We want a **full academic framework** for a *Post Graduate Diploma in Cyber Security (PGDCS)* that’s **one year**, **30 credit hours**, **two semesters**, and **Python-based practicals**, while keeping it **accessible for non-science graduates**.

I’ll make sure it’s:

* **Light but rigorous enough** for postgrad level.
* **Practical-heavy** (Python labs in every course).
* **Well-structured** with prerequisites handled inside the courses.
* **Aligned to industry pathways** after completion.

**Post Graduate Diploma in Cyber Security (PGDCS)**

**Duration:** 1 Year (Two Semesters)  
**Total Credits:** 30 (15 per semester)  
**Structure:** 5 courses per semester, each 3 credits (2 theory + 1 lab)  
**Delivery Mode:** Lectures + Practical Labs (Python-based) + Project Work

**Program Rationale**

Cybersecurity has emerged as one of the most critical skill areas in the digital age, with organizations worldwide seeking skilled professionals who can protect systems, networks, and data.  
Nepal, like other nations, is rapidly adopting digital infrastructure in banking, governance, education, and industry, making **cyber defense capability a national priority**.

This Post Graduate Diploma in Cyber Security aims to provide **foundational to intermediate-level knowledge** in key areas such as cryptography, network security, malware analysis, wireless security, and regulatory frameworks.  
The curriculum is **practical-first**, with a strong emphasis on **hands-on Python programming** for implementing and testing security concepts.

The program is designed to **accommodate non-science graduates** by starting with fundamental concepts in computing, networking, and programming, and gradually moving to advanced cybersecurity topics.

**Eligibility**

* **Minimum Qualification:** Bachelor's degree in any discipline (non-science graduates welcome).
* **Prerequisites:** None in programming or computer science — all essential computing foundations will be taught as part of the curriculum.
* **Ideal Candidates:**
  + Fresh graduates seeking a career in cybersecurity.
  + IT professionals wishing to specialize in security.
  + Employees from banking, law enforcement, or government looking to understand and mitigate cyber threats.

**Pathways After Completion**

Graduates can:

* **Join Academic Programs:**
  + MSc in Cybersecurity
  + MSc in Computer Science (with Security specialization)
  + MBA in Information Security Management
  + International Certifications (CEH, CompTIA Security+, CISSP)
* **Work in Professional Roles:**
  + Cybersecurity Analyst
  + Security Operations Center (SOC) Analyst
  + Penetration Tester
  + Digital Forensics Investigator
  + Network Security Engineer

**Post Graduate Diploma in Cyber Security (PGDCS)**

**Duration:** 1 Year (Two Semesters)  
**Total Credits:** 30 (15 per semester)  
**Structure:** 5 courses per semester, each 3 credits (2 theory + 1 lab)  
**Focus:** Hands-on Python-based practicals in every course

**Program Rationale**

Cybersecurity is a vital field for national and organizational digital safety. This program bridges the gap for **graduates from any discipline**, introducing computing fundamentals and advancing toward applied cybersecurity.  
Python is used throughout for implementing cryptography, network analysis, malware detection, and automation scripts — ensuring students **learn by doing**.

**Eligibility**

* Bachelor’s degree in any discipline.
* No prior coding or technical background required — foundational computing and programming skills are taught in the first semester.

**Course Structure**

**Semester 1 — Foundations**

| **Code** | **Course Title** | **Credits** | **Description** |
| --- | --- | --- | --- |
| PGDCS 101 | Fundamentals of Cyber Security | 3 | Overview of cybersecurity concepts, threats, policies, and ethical hacking basics. |
| PGDCS 102 | Python Programming for Cybersecurity | 3 | Python fundamentals with a focus on cybersecurity applications. |
| PGDCS 103 | Fundamentals of Cryptography | 3 | Symmetric/asymmetric encryption, hashing, digital signatures, all implemented in Python. |
| PGDCS 104 | Database Management Systems (with Security) | 3 | SQL basics, database security, Python-DB integration. |
| PGDCS 105 | Computer Networks & Operating Systems | 3 | Basics of networking, OSI/TCP-IP, routing, OS functions, file systems, and scripting for security tasks. |

**Semester 2 — Advanced Applications**

| **Code** | **Course Title** | **Credits** | **Description** |
| --- | --- | --- | --- |
| PGDCS 201 | Network Security | 3 | Firewalls, IDS/IPS, VPN, secure protocols, Python network scanning. |
| PGDCS 202 | Wireless Security | 3 | Wi-Fi protocols, attacks, and wireless penetration testing using Python. |
| PGDCS 203 | Malware Analysis & Digital Forensics | 3 | Malware types, forensic techniques, Python for log analysis. |
| PGDCS 204 | Cybersecurity Regulations & Compliance | 3 | Laws, compliance frameworks, and incident reporting. |
| PGDCS 205 | Project Work | 3 | Capstone project integrating knowledge from multiple courses. |

**Detailed Course Content**

**Semester 1**

**PGDCS 101: Fundamentals of Cyber Security**

* Cybersecurity basics, CIA triad
* Threats, vulnerabilities, risk management
* Ethical hacking phases
* Lab: Password strength checker, basic port scanner

**PGDCS 102: Python Programming for Cybersecurity**

* Python syntax, data types, control structures
* File handling, exceptions
* Libraries for security: hashlib, socket, scapy
* Lab: Packet sniffer, log file parser

**PGDCS 103: Fundamentals of Cryptography**

* Classical ciphers, modern encryption
* AES, RSA, ECC, hashing
* Digital signatures, PKI
* Lab: Implement AES & RSA in Python (PyCryptodome)

**PGDCS 104: Database Management Systems (with Security)**

* SQL basics, relational model
* SQL injection, prevention techniques
* Python + MySQL integration
* Lab: Secure login system with password hashing

**PGDCS 105: Computer Networks & Operating Systems**

* OSI & TCP/IP models
* IP addressing, subnetting, routing
* OS functions, file permissions, process management
* Lab: Python socket programming, process monitoring

