

SEMESTER-5

DEPARTMENT OF COMPUTER SCIENCE

[UG Programme for Bachelor in Computer Science (Honours)]

DISCIPLINE SPECIFIC CORE COURSE - 13 (DSC-13) : Algorithms and Advanced Data Structures

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		
DSC 13 Algorithms and Advanced Data Structures	4	3	0	1	Pass in Class XII	DSC 07 Data Structures with C++, DSC 10 Design and Analysis of Algorithms

Learning Objectives

This course is designed to build upon the fundamentals in data structures and algorithm design and gain exposure to more data structures and algorithms for new problems.

Learning outcomes

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

- Comprehend and use data structures for lists.
- Use hash tables for dictionaries.
- Comprehend and use data structures and algorithms for string matching.
- Apply disk based data structures.
- Implement and analyze advanced data structures and algorithms for graphs.
- Describe the purpose of randomization in data structures and algorithms.

Unit 1 (4 hours)

List and Iterator ADTs: Vectors, Lists, Sequences

Unit 2 (6 hours)

Hash Tables, Dictionaries: Hash Functions, Collision resolution schemes.

Unit 3 (8 hours)

Strings: String Matching: KMP algorithm; Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engines

Unit 4 (8 hours)

More on Trees: 2-4 Trees, B Trees

Unit 5 (8 hours)

More on Graphs: Bellman Ford Algorithm, Union Find Data Structures - application
Kruskal's algorithm

Unit 6 (6 hours)

Randomization: Randomized Quicksort, Randomized Select, Skip lists

Unit 7 (5 hours)

Network Flows: Ford Fulkerson algorithm for max flow problem.

Essential/recommended readings

1. Goodrich, M.T, Tamassia, R., & Mount, D. *Data Structures and Algorithms Analysis in C++*, 2nd edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4th edition, Prentice Hall of India, 2022.
3. Kleinberg, J., Tardos, E. *Algorithm Design*, 1st edition, Pearson, 2013.
4. Drozdek, A. *Data Structures and Algorithms in C++*, 4th edition, Cengage Learning. 2012.

Practical List : (30 Hours)

Practical exercises such as

1. Write a program to sort the elements of an array using Randomized Quick sort (the program should report the number of comparisons).
2. Write a program to find the ith smallest element of an array using Randomized Select.
3. Write a program to determine the minimum spanning tree of a graph using Kruskal's algorithm.
4. Write a program to implement the Bellman Ford algorithm to find the shortest paths from a given source node to all other nodes in a graph.

5. Write a program to implement a B-Tree.
6. Write a program to implement the Tree Data structure, which supports the following operations:
 - I. Insert
 - II. Search
7. Write a program to search a pattern in a given text using the KMP algorithm.
8. Write a program to implement a Suffix tree.

DISCIPLINE SPECIFIC CORE COURSE – 14 (DSC-14): Theory of Computation

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 14 Theory of Computati on	4	3	0	1	Pass in Class XII	DSC04 Object Oriented Programming with C++ / GE1a Programming using C++ /A course in C/C++ at plus 2 level

Learning Objectives

This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine; and their relationships with formal languages. make students aware of the notion of computation using abstract computing devices. Students will also learn about the limitations of computing machines as this course addresses the issue of which problems can be solved by computational means (decidability vs undecidability)

Learning outcomes

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

- design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.

- apply pumping lemma to prove that a language is non-regular/non-context-free.
- describe limitations of a computing machines and
- recognize what can be solved and what cannot be solved using these machines.

SYLLABUS OF DSC 14

Unit 1 (7 hours)

Introduction: Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

Unit 2 (15 hours)

Finite Automata and Regular: Regular expressions, Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA), relationship between NFA and DFA, Transition Graphs (TG), properties of regular languages, the relationship between regular languages and finite automata, pumping lemma, Kleene's theorem.

Unit 3 (15 hours)

Context-Free Languages (CFL): Context-Free Grammars (CFG), deterministic and non-deterministic Pushdown Automata (PDA), relationship between CFG and PDA, parse trees, leftmost derivation, Ambiguities in grammars, pumping lemma for CFL, properties of CFL, Chomsky Normal Form.

Unit 4 (8 hours)

Turing Machines and Models of Computations: Turing machine as a model of computation, configuration of Turing machine, Recursive and recursively enumerable languages, Church Turing Thesis, Universal Turing Machine, decidability, Halting problem.

Essential/recommended readings

1. Harry R. Lewis and Christos H. Papadimitriou, *Elements of the Theory of Computation*, 2nd Edition, Prentice Hall of India (PHI), 2002
2. Daniel I.A. Cohen, *Introduction to Computer Theory*, 2nd Edition, Wiley India Pvt. Ltd., 2011.

Additional References

1. J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3rd edition, Addison Wesley, 2006.
2. Peter Linz, *An Introduction to Formal Languages and Automata*, 6th edition, Jones & Bartlett Learning, 2017.
3. Michael Sipser, *Introduction to the Theory of Computation*, Cengage, 2014

DISCIPLINE SPECIFIC CORE COURSE– 15 (DSC-15): Software Engineering

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		
DSC 15 Software Engineering		3	0	1	Pass in Class XII	DSC01 Programming using Python/ DSC04 Object Oriented Programming with C++/A course in C/C++ or Python at plus 2 level

Learning Objectives

This course will acquaint the student with different approaches and techniques used to develop good quality software. The course includes learning of various software development process frameworks, requirement analysis, design modeling, qualitative and quantitative software metrics, risk management, and testing techniques.

