Suggested Practicals

It is suggested that the following tools/e-resources can be explored during the practical sessions • Wireshark • COFEE Tool • Magnet RAM Capture • RAM Capture • NFI Defragger • Toolsley • Volatility

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.
5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image
6. Using System internal tools for network tracking and process monitoring do the following:
    a. Monitor live processes
    b. Capture RAM
    c. Capture TCP/UDP packets
    d. Monitor Hard disk
    e. Monitor Virtual Memory
    f. Monitor Cache Memory

DSC20/DSC08/GE8a: INFORMATION SECURITY

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Information Security	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objective

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

Learning Outcomes

On successful completion of this course, a student will be able to

- Identify the major types of threats to information security.
- Describe the role of cryptography in security.
- Discover the strengths and weaknesses of private and public key cryptosystems.
- Identify and apply various access control and authentication mechanisms.
- Discuss data and software security and related issues.
- Explain network security threats and attacks.
- Articulate the need for security in cloud and IoT.

Syllabus

Unit 1 **(3 hours)**

Overview: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles.

Unit 2 **(6 hours)**

Cryptographic tools: Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication

Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

The Need for Database Security, SQL Injection Attacks, Database Access Control.

Unit 5 **(8 hours)**

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. **Overflow Attacks** - Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. **Software Security Issues** - Handling Program Input, Writing Safe Program Code, Handling Program Input.

Unit 6 **(6 hours)**

Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Public-Key Infrastructure.

Unit 7 **(7 hours)**

Wireless, Cloud and IoT Security: Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security.

References

1. W. Stallings, L. Brown, *Computer Security: Principles and Practice*, 4th edition, Pearson Education, 2018.

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications*, 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.

5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

COMMON POOL OF DISCIPLINE ELECTIVE COURSES (DSE) COURSES

Computer Science Courses for all Undergraduate Programmes of study with Computer Science as Discipline Elective

DISCIPLINE SPECIFIC ELECTIVE COURSE: Information and Image

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE8a: Information and Image Retrieval	**4**	**3**	**0**	**1**	Pass in Class XII	Digital Image Processing

Course Objective

This course introduces students to the fundamentals of information retrieval extending into image retrieval. It lays the theoretical foundation of various essential concepts related to image searches, together with examples of natural and texture image types. It will provide insight to content-based image retrieval, understanding of the technologies, and solutions of content-based image retrieval.

Course Learning Outcomes

On the successful completion of the course, the student would be able to:

1. Understand the concept of information retrieval and the information retrieval models.
2. Understand the working of Text based and content based image retrieval systems.
3. Identify and evaluate the use of content-based features in indexing and retrieval of various types of media content
4. Extract different visual features from images
5. Understand indexing and the semantics of visual data
6. Understand query specification and evaluate the retrieval

Syllabus

Unit 1 Introduction to IR: An example information retrieval problem, the extended Boolean model versus ranked retrieval, The term vocabulary and postings lists: Tokenization, stop words, Normalization (equivalence classing of terms), Stemming and lemmatization, term weighting model: Inverse document frequency, Tf-idf weighting, Information retrieval system evaluation.

Unit 2 CBIR and feature extraction: Image Retrieval: Multimedia Information retrieval , Text Based Image Retrieval (TBIR), Content Based Image Retrieval (CBIR), Hybrid systems. Architecture of a typical CBIR system, Low-level features of an image: Color – color space, color moments, color histogram, color coherence vector (CCV), color correlogram, invariant color features.

Texture – Tamura features, coarseness, contrast, SAR Model, Wavelet transform feature. Shape- Moment invariants, turning angles, Fourier descriptors.

Unit 3 Similarity measures and Performance evaluation: Similarity measures used in content-based image retrieval: Minkowski-form distance, Mahalanobis distance, Canberra distance, Earth Mover distance, Quadratic form distance

Performance evaluation used in content-based image retrieval: user Comparison, precision and recall, P-R graph, Average Precision, F-measure, Average Normalized Modified Retrieval Rank (ANMRR)

Unit 4 CBIR systems: QBIC: Query by Image Content, VIR, VisualSEEK, WebSEEK, NeTRA, MARS: Multimedia Analysis and Retrieval System, SIMPLIcity.

Unit 5 Content-Based Image Retrieval-Challenges: Semantic gap: Introduction to semantic gap. Bridging the semantic gap: Relevance feedback, multi-modal fusion. Semantic similarity: WordNet.

"Curse of Dimensionality": Feature Dimensionality reduction, Methods for dimensionality reduction: Principal Component Analysis (PCA), Fisher Linear Discriminant Analysis (FLDA), Local Fisher Discriminant Analysis (LFDA), Isometric Mapping (ISOMAP), Locally Linear Embedding (LLE), and Locality Preserving Projections (LPP).

References

1. C. Manning, P. Raghavan, and H. Schutze *Introduction to Information Retrieval* Cambridge University Press, 2009 .

2. Vipin Tyagi *Content-Based Image Retrieval: Ideas, Influences, and Current Trends*, Springer, 2018.

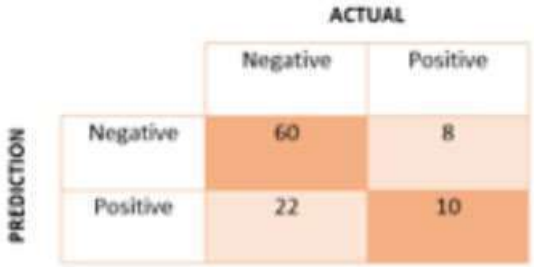
Suggested Practical List

To be implemented in Python

1. Write a program to compute the edit distance between strings s1 and s2. (Hint. Levenshtein Distance)
2. Write a program to Compute Similarity between two text documents.
3. Write a program for Pre-processing of a Text Document: stop word removal.
4. Consider 3 documents as below:-

   Doc 1: Ben studies about computers in Computer Lab.
   Doc 2: Steve teaches at Brown University.
   Doc 3: Data Scientists work on large datasets.

