

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

DSC19/GE7d/DSE8e: CLOUD COMPUTING

Credit distribution, Eligibility and Pre-requisites of the Course

	Credits	Credit distribution of the course		
--	----------------	--	--	--

Course title & Code		Lecture	Tutorial	Practical/ Practice	Eligibility criteria	Pre-requisite of the course
Cloud Computing	4	3	0	1	Pass in Class XII	NIL

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:

Learning outcomes for an undergraduate course on cloud computing may include:

1. Knowledge of the fundamental concepts and principles of cloud computing, including virtualization, scalability, reliability, and security.
2. Ability to design, develop, and deploy cloud-based applications using popular cloud platforms and services.
3. Familiarity with cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. Visualize the economic, legal, and ethical implications of cloud computing, including issues related to data privacy, ownership, and security.
5. Ability to evaluate and select cloud-based solutions based on their technical, economic, and business requirements.
6. Understanding of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: Overview of Computing Paradigm

Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing

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

Unit 3: 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: Case Studies

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

Unit 5: Cloud Computing Management

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

Unit 6: 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. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms .
3. Working on tools used in cloud computing online-
 - a) Storage
 - b) Sharing of data
 - c) manage your calendar, to-do lists,
 - d) a document editing tool
4. Exploring Google cloud
5. Exploring microsoft cloud
6. Exploring amazon cloud

Additional Suggestive list of Practical's (can be implemented in C++/Python)

1. Write a program to implement DFAs that recognize identifiers, constants, and operators of the mini language.
2. Write a program to design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. Identifiers may be of restricted length.
3. Write a program to check the types of expressions in a language.
4. Write a translator to translate a 3-address code into assembly code.

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: DIGITAL IMAGE 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		
DSE7a: Digital Image Processing	4	3	0	1	Pass in Class XII	One course in any Programming Language

Course Objective

This course introduces students to the fundamentals of digital image processing, It introduces image processing in the Spatial and frequency domains including techniques for various image

transformations, image enhancement/filtering, image restoration, image compression and segmentation and morphological image processing.

Course Learning Outcomes

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

1. Understand the fundamentals of Image Processing and its role and importance in a variety of applications.
2. Write programs to read/write and manipulate images for the purpose of enhancement.
3. Understand the need for image transforms and their properties.
4. Understand different causes for image degradation and use various techniques to restore images.
5. Understand the need and techniques for image compression.
6. Perform morphological image processing and image segmentation.
7. Develop an image processing application.

Syllabus

Unit 1 Introduction

(5 hours)

Digital Image Fundamentals, Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

Unit 2 Spatial Domain Filtering

(10 hours)

Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothing Spatial Filters-Low pass filters, Order Statistics filters; Sharpening Spatial Filters- Laplacian filter.

Unit 3 Filtering in Frequency Domain

(6 hours)

The Discrete Fourier Transformation (DFT), Frequency Domain Filtering:-Ideal and Butterworth Low pass and High pass filters

Unit 4 Image Degradation/Restoration Process (4 hours)

Noise models, Noise Restoration Filters

Unit 5 Image Compression (5 hours)

Fundamentals of Image Compression, Huffman Coding, Run Length Coding

Unit 6 Morphological Image Processing (10 hours)

Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

Unit 7 Image Segmentation (5 hours)

Point, Line and Edge Detection, Thresholding.

References

1. Gonzalez, R. C., & Woods, R. E. *Digital Image Processing*, 4th edition, Pearson education, 2017 .

Additional References

1. Castleman, K. R. *Digital Image Processing*, 1st edition, Pearson Education, 2007.
2. Gonzalez, R. C., Woods, R. E., & Eddins, S. *Digital Image Processing using MATLAB*, Pearson Education Inc., 2004.
3. Jain, A. K. *Fundamentals of Digital Image Processing*, 1st edition, Prentice Hall of India, 1988.

Suggested Practical List

The practicals are to be conducted using Python. The objective is to become familiar with basic Python libraries for Image Processing, like OpenCV, Scikit-Image, etc.

1. Perform the following:
 - a. Read and display an image.
 - b. Resize a given image.
 - c. Convert a given color image into a corresponding gray-scale image.
 - d. Convert a given color/gray-scale image into black & white image
 - e. Draw the image profile.
 - f. Separate a given color image into three R, G & B planes.
 - g. Create a color image using separate three R, G and B planes.
 - h. Write given 2-D data in an image file.