Learning outcomes

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

- describe the software development models.
- analyse and model customer requirements and build design models.
- estimate and prepare schedule for software projects.
- analyse the impact of risks involved in software development.
- design and build test cases, and perform software testing.

SYLLABUS OF DSC 15

Unit 1 (9 hours)

Introduction: Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile, Agile Model – Scrum.

Unit 2 (6 hours)

Software Requirements Analysis and Specification: Use Case Approach, Software Requirement Specification Document, Flow-oriented Model, Data Flow Model

Unit 3 (8 hours)

Design Modeling: Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Structure Charts.

Unit 4 (7 hours)

Software Metrics and Project Estimation: Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations); Project Scheduling (Timeline charts, tracking the schedule).

Unit 5 (5 hours)

Quality Control and Risk Management: Quality Control and Quality Assurance, Software Process Assessment and Improvement; Software Risks, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management.

Unit 6 (10 hours)

Software Testing: Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

Essential/recommended readings

1. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 9th edition, McGraw-Hill, 2020.
2. Aggarwal, K.K., Singh, Y. *Software Engineering*, 3rd edition, New Age International Publishers, 2007.
3. Jalote, P. *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2005.

Additional References

1. Sommerville, I. *Software Engineering*, 9th edition, Addison Wesley, 2011.
2. *The Definitive Guide to Scrum: The Rules of the Game*, Ken Schwaber, Jeff Sutherland, July 2016.

Suggested Practical List :(30 Hours)

Practical exercises such as

The students document, design and code a module of a Software Project using an appropriate Software Process model. The Software Project should include the use of software engineering tools and include.

1. Problem Statement, Process Model
2. Requirement Analysis: Create Data Flow, Data Dictionary, Use Cases, Sequence Diagram, Software Requirement Specification Document
3. Project Management: Timeline Chart, Compute FP, Effort, Cost, Risk Table.
4. Design Engineering: Architectural Design, Pseudocode of a small module.
5. Coding: Develop at least a single module using any programming Language
6. Testing: Compute Basic path set for at least one module from a project, Generate test cases.

Some of the sample projects are given below:

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers
2. DTC Route Information: Online information about the bus routes and their frequency and fares.
3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation System
8. Automatic Internal Assessment System
9. Parking Allocation System

Computer Science Courses for Undergraduate Programme of study with **Computer Science** discipline Elective

DISCIPLINE SPECIFIC ELECTIVE COURSE: Data Mining-II

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		
Data Mining-II	4	3	0	1	Pass in Class XII	DSC01 Programming using Python, / GE1b Programming with Python / AI Programming Fundamentals using Python ,Data Mining-I

Learning Objectives

The course introduces the students to the important supervised and unsupervised learning techniques. Students will learn about the importance of ensemble methods, cluster analysis, anomaly detection and their applicability in mining patterns in real applications. At the end students will be exposed to two advanced topics: text mining and time-series mining. Students will use the learned topics in solving real applications using Open-source software.

Learning outcomes

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

- Differentiate between partition-based, density-based and hierarchical clustering.
- Build ensemble models to improve predictive performance of the classifier.
- Identify anomalies and outliers using supervised and unsupervised techniques.
- Analyze time-series data and extract patterns from the stamped data.
- Mine textual data and do topic modelling.

SYLLABUS OF DSE

Unit 1 (10 hours)

Clustering: Partitioning Methods, Hierarchical Methods, Density-Based Methods, Comparison of different methods

Unit 2 (8 hours)

Ensemble Methods: Need of ensemble, Random Forests, Bagging and Boosting

Unit 3 (10 hours)

Anomaly Detection: Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-based and density-based outlier detection, Clustering-based approaches

Unit 4 (7 hours)

Mining Text Data: Document Preparation and Similarity, Clustering Methods for Text, Topic Modeling

Unit 5 (10 hours)

Stream Mining: Time series basics, Date Ranges, Frequencies, and Shifting, Resampling and moving windows functions, Decay function, Clustering stamped data: STREAM and CluStream

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. *Introduction to Data Mining*, 2nd edition, Pearson, 2019.
2. Zaki M. J. and Meira J. Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2nd edition, Cambridge University Press, 2020.
3. Aggarwal C. C. *Data Mining: The Textbook*, Springer, 2015.

Additional References

1. Han J. Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2011.
2. Dunham M. *Data Mining: Introductory and Advanced Topics*, Pearson, 2006.

Suggested Practical List : (30 Hours)

Practical exercises such as

Use a dataset of your choice from Open Data Portal ([https:// data.gov.in/](https://data.gov.in/), UCI repository or a dataset verified by the teacher) or load from scikit, seaborn library for the following exercises to practice the concepts learnt.

1. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics.
2. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance.

3. Apply different outlier-detection methods on a noisy dataset and compare their effectiveness in terms of outliers reported.
4. Compute similarity between two documents after required document preparation.
5. Considering a time-stamped data (sales data/weather data), compare the aggregate values visually using different moving windows function.
6. Write a program to find the latent topics in a document using any topic modeling method and display top 5 terms that contribute to each topic along with their strength. Also, visualize the distribution of terms contributing to the topics.