   Perform search on these documents with the following query: Data Scientists and, calculate tf * idf for data and Scientists in all the documents.
5. Write a program to find out the similarity between document d1 and d2 (refer question#4) using cosine similarity method.
6. Write a program to calculate the color moments, color histogram, color coherence vector (CCV), color correlogram for an image.
7. Write a program to find out the similarity between two images using:-
   a. Minkowski-form distance
   b. Mahalanobis distance
   c. Canberra distance
   d. Earth Mover distance
   e. Quadratic form distance
8. Given a confusion matrix



Write a program to find precision and recall, Average Precision, F-measure, Average Normalized Modified Retrieval Rank (ANMRR).

DISCIPLINE SPECIFIC ELECTIVE COURSE: Natural Language Processing

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE8b: Natural Language Processing	**4**	**3**	**0**	**1**	Pass in Class XII	Machine Learning

Course Objective

The objectives of this course are:

1. To introduce foundational understanding in natural language
2. To understand the principles and methods of statistical natural language processing
3. To develop an in-depth understanding of the algorithms available for the processing and analysis of natural languages
4. To perform statistical analysis of textual data and find useful patterns from the data

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Grasp the significance of natural language processing in solving real-world problems
2. Preprocess and Analyze text using mathematical techniques.
3. Apply machine learning techniques used in NLP - HMM, RNN
4. Understand approaches to syntax and semantics analysis in NLP
5. Gain practical experience of using NLP toolkits

Syllabus

Unit 1 Introduction and Basic Text Processing: Knowledge in Speech and Language Processing, The problem of ambiguity, Typical NL Tasks, Tokenization, Stemming, Lemmatization, Stop-word removal

Unit 2 Formal Language Modeling: Regular Expressions, Text Normalization, and Edit Distance, Unigrams, Bigrams, N-grams, N-gram Language Models, Smoothing and Entropy

Unit 3 Sequence Labeling for Parts of Speech Tagging: Part-of-Speech Tagging, Named Entities and Named Entity Tagging/Recognition, Hidden Markov Model (Forward and Viterbi algorithms and EM training)

Unit 4 Vector Semantics and Embedding: Lexical Semantics, Vector Semantics, Words and Vectors, TF-IDF: Weighing terms in the vector and its applications, Learning Word Embeddings - Word2vec and Gensim, Vector Space Models

Unit 5 Applications of Text Mining: Text classification, Sentiment Analysis

Unit 6 Deep Learning Models for NLP: Feedforward Neural Networks, Recurrent Neural Networks, and LSTM

References

1. Daniel Jurafsky and James H. Martin *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 3rd edition, Pearson, 2022.
2. Christopher D. Manning and Hinrich Schütze *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
3. Steven Bird, Ewan Klein, and Edward Loper *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*, 1st edition, O'Reilly Media, 2009.

Additional Reference

(i) Yoav Goldberg *A Primer on Neural Network Models for Natural Language Processing, 2022.*

Suggested Practical List

Python Packages like Scikit (SKLearn), NLTK, spaCy, gensim, PyTorch, transformers (HuggingFace) etc. may be used for programming

1. Prepare/Pre-process a text corpus to make it more usable for NLP tasks using tokenization, filtration of stop words, removal of punctuation, stemming and lemmatization.
2. List the most common words (with their frequency) in a given text excluding stopwords.
3. Extract the usernames from the email addresses present in a given text. .
4. Perform POS tagging in a given text file. Extract all the nouns present in the text. Create and print a dictionary with frequency of parts of speech present in the document. Find the similarity between any two text documents
5. Perform dependency analysis of a text file and print the root word of every sentence.
6. Create the TF-IDF (Term Frequency -Inverse Document Frequency) Matrix for the given set of text documents
7. Extract all bigrams , trigrams using ngrams of nltk library
8. Identify and print the named entities using Name Entity Recognition (NER) for a collection of news headlines.
9. Find the latent topics in a document using any LDA and display top 5 terms that contribute to each topic along with their strength. Also visualize the distribution of terms contributing to the topics.
10. Classify movie reviews as positive or negative from the IMDB movie dataset of 50K movie reviews. (Link for dataset:

    https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews)

0. Build and train a text classifier for the given data (using textbob or simpletransformers library)

0. Generate text using a character-based RNN using an appropriate dataset. Given a sequence of characters from a given data (eg "Shakespear"), train a model to predict the next character in the sequence ("e").

DISCIPLINE SPECIFIC ELECTIVE COURSE: Blockchain and its Applications

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE8c: Blockchain and its Applications	**4**	**3**	**0**	**1**	Pass in Class XII	A course in any Programming Language, Database Management Systems

Course Objective

This course covers the basic concepts behind blockchain and presents Bitcoin and other cryptocurrencies as the motivation for blockchain technologies. It provides a substantive discussion about different technologies behind blockchain and cryptocurrencies.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. understand the applications of blockchain in different domains
2. learn the practical applications of cryptocurrency such as Bitcoin and Ethereum
3. understand basic technologies like cryptographic hash functions, blocks, merkel trees, elliptic curve cryptography and digital signatures.
4. to have knowledge of decentralized consensus algorithms like proof of work, proof of stack, proof of capacity etc.
5. to learn how to record transactions in blockchain, computing bitcoin address etc.
6. to learn about smart contracts and their applications
7. to learn about permissioned and permission less blockchain and hyperledgers.
8. to gain knowledge of real world aspects of Bitcoin, such as wallets and mining techniques. with the Bitcoin network.