2. To write and execute image processing programs using point processing method:
 - a. Obtain Negative image
 - b. Obtain Flip image
 - c. Thresholding
 - d. Contrast stretching
3. To write and execute programs for image arithmetic operations:
 - a. Addition of two images
 - b. Subtract one image from other image
 - c. Calculate mean value of image
 - d. Different Brightness by changing mean value
4. To write and execute programs for image logical operations:
 - a. AND operation between two images
 - b. OR operation between two images
 - c. Calculate intersection of two images
 - d. Water Marking using X-OR operation
 - e. NOT operation (Negative image)
5. To write and execute a program for histogram calculation and equalization:
 - a. Using inbuilt function
 - b. Without using inbuilt function
6. To write and execute a program performing the following geometric transformations on an image:
 - a. Translation
 - b. Scaling
 - c. Rotation
 - d. Shrinking
 - e. Zooming
7. To understand various image noise models and to write programs for:
 - a. Image restoration
 - b. Remove Salt and Pepper Noise
 - c. Minimize Gaussian noise
 - d. Median filter and Weiner filter
8. Write and execute programs to remove noise from images using spatial filtering.
 - a. Understand 1-D and 2-D convolution process

- b. Use 3x3 Mask for low pass filter and high pass filter
9. Write and execute programs for image frequency domain filtering.
 - a. Apply FFT on given image
 - b. Perform low pass and high pass filtering in frequency domain
 - c. Apply IFFT to reconstruct image
 10. Write and execute a program for edge detection using different edge detection mask.
 11. Write and execute a program for image morphological operations erosion and dilation

DISCIPLINE SPECIFIC ELECTIVE COURSE: Advanced Algorithms

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		
DSE7b: Advanced Algorithms	4	3	1	0	Pass in Class XII	Design and Analysis of Algorithms

Course Objective

This course is designed to provide exposure to more sophisticated algorithms for some tractable problems, some advanced topics in algorithms such as NP Completeness and how to handle NP hard problems in practice.

Learning Outcomes

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

1. Understand and develop more sophisticated algorithms using some of the known design techniques.
2. Identify NP hard problems.

- c. Rotation
 - d. Shrinking
 - e. Zooming
0. To understand various image noise models and to write programs for:
 - a. Image restoration
 - b. Remove Salt and Pepper Noise
 - c. Minimize Gaussian noise
 - d. Median filter and Weiner filter
 0. Write and execute programs to remove noise from images using spatial filtering.
 - a. Understand 1-D and 2-D convolution process
 - b. Use 3x3 Mask for low pass filter and high pass filter
 0. Write and execute programs for image frequency domain filtering.
 - a. Apply FFT on given image
 - b. Perform low pass and high pass filtering in frequency domain
 - c. Apply IFFT to reconstruct image
 0. Write and execute a program for edge detection using different edge detection mask.
 0. Write and execute a program for image morphological operations erosion and dilation

DISCIPLINE SPECIFIC ELECTIVE COURSE: Advanced Algorithms

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		
DSE7b: Advanced Algorithms	4	3	1	0	Pass in Class XII	Design and Analysis of Algorithms

Course Objective

This course is designed to provide exposure to more sophisticated algorithms for some tractable problems, some advanced topics in algorithms such as NP Completeness and how to handle NP hard problems in practice.

Learning Outcomes

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

1. Understand and develop more sophisticated algorithms using some of the nown design techniques.
2. Identify NP hard problems.
3. Use polynomial time reductions to prove NP hardness of problems.
4. Design approximation algorithms for NP hard problems and find their approximation ratio.

Syllabus

Unit 1 More applications of Divide and Conquer, Greedy and Dynamic Programming approaches: Counting Inversions, Closest pair of points, Integer Multiplication, Huffman Code, Segmented Least Squares etc.

Unit 2 Network Flows: Ford Fulkerson algorithm for max flow problem.

Unit 2 Backtracking: Constructing All Subsets, Constructing All Permutations, Constructing all paths in a graph.

Unit 3 Polynomial time reductions via gadgets: SAT and 3-SAT problems; Reducing 3-SAT to Independent set, Clique and Vertex cover.

Unit 4 Proving NP completeness: Circuit satisfiability, 3-SAT, Sequencing Problems, Graph coloring, Subset sum.