**Semester 2**

**PGDCS 201: Network Security**

* Firewalls, IDS, IPS
* VPN and secure tunneling
* SSL/TLS basics
* Lab: Intrusion detection prototype in Python

**PGDCS 202: Wireless Security**

* Wireless standards, WPA/WPA2/WPA3
* Common wireless attacks
* Lab: Wireless packet capture & analysis

**PGDCS 203: Malware Analysis & Digital Forensics**

* Malware classification, indicators of compromise
* Basics of reverse engineering
* Forensic imaging, evidence handling
* Lab: Python anomaly detection from logs

**PGDCS 204: Cybersecurity Regulations & Compliance**

* Nepalese cyber laws
* International laws: GDPR, HIPAA, ISO 27001
* Lab: Drafting an incident report in Python template

**PGDCS 205: Project Work**

* Real-world problem-solving project in cybersecurity
* Examples: Vulnerability scanner, secure messaging app, forensic investigation

**Pathways After Completion**

**Academic:**

* MSc in Cybersecurity
* MSc in Computer Science (Security Specialization)
* MBA in Information Security Management
* Professional Certifications: CEH, CompTIA Security+, CISSP

**Professional Roles:**

* Cybersecurity Analyst
* SOC Analyst
* Penetration Tester
* Forensic Investigator
* Network Security Engineer

**Course Title: Fundamentals of Cyber Security**

**Course Code:** PGDCS 101  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Provide a broad understanding of cybersecurity principles and practices.
2. Familiarize students with threats, vulnerabilities, and risk management.
3. Introduce foundational cryptographic techniques and applications.
4. Present relevant laws, ethical guidelines, and professional responsibilities.
5. Introduce Python scripting for basic security tasks.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain fundamental cybersecurity concepts and terminology.
2. Identify and classify cyber threats and vulnerabilities.
3. Apply basic security controls and policy frameworks.
4. Understand and explain basic cryptographic methods.
5. Write Python scripts for simple cybersecurity tasks.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Cyber Security – 6 hours**  
Definition, scope, and importance of cybersecurity; CIA triad; Security goals; Cybersecurity lifecycle; Overview of recent global and Nepal-specific cyber incidents; Career opportunities in cybersecurity.

**Unit 2: Cyber Threats and Attack Vectors – 8 hours**  
Types of threats: malware, phishing, ransomware, DoS/DDoS, MITM, SQL injection; Threat actors and motivations; Social engineering techniques; Real-world attack case studies.

**Unit 3: Vulnerabilities and Risk Management – 7 hours**  
Types of vulnerabilities (software, hardware, human); Vulnerability assessment tools; Risk assessment methodology; Risk mitigation strategies; Security controls — administrative, technical, and physical.

**Unit 4: Security Policies, Standards, and Frameworks – 7 hours**  
Role of policies and guidelines; Developing security policies; Overview of ISO 27001, NIST Cybersecurity Framework, CERT; Incident response planning; Basics of business continuity and disaster recovery.

**Unit 5: Basics of Cryptography – 8 hours**  
Importance of cryptography; Symmetric vs asymmetric encryption; Overview of AES, RSA, ECC; Hashing algorithms (SHA family, MD5); Digital signatures; PKI; Common Python libraries for cryptography (hashlib, PyCryptodome).

**Unit 6: Ethical, Legal, and Professional Issues – 6 hours**  
Cyber laws in Nepal; International regulations (GDPR, HIPAA); Privacy protection; Intellectual property rights; Ethical hacking principles; Responsible vulnerability disclosure.

**Unit 7: Introduction to Python for Cybersecurity – 6 hours**  
Python basics: syntax, variables, loops, functions; File handling; Exception handling; Networking with socket; Hashing with hashlib; Example scripts — password strength checker, port scanner, file integrity checker.

**Laboratory Experiments (Python-based)**

1. **Password Strength Checker** — Write a program to evaluate password strength using length, character diversity, and common word checks.
2. **Simple Port Scanner** — Use socket to scan a range of ports on a given IP and identify open ports.
3. **File Hash Generator** — Generate MD5, SHA1, and SHA256 hashes for a given file using hashlib.
4. **File Integrity Verification** — Compare original and current file hashes to detect file tampering.
5. **Basic Packet Sniffer** — Capture and display packet details using scapy.
6. **Phishing Email Detector** — Identify suspicious keywords and URLs from sample email text.
7. **Log File Analyzer** — Parse sample log files to detect failed login attempts or unusual access patterns.

**Reference Books**

1. Stallings, W. (2018). *Computer Security: Principles and Practice* (4th ed.). Pearson.
2. Pfleeger, C., Pfleeger, S., & Margulies, J. (2015). *Security in Computing* (5th ed.). Pearson.
3. Andress, J. (2019). *Foundations of Information Security: A Straightforward Introduction*. No Starch Press.
4. Grimes, R. (2021). *Cybersecurity for Beginners*. Apress.
5. Dowd, M., McDonald, J., Schuh, J. (2009). *The Art of Software Security Assessment*. Addison-Wesley.

**Course Title: Fundamentals of Cryptography**

**Course Code:** PGDCS 103  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce students to the core concepts and principles of cryptography.
2. Provide an understanding of both classical and modern cryptographic techniques.
3. Explain how cryptography ensures confidentiality, integrity, and authentication.
4. Equip students to implement cryptographic algorithms using Python.
5. Prepare students for advanced security courses and applied cryptography applications.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain fundamental cryptographic concepts and terminology.
2. Differentiate between symmetric and asymmetric encryption.
3. Apply hashing, encryption, and digital signatures for security.
4. Implement cryptographic algorithms in Python.
5. Evaluate the strengths and weaknesses of different cryptographic methods.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Cryptography – 6 hours**  
Definition and importance of cryptography; Historical background; Role in cybersecurity; Terminology (plaintext, ciphertext, key, algorithm); CIA triad in relation to cryptography.

**Unit 2: Classical Cryptography – 6 hours**  
Substitution ciphers (Caesar, Monoalphabetic, Vigenère); Transposition ciphers; Cryptanalysis of classical ciphers; Python implementation of basic ciphers.