Syllabus

Unit 1 Introduction: History of money, Digital Currencies, Ledgers, Cryptography, Centralized and Decentralized systems, peer to peer systems, the purpose of Blockchain, types of blockchain (public, private and semi-private blockchain), application of blockchain (in government, healthcare, real estate, voting, insurance, non-fungible tokens, metaverse, Web 3.0).

Unit 2 Cryptocurrency and Design: Concept of cryptocurrency, History of Bitcoin, concept of mining, challenges of blockchain/bitcoin design (performance, scalability, efficiency, security, governance, public policy and legal framework).

Unit 3 Blockchain Technology: Properties of hash functions, Cryptographic hash functions, hashes (as names, references and commitments), Blocks, Block Headers, Merkel Trees, chain forks, Asymmetric Cryptography, Digital signatures.

Unit 4 Decentralized Network Consensus: Introduction to decentralized networks, Native Currency, consensus, proof of work (PoW), proof of stake (PoS), proof of capacity (PoC), proof of burn (PoB), Practical Byzantine Fault Tolerance (pBFT), Proof of Elapsed Time (PoET).

Unit 5 Permissioned and Permissionless blockchain: Blockchain systems vs. traditional databases, introduction to permissioned/permissionless blockchains and their applications, Advantages and disadvantages, Solidity.

Unit 6 Blockchain and Money Transactions: Satoshi and Bitcoin, Recording of transactions in blockchain, transaction inputs, outputs and format, Bitcoin address.

Unit 7 Smart contracts (Ethereum and other currencies): Overview of smart contracts, tokens and Ethereum as a platform for smart contracts, blockchain technology as regulatory authority.

References

1. Imran Bashir *Mastering blockchain Distributed ledger technology, decentralization, and smart contracts explained*, 2nd edition, Packt Publication, 2018.
2. Lorne Lantz and Daniel Cawrey *Mastering Blockchain Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications*, 1st edition, O'Reilly Publication, 2020.
3. Chris Dannen *Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners* ,1st edition, Apress Publication, 2017.

Additional Reference

(i) Daniel Drescher *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 1st edition, Apress Publication, 2017.

Suggested Practical List

Use any programming language to implement the following:

1. Using SHA256, obtain the message digest of string "Blockchain Developer".
2. Write a program to encrypt and decrypt the message "Hello World" using SHA256.
3. Implement RSA cryptographic algorithm.
4. Create a simple blockchain using Proof of Work (PoW).
5. Demonstrate sending of a digitally signed document.
6. Create a blockchain block containing block hash, transaction history, time of creation.
7. Create a blockchain having 5 nodes and print the hash values of each block.
8. Create a blockchain having 5 nodes and check its validity.
9. Implement a smart contract using solidity programming language.
10. Create a simple permissioned blockchain using Hyperledger Fabric.

2. Q. Kong, T. Siauw, A. Bayen, *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*, 1st edition, 2020.

Suggested Practical List (If any)

:(30 Hours)

Practical exercises such as

Write programs to implement the following methods:

Constrained and Unconstrained Optimization, Global and Local Optimization, Line Search and Trust Region, Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent, Newton's Method, Quasi-Newton Methods, The Cauchy Point algorithm, Finite-Difference Derivative Approximations, Convergence to Stationary Points, Conjugate Gradient Method, Rate of Convergence, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions, and Projected Hessians. Linear and non-linear constrained optimization Augmented Lagrangian Methods.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GE7e/DSE: ETHICAL HACKING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Ethical Hacking	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objectives

The objective of this course is to enable students to be part of such a team that can conduct the security assessment of an organization through the process of ethical hacking. This course will introduce the students, the idea of security assessment of systems and networks under investigation and how to perform them under the legal and ethical framework. Further, this course will outline the importance of various stages of ethical hacking, including but not limited to tasks such as penetration testing, and usage of various tools at each stage.

Learning outcomes

On successful completion of the course, students will be able to:

1. Understand and acknowledge the relevance of legal, ethical, and professional challenges faced by an ethical hacker.
2. Apply fundamental principles of system, application, and network security to ethically attack / penetrate the system to uncover the security flaws.
3. Perform evaluation of security systems through a systematic ethical hacking process and recommend countermeasures to improve security.
4. Understand and use various tools and techniques used in various stages of the ethical hacking process.

Syllabus

Unit 1 **(4 Hours)**

Introduction: Overview of information security threats and attack vectors, vulnerability assessment and penetration testing concepts, information security controls, security laws and standards. OWASP top 10 vulnerabilities

Unit 2 **(6 hours)**

Footprinting and Reconnaissance: Introduction to network reconnaissance tools such ipconfig, ifconfig, domain tools, nmap, Wireshark, etc.

Unit 3 **(8 hours)**

Scanning and Enumeration: Network penetration testing, Password cracking techniques and countermeasures, NetBIOS tools

Unit 4 **(8 hours)**

Gaining and Maintaining Access: Network level attacks and countermeasures, Metasploit framework, Burp Suite

Unit 5 **(8 hours)**

Exploitation and Covering Tracks: Privilege escalation, social Engineering**,** identity theft, countermeasures, Covering tracks using attrib command and creating Alternate Data Stream (ADS) in Windows, Erasing evidence from Windows logs, Strategies for maintaining access.