Unit 5 Introduction to Approximation Algorithms: Definition, Concept of approximation factor, Bounding the optimal solution, concept of tight example.

Unit 6 Combinatorial Approximation Algorithms: Set cover, Minimizing makespan, k-center.

Unit 7 LP based Approximation Algorithms: Approximation algorithms for Vertex cover/Set cover via LP rounding.

References

1. Kleinberg, J., Tardos, E. *Algorithm Design*, 1st edition, Pearson, 2013.
2. Vazirani, V. V. *Approximation Algorithms*, 1st edition, Springer, 2001.

Additional References

- (i) Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4th edition, Prentice Hall of India, 2022.
- (ii) Williamson, D. P., Shmoys, D. B. *The Design of Approximation Algorithms*, 1st edition, Cambridge University Press, 2011.

Tutorials

Tutorials based on Theory

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, Endianness, 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 Bejtlich, Curtis W. Rose, Addison Wesley 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		

Syllabus of Discipline Specific Elective VI/VII Semester (NEP UGCFC 2022)

(Effective from Academic Year 2024-25)

DISCIPLINE SPECIFIC ELECTIVE (DSE): Research Methodology

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

Course Objective

This course allows the students to acquire the necessary skills to conduct research in computer science. It enables the students to understand the entire process of research from problem identification, literature review, designing the project to documenting the outcome.

Course Learning Outcomes

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

- Identify the problem after conduct of a literature survey.
- Define goals, approach, and scope of the research.
- Explore, download and interpret datasets.
- Effectively record study findings in a research paper format.

Syllabus of DSE

Unit 1 : 8 Hours

Research Fundamentals: Meaning and significance of research, requirements and characteristics of research, types of research - basic, applied analytical, conceptual, empirical, experimental, non-experimental, prospective, retrospective, exploratory / descriptive, qualitative, quantitative, mixed method. Research process, induction and deduction in research, introduction to research tools, qualities of a good researcher.

Unit 2 : 5 Hours

Problem Identification: Choosing an appropriate problem area, identifying sources of research articles, literature review – stating and evaluating the research problem, techniques and methodologies, state of the art.

Unit 3 : 12 Hours

Data Analytics: Exploring and organizing data sets, pre-processing data, interpreting the data. Choosing appropriate statistics. Descriptive statistics - measures of central tendency and variability, measures of association. Inferential statistics – estimating population parameters, testing hypothesis.

Unit 4 : 10 Hours

Presenting research outcomes: Essential elements of a research paper - explanation of the research problem, description of methods and data, data analysis and its interpretation, identification of possible weaknesses of the study, presenting and summarizing the research output, drawing conclusions.

Unit 5 : 5 Hours

Publication: Process of journal submission and review. Peer review process - single, blind and double blind. Professional research societies, scientometric analysis - citation index and analysis, plagiarism, plagiarism checker.

Unit 6 : 5 Hours

Research Ethics: Ethical issues in research, protection from harm, voluntary and informed participation, right to privacy, conflict of interest, honesty with professional colleagues, professional code of ethics, intellectual property rights, fraud and misconduct in science.

Essential/recommended readings

1. Thomas, C. G. (2021). *Research Methodology and Scientific Writing*, 2nd Ed. Springer.
2. Leedy, P. D., & Ormrod, J. E. (2016). *Practical Research: Planning and Design*, 11th Ed. Pearson.

Additional References:

1. Ghezzi, C. *Being a Researcher: An Informatics Perspective*. Springer
2. Locharoenrat, K. (2018). *Research Methodologies for Beginners*. PAN Stanford Publication.
3. <https://www.unesco.org/en/articles/what-you-need-know-about-unescos-new-ai-competency-frameworks-students-and-teachers?hub=32618>

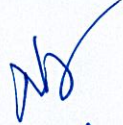
Suggested Practical List

Capstone Project: Students must choose an area of interest for research, based on the curriculum (but not limited by it) covered in the program. They should identify a research problem to solve. During the semester the students must document the research journey in the form of a report, which will be evaluated at the end of the semester. The students are encouraged to write a research paper based on the report, under the guidance of the teacher. The practical class for research methodology course should be utilized to perform the following tasks in the research process.

1. Search the research papers related to the chosen problem using academic search engines like Google Scholar, Scopus search, Web of Science database, etc.