**Unit 3: Symmetric Key Cryptography – 8 hours**  
Stream and block ciphers; DES, 3DES, AES; Modes of operation (ECB, CBC, CTR); Key generation and management; Security considerations.

**Unit 4: Asymmetric Key Cryptography – 8 hours**  
RSA algorithm: key generation, encryption, decryption; Diffie–Hellman key exchange; Elliptic Curve Cryptography (ECC) basics; Applications in secure communications.

**Unit 5: Hash Functions and Message Authentication – 8 hours**  
Hashing algorithms: MD5, SHA-1, SHA-2, SHA-3; HMAC; Collision resistance; Password hashing and salting; Integrity verification.

**Unit 6: Digital Signatures and Certificates – 6 hours**  
Digital signature process; RSA-based signatures; ECDSA basics; Certificate Authorities (CA); Public Key Infrastructure (PKI); SSL/TLS basics.

**Unit 7: Cryptography in Practice – 6 hours**  
Python cryptography libraries (hashlib, PyCryptodome); Encrypting/decrypting files; Secure password storage; Best practices and common pitfalls in cryptographic implementations.

**Laboratory Experiments (Python-based)**

1. **Implement Caesar Cipher** — Encrypt and decrypt text using Caesar Cipher and allow variable key shifts.
2. **Vigenère Cipher in Python** — Implement encryption and decryption with a given keyword.
3. **AES File Encryption/Decryption** — Use PyCryptodome to encrypt and decrypt a text file.
4. **RSA Key Generation and Encryption** — Generate RSA keys, encrypt a message, and decrypt it.
5. **Hashing Utility** — Generate MD5, SHA1, and SHA256 hashes of files using hashlib.
6. **Password Storage with Salting** — Store user passwords securely with salt and hash.
7. **Digital Signature Verification** — Sign a message with a private key and verify it with the public key.

**Reference Books**

1. Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.
2. Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer.
3. Schneier, B. (2015). *Applied Cryptography* (20th Anniversary ed.). Wiley.
4. Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.
5. Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley.

**Course Title: Database Management Systems (with Security)**

**Course Code:** PGDCS 104  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce fundamental concepts of database systems and relational databases.
2. Teach students how to design, implement, and query databases.
3. Explain database security principles and common vulnerabilities.
4. Enable students to integrate databases with Python for secure applications.
5. Prepare students to implement practical security controls in database management.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain the basic concepts of relational database systems.
2. Design relational database schemas using normalization techniques.
3. Write SQL queries to retrieve and manipulate data.
4. Identify common database security threats and implement mitigation measures.
5. Integrate Python with databases for secure data access.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Database Systems – 6 hours**  
Database concepts and architecture; Types of databases (relational, NoSQL); DBMS vs file-based systems; Roles of DBAs and security administrators.

**Unit 2: Relational Model and Database Design – 7 hours**  
Entities, attributes, relationships; Primary and foreign keys; Normalization (1NF, 2NF, 3NF); ER diagrams; Designing secure databases.

**Unit 3: SQL Basics – 8 hours**  
Data definition language (DDL); Data manipulation language (DML); SELECT, INSERT, UPDATE, DELETE statements; Joins, subqueries, aggregation functions; Views and stored procedures.

**Unit 4: Advanced SQL and Transactions – 7 hours**  
Transactions and ACID properties; Concurrency control; Locks; Indexing; Triggers; Stored procedures for security enforcement.

**Unit 5: Database Security Concepts – 8 hours**  
Authentication and authorization; User roles and privileges; SQL injection attacks; Preventing SQL injection; Data encryption at rest and in transit; Backup and recovery strategies.

**Unit 6: Database Integration with Python – 6 hours**  
Python libraries: sqlite3, mysql.connector, SQLAlchemy; Connecting and querying databases; Error handling and transaction management; Safe data access and parameterized queries.

**Unit 7: Case Studies and Best Practices – 6 hours**  
Real-world database security breaches; Lessons learned; Best practices for secure database design; Implementing logging and audit trails.

**Laboratory Experiments (Python-based)**

1. **Database Creation and Table Design** — Create tables using SQL commands and define primary/foreign keys.
2. **Data Manipulation** — Insert, update, delete, and query records using Python and SQL.
3. **Joins and Aggregations** — Perform inner, outer, and cross joins; aggregate data using Python.
4. **Stored Procedures and Triggers** — Implement triggers for automatic logging of changes.
5. **SQL Injection Demonstration** — Demonstrate and prevent SQL injection attacks.
6. **User Role Management** — Create users with different privileges and test access control.
7. **Database Backup and Recovery** — Perform backup, restore, and verify data integrity using Python scripts.

**Reference Books**

1. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
2. Silberschatz, A., Korth, H., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill.
3. Rob, P., & Coronel, C. (2018). *Database Systems: Design, Implementation, & Management* (13th ed.). Cengage.
4. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). *Database Systems: The Complete Book* (2nd ed.). Pearson.
5. Harrison, J. (2016). *SQL & Python for Beginners*. Packt Publishing.

**Course Title: Computer Networks & Operating Systems**

**Course Code:** PGDCS 105  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce fundamental concepts of computer networks and operating systems.
2. Teach network architectures, protocols, and OS services relevant to cybersecurity.
3. Explain how operating systems manage resources and provide security features.
4. Enable students to implement basic network and OS-related security tasks using Python.
5. Prepare students for advanced network and system security courses.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain basic concepts of computer networks and operating systems.
2. Understand network layers, protocols, and OS functions.
3. Analyze and troubleshoot network and OS-related security issues.
4. Implement basic Python scripts for network scanning, monitoring, and OS automation.
5. Apply OS and network security concepts in real-world scenarios.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Computer Networks – 6 hours**  
Network definition and types (LAN, WAN, MAN, WLAN); Network topologies; OSI and TCP/IP models; Networking devices (routers, switches, hubs); IP addressing and subnetting basics.

**Unit 2: Network Protocols and Communication – 7 hours**  
TCP, UDP, ICMP, HTTP, HTTPS, FTP, DNS; Packet structure; Protocol operations; Introduction to sockets; Python socket programming examples.