Unit 6 **(8 hours)**

Advanced stages: Denial of service, Session hijacking, hacking web servers, hacking web applications, sql injection etc.

Unit 7 **(8 hours)**

NIST Cybersecurity framework and ISO standards: NIST cybersecurity framework, Cyber Kill chain, ISO/IEC 27001 and related standards.

Unit 8 **(4 Hours)**

Cyber Defense and Reporting: Preparing vulnerability assessment reports, presenting post testing findings, preparing recommendations

References

1. Patrick Engbretson, The Basics of Hacking and Penetration Testing, 2nd Edition, Syngress, 2013.
2. Georgia Weidman, Penetration TEsting: A Hands-On Introduction to Hacking, 1st Edition, No Starch Press, 2014.

Additional References

1. Peter Kim, The Hacker Playbook 3: Practical Guide to Penetration Testing, Zaccheus Entertainment, 2018.
2. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press, 2008.
3. Online Resources:

https://www.sans.org/cyberaces/

https://skillsforall.com/

https://www.hackingloops.com/ethical-hacking/

Suggested Practical List (If any): (30 Hours)

Perform the following activities, record and report in standard form.

(NOTE: Exercise extra caution while performing these exercises and codes)

1. Perform various Virtual Machine based exercises on https://vulnhub.com/
2. Perform Capture the Flag (CTF) exercises from https://www.hacker101.com/
3. Follow the lessons and activities from https://www.hackingloops.com/ethical-hacking/
4. Google site for hacking https://google-gruyere.appspot.com/
5. OWASP WebGoat https://github.com/WebGoat/WebGoat

GE8d/DSE: CYBER FORENSICS

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

3. Implement Breadth-First Search (Synchronous Network)
4. Implement Maximal Independent Set (Synchronous Network)
5. Implement Leader Election in an Asynchronous Ring.
6. Implement Asynchronous Banking System (Optional)
7. Implement distributed consensus with link failure (Synchronous Network)
8. Implement distributed consensus with Process failure (Synchronous Network)

DISCIPLINE SPECIFIC ELECTIVE COURSE: Cloud Computing

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE8e: Cloud Computing	**4**	**3**	**0**	**1**	Pass in Class XII	

Course Objective:

The objective of an undergraduate cloud computing course is to provide students with a comprehensive understanding of cloud computing technologies, services, and applications.

Course Learning Outcomes:

On successful completion of this course, the student will be able to:

1. Apply the fundamental concepts and principles of cloud computing, including virtualisation, scalability, reliability, and security.
2. design, develop, and deploy cloud-based applications using popular cloud platforms and services.

3. understand cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. Visualise the economic, legal, and ethical implications of cloud computing, including issues related to data privacy, ownership, and security.
5. evaluate and select cloud-based solutions based on their technical, economic, and business requirements.
6. understand of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: **(6 hours)**

Overview of Computing Paradigm: Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: **(7 hours)**

Introduction to Cloud Computing :History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

Unit 3: **(12 hours)**

Cloud Computing Architecture: Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

Unit 4: **(7 hours)**

Case Studies :Case study of Service model using Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, Eucalyptus.

Unit 5: **(6 hours)**

Cloud Computing Management : Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

Unit 6:

(7 hours)

Cloud Computing Security: Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

References:

1. Thomas Erl, Ricardo Puttini and Zaigham Mahmood, Cloud Computing: Concepts, Technology and Architecture, Publisher: PHI, 2013.
2. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2013.
3. Boris Scholl, Trent Swanson, and Peter Jausovec, Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications, Publisher : Shroff/O'Reilly, 2019.

Additional References:

1. *Cloud Computing Bible,* Barrie Sosinsky, *Wiley-India,* 2010
2. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, *Wile,* 2011
3. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam, *Springer,* 2012
4. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing,* Ronald L. Krutz, Russell Dean Vines, *Wiley-India,* 2010

Suggested Practical List:

1. Introduction to Cloud Platforms

   Objective: Familiarize students with cloud platforms and their interfaces.

   Steps:

   d) Create free-tier accounts on AWS, Azure, and GCP.

   e) Explore dashboards and identify key services (compute, storage, networking).

   f) Understand pricing calculators on each platform.

2. Launch Your First Amazon EC2 Instance

   Objective: Deploy a virtual machine on AWS using Amazon EC2.

Steps:

e) Launch an EC2 instance from the AWS Management Console.

f) Use a pre-configured AMI (e.g., Amazon Linux 2).

g) Configure security groups to allow SSH access.

h) Connect to the instance using SSH.

3. Set Up a VPC

Objective: Create and configure a Virtual Private Cloud (VPC).

Steps:

a) Create a custom VPC with a public and private subnet.

b) Launch an EC2 instance in the public subnet and another in the private subnet.

c) Configure an Internet Gateway for Internet access in the public subnet.

d) Use a NAT Gateway to provide internet access for instances in the private subnet.

4. Configure Auto Scaling and Load Balancing

Objective: Set up an auto-scaling group and a load balancer

Steps:

e) Create an Auto Scaling Group and define a launch template.

f) Configure scaling policies (e.g., scale up when CPU utilization exceeds 70%).

g) Deploy an Application Load Balancer (ALB) to distribute traffic.

h) Test auto-scaling by simulating high traffic.

5. Deploying a Static Website on the Cloud

Objective: Host a static website using cloud storage services.