Head
संगणक विज्ञान विभाग
Department of Computer Science
दिल्ली विश्वविद्यालय, दिल्ली-110007
University of Delhi, Delhi-110007

- a. Evaluate the venue of the source of research paper - Journals using citation metrics like CiteScore, SCImago Journal Rank (SJR)), Source Normalized Impact per Paper (SNIP) etc., Conferences venues are evaluated using indexing information, Core Ranking etc.
 - b. Summarize the reviewed papers in a tabular format with columns: Paper Title, Author(s), Year, Key Findings, and Citation Count.
 - c. Explore reference management tools like Mendley / Zotero / EndNote to organize, store, and manage references.
2. Practice data analysis techniques taught in the class and identify a suitable technique required to solve the chosen research problem.
3. Write the research report and prepare to write the research paper.
- a) Choose a document writing software and prepare the report as per the format given by the teacher.
 - b) Use the plagiarism check tool to assess the similarity index of the report and ensure that it is less than 10%.
 - c) Explore the journal finder tools available for the publishers and select a suitable journal to submit the manuscript


अध्यक्ष/Head
संगणक विज्ञान विभाग
Department of Computer Science
दिल्ली विश्वविद्यालय, दिल्ली-110009
University of Delhi, Delhi-110007

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, Yoshua 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 CORE COURSE – 17 (DSC-17): Machine Learning

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC-17 Machine Learning	4	3	0	1	Pass in Class XII	DSC01 Programming using Python_ / A course in Python at plus 2 level

Learning Objectives

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Learning outcomes

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

- Differentiate between supervised and unsupervised learning tasks.
- State the need of preprocessing, feature scaling and feature selection.
- Formulate classification, regression and clustering problems as optimization problems
- Implement various machine learning algorithms learnt in the course.

SYLLABUS OF DSC-17

Unit 1 (5 Hours)

Introduction: Basic definitions and concepts, key elements, supervised and unsupervised learning, introduction to reinforcement learning, applications of ML.

Unit 2 (8 Hours)

Preprocessing: Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis).

Unit 3 (12 Hours)

Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

Unit 4 (12 Hours)

Classification: Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics.

Unit 5 (8 Hours)

Clustering: Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

Essential/recommended readings

1. Mitchell, T.M. *Machine Learning*, McGraw Hill Education, 2017.
2. James, G., Witten. D., Hastie. T., Tibshirani., R. *An Introduction to Statistical Learning with Applications in R*, Springer, 2014.
3. Alpaydin, E. *Introduction to Machine Learning*, MIT press, 2009.

Additional References

1. Flach, P., *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2015.
2. Christopher & Bishop, M., *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2016.
3. Sebastian Raschka, *Python Machine Learning*, Packt Publishing Ltd, 2019

Suggested Practical List:

Practical exercises such as

Use Python for practical labs for Machine Learning. Utilize publicly available datasets from online repositories like <https://data.gov.in/> and <https://archive.ics.uci.edu/ml/datasets.php>

For evaluation of the regression/classification models, perform experiments as follows:

- Scale/Normalize the data
- Reduce dimension of the data with different feature selection techniques
- Split datasets into training and test sets and evaluate the decision models
- Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- MSE and R^2 score for regression models
- Accuracy, TP, TN, FP, FN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier
2. Simple Linear Regression multiple linear regression
3. Polynomial Regression
4. Lasso and Ridge Regression
5. Logistic regression
6. Artificial Neural Network
7. k -NN classifier
8. Decision tree classification
9. SVM classification
10. K-Means Clustering
11. Hierarchical Clustering

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.

DISCIPLINE SPECIFIC ELECTIVE COURSES

DISCIPLINE SPECIFIC ELECTIVE COURSE: Social Network Analytics

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		
Social Network Analytics	4	3	0	1	Pass in Class XII	DSC 01 Programming using Python, DSC03 Mathematics for Computing

Learning Objectives

The course introduces basic graph theory and draws distinction between graph as an abstract structure and real-life situation modelled as network. This course aims to expose the students to the strengths and capabilities of network analysis and their applications through the use of open source software.