**Unit 3: Network Security Basics – 7 hours**  
Firewalls, NAT, VPNs, IDS/IPS; Common attacks (DoS, sniffing, spoofing); Secure communication protocols; Python scripts for simple network scanning and monitoring.

**Unit 4: Introduction to Operating Systems – 6 hours**  
Definition, types of OS (Windows, Linux, macOS); OS architecture and components; Processes, threads, and multitasking; Memory management; File systems.

**Unit 5: OS Security and Access Control – 7 hours**  
User accounts and authentication; File and directory permissions; OS-level security tools; Logging and auditing; Python scripts for checking system status and access permissions.

**Unit 6: Process and Resource Management – 7 hours**  
Process lifecycle, scheduling, CPU and memory allocation; Deadlocks and resource management; Monitoring system resources; Automating resource monitoring with Python.

**Unit 7: Case Studies and Practical Applications – 8 hours**  
Real-world network and OS security breaches; Lessons learned; Best practices for securing networks and systems; Integrating Python for automation, monitoring, and basic incident response.

**Laboratory Experiments (Python-based)**

1. **Ping Sweep Script** — Use Python to check the availability of multiple hosts in a network.
2. **Port Scanner** — Scan a range of TCP/UDP ports on a host using Python socket.
3. **Packet Sniffer** — Capture and display network packet headers using Python (scapy).
4. **File Permission Checker** — Python script to verify and report file/directory permissions.
5. **Process Monitor** — Monitor running processes and CPU/memory usage using Python (psutil).
6. **Simple Network Chat Program** — Implement a TCP-based chat to understand sockets.
7. **Log Analyzer** — Parse system logs to detect failed logins or unusual activities using Python.

**Reference Books**

1. Tanenbaum, A. S., & Wetherall, D. (2013). *Computer Networks* (5th ed.). Pearson.
2. Stallings, W. (2016). *Operating Systems: Internals and Design Principles* (8th ed.). Pearson.
3. Comer, D. E. (2018). *Computer Networks and Internets* (6th ed.). Pearson.
4. Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux Kernel* (3rd ed.). O’Reilly.
5. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.

**Course Title: Network Security**

**Course Code:** PGDCS 201  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce core concepts and principles of network security.
2. Teach common network attacks and defense mechanisms.
3. Explain security protocols for secure communication.
4. Enable students to implement basic network security tasks using Python.
5. Prepare students for advanced cybersecurity roles involving network defense.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain fundamental network security concepts and terminology.
2. Identify and analyze common network threats and attacks.
3. Implement security mechanisms such as firewalls, VPNs, and IDS/IPS.
4. Use Python to develop simple network monitoring and security scripts.
5. Evaluate network security measures and recommend best practices.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Network Security – 6 hours**  
Overview of network security; Importance and goals; Threats, vulnerabilities, and risk in networks; Network security policies and models; Security lifecycle.

**Unit 2: Network Threats and Attacks – 8 hours**  
Common attacks: DoS/DDoS, MITM, sniffing, spoofing, ARP poisoning, session hijacking; Attack vectors and case studies; Social engineering attacks on networks.

**Unit 3: Firewalls and Intrusion Detection/Prevention Systems – 7 hours**  
Types of firewalls (packet filtering, stateful, application layer); IDS and IPS concepts; Network monitoring; Implementing basic rules and policies; Logging and alerting.

**Unit 4: Virtual Private Networks (VPNs) – 6 hours**  
Concept and types of VPNs (site-to-site, remote access); Tunneling protocols (IPSec, SSL/TLS); VPN configuration principles; VPN security best practices.

**Unit 5: Secure Network Protocols – 7 hours**  
SSL/TLS, HTTPS, SSH, S/MIME, IPsec; Encryption in transit; Certificate-based authentication; Protocol vulnerabilities and mitigations.

**Unit 6: Wireless Network Security – 7 hours**  
WEP, WPA, WPA2, WPA3; Wireless attacks (eavesdropping, evil twin, rogue access points); Securing Wi-Fi networks; Monitoring wireless networks using Python scripts.

**Unit 7: Network Security Management and Case Studies – 7 hours**  
Network security audit and monitoring; Security policies and compliance; Incident response in network breaches; Lessons from real-world network attacks; Integration of Python scripts for monitoring and alerting.

**Laboratory Experiments (Python-based)**

1. **Port Scanning Tool** — Python script to scan TCP/UDP ports on a host.
2. **Packet Sniffer** — Capture and analyze network packets using scapy.
3. **ARP Spoofing Detection** — Detect potential ARP spoofing attacks using Python.
4. **Simple IDS Simulation** — Create a basic intrusion detection system for network traffic.
5. **VPN Connectivity Test** — Use Python to verify VPN connectivity and encryption.
6. **SSL/TLS Certificate Checker** — Script to check validity and configuration of SSL certificates.
7. **Network Log Analyzer** — Parse firewall or router logs to detect suspicious activities.

**Reference Books**

1. Stallings, W. (2016). *Network Security Essentials: Applications and Standards* (6th ed.). Pearson.
2. Kaufman, C., Perlman, R., & Speciner, M. (2016). *Network Security: Private Communication in a Public World* (2nd ed.). Pearson.
3. Bishop, M. (2018). *Computer Security: Art and Science* (2nd ed.). Addison-Wesley.
4. Whitman, M. E., & Mattord, H. J. (2018). *Principles of Information Security* (6th ed.). Cengage.
5. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.

.

**Course Title: Wireless Security**

**Course Code:** PGDCS 202  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce the principles of wireless network security.
2. Explain vulnerabilities and common attacks in wireless networks.
3. Teach defense mechanisms and best practices for securing wireless communication.
4. Enable students to use Python for monitoring and testing wireless security.
5. Prepare students for roles in wireless network security administration and auditing.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Describe the architecture and protocols of wireless networks.
2. Identify threats and vulnerabilities specific to wireless networks.
3. Implement security measures such as WPA/WPA2/WPA3, access controls, and encryption.
4. Use Python scripts to monitor wireless networks and detect anomalies.
5. Evaluate and recommend strategies to secure wireless networks.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Wireless Networks – 6 hours**  
Wireless network types (Wi-Fi, Bluetooth, ZigBee, cellular); Wireless communication fundamentals; WLAN architecture; IEEE 802.11 standards overview.

