWS.

4.5 VPC Concepts
4.5.1 VPC Peering
4.5.2 VPN Connections.
4.5.3 Suggested Practice: Setting Up a Content Delivery Network with Amazon CloudFront

4.6 Final Project
4.6.1 Designing, Implementing, and Deploying an AWS-Based Solution

SUGGESTED RESOURCES:

Azure:
Microsoft training at: https://learn.microsoft.com/en-us/training/browse/ and in particular:

- Azure networking: https://learn.microsoft.com/en-us/training/paths/design-implement-microsoft-azure-networking-solutions-az-700/
- Azure storage: https://learn.microsoft.com/en-us/training/paths/secure-compute-storage-databases/
- Azure DevOps: https://learn.microsoft.com/en-us/training/paths/evolve-your-devops-practices/

AWS:
AWS Training at https://explore.skillbuilder.aws/learn and in particular:

- AWS Basics: https://explore.skillbuilder.aws/learn/public/learning_plan/view/82/cloud-essentials-knowledge-badge-readiness-path
- AWS Advanced: https://explore.skillbuilder.aws/learn/public/learning_plan/view/1044/solutions-architect-knowledge-badge-readiness-path

Google Cloud:
Google Training at https://cloud.google.com/learn/ and in particular:

- Cloud Engineer:
- https://www.cloudskillsboost.google/paths/11
- DevOps Engineer:
- https://www.cloudskillsboost.google/paths/20

SCRIPTING AND AUTOMATION II

(90 PERIODS OF WHICH 60 ARE PRACTICAL EXERCISES)

CHAPTER 1: (10 Periods of practical exercises)

Recap of Previous Year's Program

1.1 Review of Scripting Basics
1.2 Recap of fundamental scripting concepts in Bash and PowerShell.
1.3 Best practices for script structure, debugging, and error handling.
1.4 Practical exercises revisiting basic scripting tasks in Bash and PowerShell.

CHAPTER 2: (30 Periods)

Scripting Architecture and design

2.1 Scripting Best Practices and Design Patterns
- 2.1.1 Writing Maintainable Scripts
- 2.1.2 Code organization, commenting, and documentation.
- 2.1.3 Version control best practices for scripts.
- 2.1.4 Design Patterns in Scripting
- 2.1.5 Common design patterns and architectures for automation scripts.

2.2 Security Considerations in Scripting
- 2.2.1 Securing Script Execution
- 2.2.2 Managing credentials and sensitive data in scripts.
- 2.2.3 Secure scripting techniques to prevent injection attacks and data leakage.
- 2.2.4 Scripting in Compliance and Regulatory Environments
- 2.2.5 Understanding the impact of compliance (e.g., GDPR, HIPAA) on scripting practices.
- 2.2.6 Strategies for logging, auditing, and reporting script activities in regulated environments.

2.3 Scripting and Cloud
- 2.3.1 Introduction to DevOps and Automation in the Cloud
- 2.3.2 Overview of DevOps practices and tools for automation.
- 2.3.3 Cloud-based automation tools and services (e.g., AWS Lambda, Azure Functions).
- 2.3.4 Emerging Scripting Languages and Tools

CHAPTER 3: (15 Periods of practical exercises)

Advanced Bash Scripting

3.1 Bash Scripting for Linux
- 3.1.1 Text Processing and Sed/Awk
- 3.1.2 Advanced text manipulation tasks using sed and awk.
- 3.1.3 Incorporating external programs and multi-file scripts.
- 3.1.4 Suggested practice: Creating scripts for report generation and data extraction.

3.2 Bash Scripting for Network Administration
- 3.2.1 Using Bash for network diagnostics and configuration.

3.3 Suggested practice: Writing scripts to automate network scanning and configuration tasks.

CHAPTER 4: (15 Periods of practical exercises)

Advanced PowerShell Scripting

4.1 Working with APIs and Web Services
4.2 Automating interactions with RESTful APIs using PowerShell.
4.3 PowerShell DSC (Desired State Configuration)
4.4 Managing and maintaining Windows infrastructure using DSC.
4.5 Suggested practice:
  4.5.1 Scripting solutions to integrate with web services and cloud APIs.
  4.5.2 Developing DSC configurations for automated system setup and compliance.
4.6 Advanced Error Handling and Debugging
  3.1 Deep dive into robust error handling and debugging techniques.
4.7 Suggested practice: Enhancing scripts with comprehensive error handling and diagnostic logging.