Learning outcomes

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

- Model real life situation as networks
- Identify and apply quantitative network measures to characterize social networks at the local and global level
- Generate synthetic networks that satisfy properties of real world networks
- Discover, analyse and evaluate the intrinsic community structure of networks
- Model an information diffusion process for predictive analysis of networks

SYLLABUS OF DSE

Unit 1 (7 Hours)

Introduction to Social Network Analysis: Graph theory, random walk, degree distribution, mapping of real world situation into networks and applications of social network analysis, types of networks

Unit 2 (10 Hours)

Network Measures: Centrality measures, Page Rank, Hubs and Authority, Assortativity, Transitivity and Reciprocity, Similarity and Structural Equivalence

Unit 3 (10 Hours)

Network Models: Properties of Real-World Networks, Random Network Model, Small World Network Model, Preferential Attachment Model

Unit 4 (10 Hours)

Community Structure in Networks: Types of Communities, Community Detection algorithms and evaluation of communities obtained

Unit 5 (8 Hours)

Information Diffusion in Social Media: Information Cascades, Diffusion of Innovations, Basic Epidemic Models

Essential/recommended readings

1. Chakraborty T. *Social Network Analysis*, 1st edition, Wiley India Pvt. Ltd., 2021.
2. Zafarani R., Abbasi M. A., Liu H. *Social Media Mining: An Introduction*, 1st edition, Cambridge University Press, 2014.
3. Barabási A. L. , Pósfai M. *Network Science*, 1st edition, Cambridge University Press, 2016.

Additional References

1. Easley, Kleinberg J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, 1st edition, Cambridge University Press, 2012.

Suggested Practical List :

Practical exercises such as

Python Packages like igraph, NetworkX, NDlib etc. may be used for programming

1. Plot a weighted directed network such that node size and edge width is proportional to their degree and edge weight respectively
2. Compute and plot degree distribution of a real-world network. Also compute its local and global properties.
3. Generate three networks of 1000 nodes each using Random Network Model, Small World Network Model, Preferential Attachment Model and compare their characteristics.
4. Compute different centrality measures to identify top-N nodes and compare their ranks with those obtained by PageRank method.
5. Apply community detection algorithms on a small real-world network (e.g. Karate club) and compare modularity using bar plot. Also plot the communities revealed with different colors.
6. Simulate diffusion trends for different epidemic models and present results using appropriate visuals.



Department of Computer Science

COURSES OFFERED BY DEPARTMENT OF COMPUTER SCIENCE

(Provide the details of the Discipline Specific Courses offered by your department for the UG Programme with your discipline as the Single Core Discipline)
[UG Programme for **Bachelor in Computer Science (Honours)** degree]

DISCIPLINE SPECIFIC CORE COURSE -19 (DSC-19) : Compiler Design

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		
DSC19 Compiler Design	4	3	0	1	Pass in Class XII	One course in any Programming Language

Learning Objectives

The basic objective of the compiler design course is to understand the basic principles of compiler design, its various constituent parts, algorithms, and data structures required to be used in the compiler. It also aims to understand the use of basic compiler-building tools.

Learning Outcomes

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

1. Describe the concepts and different phases of compilation.
2. Represent language tokens using regular expressions and context free grammars.
3. Describe the working of lexical analyzers.
4. Understand the working of different types of parsers and parse a particular string.
5. Describe intermediate code representations using syntax trees and DAG's as well as use this knowledge to generate intermediate code in the form of three address code representations.

6. Apply optimization techniques to intermediate code and generate machine code for high level language program.
7. Use Lex and Yacc automated compiler generation tools.

Syllabus

Unit 1 Introduction: Overview of compilation, Phases of a compiler.

Unit 2 Lexical Analysis: Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, Error reporting, Regular expressions and definitions , Lexical Analyzer Generator-Lex.

Unit 3 Syntax Analysis: CFGs, left recursion, left factoring, Top-down parsing- LL parser, Bottom-up parsing- LR parser, Parser Generator-yacc.

Unit 4 Intermediate representations: Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, loops and conditional statements, Type Checking.

Unit 5 Storage organization & Code generation: Activation records, stack allocation, Issues in Code Generation – Design of a simple Code Generator.

Unit 6 Code optimization : Principal sources of optimization, Peephole optimization.

References

1. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. *Compilers: Principles, Techniques, and Tools*, 2nd edition, Addison Wesley, 2006.

Additional References

- (i) V Raghvan, *Principles of Compiler Design*, TMH, 2010.
- (ii) Santanu Chattopadhyay, *Compiler Design*, PHI, 2005.