Steps:

c) Deploy a static website using any of the following:

- AWS S3
- Azure Blob Storage
- GCP Cloud Storage

d) Configure permissions and enable public access.

6. Monitor Resources Using AWS CloudWatch

Objective: Use CloudWatch to monitor AWS resources

Steps

e) Set up CloudWatch metrics for an EC2 instance (e.g., CPU utilization).

f) Create a CloudWatch Alarm to send notifications when a threshold is exceeded.

g) Configure an SNS topic for email notifications.

h) Test the setup by simulating high CPU usage.

7. Install OpenStack

Objective: Set up a local OpenStack environment for practice.

8. Launch Your First Instance

Objective: Create a virtual machine (VM) using OpenStack.

Steps:

e) Create a project and assign roles to users.

f) Upload an image (e.g., Ubuntu cloud image) to the Glance service.

g) Define a flavor to specify VM configurations.

h) Launch an instance using the Horizon dashboard or CLI.

Resources Needed:

- OpenStack Horizon access or CLI setup.
- Sample Ubuntu or CentOS cloud image (from Ubuntu Cloud Images).

9. Set Up Networking

Objective: Configure OpenStack Neutron to provide networking for instances.

Steps:

d) Create a private network and a public network.

e) Attach a router to connect the private network to the public network.

f) Assign floating IPs to instances for external access.

10. Cloud Security

Objective: Understand security practices in the cloud.

Steps:

e) Implement IAM roles and policies for a cloud platform.

f) Create and assign least-privilege roles to users.

g) Configure data encryption for storage (e.g., S3 bucket encryption).

h) Set up a firewall rule and test its functionality.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Reinforcement Learning

Credit distribution, Eligibility, and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7c: Reinforcement Learning	**4**	**3**	**1**	**0**	Pass in Class XII	Machine Learning/Artificial Intelligence

Learning Objectives

The objectives of this course are:

1. to prepare students to visualize reinforcement learning problems
2. to introduce students to the concepts based on Markov Decision Process, Dynamic Programming, Monte Carlo methods, and Temporal-Difference learning.
3. recognize current advanced techniques and applications in Reinforcement Learning

Learning Outcomes

On successful completion of the course, students will be able to:

1. learn Reinforcement Learning task formulations and the core principles behind Reinforcement Learning.
2. work on problem-solving techniques based on Dynamic Programming, Monte Carlo, and Temporal-Difference.
3. implement in code common algorithms following code standards and libraries used in Reinforcement Learning.
4. learn the policy gradient methods from vanilla to relatively complex cases.

Syllabus

Unit 1 Introduction: Historical perspective of Reinforcement Learning (RL), Basics of RL: definition, how reinforcement learning happens, examples, terminology, notation, and assumptions, Elements of RL: polices, value function, reward Functions and Bellman Equation, different techniques for solving RL problem, Code Standards and Libraries used in RL using Python/Keras/TensorFlow/MATLAB.

Unit 2 Markov Decision Process (MDP) and Dynamic Programming (DP): Markov property, Introduction to Markov decision process (MDP), creating MDPs, goals and rewards, returns and episodes, optimality of value functions and policies, Bellman optimality equations. Overview of dynamic programming for MDP, principle of optimality, iterative policy evaluation, Policy Improvement, policy iteration, value iteration, generalized policy iteration, Asynchronous DP, Efficiency of DP.

Unit 3 Monte Carlo (MC) Methods: Monte Carlo methods (First visit and every visit Monte Carlo), Monte Carlo control, On policy and off policy learning, Importance sampling.

Unit 4 Temporal Difference (TD) Learning: Temporal-Difference learning methods - TD (0), SARSA, Q-Learning and their variants. Markov reward process (MRP), Overview of TD (1) and TD(λ).

Unit 5 Approximation Methods and Policy Gradient: Function approximation methods (Gradient MC and Semi-gradient TD (0) algorithms), Eligibility traces, After-states, Least squares TD. Policy Approximation and its advantages, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods, an introduction to Deep Reinforcement Learning

References

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction* 2nd Edition, MIT Press, 2018.
2. Enes Bilgin *Mastering Reinforcement Learning with Python: Build next-generation, self-learning models using reinforcement learning techniques and best practices*, 1st edition, Packt Publishing, 2020.

Additional References

(i) Phil Winder *Reinforcement Learning: Industrial Applications of Intelligent Agents*, O'Reilly Media, 2020.
(ii) Alexander Zai, Brandon Brown *Deep Reinforcement Learning in Action*, 1st edition, Manning Publications, 2020.

Suggested Practical List

Implement the following exercises using Python/Keras/TensorFlow/MATLAB.

1. Dynamic Programming Policy Evaluation algorithm.
2. Dynamic Programming Policy Iteration algorithm.
3. Dynamic Programming Value Iteration algorithm.
4. Monte Carlo Prediction
5. Off-Policy Monte Carlo Control with Importance Sampling
6. SARSA On policy TD learning algorithm
7. Q-learning OFF policy TD learning algorithm.
8. Policy Gradient REINFORCE algorithm
9. Policy Gradient Actor-Critic method algorithm

**For exercises 1 to 7, consider the following environments for testing: GridWorld, Blackjack, WindyGridWorld*

**For exercises 8 onward, consider the following environments for testing: CartPole, CartPoleRaw*

DSC-A6/DSE: DEEP LEARNING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Deep Learning	**4**	**3**	**0**	**1**	Pass in Class XII	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing

Course Objectives

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Learning outcomes

On successful completion of this course, the student will be able to:

• Describe the feed-forward and deep networks.