**Unit 2: Wireless Threats and Vulnerabilities – 8 hours**  
Eavesdropping, rogue access points, evil twin attacks, replay attacks, MAC spoofing; Denial of Service attacks in wireless networks; Case studies of real-world wireless attacks.

**Unit 3: Wireless Security Protocols – 7 hours**  
WEP, WPA, WPA2, WPA3; Encryption mechanisms; Authentication methods (PSK, Enterprise, 802.1X); Key management in wireless networks.

**Unit 4: Wireless Access Control – 6 hours**  
SSID management, MAC filtering, network segmentation; Captive portals; VPNs for wireless security; Best practices for enterprise wireless networks.

**Unit 5: Wireless Intrusion Detection and Prevention – 7 hours**  
IDS/IPS for wireless networks; Monitoring tools; Anomaly detection; Wireless audit and scanning techniques; Python scripts for monitoring Wi-Fi traffic.

**Unit 6: Mobile Device Security – 7 hours**  
Security risks for smartphones, tablets, and IoT devices; Mobile OS vulnerabilities; Encryption and authentication on mobile devices; MDM (Mobile Device Management) overview.

**Unit 7: Case Studies and Best Practices – 7 hours**  
Analysis of real-world wireless security breaches; Lessons learned; Developing security policies for wireless networks; Integration of Python tools for network monitoring and testing.

**Laboratory Experiments (Python-based)**

1. **Wi-Fi Network Scanner** — Scan nearby wireless networks and display SSID, signal strength, and encryption type.
2. **SSID Spoofing Detection** — Python script to detect duplicate or rogue SSIDs.
3. **Packet Capture for Wi-Fi** — Capture and analyze wireless packets using scapy.
4. **WEP/WPA Cracking Simulation** — Demonstrate vulnerabilities in outdated protocols in a controlled lab.
5. **MAC Address Spoofing Detection** — Detect unauthorized MAC addresses on the network.
6. **Wireless Traffic Analyzer** — Analyze Wi-Fi traffic for unusual patterns or anomalies.
7. **Wireless Security Audit Script** — Automate basic security checks for SSID configuration, encryption, and access controls.

**Reference Books**

1. Gast, M. S. (2013). *802.11 Wireless Networks: The Definitive Guide* (2nd ed.). O’Reilly.
2. He, W., Yan, Q., & Yang, X. (2017). *Wireless Network Security: A Beginner’s Guide*. Wiley.
3. Stallings, W. (2016). *Network Security Essentials: Applications and Standards* (6th ed.). Pearson.
4. Whitman, M. E., & Mattord, H. J. (2018). *Principles of Information Security* (6th ed.). Cengage.
5. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.

**Course Title: Malware Analysis and Digital Forensics**

**Course Code:** PGDCS 203  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce students to malware types, behavior, and propagation methods.
2. Teach fundamentals of digital forensics and evidence handling.
3. Explain tools and techniques for malware detection, analysis, and mitigation.
4. Enable students to perform basic digital forensics and malware analysis using Python.
5. Prepare students for careers in cybersecurity, forensic analysis, and incident response.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Identify different types of malware and their attack vectors.
2. Understand digital forensics concepts and legal considerations.
3. Use Python to analyze malware behavior and automate forensic tasks.
4. Collect, preserve, and analyze digital evidence securely.
5. Apply best practices for incident response and malware mitigation.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Malware – 6 hours**  
Definition of malware; Types (viruses, worms, trojans, ransomware, spyware, adware); Malware life cycle; Malware distribution techniques; Recent case studies.

**Unit 2: Malware Analysis Techniques – 8 hours**  
Static analysis; Dynamic analysis; Sandboxing; Reverse engineering basics; Tools and environments for malware analysis; Python scripts for basic file inspection and signature detection.

**Unit 3: Digital Forensics Fundamentals – 7 hours**  
Overview of digital forensics; Principles of evidence handling; Forensic process (identification, preservation, analysis, presentation); Forensic readiness; Chain of custody.

**Unit 4: File and Memory Forensics – 7 hours**  
Analyzing file systems (FAT, NTFS, ext); Recovering deleted files; Memory analysis basics; Python for forensic file inspection and memory snapshot parsing.

**Unit 5: Network Forensics – 6 hours**  
Capturing and analyzing network traffic; Detecting malware communication patterns; Analyzing logs; Python scripts for packet inspection and anomaly detection.

**Unit 6: Malware Mitigation and Incident Response – 7 hours**  
Containment, eradication, and recovery; Patch management; Antivirus and endpoint protection strategies; Incident response planning; Best practices for mitigation.

**Unit 7: Case Studies and Practical Applications – 7 hours**  
Analysis of real-world malware incidents; Lessons learned; Integration of Python for malware scanning and forensic automation; Developing security policies based on forensic insights.

**Laboratory Experiments (Python-based)**

1. **File Hash Checker** — Generate hashes of files and detect changes for malware detection.
2. **Simple Malware Signature Scanner** — Identify known malware patterns in files.
3. **Memory Dump Analysis** — Parse and analyze memory dumps for suspicious processes.
4. **Network Traffic Analysis** — Capture and analyze packets for malware communication patterns.
5. **Recover Deleted Files** — Use Python scripts to recover deleted files from a test disk image.
6. **Automated Log Analyzer** — Detect anomalies in system or application logs.
7. **Malware Behavior Simulation** — Analyze and report on controlled malware behavior in a sandbox.

**Reference Books**

1. Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.
2. Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet* (3rd ed.). Academic Press.
3. Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2010). *Malware Analyst’s Cookbook and DVD: Tools and Techniques for Fighting Malicious Code*. Wiley.
4. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.
5. Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley.