Suggested Practical List

1. Write a Lex program to count the number of lines and characters in the input file.
2. Write a Lex program to count the number of vowels and consonants in a given string
3. Write a Lex program that implements the Caesar cipher: it replaces every letter with the one three letters after in alphabetical order, wrapping around at Z. e.g. a is replaced by d, b by e, and so on z by c.
4. Write a Lex program that finds the longest word (defined as a contiguous string of upper and lower case letters) in the input.
5. Write a Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments in any simple programming language.
6. Write a Lex program to count the number of words, characters, blank spaces and lines in a C file.
7. Write a Lex specification program that generates a C program which takes a string “abcd” and prints the following output


```
abcd
abc
a
```
8. Write a Lex program to recognize a valid arithmetic expression.

9. Write a YACC program to find the validity of a given expression (for operators + - * and /)A program in YACC which recognizes a valid variable which starts with a letter followed by a digit. The letter should be in lowercase only.
10. Write a program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).
11. Write a program in YACC to recognize the string „abbb“, „ab“ „a“ of the language (an b n , n>=1).
12. Write a program in YACC to recognize the language (an b , n>=10). (output to say input is valid or not)

Additional Suggestive list of Practicals (Can be implemented in C++/Python)

1. Write a program to implement DFAs that recognize identifiers, constants, and operators of the mini language.
2. Write a program Design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. Identifiers may be of restricted length.
3. Write a program to check the types of expressions in a language.
4. Write a translator to translate a 3-address code into assembly code.

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: Data Analysis and Visualization

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		
DSE7a: Digital Image Processing	4	3	0	1	Pass in Class XII	One course in any Programming Language

Course Objective

This course introduces students to the fundamentals of digital image processing, It introduces image processing in the Spatial anfrequency domains including techniques for various image

DSC12/DSC06/GE6a: COMPUTER NETWORKS

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		
Computer Networks	4	3	0	1	Pass in Class XII	NIL

Course Objectives

The course objectives of this paper are to:

- Understand the concepts behind computer networks and data communication.
- Learn the different types of networks, network topologies and their characteristics.
- Learn the working of protocols used at various layers.
- Understand the utility of different networking devices.

Learning Outcomes

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

- differentiate between various types of computer networks and their topologies.
- understand the difference between the OSI and TCP/IP protocol suit.
- distinguish between different types of network devices and their functions.
- design/implement data link and network layer protocols in a simulated networking environment.

Syllabus**Unit 1****(8 hours)****Introduction:**

Types of computer networks, Internet, Intranet, network topologies (bus, star, ring, mesh, tree, hybrid topologies), network classifications. layered architecture approach, OSI Reference Model, TCP/IP Reference Model. Transmission Modes: simplex, half duplex and full duplex, network devices and their role.

Unit 2**(9 hours)****Physical Layer:**

Analog signal, digital signal, the maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time-division multiplexing, wavelength division multiplexing). Guided Media (Wired) (Twisted pair, Coaxial Cable, Fiber Optics. Unguided Media (Radio Waves, Infrared, Micro-wave, Satellite).

Unit 3**(10 hours)****Data Link and MAC Layer:**

Data link layer services, error detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols with collision detection, MAC addressing, Ethernet..

Unit 4**(8 hours)****Network layer:**

Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Overview of Network Layer protocols- (ARP, IPV4, ICMP, RARP, IPV6)

Unit 5**(10 hours)****Transport and Application Layer:**

Process to process Delivery- (client-server paradigm, connectionless versus connection-oriented service); User Datagram Protocols, TCP/IP protocol, Flow Control. FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Remote login protocol), WWW (World Wide Web), HTTP (HyperText Transfer Protocol), URL (Uniform Resource Locator), DNS, DHCP, BOOTP.

Essential/recommended readings

1. Tanenbaum, A.S. & Wethrall, D.J.. Computer Networks, 5th edition, Pearson Education, 2012.
2. Forouzan, B. A.. Data Communication and Networking, 4th edition, McGraw-Hill Education, 2017.

Additional References

1. Comer, D. E.. Computer Networks and Internet, 6th edition, Pearson education, 2015.
2. Stallings, W., Data and Computer Communications, 10th edition, Pearson education India, 2017.

Practicals.

Introduce students to any network simulator tool and do the following:

1. To Study basic network command and Network configuration commands.
2. To study and perform PC to PC communication.
3. To create Star topology using Hub and Switch.
4. To create Bus, Ring, Tree, Hybrid, Mesh topologies.
5. Perform an initial Switch configuration.
6. Perform an initial Router configuration.
7. To implement Client Server Network.
8. To implement connection between devices using a router.
9. To perform remote desktop sharing within LAN connection.