Project: Students are encouraged to work on a good dataset in consultation with their faculty and apply the concepts learned in the course.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Data Privacy

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		
Data Privacy	4	3	0	1	Pass in Class XII	NIL

Learning Objectives

This course aims to provide students with an ability to identify privacy related aspects of data uses (including attacks on data privacy), evaluate proposed mechanisms for privacy protection and relate to ethical issues related to data privacy.

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

- Understand the basic principles of data privacy and the implications of data breaches.
- Identify and evaluate different methods of protecting sensitive data.
- Explain the role of privacy regulations in safeguarding personal information.
- Implement basic cryptographic techniques to secure data.
- Apply data anonymization techniques to protect personal information.
- Analyze the ethical considerations in data privacy

SYLLABUS OF DSE

Unit 1 (10 hours)

Introduction to Data Privacy and Privacy Regulations: Notion of data privacy, Historical context of data privacy, Types of sensitive data, Privacy laws and regulations.

Unit 2 (15 hours)

Data Privacy Attacks, Cryptography and Data Protection: Type of Attacks/ Data Breaches on Data Privacy, Impact of Data Breaches / Attacks, Introduction to cryptography, Symmetric and asymmetric encryption, Hashing and digital signatures.

Unit 3 (10 hours)

Data Collection, Use and Reuse: Harms Associated with Data collections, use and reuse, Introduction to data anonymization, Data Anonymization Techniques for anonymizing data, Challenges in anonymizing data

Unit 4 (10 hours)

Ethical considerations in Data Privacy: Privacy and Surveillance, Ethics of Data Collection and Use, Bias and discrimination in data analysis

Essential/recommended readings

1. Ronald Leenes, Rosamunde van Brakel, and Serge Gutwirth: *Data Protection and Privacy: The Age of Intelligent Machines*, Hart Publishing, 2017.
2. Naavi: *Personal Data Protection Act of India (PDPA 2020)*: Be Aware, Be Ready and Be Compliant, Notion Press, 2020.
3. Ravinder Kumar Gaurav Goyal, *The Right to Privacy in India: Concept and Evolution*, Publisher: Lightning Source, 2016.

Additional References

1. https://onlinecourses.nptel.ac.in/noc22_cs37/preview
2. <https://www.coursera.org/learn/northeastern-data-privacy/home/info>

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Data Privacy Audit: Students can conduct a data privacy audit of a company or organization to identify potential vulnerabilities and risks in their data privacy practices.

2. Privacy Impact Assessment: Students can conduct a privacy impact assessment (PIA) of a new technology or system to identify potential privacy risks and develop strategies to mitigate them.
3. Regulation Compliance: Students can explore the requirements of the Data Protection Regulations and develop a plan for ensuring compliance with the regulation.
4. Cryptography: Students can learn about different cryptographic techniques and tools, such as encryption, hashing, and digital signatures, and implement them in practice.
5. Anonymization Techniques: Students can learn about data anonymization techniques, such as k-anonymity, differential privacy, and data masking, and apply them to a real-world dataset.
6. Privacy Policy Analysis: Students can analyze the privacy policies of different companies and identify gaps or areas for improvement.
7. Privacy-Enhancing Technologies: Students can explore privacy-enhancing technologies (PETs), such as virtual private networks (VPNs), Tor, and secure messaging apps, and evaluate their effectiveness in protecting privacy.
8. Privacy Breach Response Plan: Students can develop a privacy breach response plan for a company or organization, including steps to take in the event of a data breach and strategies for communicating with affected parties.
9. Ethical Considerations: Students can explore ethical considerations in data privacy, such as the balance between privacy and security, the impact of data collection and analysis on marginalized communities, and the role of data ethics in technology development.
10. Case Studies: Students can analyze case studies of privacy breaches or successful privacy protection strategies, and identify key lessons and takeaways.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Unix Network Programming

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	
		Lecture	Tutorial	Practical/ Practice		
Unix Network Programming	4	3	0	1	Pass in Class XII	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 the concepts of Internet protocols, ports used during communication, Client/Server concepts and various transport protocols used in computer network applications and services. The objective is to equip the students with technical knowledge of it comprises of the study of the sockets used with TCP and UDP.

Learning outcomes

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

- Describe and analyze the various Internet Transport layer protocols used in TCP AND UDP.
- Comprehend the concepts and structures of both TCP based connection-oriented and UDP based connectionless client server applications.
- Write various real-life client-server applications using socket programming.
- Modify, maintain and extend the present internet client-server applications and write any new type of internet applications to suit the current needs of Internet users.

SYLLABUS OF DSE

Unit 1 (6 hours)

Introduction Basics of Client Server applications, Example of day time client server, concurrent servers, protocols, sockets, port numbers.

Unit 2 (17 hours)

Connection-oriented Socket Applications: Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, close function, Socket Address Structures, Byte Ordering and Manipulation Functions, TCP Client and Server for Echo, Signal Handling in case of crashing and rebooting of server, Shutdown process function. Socket Options: Getsockopt and stockpot functions, Socket states, Generic socket option.

Unit 3 (15 hours)

Connectionless Socket Applications: TCP-oriented basic concurrent client server applications, UDP oriented Echo client and server application, Handling of errors like lost datagram, Lack of flow control with UDP, determining outgoing interface with UDP.