**Course Title: Cyber Security Regulation**

**Course Code:** PGDCS 204  
**Credits:** 3 (2 Theory + 1 Practical)  
**Total Hours:** 48 (Theory) + Lab Experiments

**Course Objectives**

1. Introduce students to cybersecurity laws, regulations, and compliance frameworks.
2. Explain national and international legal requirements for data protection and privacy.
3. Teach students how to develop security policies in compliance with regulations.
4. Familiarize students with ethical issues and professional responsibilities in cybersecurity.
5. Prepare students to implement regulatory requirements in organizations using practical tools.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Explain major cybersecurity laws and regulations relevant in Nepal and globally.
2. Identify compliance requirements for data protection and privacy.
3. Develop and implement security policies in accordance with regulations.
4. Understand ethical and professional responsibilities in cybersecurity.
5. Use Python scripts and tools to automate compliance monitoring and reporting.

**Detailed Syllabus (48 Theory Hours)**

**Unit 1: Introduction to Cybersecurity Laws – 6 hours**  
Overview of cybersecurity laws; Need for regulation; Legal terminology; Cybercrime types; Historical perspective of cyber laws.

**Unit 2: National Cybersecurity Regulations – 7 hours**  
Cyber laws in Nepal; IT Act, Data Protection and Privacy Laws; Offenses and penalties; Roles of regulatory authorities; Compliance requirements.

**Unit 3: International Cybersecurity Regulations – 7 hours**  
General Data Protection Regulation (GDPR); HIPAA; ISO/IEC 27001; NIST Cybersecurity Framework; Cross-border data regulations; Comparative analysis.

**Unit 4: Data Privacy and Protection – 6 hours**  
Personal data and sensitive information; Data classification; Privacy principles; Encryption and access control; Policies for data protection.

**Unit 5: Cybercrime and Legal Procedures – 8 hours**  
Types of cybercrime (hacking, phishing, identity theft, financial fraud); Investigation procedures; Evidence collection and admissibility; Reporting and prosecution; Role of law enforcement.

**Unit 6: Security Policies and Compliance – 7 hours**  
Developing organizational security policies; Standards and frameworks; Audit and monitoring; Risk management and compliance checks; Using Python scripts for automated monitoring.

**Unit 7: Ethics, Professional Responsibility, and Case Studies – 7 hours**  
Ethical hacking principles; Professional codes of conduct; Intellectual property rights; Ethical dilemmas in cybersecurity; Case studies of regulatory non-compliance and lessons learned.

**Laboratory Experiments (Python-based)**

1. **Policy Compliance Checker** — Python script to verify system configuration against security policy rules.
2. **GDPR Data Mapping Simulation** — Create a simulated database and identify sensitive personal data.
3. **Access Control Audit** — Verify user permissions and access logs using Python scripts.
4. **Log Review for Legal Compliance** — Parse system logs to check for suspicious activities and policy violations.
5. **Data Encryption Validation** — Encrypt and decrypt sensitive data to demonstrate compliance.
6. **Cybercrime Scenario Simulation** — Analyze a controlled incident to understand legal and procedural requirements.
7. **Reporting and Documentation Automation** — Generate automated compliance reports using Python.

**Reference Books**

1. Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.
2. Whitman, M. E., & Mattord, H. J. (2018). *Principles of Information Security* (6th ed.). Cengage.
3. Brenner, S. W. (2011). *Cybercrime: Criminal Threats from Cyberspace* (2nd ed.). Praeger.
4. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.
5. Reed, C., & Zorz, J. (2016). *Cyber Law Handbook: A Guide to Cybersecurity and Privacy Compliance*.

**Course Title: Cyber Security Project**

**Course Code:** PGDCS 205  
**Credits:** 3 (Fully Practical)  
**Total Hours:** 48 (Practical)

**Course Objectives**

1. Provide students with an opportunity to apply theoretical knowledge and practical skills to a real-world cybersecurity problem.
2. Develop problem-solving, analytical, and project management skills.
3. Encourage independent research, experimentation, and reporting in cybersecurity.
4. Promote the integration of Python programming for implementing security solutions.
5. Prepare students for professional roles or higher studies in cybersecurity.

**Course Learning Outcomes (CLOs)**

By the end of this course, students will be able to:

1. Identify and define a cybersecurity problem or research area.
2. Design and implement a practical solution using Python and cybersecurity tools.
3. Conduct testing, validation, and evaluation of the implemented solution.
4. Document project methodology, results, and recommendations professionally.
5. Present and defend their project work to an academic or professional audience.

**Project Guidelines**

**Unit 1: Project Proposal – 6 hours**  
Identify project topic; Define objectives, scope, and expected outcomes; Conduct preliminary research; Submit a project proposal for approval.

**Unit 2: Project Design and Planning – 8 hours**  
Develop system architecture; Define workflow, modules, and tools; Plan data collection and test scenarios; Prepare Gantt chart or timeline for project execution.

**Unit 3: Implementation – 18 hours**  
Code and develop the project solution using Python; Implement security mechanisms, analysis tools, or simulations; Integrate lab techniques learned from other courses.

**Unit 4: Testing and Evaluation – 8 hours**  
Test functionality, performance, and security; Validate results; Identify improvements; Conduct peer review and feedback sessions.

**Unit 5: Documentation and Presentation – 8 hours**  
Prepare detailed report including problem definition, methodology, implementation, results, and conclusions; Develop a presentation; Defend the project before faculty panel.

**Example Project Ideas**

1. **Python-based Network Intrusion Detection System** — Implement simple IDS with alerting functionality.
2. **Malware Analysis Tool** — Automate malware signature detection using Python.
3. **Secure File Storage System** — Encrypt and manage sensitive files with Python.
4. **Wi-Fi Security Analyzer** — Scan and report vulnerabilities in wireless networks.
5. **Cybersecurity Policy Compliance Checker** — Automate auditing of system configurations.
6. **Digital Forensics Toolkit** — Collect and analyze evidence from file systems or network logs.