GENERIC ELECTIVES (GE-6b): Internet Technologies: Web App Design and Development

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		
GE6b: Internet Technologies: Web App Design and Development	4	3	0	1	Pass in Class XII	NIL

Learning Objectives

The course aims at:

- Develop understanding of Web Development Architecture.
- Using React components in Web applications
- Introduce REST APIs Design
- Understanding of Angular Architecture, data-binding and dependency injection
- Understand form validations and application of templates

Learning outcomes

On successful completion of the course students will be able to

- Develop interfaces for single page applications
- Develop a complete client side solutions using angular js
- Develop a RESTful web services.
- Apply form validations

SYLLABUS OF GE-6b

Unit 1

Introduction to React: Definition of React, React library, React Developer tools, Introduction to ES6, Declaring variables, Arrow Functions, Objects and Arrays, modules, Introduction to AJAX, Functions in AJAX Pure React: Page setup, virtual DOM, React Element, React DOM, Constructing Elements with Data, React Components, DOM Rendering, First React Application using Create React App, React with JSX, React Element as JSX Props, State and Component Tree: Property Validation, Validating Props with createClass, Default Props, ES6 Classes and stateless functional components, React state management, State within the component tree, state vs props, Forms in React

Unit 2

Rest APIs: JSON: Introduction, Syntax, Data Types, Objects, Schema. REST API:

Introduction, WRML, REST API Design, Identifier Design with URIs, Interaction Design with HTTP, Representation Design, Caching, Security.

Unit 3

Angular.js.: Introduction to Angular: Angular architecture; introduction to components, component interaction and styles; templates, interpolation and directives; forms, user input, form validations; data binding and pipes; retrieving data using HTTP; Angular modules

Essential/recommended readings

1. D. Brad, B. Dayley and C. Dayley, *Node.js, MongoDB and Angularjs Web Development: The definitive guide to using the MEAN stack to build web applications*, 2nd edition, Addison-Wesley, 2018.
2. D. Herron, *Node.js Web Development*, 5th edition, Packt Publishing, 2020.
3. A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 1st edition, O'Reilly, 2017.
4. M. Masse, *REST API – Design Rulebook*, 1st edition, O'Reilly, 2011.

Additional References

No additional references mentioned.

Suggested Practical List :

Practical exercises such as

1. Angular.js:

- a. Build a simple Angular.js application that displays a list of items.
- b. Create a form in Angular.js to add new items to the list.
- c. Implement filtering and sorting functionality in Angular.js to manipulate the displayed list.
- d. Integrate Angular.js with a RESTful API to fetch data and display it in the application.
- e. Implement authentication and authorization using Angular.js routing and services.

2. React:

- a. Create a basic React component that displays "Hello, World!" on the screen.
- b. Build a React application that fetches data from a REST API and renders it in a list.
- c. Implement form handling in React to create, update, and delete items from the list.
- d. Create a search functionality using React to filter the displayed list based on user input.
- e. Implement routing in React to navigate between different pages within the application.

3. REST API:

- a. Build a simple REST API using a framework like Node.js and Express.
- b. Create endpoints to perform CRUD operations (Create, Read, Update, Delete) on a specific resource (e.g., users, products).
- c. Implement authentication and authorization mechanisms using JSON Web Tokens (JWT) to secure the API.
- d. Develop endpoints that handle file uploads and downloads.

GENERIC ELECTIVES (GE-7d): 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
		Lecture	Tutorial	Practical/ Practice		
GE 7d: Cloud Computing	4	3	0	1	Pass in Class XII	NIL

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 the course, students will be able to:

1. Apply knowledge of the fundamental concepts and principles of cloud computing, including virtualization, scalability, reliability, and security.
2. to design, develop, and deploy cloud-based applications using popular cloud platforms and services.
3. apply knowledge of cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. visualize 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. gain an understanding of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: Overview of Computing Paradigm (6 hours)

Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing (7 hours)

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

Unit 3: Cloud Computing Architecture (12 hours)

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: Case Studies (7 hours)

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

Unit 5: Cloud Computing Management (6 hours)

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

Unit 6: Cloud Computing Security (7 hours)

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:

- a) Create free-tier accounts on AWS, Azure, and GCP.
- b) Explore dashboards and identify key services (compute, storage, networking).
- c) Understand pricing calculators on each platform.