Unit 4 (7 hours)

Elementary name and Address conversions: Domain Name System, socket functions like gethostbyname, gethostbyname2, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions.

Essential/recommended readings

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, *Unix Network Programming*, The sockets Networking API, Vol. 1, 3rd Edition, PHI.
2. B. A. Forouzan: *Data Communications and Networking*, THM Publishing Company Ltd
3. R. Stevens, *Unix Network Programming*, PHI 2nd Edition

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Implement TCP Echo client and TCP Echo server (Iterative).
2. Implement TCP Echo client and TCP Echo server (Concurrent).
3. Implement TCP daytime client and TCP daytime server (Iterative).
4. Implement TCP daytime client and TCP daytime server (concurrent).
5. Implement UDP Echo Client and UDP Echo Server.
6. Implement UDP daytime Client and UDP daytime server.
7. Implement TCP client and server (concurrent) where client gets input from the user and sends it to server. Server displays it on the screen. Server then gets another input from the user and sends it to client. Client displays it on the screen. The process continues till server or client sends “bye” to the other party.
8. Implement TCP client and server (concurrent) where client requests server to transfer a file. Assume file is smaller than 1K size. If the file is present on the server, it is sent to the client otherwise an error message is sent to client. Client copies the file on the hard disk and disconnects.
9. Implement UDP client and UDP server where server displays the IP address and port number of the client sending the datagram. Client sends a datagram (size 64 bytes) three times to the same server. Server sends the message back to client. Client reports the time elapsed in sending and receiving of the message. Use connected UDP sockets.
10. Write a program to
 - a) display name of the host
 - b) all IP addresses of the host.
 - c) Check whether FTP and HTTP services are running on the system.
 - d) Display the name of the service running on port number specified by user.

COMMON POOL OF GENERIC ELECTIVES (GE) COURSES

(For all the Generic Elective courses offered by your department, please put it in the format provided below)

GENERIC ELECTIVES (GE-5a): Operating Systems

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-5a Operating Systems	4	3	0	1	Pass in Class XII	Knowledge of Programming in C/C++/Java/Python at class XII level or above

Learning Objectives

The course introduces Operating System and its importance in computer systems. The focus is to explain the common services provided by an operating system like process management, memory (primary, secondary & virtual) management, I/O management, file management. The course talks about the various functional components of the operating and their design.

Learning outcomes

On successful completion of the course, students will be able to gain knowledge of different concepts of the operating System and its components. They would learn about shell scripts and would be able to use the system in an efficient manner.

SYLLABUS OF GE-5a

Unit 1 (5 hours)

Introduction: Operating Systems (OS) definition and its purpose, Multiprogrammed and Time-Sharing Systems, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

Unit 2 (6 hours)

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

Unit 3 (10 hours)

Process Management: Process Concept, States. Process Control Block, Context Switch, Process scheduling, Schedulers, Overview of threads and Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin.

Unit 4 (10 hours)

Memory Management: Physical and Logical address space, Swapping Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging, virtual memory: Demand Paging.

Unit 5 (8 hours)

File and Input / Output Device Management: File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories, Magnetic Disks, Solid-State Disks, Magnetic Tapes.

Unit 6 (6 hours)

Shell Scripting: Shell variables, parameter passing conditional statements, iterative statements, writing and executing shell scripts, utility programs (cut, paste, grep, echo, pipe, filter etc.)

Essential/recommended readings

1. Galvin, S. P. B., Gagne, G., *Operating System Concepts*, 9th edition, John Wiley Publications, 2016.
2. G. Nutt, *Operating Systems*, Pearson, 2009

Additional References

1. Dhamdhere, D. M., *Operating Systems: A Concept-based Approach*, 2nd edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Pike, R., *The Unix Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
3. Stallings, W., *Operating Systems: Internals and Design Principles*, 9th edition, Pearson Education, 2018.
4. Tanenbaum, A. S., *Modern Operating Systems*. 3rd edition. Pearson Education, 2007.

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Usage of following commands: ls, pwd, cat, who, rm, mkdir, rmdir, cd.

2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, bc.
4. Write a shell script to display date in the mm/dd/yy format.
5. Write a shell script to display the multiplication table any number.
6. Write a shell script to find the factorial of a given number.
7. Program to show the pyramid of special character “*”.
8. Write a shell script to find the sum of digits of a given number.
9. Write a shell script to perform the tasks of basic calculator.
10. Write a shell script to find the power of a given number.
11. Write a shell script to check whether the number is Armstrong or not.
12. Write a shell script to find the GCD (greatest common divisor) of two numbers.
13. Write a shell script to check if the number entered at the command line is prime or not.
14. Write a shell script to display on the screen sorted output of “who” command along with the total number of users.
15. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.
16. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
17. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
18. Write a shell script to check whether the file have all the permissions or not.
19. Write a shell script to modify “cal” command to display calendars of the specified months.
20. Write a shell script to modify “cal” command to display calendars of the specified range of months.

GENERIC ELECTIVES (GE-5b): Advanced Web 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		
GE5b: Advanced Web Programming	4	3	0	1	Pass in Class XII	Knowledge of HTML, CSS

Learning Objectives

The course aims to familiarize the students with the concepts and techniques of server side web programming. This will enable the students to create dynamically generated web pages using HTML, PHP, MySQL and JQuery.