**Assessment Criteria**

* **Project Proposal:** 10%
* **Design and Implementation:** 40%
* **Testing and Evaluation:** 20%
* **Documentation:** 20%
* **Presentation & Defense:** 10%

**Reference Books**

1. Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.
2. Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis*. No Starch Press.
3. Al-Sakib Khan Pathan (2019). *Python for Cybersecurity*.
4. Stallings, W. (2018). *Computer Security: Principles and Practice* (4th ed.). Pearson.
5. Casey, E. (2011). *Digital Evidence and Computer Crime*. Academic Press.

**Post Graduate Diploma in Cyber Security (PGDCS)**

**Duration:** 1 Year (2 Semesters)  
**Total Credits:** 30 (5 courses per semester, 3 credits each)  
**Mode:** Practical-oriented (Python-based labs)

**Program Rationale**

The rapid growth of digital technologies and increasing cyber threats has created a strong demand for skilled cybersecurity professionals. This program aims to equip students — including non-science undergraduates — with practical and foundational knowledge in cybersecurity. The focus on hands-on Python-based labs ensures that learners can implement, test, and evaluate cybersecurity solutions in real-world scenarios.

The program emphasizes:

* Understanding cyber threats, vulnerabilities, and defense mechanisms.
* Applying cryptography and network security principles.
* Conducting malware analysis, digital forensics, and secure database management.
* Understanding cybersecurity laws, regulations, and ethical responsibilities.
* Developing a practical project demonstrating integrated skills.

**Eligibility**

* Bachelor’s degree in any discipline from a recognized university.
* Basic computer literacy (handling operating systems, MS Office, internet usage).
* No prior programming or science background is required, but interest in technology is preferred.

**Program Structure**

| **Semester** | **Course Code** | **Course Title** | **Credits** | **Hours (Theory + Lab)** |
| --- | --- | --- | --- | --- |
| I | PGDCS 101 | Fundamentals of Cyber Security | 3 | 48 + 24 |
| I | PGDCS 102 | Python Programming for Cybersecurity | 3 | 48 + 24 |
| I | PGDCS 103 | Fundamentals of Cryptography | 3 | 48 + 24 |
| I | PGDCS 104 | Database Management Systems (with Security) | 3 | 48 + 24 |
| I | PGDCS 105 | Computer Networks & Operating Systems | 3 | 48 + 24 |
| II | PGDCS 201 | Network Security | 3 | 48 + 24 |
| II | PGDCS 202 | Wireless Security | 3 | 48 + 24 |
| II | PGDCS 203 | Malware Analysis & Digital Forensics | 3 | 48 + 24 |
| II | PGDCS 204 | Cyber Security Regulation | 3 | 48 + 24 |
| II | PGDCS 205 | Cyber Security Project | 3 | 48 (Practical only) |

**Total Credits:** 30  
**Total Contact Hours:** 480 (Theory + Lab + Project)

**Teaching and Learning Approach**

* **Lectures:** Conceptual understanding and theory.
* **Practical Labs:** Python-based hands-on exercises for each course.
* **Case Studies:** Real-world examples of cyber attacks and mitigation.
* **Project:** Application of learned concepts to a practical cybersecurity problem.
* **Assessment:** Mix of theory, lab exercises, and project evaluation.

**Career Pathways after Completion**

Graduates of this program can pursue:

**Professional Roles:**

* Cybersecurity Analyst / Specialist
* Network Security Administrator
* Information Security Officer
* Digital Forensics Analyst
* Malware Analyst
* Penetration Tester (Entry-level)

**Further Studies / Certifications:**

* Master’s programs in Cybersecurity, Information Security, or Computer Science
* Professional certifications such as:
  + CompTIA Security+
  + Certified Ethical Hacker (CEH)
  + Certified Information Systems Security Professional (CISSP)
  + Offensive Security Certified Professional (OSCP)
  + Certified Digital Forensics Examiner (CDFE)

This **comprehensive structure** ensures that students with non-technical backgrounds can gain practical cybersecurity expertise and are prepared for both employment and higher studies.

**PGDCS Program: Semester-wise Course Map**

**Semester 1: Foundation & Core Skills (15 Credits)**

| **Course Code** | **Course Title** | **Credits** | **Theory Hours** | **Lab Hours** | **Focus** |
| --- | --- | --- | --- | --- | --- |
| PGDCS 101 | Fundamentals of Cyber Security | 3 | 48 | 24 | Cybersecurity basics, threats, security principles |
| PGDCS 102 | Python Programming for Cybersecurity | 3 | 48 | 24 | Python programming and automation |
| PGDCS 103 | Fundamentals of Cryptography | 3 | 48 | 24 | Encryption, symmetric/asymmetric cryptography |
| PGDCS 104 | Database Management Systems (with Security) | 3 | 48 | 24 | Secure database design, SQL, Python integration |
| PGDCS 105 | Computer Networks & Operating Systems | 3 | 48 | 24 | Network concepts, OS principles, system security |

**Total Credits Semester 1:** 15  
**Total Hours (Theory + Lab):** 240

**Semester 2: Advanced Security & Project (15 Credits)**

| **Course Code** | **Course Title** | **Credits** | **Theory Hours** | **Lab Hours** | **Focus** |
| --- | --- | --- | --- | --- | --- |
| PGDCS 201 | Network Security | 3 | 48 | 24 | Network attacks, firewalls, IDS/IPS, Python-based monitoring |
| PGDCS 202 | Wireless Security | 3 | 48 | 24 | Wi-Fi threats, encryption, wireless monitoring with Python |
| PGDCS 203 | Malware Analysis & Digital Forensics | 3 | 48 | 24 | Malware types, analysis, memory/file/network forensics |
| PGDCS 204 | Cyber Security Regulation | 3 | 48 | 24 | Laws, compliance, ethical frameworks, policy enforcement |
| PGDCS 205 | Cyber Security Project | 3 | 0 | 48 | Practical integration of all learned skills in a real-world project |

**Total Credits Semester 2:** 15  
**Total Hours (Theory + Lab + Project):** 240

**Visual Flow of Skills Development**

1. **Semester 1 (Foundation) →**
   * Cybersecurity basics → Python skills → Cryptography → Secure databases → Networks & OS fundamentals  
     *(Builds the core technical knowledge required for advanced security tasks)*
2. **Semester 2 (Advanced & Application) →**
   * Network Security → Wireless Security → Malware & Forensics → Regulations → Project  
     *(Applies knowledge in hands-on labs and real-world scenarios, culminating in a project)*

**Key Features Highlighted in the Map**

* **Hands-on learning:** Every course includes Python-based lab exercises.
* **Progressive learning:** Starts from fundamentals and moves to advanced topics.
* **Integrated project:** Students consolidate all practical skills in Semester 2.
* **Non-technical friendly:** Designed for non-science graduates with practical-oriented learning.
* **Career-ready:** Prepares students for jobs, certifications, or higher studies.