CHAPTER 5: (20 Periods of practical exercises)

Introduction to Python for Automation

5.1 Introduction to Python for Automation
  5.1.1 Python Basics for Scripting
  5.1.2 Introduction to Python syntax and setting up the environment.
5.2 Suggested practice: Writing simple Python scripts for file and directory management.
5.3 Automating Administrative Tasks with Python
  5.3.1 Using Python for system administration, network monitoring, and configuration.
  5.3.2 Developing Python scripts for log analysis, system health checks, and automated reporting.
  5.3.3 Leveraging Python Libraries for Automation

REMOTE MAINTAINANCE II
(30 PERIODS OF PRACTICAL EXERCISES)

CHAPTER 1 (10 Periods)

Recap of Remote Maintenance

1.1 Review of concepts
- 1.1.1 Importance, problems and solutions.
- 1.1.2 RDP, SSH, Authentication and Tunneling.
- 1.1.3 TeamViewer or other tool (Benefits, Practical and Troubleshooting)
- 1.1.4 Remote Policies (Security, Protection, Encryption and Authentication)

CHAPTER 2 (12 Periods)

RUST Setup and Technologies

2.1 Overview of Configuring Remote Desktop Protocol.

2.2 RUST Tools:
- 2.2.1 Introduction and Benefits.
- 2.2.2 Comparing RUST with other Remote Tools (Features, Advantages and Disadvantages).
- 2.2.3 Install and Configure RUST on different platforms (Mac, Windows, Linux).
- 2.2.4 RUST Architecture and Components analysis.
- 2.2.5 Practical Exercises for RDP, SSH and RUST.
- 2.2.6 Troubleshooting and Q&A.

2.3 Configuring Remote Access Policies.

CHAPTER 3 (8 Periods)

RUST Security and Solutions

3.4 Security Overview and Importance
- 3.4.1 User Authentication.
- 3.4.2 Device Security.

3.5 Data Protection and Encryption.
- 3.5.1 Secure Data.
- 3.5.2 Encryption Overview and Role.

3.6 Identifying Issues and Solutions.

3.7 Plan and Develop a RUST project.

END OF STUDIES' PROJECT AND INTERNSHIP
(180 PERIODS)

END OF STUDIES' PROJECT (120 Periods)

Description of the material

This course is a design project designed to allow students to put into practice the concepts and skills acquired during their technical training in systems and networks.
The student must submit a personal project for approval. It is also possible for him to carry out a research project.
The graduation project is presented orally and in the form of a professional caliber technical report. The project can be carried out in single, pair or trinomial.
This project must demonstrate mastery of all the knowledge acquired in the study program in question and the ability to adapt knowledge to a specific and real case. In doing so, the student must demonstrate autonomy and initiative. In all cases, the work must be carried out in accordance with the regulations for end-of-study projects.
This work must be done relatively independently; the teacher guides the students, provides possible solutions, without specifying the entire process. It is during this course that students fulfill the requirements of the program summary test according to procedures to be specified during the session.

Objective of the subject

At the end of the project, the student will be able to demonstrate his ability to:

- design elements, systems and processes that correspond to specific needs;
- design a work plan and timelines using project planning documents;
- formulate a problem, find information relevant to the solution of the problem raised in the problem, analyze this information, synthesize it, design and state recommendations;
- write a professional-calibre technical report including: the problem, the objectives, the hypotheses, the review of the documentation, the methodology, the design process, the presentation of the results and their analysis, the conclusions and the recommendations.
- The student must devote, on average, six hours per week (for a total of 180 120 hours) to the activities necessary for the realization of the study or the project, the drafting of reports and the preparation of his oral presentation.
- He will have discussion sessions with his supervisor during the session in order to receive advice, report on the progress made, receive feedback on the various deliverables that have been requested of him, etc.
- When in teams, students will perform tasks according to roles that will be determined during project planning

Team work

- Students carry out a graduation project based on a subject of common interest.
- A teacher-supervisor is appointed to ensure the general supervision of the team.