Learning outcomes

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

- develop interactive and dynamic websites.
- write programs to communicate with the server using GET and POST methods
- learn to connect and manipulate databases using MySql
- validate server-side/backend data

SYLLABUS OF GE-5b

Unit 1 (7 hours)

Introduction to PHP: Basic syntax, defining variables and constants, data types including arrays, operators and expressions, decision making statements, constructs for iterations.

Unit 2 (5 hours)

String Handling: Creating a string and accessing its content, searching and replacing content of a string, and other built-in functions.

Unit 3 (12 hours)

Handling HTML Form with PHP: Creating a form, submitting data to the server at the backend using GET and POST methods, GET vs POST methods, PHP global functions.

Unit 4 (15 hours)

Database: Connectivity with MySQL: Connectivity with database, database creation, creating tables, create, retrieve, update, and delete (CRUD) operations

Unit 5 (6 hours)

jQuery and JSON: Introduction to jQuery syntax (selectors, events, AJAX, JSON).

Essential/recommended readings

1. Nixon, R. Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5, O'Reilly, 2018.
2. Holzner S. PHP: The Complete Reference, McGraw Hill, 2017
3. Murach J, Murach's PHP and MySQL, 2nd edition, Mike Murach & Associates, 2014.
4. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, Wiley, 2014.

Additional References

1. <https://www.w3schools.com/php/default.asp>

2. <https://www.tutorialspoint.com/php/index.htm>

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Write a PHP script to reverse the digits of a number.
2. Create a web page containing two text boxes and a button named “Evaluate”. When the user enters numbers in the text boxes and clicks on the “Evaluate” button, a function should evaluate the sum of the numbers and display the result.
3. Write a PHP script to perform following string operations using in-built functions and built an interactive web page having buttons for each of the following operation:
 - a. Find the length of a string
 - b. Find a substring from a string
 - c. Replace text within a string
 - d. Remove whitespace and other predefined characters from both sides of a string.
 - e. Check if a value is a string
 - f. Convert the first character of each word in a string into uppercase.
4. Design a Login form and validate that form using PHP code. Display error message box when data is not valid otherwise redirect to the next page and display “Welcome username!”.
5. Design a student registration form, using appropriate input fields consisting of following:
 - a. First Name
 - b. Last Name
 - c. Gender
 - d. Roll Number
 - e. Phone Number
 - f. Course

Submit and retrieve the form data using \$_POST, \$_GET variable.

6. Write PHP Code to make connection to MySQL database, create database and tables and perform insertion, deletion, and retrieval of the data (Using SQL operations like .JOIN, ORDER BY, GROUP BY) Display the messages such as “The record is added in the database!” when data is inserted into the database, “A record is deleted from the database” when data is deleted from the database. Use appropriate button names such as Add Data, Delete Data, and Display Data.

jQuery and JSON

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery

4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

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		
GE5c: Java Based Web App Development	4	3	0	1	Pass in Class XII	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 / A programming course at class XII level

Learning Objectives

The course aims to familiarize the students with the concepts and techniques of web app development based on Java. The students will learn about database connectivity, use of HTTP protocol, client side programming, and use of servlets and JSP for server side programming.

Learning outcomes

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

- develop an understand of client-server architecture, HTTP protocol, and web application components.
- connect an application to database and perform basic database operations.
- create servlets and JSP for web applications
- deploy web applications

SYLLABUS OF GE-5c

Unit 1 (8 hours)

Review of Programming Language: Programming Constructs, Data types, Operators, Concepts of Class, Interface, Inheritance, Exception Handling, Util package, Multithreading, event handling.

Unit 2 (10 hours)

Java Database Connections: Database connectivity, Connection, statement, result set object, Metadata, Connection pooling, CRUD operations, Prepared and callable statements

Unit 3 (15 hours)

Introduction to servlets: Concepts of Streams, events and listener, recap of HTML, CSS, XML, Servlet package and interface, life cycle of servlet, deployment descriptor, Filters, HTTP and Generic servlet, request dispatcher, Request Response classes, Dynamic page designing using servlet.

Unit 4 (12 hours)

Introduction to JSP: JSP Life cycle, tags in JSP, custom tags, Expression Language, Introduction to Struts Framework, Implicit objects, database access using JSP

Essential/recommended readings

1. Herbert Schildt, *Java : The Complete Reference*, 12th edition, McGraw-Hill Education, 2021.
2. Hans Bergsten, *Java Server Pages*, 3rd edition, O'Reilly, 2003.
3. Jim Keogh, *The Complete Reference J2EE*, 1st edition, McGraw-Hill Education, 2017.

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Setting up the development environment: Install Java Development Kit (JDK), Eclipse IDE, and Apache Tomcat web server. Create a new web project in Eclipse.
2. Writing and deploying a "Hello World" servlet: Create a simple servlet that prints "Hello World" on the web page. Deploy the servlet on Tomcat and test it in a web browser.
3. Handling HTTP requests and responses: Write a servlet that reads input from HTTP requests and sends output as HTTP responses.
4. Creating a JSP page: Create a JSP page that displays dynamic.
5. Write a servlet that handles form submissions and saves the data to a database using JDBC.
6. Write a servlet that implements user authentication and authorization using a database.
7. Creating a web application using MVC architecture: Create a web application using Model-View-Controller (MVC) architecture. Use servlets as controllers, JSP pages as views.
8. Deploying a web application to a server: Configure and deploy a web application to a server using Apache Tomcat Manager or other deployment tools.