• Design single and multi-layer feed-forward deep networks and tune various hyper parameters.

• Implement deep neural networks to solve a problem

• Analyze performance of deep networks.

• Use pre-trained models to solve a problem.

SYLLABUS

Unit 1 **(8 Hours)**

Introduction to neural networks:

Artificial neurons, perceptron, computational models of neurons, Structure of neural networks, Multilayer feedforward neural networks (MLFFNN), Backpropagation learning, Empirical risk minimization, bias-variance tradeoff, Regularization, output units: linear, softmax , hidden units:tanh, RELU

Unit 2 **(8 Hours)**

Deep neural networks:

Difficulty of training DNNs, Greedy layerwise training, Optimization for training DNN's, Newer optimization methods for neural networks(AdaGrad, RMSProp, Adam), Regularization methods(dropout, drop connect, batch normalization).

Unit 3 **(8 Hours)**

Convolution neural networks(CNNs):

Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Using a pre trained convnet

Unit 4 **(8 Hours)**

Recurrent neural networks (RNNs):

Sequence modeling using RNNs, Backpropagation through time, LongShort Term Memory (LSTM), Bidirectional RNN

Unit 5 **(8 Hours)**

Unsupervised deep learning:

Autoencoders, Generative Adversarial Networks.

Unit 6 **(5 Hours)**

Applications:

Computer vision, Speech recognition and NLP.

Essential/recommended readings

1. Ian Goodfellow, Yodhua Bengio and Aaron Courville, Deep Learning, MIT Press Book, 2016.

2. Francois Chollet, Deep Learning with python, 2nd edition, Meaning Publications Co, 2021.

Additional References

1. Bunduma, N., Fundamentals of Deep Learning, 1st edition, O'reilly Books, 2017.

2. Heaton, J., Deep Learning and Neural Networks, 1st edition, Heaton Research Inc., 2015.

Suggested Practical List :

Practical exercises such as

The following practicals are to be conducted using Python.

1. Implement a feed-forward neural networks for classifying movie reviews as positive or negative(using IMDB dataset)
2. Implement a deep-neural feed-forward network for estimating the price of house, given real-estate data(Boston Housing Price)
3. Implement a deep-neural network for classifying news wires by topic (Reuters dataset).
4. Implement CNN for classifying MNIST dataset
5. Create a model for time-series forecasting using RNN/LSTM 6. Implement an auto-encoder

DSE: NUMERICAL OPTIMIZATION

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course	Eligibility criteria	

DISCIPLINE SPECIFIC ELECTIVE COURSE: Computer Graphics

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Computer Graphics	**4**	**3**	**1**	**0**	Pass in Class XII	DSC 03 (Mathematics for Computing - I), DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b Java Programming

Learning Objectives

This course introduces fundamental concepts of Computer Graphics with focus on modeling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics system.

Learning outcomes

On successful completion of the course, students will be able to:

- Describe Standard raster and vector scan devices as well as Graphical Input and output devices
- Implement algorithms for drawing basic primitives such as line, circle and ellipse.
- Implement algorithms for line clipping, polygon clipping and polygon filling.
- Implement a 3D object representation scheme, carryout 2D and 3D transformation, 3D projections
- Implement visible surface determination algorithms, Illumination models and surface rendering methods
- Implement a simple computer animation algorithm

SYLLABUS OF DSE

Unit 1 (8 Hours)

Introduction: Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, input/output devices.

Unit 2 (8 Hours)
Drawing and clipping primitives: Raster scan line, circle and ellipse drawing algorithms, Polygon filling, line clipping and polygon clipping algorithms

Unit 3 (12 Hours)
Transformation and Viewing: 2D and 3D Geometric Transformations, 2D and 3D Viewing transformations (Projections- Parallel and Perspective), Vanishing points.

Unit 4 (9 Hours)
Geometric Modeling: Polygon Mesh Representation, Cubic Polynomial curves (Hermite and Bezier).

Unit 5 (8 Hours)
Visible Surface determination and Surface Rendering: Z-buffer algorithm, List-priority algorithm and area subdivision algorithm for visible surface determination. Illumination and shading models, RGB Color model and Basics of Computer Animation.

Essential/recommended readings

1. Hearn, D & Baker, M.P. *Computer Graphics*, 2nd edition, Prentice Hall of India, 2009.
2. Foley, J. D., Dam, A.V, Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice in C*, 2nd edition, Pearson education, 2002.
3. Rogers, D. F. *Mathematical Elements for Computer Graphics*, 2nd edition, McGraw Hill Education, 2017.

Additional References

1. Bhattacharya, S. *Computer Graphics*, Oxford University Press, 2018.
2. Marschner, S., & Shirley, P. *Fundamentals of Computer Graphics,* 4th edition CRC Press, 2017.

Suggested Practical List :

Practical exercises such as

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement a midpoint circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using the Scan line fill algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogeneous Coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
8. Write a program to draw Hermite /Bezier curve.

GENERIC ELECTIVES (GE-8b): Digital marketing and Social Media Analystics

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
GE8b: Digital Marketing and Social Media Analytics	**4**	**3**	**0**	**1**	Pass in Class XII	Knowledge of HTML and Python Programming

Course Objective:
This Course provides introduction of various tools and technologies required to extract social media data. After completing this course, students will be able to execute end-to-end social media analytics projects and integrate them with existing business applications.