2. Launch Your First Amazon EC2 Instance

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

Steps:

- a) Launch an EC2 instance from the AWS Management Console.
- b) Use a pre-configured AMI (e.g., Amazon Linux 2).
- c) Configure security groups to allow SSH access.
- d) 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:

- a) Create an Auto Scaling Group and define a launch template.
- b) Configure scaling policies (e.g., scale up when CPU utilization exceeds 70%).
- c) Deploy an Application Load Balancer (ALB) to distribute traffic.
- d) 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:

- a) Deploy a static website using any of the following:
 - AWS S3
 - Azure Blob Storage
 - GCP Cloud Storage
- b) Configure permissions and enable public access.

6. Monitor Resources Using AWS CloudWatch

Objective: Use CloudWatch to monitor AWS resources

Steps

- a) Set up CloudWatch metrics for an EC2 instance (e.g., CPU utilization).
- b) Create a CloudWatch Alarm to send notifications when a threshold is exceeded.
- c) Configure an SNS topic for email notifications.
- d) 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:

- a) Create a project and assign roles to users.
- b) Upload an image (e.g., Ubuntu cloud image) to the Glance service.

- c) Define a flavor to specify VM configurations.
- d) 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:

- a) Create a private network and a public network.
- b) Attach a router to connect the private network to the public network.
- c) Assign floating IPs to instances for external access.

10. Cloud Security

Objective: Understand security practices in the cloud.

Steps:

- a) Implement IAM roles and policies for a cloud platform.
- b) Create and assign least-privilege roles to users.
- c) Configure data encryption for storage (e.g., S3 bucket encryption).
- d) Set up a firewall rule and test its functionality.

GENERIC ELECTIVES (GE-7e): 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		
GE 7e: Ethical Hacking	4	3	0	1	Pass in Class XII	NIL

Course Objectives

2. Q. Kong, T. Siau, 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		

DISCIPLINE SPECIFIC CORE COURSE – 17 (DSC-17): Machine Learning

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC-17 Machine Learning	4	3	0	1	Pass in Class XII	DSC01 Programming using Python/ A course in Python at plus 2 level

Learning Objectives

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Learning outcomes

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

- Differentiate between supervised and unsupervised learning tasks.
- State the need of preprocessing, feature scaling and feature selection.
- Formulate classification, regression and clustering problems as optimization problems
- Implement various machine learning algorithms learnt in the course.

SYLLABUS OF DSC-17

Unit 1 (5 Hours)

Introduction: Basic definitions and concepts, key elements, supervised and unsupervised learning, introduction to reinforcement learning, applications of ML.

Unit 2 (8 Hours)

Preprocessing: Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis).

Unit 3 (12 Hours)

Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

Unit 4 (12 Hours)

Classification: Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics.

Unit 5 (8 Hours)

Clustering: Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

Essential/recommended readings

1. Mitchell, T.M. *Machine Learning*, McGraw Hill Education, 2017.
2. James, G., Witten. D., Hastie. T., Tibshirani., R. *An Introduction to Statistical Learning with Applications in R*, Springer, 2014.
3. Alpaydin, E. *Introduction to Machine Learning*, MIT press, 2009.

Additional References

1. Flach, P., *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2015.
2. Christopher & Bishop, M., *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2016.
3. Sebastian Raschka, *Python Machine Learning*, Packt Publishing Ltd, 2019

Suggested Practical List:

Practical exercises such as

Use Python for practical labs for Machine Learning. Utilize publicly available datasets from online repositories like <https://data.gov.in/> and <https://archive.ics.uci.edu/ml/datasets.php>

For evaluation of the regression/classification models, perform experiments as follows:

- Scale/Normalize the data
- Reduce dimension of the data with different feature selection techniques
- Split datasets into training and test sets and evaluate the decision models
- Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- MSE and R^2 score for regression models
- Accuracy, TP, TN, FP, FN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier
2. Simple Linear Regression multiple linear regression
3. Polynomial Regression
4. Lasso and Ridge Regression
5. Logistic regression
6. Artificial Neural Network
7. k -NN classifier
8. Decision tree classification
9. SVM classification
10. K-Means Clustering
11. Hierarchical Clustering