(Computer Science Courses for Undergraduate Programme of study with **Computer Science** discipline as one of the **three** Core Disciplines)

DISCIPLINE SPECIFIC CORE COURSE (DSC-5): Database Management Systems

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		
DSC 5: Database Management Systems	4	3	0	1	Pass in Class XII	NIL

Learning Objectives

The course introduces the students to the fundamentals of database management system and its applications. Emphasis is given on the popular relational database system. Students will learn about the importance of database structure and its designing using Entity Relationship diagram and formal approach using normalization. Basic concepts of file indexing and transaction processing will be taught. The course would give students hands-on practice of structured query language to create, manipulate and implement a relational database.

Learning outcomes

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

- Use relational database management software to create and manipulate the database.
- Create conceptual data models using entity relationship diagrams for modeling real-life situations and map it to corresponding relational database schema.
- Use the concept of functional dependencies to remove redundancy and update anomalies.
- Apply normalization theory to get a normalized database scheme to get anomalies free database.
- Write queries in relational algebra.
- Implement relational databases and formulate queries for data retrieval and data update problems using SQL.
- Learn the importance of index structures and concurrent execution of transactions in database systems.

SYLLABUS OF DSC-5

Unit 1 (5 hours)

Introduction to Database: Database, characteristics of database approach, data models, database management system, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

Unit 2 (6 hours)

Entity Relationship Modeling: Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram as conceptual data model.

Unit 3 (6 hours)

Relational Data Model: Data anomalies, Relational Data Model - Characteristics of a relation, schema-instance distinction, types of keys, relational integrity constraints. Relational algebra operators like selection, projection, cartesian product, join and write queries using them.

Unit 4 (8 hours)

Structured Query Language (SQL): DDL to create database and tables, table constraints, DML, Querying in SQL to retrieve data from the database, aggregation functions group by and having clauses, generate and query views.

Unit 5 (10 hours)

Database Design: Mapping an Entity Relationship diagram to corresponding relational database scheme, functional dependencies and Normal forms, 1NF, 2NF, and 3NF decompositions and desirable properties of them.

Unit 6 (10 hours)

File indexing and Transaction Processing: Need of file indexes, types of indexes, file organizations, single- and multi-level indexing, concurrent execution of transactions, ACID properties, need of data recovery.

Essential/recommended readings

1. Elmasri, R., Navathe, B. S., *Fundamentals of Database Systems*, 7th edition, Pearson Education, 2016.
2. Murach, J., *Murach's MySQL*, 3th edition, Pearson, 2019.

Additional References

1. Connolly, T. M., Begg, C. E., *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th edition, Pearson, 2019.
2. Ramakrishnan, R., Gehrke, J., *Database Management Systems*, 3rd edition, McGraw-Hill, 2014.
3. Silberschatz, A., Korth, H.F., Sudarshan S., *Database System Concepts*, 7th edition, McGraw Hill, 2019.

Suggested Practical List : (30 Hours)

Practical exercises such as

Create and use the following student-course database schema for a college to answer the given queries using the standalone SQL editor.

STUDENT	<u>Roll No</u>	StudentName	CourseID	DOB
	Char(6)	Varchar(20)	Varchar(10)	Date

COURSE	<u>CID</u>	CourseName	Course Type	Teacher-in-charge	TotalSeats	Duration
	Char(6)	Varchar (20)	Char (8)	Varchar (15)	Unsigned int	Unsigned int

ADMISSION	<u>Roll No</u>	<u>CID</u>	DateOfAdmission
	Char(6)	Char(6)	Date

Here Rollno (ADMISSION) and SID (ADMISSION) are foreign keys. Note that course type may have two values viz. Fulltime and Parttime and a student may enroll in any number of courses

1. Retrieve names of students enrolled in any course.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find course details with more than N students enrolled where N is to be input by the user
7. Update student table for modifying a student name.
8. Find course names in which more than five students have enrolled
9. Find the name of youngest student enrolled in course 'BSc(P)CS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two popular part time courses (on the basis of enrolled students)
12. Find the student names who are admitted to full time courses only.
13. Find course names in which more than 30 students took admission

14. Find names of all students who took admission to any course and course names in which at least one student has enrolled
15. Find course names such that its teacher-in-charge has a name with 'Gupta' in it and the course is full time.
16. Find the course names in which the number of enrolled students is only 10% of its total seats.
17. Display the vacant seats for each course
18. Increment Total Seats of each course by 10%
19. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
20. Update date of admission of all the courses by 1 year.
21. Create a view to keep track of course names with the total number of students enrolled in it.
22. Count the number of courses with more than 5 students enrolled for each type of course.
23. Add column Mobile number in student table with default value '9999999999'
24. Find the total number of students whose age is > 18 years.
25. Find names of students who are born in 2001 and are admitted to at least one part time course.
26. Count all courses having 'science' in the name and starting with the word 'BSc'.

Students are also encouraged to implement the database given in the textbook and do the related queries.