Course Learning Outcomes:
On successful completion of the course, students will be able to:

1. Understand the importance of data available in social media platforms.
2. Collections of data from various social media platforms like YouTube, Twitter etc. using API's and Python.
3. Data processing involving cleaning, structuring and analysis.
4. Case Study involving text mining and sentiment Analysis.
5. Development of complete social media recommendation system.

Syllabus

Unit 1 Introduction: Marketing in the Digital World, Introduction to Digital Marketing, Online Marketplace analysis, Data mining, Predicting and influencing strategies, Big data concepts.
Unit 2 Online Macro Environment: Introduction to Internet Technology, URL, Web page standards, Web Application frameworks and application servers, Approaches to develop secure systems.
Unit 3 SEO and SEM: Crawling, Indexing, Ranking, SEO tools, On page optimization and off page optimization. Advertisements in social media platforms, Paid search Marketing, Search engine Analytics.
Unit 4 Social Media Analytics: Role of email marketing, types of emails, email marketing objective, Build an automated email campaign, Analytics of Social Media Platforms.
Unit 5 Analytics using Python: learn to analyze marketing campaigns data, measure customer engagement, and predict how customer approaches to buy products, develop systems to crawl and predict.

Practical Exercise:
Q1. Go to a website that you visit regularly and access the source code of the page. (Right click on the page text and select View Source Code.)

1. Complete a search in the source code by pressing Ctrl F.
   • Does the web page include an H1?
   • Is the H1 the main page headline?
   • Does the H1 include a core message for the user?
   • Is there any sign that the H1 is optimized for searching (are there any keywords included in it)?
   • Is the site using the additional headings H2 through H6 ? Is it creating correct page and content structure?

0. Search for the title. It should be placed near the top of the page (<title> title text </title>). This is the meta title for the page.
   • Is the target keyword included in the title?
   • Is the title under 60 characters?

0. Search for the description. It should be placed near the top of the page (<meta-name="description" content="description text"/>). This is the meta description for the page.
   • Is there a description visible?
   • Is the target keyword included?
   • Is it under 160 characters?

Q2. Create an email marketing campaign using split testing. Send your email to a select number of email addresses. From here test subject lines, content, and sender details. Using this information, decide which split is performing better and why.

Q3. Create an email marketing campaign for a leading Holistic Living App incorporating:

- Optimize your subject lines, preheader text, email content, CTA, and landing pages through A/B tests
- Measure performance (open rate, CTR rate, response, and bounce rate)

Q4. You are the Social Media Analyzer for a Holistic Living App. You have been asked to prepare a comparative analysis of competitive brands in market to understand the branding value and user sentiments.
To develop the analysis, perform following:

- Extract all the posts of related apps permitted by the Facebook API
- Extract the metadata for each post: Timestamp, number of likes, number of shares, and number of comments
- Extract the user comments under each post and the metadata
- Process the posts to retrieve the most common keywords, bi-grams, and hashtags
- Process the user comments using the Alchemy API to retrieve the emotions Analyze all the results obtained from the preceding steps to derive conclusions

Q5. Perform the following on current trending twitter account and establish a case study:

- Fetching data from Twitter
- Cleaning of data
- Sentiment Analysis
- Customized Sentiment Analysis

References:

1. Chaffey, D., & Ellis-Chadwick, F. (2022). Digital Marketing: Strategy, implementation and practice. Pearson.
2. Dodson, I. (2016). The Art of Digital Marketing: The definitive guide to creating strategic, targeted, and measurable online campaigns. Wiley.
3. Chatterjee, S., & Krystyanczuk, M. (2017). Python social media analytics analyze and visualize data from Twitter, YouTube, GitHub, and more. Packt.

Suggested Practicals

It is suggested that the following tools/e-resources can be explored during the practical sessions • Wireshark • COFEE Tool • Magnet RAM Capture • RAM Capture • NFI Defragger • Toolsley • Volatility

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.
5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image
6. Using System internal tools for network tracking and process monitoring do the following:
    a. Monitor live processes
    b. Capture RAM
    c. Capture TCP/UDP packets
    d. Monitor Hard disk
    e. Monitor Virtual Memory
    f. Monitor Cache Memory

DSC20/DSC08/GE8a: INFORMATION SECURITY

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Information Security	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objective

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

Learning Outcomes

On successful completion of this course, a student will be able to

- Identify the major types of threats to information security.
- Describe the role of cryptography in security.
- Discover the strengths and weaknesses of private and public key cryptosystems.
- Identify and apply various access control and authentication mechanisms.
- Discuss data and software security and related issues.
- Explain network security threats and attacks.
- Articulate the need for security in cloud and IoT.

Syllabus

Unit 1 **(3 hours)**

Overview: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles.

Unit 2 **(6 hours)**

Cryptographic tools: Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication

Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

The Need for Database Security, SQL Injection Attacks, Database Access Control.

Unit 5 **(8 hours)**

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. **Overflow Attacks** - Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. **Software Security Issues** - Handling Program Input, Writing Safe Program Code, Handling Program Input.

Unit 6 **(6 hours)**

Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Public-Key Infrastructure.

Unit 7 **(7 hours)**

Wireless, Cloud and IoT Security: Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security.

References

1. W. Stallings, L. Brown, *Computer Security: Principles and Practice*, 4th edition, Pearson Education, 2018.

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications*, 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.

5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Introduction to Parallel Programming	**4**	**3**	**0**	**1**	**Pass in Class XII**	**Computer System Architecture/A course in C++at class XII/Data Structures, Operating Systems**

Course Objective

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.

Syllabus

Unit 1

Introduction to Parallel Computing: Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

Unit 2

OpenMP programming for shared memory systems: Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks.