This project is designed to allow students to synthesize the knowledge acquired during the two years of teaching, while getting closer to the context of the company. The objective is therefore to allow the student to be operational very quickly and to facilitate his professional integration. The interest of such an organization is to oblige the students to acquire a methodology of group work.

Similarly, the objectives of the project are to introduce the group to research work; use their technical and conceptual skills in the experimental study of complex problems in systems and networks; develop practice, exercise autonomy in the laboratory in understanding

problems and interpreting results; communicate effectively in writing and orally. This project can be bibliographic and/or laboratory work.

Activities aimed at perfecting the training of students through the acquisition of new knowledge and the observation of working methods and techniques used in the systems and network laboratories and more particularly in the fields of application of the program. Projects will be crowned with a public oral presentation as well as a written report, both activities forming part of the student assessment.

The main objective of the project is creating a network system in a real world scenario. This should include the topics covered during TS1 and TS2, and in particular include network design, implementation, automation, security and virtualization.

The steps to follow are:

1 Participate in the development of the project:
  1.1 Feasibility study.
  1.2 Drafting specifications.
  1.3 Development of technical files.
2 Participate in the decision and the choice:
  2.1 From the wiring method.
  2.2 Equipment.
  2.3 The operating system on each station (Two separate operating systems must exist on the stations)
3 Participate in the follow-up of the works:
  3.1 Conformity of the cabling to the specifications.
  3.2 Acceptance of the site (if possible)
4 Install network hardware and software
  4.1 Indispensable participation in the preparation of the server (hardware and software) and the installation of network cards on the stations.
  4.2 Regularly: installation of applications on the network.
  4.3 Installation of users and their access rights.
5 Network Maintenance
  5.1 Update of the network monitoring book.
6 Network protection:
  6.1 Server Protection
    6.1.1 Power cut protection.
  6.2 Network security:
    6.1.2 Essential discriminating access rights. (Not everyone has access to just anything.)
    6.1.3 Confidentiality ensured by a password specific to each user.
    6.1.4 Essential installation of a network anti-virus.
7 Network resources must be easily usable:
  7.1 Installation of a print server allowing simplified print management. (If applicable)
  7.2 Installation of an automatic backup on the network. (SAN, NAS, RAID, etc.)
8 Backup Planning:
  8.1. Create a detailed backup plan that establishes when backups should be performed, their frequency and retention time.
  8.2. Server Configuration: Configure both servers (Windows and Linux) to perform data backups to a cloud service. Integrated operating system backup tools or third-party software can be used.
  8.3. Choosing a Cloud Provider: Identify a service provider that offers storage and backup services. Describe your choice based on factors such as reliability, security, cost and features offered.

9 Programming and scripting automation :

9.1. Create scripts to automate network configuration tasks, such as setting up IP addresses, configuring firewalls, and managing network services.

9.2. Utilize scripting skills to develop custom scripts for monitoring network performance and security. This involves creating scripts that can collect data from various sources, process it, and present it in a useful format or alert administrators to potential issues.

9.3. Scripting for Backup and Recovery: Implement scripting solutions for automating the backup and recovery process. This includes writing scripts that can automatically perform backups according to the schedule defined in the backup plan, verify the integrity of backups, and automate the recovery process when needed.

9.4. Use scripting to enhance network security by automating the deployment of security policies, scanning for vulnerabilities, and managing security updates.

Documentation

The thesis document must contain the following parts:

- Cover page.
- Summary.
- Foreword.
- Study of the existing: Positions, activities, physical plan, reviews, solution.
- Physical topology: cabling, distance, topology, equipment
- Logical Topology: Addressing and Topology
- Security: VLAN, VPN, Access list, Firewall, IPD, IDS,… (if applicable)
- Redundancy: STP, RSTP, SAN, NAS, … (if applicable)
- Configuration: Operating system, routers, switch, ACL, ISA,…
- Use of a network simulator: Packet Tracer, NetSim, Ns-3, etc.
- Conclusion.
- Appendices (screens, reports, programs).
- Bibliography.

The size of the project implies the constitution of a team made up of a maximum of three students.

Each group of students must submit a written report and a digital support (if applicable).