Computer Science Courses for Undergraduate Programme of study with **Computer Science** discipline as one of the **two** Core Disciplines
(For e.g. courses for B.A. Programmes with Computer Science as Major discipline)

Computer Science Major

DISCIPLINE SPECIFIC CORE COURSE (DSC-5): Database Management System

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		
DSC05: Database Management System	4	3	0	1	Class XII Passed	NIL

Learning Objectives

The course will give an overview of categories of data models used by database management systems and writing queries in relational algebra. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice to write complex queries, nested queries using aggregate functions in SQL and to use basic database administration commands.

Learning outcomes

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

- Write queries using relational algebra, a procedural language.
- Use the concept of functional dependencies to remove data anomalies and arrive at normalized database design.
- Understand the data storage system, file organization and need of file indexing.
- Learn the importance of transaction processing and concurrency control.
- Write complex and nested SQL queries, and learn basic database administration commands.
- Acquire information about emerging technologies.

SYLLABUS OF DSC 05

Unit 1 (6 hours)

Data models and Relational Algebra: Categories of data models, types of database users, Relational Algebra Operations from SET Theory, SELECT, PROJECT, JOIN, DIVISION Operations.

Unit 2 (8 hours)

Normalization: Functional dependencies, minimal cover, normalizing database with multiple keys till 3NF, Boyce-Codd Normal Form

Unit 3 (8 hours)

Database Storage and index Structures: Storage of data, file structure, file organization and its types: Sequential, Heap and Indexed, Hash file, need for file Indexing, single- and multi-level indexing.

Unit 4 (8 hours)

Transaction Processing: Concurrent execution of transactions and their handling, ACID properties, need of data recovery and log files.

Unit 5 (8 hours)

Advanced SQL: Nested and complex queries using Inner JOIN, Left JOIN, Right JOIN, Full JOIN, views, Database Administration Commands: COMMIT, ROLLBACK, drop database, control permissions etc.

Unit 6 (7 hours)

Emerging Technology: Distributed Database Concepts, Introduction to emerging technologies like Data Warehousing and OLAP, Data Mining.

Essential/recommended readings

1. Elmasri R. and Navathe B. S. *Fundamentals of Database Systems*, 7th Edition, Pearson Education, 2016.
2. Murach J. *Murach's MySQL*, 3th Edition, Pearson, 2019.

Additional References

1. Silberschatz, A., Korth, H.F., & Sudarshan, S. Database System Concepts, 8th Edition, McGraw Hill, 2019.
2. Ramakrishnan, R. & Gehrke, J. Database Management Systems, 3rd Edition, Tata McGraw Hill Education, 2014.

Suggested Practical List : (30 Hours)

MySQL might be used for practical purposes.

The following concepts must be introduced to the students:

1. Multiple table queries (join on different and same tables)
2. Nested select statements
3. Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists union, intersect, minus, etc.)
4. Categorization using group by... having
5. Arranging using order by

List of sample queries to be done using the above schema.

Create tables with relevant column and integrity constraints and populate the tables with data.

Perform the following queries on the database:

1. Display all the details of all employees working in the company.
2. Display ssn, lname, fname, address of employees who work in department no 7.
3. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong' and who is a manager
4. Retrieve the name and salary of every employee working in department 'Production'
5. Display maximum and minimum salaries of each department
6. Retrieve all employee names whose address is in 'Bellaire' and working on at least one project
7. Modify the datatype of hours column to float in works_on table.
8. Retrieve all employees who age is more than 35 years
9. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
10. Add another column named "Join_Date" in the employee table and "Address" in the department table
11. Retrieve the names of all employees who do not have supervisors
12. Retrieve SSN and department name for all employees
13. Retrieve the name and address of all employees who work for the 'Research' department
14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate

15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor
16. Retrieve all combinations of Employee Name and Department Name
17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project
18. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees
19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name
20. Select the names of employees whose salary does not match with salary of any employee in department 10
21. Drop the column "Join_Date" from the employee table
22. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee
23. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
24. Change the name of existing table DEPT_LOCATIONS to DLOCATIONS
25. Find the sum of the salaries of all employees, the maximum salary, the minimum salary and the average salary for each department. Display with proper headings
26. Find the sum of the salaries and number of employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department
27. Select the names of employees whose salary is greater than the average salary of all employees in department 10
28. For each department, retrieve the department number, the number of employees in the department and their average salary
29. For each project, retrieve the project number, the project name, and the number of employees who work on that project for more than 10 hours
30. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively
31. For each department having more than 10 employees, retrieve the department name, number of employees drawing more than 40,000 as salary
32. Display employee names having no dependent in descending order along with their age
33. For each department, find the number of female and number of male employees along with name of manager of that department
34. Find the name and age of youngest employee in each department
35. Change the name of MINIT attribute of an employee table to MNAME using alter command
36. Create a view to keep names of employees and their department names if they are not working on any project
37. Create a view to keep track of all employee names who are working on same projects as that of employee 'Franklin T. Wong'
38. Execute system administrative commands like commit, rollback, granting control permissions, etc.

DISCIPLINE SPECIFIC CORE COURSE (A5): Machine Learning

0 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		
A5: Machine Learning	4	3	0	1	Pass in Class XII	A1 Programming Fundamentals using Python, A2 Data Visualization using Python

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.
- Appreciate the need of preprocessing, feature scaling and feature selection.
- Understand the fundamentals of classification, regression and clustering
- Implement various machine learning algorithms learnt in the course.