**Post Graduate Diploma in Cyber Security (PGDCS)**

**Duration:** 1 Year (2 Semesters)  
**Total Credits:** 30 (5 courses per semester, 3 credits each)  
**Mode:** Practical-oriented (Python-based labs)

**1. Program Rationale**

The rapid digitalization of services and increasing cyber threats demand skilled professionals capable of preventing, detecting, and mitigating cyber attacks. This program equips students — including non-science graduates — with practical skills in cybersecurity fundamentals, cryptography, network security, malware analysis, forensics, and regulations. Hands-on Python-based labs ensure that learners gain real-world problem-solving skills applicable in workplaces and research.

**2. Eligibility**

* Bachelor’s degree in any discipline from a recognized university.
* Basic computer literacy (OS, internet, MS Office).
* No prior programming or science background required.

**3. Program Structure**

**Semester 1 (15 Credits)**

| **Course Code** | **Course Title** | **Credits** | **Theory Hours** | **Lab Hours** | **Focus** |
| --- | --- | --- | --- | --- | --- |
| PGDCS 101 | Fundamentals of Cyber Security | 3 | 48 | 24 | Cybersecurity basics, threats, security principles |
| PGDCS 102 | Python Programming for Cybersecurity | 3 | 48 | 24 | Python programming and automation |
| PGDCS 103 | Fundamentals of Cryptography | 3 | 48 | 24 | Encryption, symmetric/asymmetric cryptography |
| PGDCS 104 | Database Management Systems (with Security) | 3 | 48 | 24 | Secure database design, SQL, Python integration |
| PGDCS 105 | Computer Networks & Operating Systems | 3 | 48 | 24 | Network concepts, OS principles, system security |

**Total Credits Semester 1:** 15  
**Total Hours (Theory + Lab):** 240

**Semester 2 (15 Credits)**

| **Course Code** | **Course Title** | **Credits** | **Theory Hours** | **Lab Hours** | **Focus** |
| --- | --- | --- | --- | --- | --- |
| PGDCS 201 | Network Security | 3 | 48 | 24 | Network attacks, firewalls, IDS/IPS, Python-based monitoring |
| PGDCS 202 | Wireless Security | 3 | 48 | 24 | Wi-Fi threats, encryption, wireless monitoring with Python |
| PGDCS 203 | Malware Analysis & Digital Forensics | 3 | 48 | 24 | Malware types, analysis, memory/file/network forensics |
| PGDCS 204 | Cyber Security Regulation | 3 | 48 | 24 | Laws, compliance, ethical frameworks, policy enforcement |
| PGDCS 205 | Cyber Security Project | 3 | 0 | 48 | Practical integration of all learned skills in a real-world project |

**Total Credits Semester 2:** 15  
**Total Hours (Theory + Lab + Project):** 240

**4. Teaching and Learning Methods**

* Lectures for conceptual understanding
* Python-based hands-on labs
* Case studies of real-world cybersecurity incidents
* Semester project integrating knowledge from all courses
* Assessment through theory exams, lab exercises, and project evaluation

**5. Career Pathways**

**Professional Roles**

* Cybersecurity Analyst / Specialist
* Network Security Administrator
* Information Security Officer
* Digital Forensics Analyst
* Malware Analyst
* Penetration Tester (Entry-level)

**Further Studies / Certifications**

* Master’s programs in Cybersecurity, Information Security, or Computer Science
* Certifications: CompTIA Security+, CEH, CISSP, OSCP, CDFE

**6. Course-wise Summary (Theory + Labs)**

| **Course** | **Theory Hours** | **Lab Hours** | **Total Hours** | **Key Focus** |
| --- | --- | --- | --- | --- |
| Fundamentals of Cyber Security | 48 | 24 | 72 | Cyber threats, security principles, risk management |
| Python Programming | 48 | 24 | 72 | Python basics, scripting for security tasks |
| Fundamentals of Cryptography | 48 | 24 | 72 | Symmetric & asymmetric encryption, hashing, digital signatures |
| Database Management Systems | 48 | 24 | 72 | Secure databases, SQL, Python integration |
| Computer Networks & OS | 48 | 24 | 72 | Networks, OS basics, security configurations |
| Network Security | 48 | 24 | 72 | Firewalls, IDS/IPS, secure protocols, Python-based monitoring |
| Wireless Security | 48 | 24 | 72 | Wi-Fi security, attacks, monitoring, Python tools |
| Malware & Digital Forensics | 48 | 24 | 72 | Malware types, analysis, file/network/memory forensics |
| Cyber Security Regulation | 48 | 24 | 72 | Cyber laws, compliance, ethical frameworks, policy enforcement |
| Project | 0 | 48 | 48 | Practical integration of all knowledge and skills |

**7. Assessment Methods**

* **Theory exams:** 40%
* **Laboratory exercises:** 30%
* **Project report & defense:** 20%
* **Assignments and case studies:** 10%

**8. Program Outcomes**

1. Graduates will have practical and theoretical knowledge of cybersecurity concepts.
2. They will be proficient in Python for automating cybersecurity tasks.
3. They will understand network, database, OS, and wireless security fundamentals.
4. They will be capable of performing malware analysis, digital forensics, and compliance auditing.
5. Graduates will be prepared for professional roles or further studies in cybersecurity.