The style to be used in the realization of the memory is obligatorily the Font Times New Roman, size 12.

The pagination of the document is mandatory.

The document must contain at least 50 pages without annexes.

INTERNSHIP (60 Periods)

Training internship in a IT company

The internship is a period of practical training and professional experimentation that students will undertake in a company to acquire specific IT and/or Network sector skills, apply the knowledge gained during their academic studies, and gain direct experience in their field of interest. Characterized by a practical and field-oriented approach, an internship aims to integrate academic training with practical experience in a real working environment. During the internship, students will have the opportunity to work on concrete projects, collaborate with industry professionals, and develop cross-functional and soft skills.

1.1 Internship Objectives:

- Development of Technical Skills: Acquire a practical knowledge in the field of study through the practical application of concepts learned in the classroom.
- Development of Professional Skills: improve skills necessary to operate effectively in a professional environment.
- Reinforcement of Soft Skills: enhance soft skills such as leadership, problem-solving, and adaptability, essential for success in a professional career.
- Networking and Professional Connections: establish professional relationships with colleagues, supervisors, and other industry professionals to foster future job opportunities.
- Career Guidance: gain practical experience in the field of interest, allowing interns to confirm or redirect their professional choices.
- Development of Autonomy and Initiative: promote independence and the ability to take initiative in tackling new tasks and problem-solving.

1.2 Internship Tools:

The internship agreement is a formal agreement between three parties: the student (intern), the educational institution (school, university, or training institute), and the host organization (company, organization, or entity) that welcomes the student for a period of practical training. This document establishes the terms and conditions of the internship experience, providing a legal and organizational framework for the activities carried out during the internship period.

Internship Agreement: is an essential tool to ensure clarity, transparency, and compliance with regulations during the internship experience, protecting the interests of all parties involved.

The internship agreement typically includes the following elements:

- Student Information
- School Information and Responsible Contact Person
- Host Company Information
- Ethical and Behavioral Norms
- Monitoring and Evaluation Procedures
- Signatures of the Involved Parties (School, Student, Company)

Internship Training Plan: is a key tool to ensure a structured learning experience consistent with the student's educational and professional objectives. Its drafting requires effective collaboration among all parties involved. The internship training plan is a document outlining the objectives, activities, and implementation methods of a period of practical training within an organization. This document is usually prepared in collaboration between the student, the school, and the hosting company.

The training plan should include:

- Educational Objectives: a detailed description of the objectives the student should achieve during the internship.
- Contents and Activities: a list of specific activities the student will perform during the internship.
- Duration and Schedule:
- Tutoring and Supervision: appointment of a company tutor or supervisor responsible for supporting and guiding the student during the internship.
- Evaluation Methods: description of how the student's performance will be assessed.
- Obligations and Responsibilities of the Involved Parties: clear definition of the obligations and responsibilities of the student, the school, and the hosting company during the internship period.
- Ethical and Behavioral Norms: ethical guidelines and behavioral norms that the student must follow during the internship.
- Resources and Tools: a list of resources, tools, and access that the student will have available to carry out the activities outlined in the training plan.

Evaluation tools for an internship are devices used to measure, assess, and document a student's performance and progress during the internship period. These tools are designed to provide accurate and constructive feedback on the internship experience and to evaluate the achievement of educational objectives.
Typically, the following are used:

- Evaluation Sheets filled out by the hosting company to assess technical, behavioral, and cross-functional skills acquired by the student.
- Self-assessment Questionnaire that allows the student to reflect on their performance.
-

Final Student Report: is a written document summarizing the student's experience during the internship period. This report provides an overall view of the activities carried out, the skills acquired, the results achieved, and personal reflections on lessons learned.
Here is a possible format and content for a final intern report:

- Introduction: a brief presentation of the student, indicating name, course of study, and educational institution.
- Context and Purpose of the Internship: a description of the hosting company and the context in which the internship took place.
- Activities Carried Out
- Skills Acquired:
- Challenges Faced: a discussion of the challenges and obstacles encountered during the internship.
- Feedback from the Company Tutor and Colleagues:
- Personal Reflections
- Integration with the Educational Path: a discussion of how the internship experience integrates with the student's academic path.
- Suggestions and Recommendations
- Conclusions