Unit 3

MPI programming for distributed memory systems: MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, Introduction to MPI Collectives, Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

Unit 4

Applications: Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm.

References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1st Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

Additional references

(i) B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999, Springer.

Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP

DSC17/GE7d/DSE8e: CLOUD COMPUTING

Credit distribution, Eligibility and Pre-requisites of the Course

	Credits	Credit distribution of the course		

1. Peter Kim, The Hacker Playbook 3: Practical Guide to Penetration Testing, Zaccheus Entertainment, 2018.
2. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press, 2008.
3. Online Resources:

https://www.sans.org/cyberaces/

https://skillsforall.com/

https://www.hackingloops.com/ethical-hacking/

Suggested Practical List (If any): (30 Hours)

Perform the following activities, record and report in standard form.

(NOTE: Exercise extra caution while performing these exercises and codes)

1. Perform various Virtual Machine based exercises on https://vulnhub.com/
2. Perform Capture the Flag (CTF) exercises from https://www.hacker101.com/
3. Follow the lessons and activities from https://www.hackingloops.com/ethical-hacking/
4. Google site for hacking https://google-gruyere.appspot.com/
5. OWASP WebGoat https://github.com/WebGoat/WebGoat

GE8d/DSE: CYBER FORENSICS

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		

Cyber Forensics	**4**	**3**	**0**	**1**	**Pass in Class XII**	**NIL**

Course Objective:

This course is to equip students with the knowledge and skills necessary to identify, collect, analyze and present digital evidence in a manner that is admissible in legal proceedings. Students will be able to conduct a thorough investigation of cybercrime incidents, preserve digital evidence, and report findings to relevant stakeholders.

Course Learning Outcomes:

- Students will be able to demonstrate an understanding of the principles of digital forensics, including legal considerations, recognition, collection, and preservation of digital evidence.
- Students will develop skills in using digital forensics tools and techniques, such as creating disk images, conducting keyword and grep searches, and examining Windows registry.
- Students will learn evidence recovery methods, including deleted file recovery, formatted partition recovery, and data recovery procedures, and ethical considerations.
- Students will gain knowledge of cyber forensic investigation tools and techniques, including digital evidence collection, preservation, and password cracking.
- Students will understand cyber laws and crimes, including hacking, viruses, intellectual property, and e-commerce, and the legal system of information technology, including jurisdiction issues and security and evidence in e-commerce.

Unit 1 – Digital Forensics: Introduction to digital forensics, legal considerations, recognising and collecting digital evidence, preservation of evidence, hash values and file hashing, creating disk images, keyword and grep searches, network basics, reporting and peer review, digital forensics report.

Unit 2 – Windows OS Forensics: Bits, bytes, Endieness, Disk partition schema, File systems – FAT, NTFS, ex-FAT, windows registry forensics, examining windows registry, NTUser.Dat Hive File Analysis, SAM Hive file, Software Hive file, System Hive File, USRClass.dat Hive File, AmCache Hive File.

Unit 3 – Evidence Recovery: Introduction to Deleted File Recovery, Formatted Partition Recovery, Data Recovery Tools, Data Recovery Procedures and Ethics, Complete time line analysis of computer files based on file creation, File modification and file access, Recover Internet Usage Data, Recover Swap Files/Temporary Files/Cache Files, Introduction to Encase Forensic Edition, Forensic Tool Kit (FTK), Use computer forensics software tools to cross validate findings in computer evidence.

Unit 4 – Investigation: Introduction to Cyber Forensic Investigation, Investigation Tools, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Encryption and Decryption methods, Search and Seizure of Computers, Recovering deleted evidences, Password Cracking.

Unit 5 – Cyber Crimes and Cyber Laws: Introduction to IT laws & Cyber Crimes, Internet, Hacking, Cracking, Viruses, Software Piracy, Intellectual property, Legal System of Information Technology, Understanding Cyber Crimes in context of Internet, Indian Penal Law & Cyber Crimes Fraud Hacking Mischief, International law, E-Commerce-Salient Features On-Line contracts Mail Box rule Privities of, Contracts Jurisdiction issues in E-Commerce Electronic Data Interchange, Security and Evidence in E-Commerce Dual Key encryption Digital signatures security issues.

References:

1. Marjee T. Britz, Computer Forensics and Cyber Crime: An Introduction, Pearson Education, 2013.

2. C. Altheide& H. Carvey Digital Forensics with Open Source Tools, Syngress, 2011. ISBN: 9781597495868.

Additional References:

1. Computer Forensics: Investigating Network Intrusions and Cybercrime" by Cameron H. Malin, Eoghan Casey, and James M. Aquilina

2. Online Course management System: https://esu.desire2learn.com/

3. Computer Forensics, Computer Crime Investigation by John R,Vacca, Firewall Media, New Delhi.

4. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning

5. Real Digital Forensics by Keith j.Jones, Richard Bejitlich,Curtis W.Rose ,AddisonWesley Pearson Education

Suggested Practicals

It is suggested that the following tools/e-resources can be explored during the practical sessions • Wireshark • COFEE Tool • Magnet RAM Capture • RAM Capture • NFI Defragger • Toolsley • Volatility

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.
5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image
6. Using System internal tools for network tracking and process monitoring do the following:
    a. Monitor live processes
    b. Capture RAM
    c. Capture TCP/UDP packets
    d. Monitor Hard disk
    e. Monitor Virtual Memory
    f. Monitor Cache Memory

DSC20/DSC08/GE8a: INFORMATION SECURITY

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		