SYLLABUS OF A5

Unit 1 (8 hours)

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

Unit 2 (10 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 (15 hours)

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

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 Practicals List : (30 Hours)

Practical exercise such as

Use Python for practical labs for Machine Learning. Utilize publically 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:

- i. Scale/Normalize the data
- ii. Reduce dimension of the data with different feature selection techniques
- iv. Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- i. MSE and R^2 score for regression models
- ii. 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

Computer Science Courses for Undergraduate Programme of study with **Computer Science discipline as one of the **two** Core Disciplines**
(For e.g. courses for B.A. Programmes with Computer Science as Non-major discipline)

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		
DSC05: Database Management System	4	3	0	1	Pass in Class XII	NIL

Learning Objectives

The course will give an overview of categories of data models used by database management systems and writing queries in relational algebra. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice to write complex queries, nested queries using aggregate functions in SQL and to use basic database administration commands.

Learning outcomes

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

- Write queries using relational algebra, a procedural language.
- Use the concept of functional dependencies to remove data anomalies and arrive at normalized database design.
- Understand the data storage system, file organization and need of file indexing.
- Learn the importance of transaction processing and concurrency control.
- Write complex and nested SQL queries, and learn basic database administration commands.
- Acquire information about emerging technologies.

SYLLABUS OF DSC 05

Unit 1 (6 hours)

Data models and Relational Algebra: Categories of data models, types of database users, Relational Algebra Operations from SET Theory, SELECT, PROJECT, JOIN, DIVISION Operations.

Unit 2 (8 hours)

Normalization: Functional dependencies, minimal cover, normalizing database with multiple keys till 3NF, Boyce-Codd Normal Form

Unit 3 (8 hours)

Database Storage and index Structures: Storage of data, file structure, file organization and its types: Sequential, Heap and Indexed, Hash file, need for file Indexing, single- and multi-level indexing.

Unit 4 (8 hours)

Transaction Processing: Concurrent execution of transactions and their handling, ACID properties, need of data recovery and log files.

Unit 5 (8 hours)

Advanced SQL: Nested and complex queries using Inner JOIN, Left JOIN, Right JOIN, Full JOIN, views, Database Administration Commands: COMMIT, ROLLBACK, drop database, control permissions etc.

Unit 6 (7 hours)

Emerging Technology: Distributed Database Concepts, Introduction to emerging technologies like Data Warehousing and OLAP, Data Mining.

Essential/recommended readings

1. Elmasri R. and Navathe B. S. *Fundamentals of Database Systems*, 7th Edition, Pearson Education, 2016.
2. Murach J. *Murach's MySQL*, 3th Edition, Pearson, 2019.

Additional References

1. Silberschatz, A., Korth, H.F., & Sudarshan, S. *Database System Concepts*, 8th Edition, McGraw Hill, 2019.

Suggested Practical List : (30 Hours)

MySQL might be used for practical purposes.

The following concepts must be introduced to the students:

1. Multiple table queries (join on different and same tables)
2. Nested select statements
3. Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists union, intersect, minus, etc.)
4. Categorization using group by.....having
5. Arranging using order by

List of sample queries to be done using the above schema.

Create tables with relevant column and integrity constraints and populate the tables with data.

Perform the following queries on the database:

1. Display all the details of all employees working in the company.
2. Display ssn, lname, fname, address of employees who work in department no 7.
3. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong' and who is a manager
4. Retrieve the name and salary of every employee working in department 'Production'
5. Display maximum and minimum salaries of each department
6. Retrieve all employee names whose address is in 'Bellaire' and working on at least one project
7. Modify the datatype of hours column to float in works_on table.
8. Retrieve all employees who age is more than 35 years
9. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
10. Add another column named "Join_Date" in the employee table and "Address" in the department table
11. Retrieve the names of all employees who do not have supervisors
12. Retrieve SSN and department name for all employees
13. Retrieve the name and address of all employees who work for the 'Research' department
14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate
15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor
16. Retrieve all combinations of Employee Name and Department Name
17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project
18. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees
19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name

20. Select the names of employees whose salary does not match with salary of any employee in department 10
21. Drop the column "Join_Date" from the employee table
22. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee
23. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
24. Change the name of existing table DEPT_LOCATIONS to DLOCATIONS
25. Find the sum of the salaries of all employees, the maximum salary, the minimum salary and the average salary for each department. Display with proper headings
26. Find the sum of the salaries and number of employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department
27. Select the names of employees whose salary is greater than the average salary of all employees in department 10
28. For each department, retrieve the department number, the number of employees in the department and their average salary
29. For each project, retrieve the project number, the project name, and the number of employees who work on that project for more than 10 hours
30. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively
31. For each department having more than 10 employees, retrieve the department name, number of employees drawing more than 40,000 as salary
32. Display employee names having no dependent in descending order along with their age
33. For each department, find the number of female and number of male employees along with name of manager of that department
34. Find the name and age of youngest employee in each department
35. Change the name of MINIT attribute of an employee table to MNAME using alter command
36. Create a view to keep names of employees and their department names if they are not working on any project
37. Create a view to keep track of all employee names who are working on same projects as that of employee 'Franklin T. Wong'
38. Execute system administrative commands like commit, rollback, granting control permissions, etc.

Semester-VI

DEPARTMENT OF COMPUTER SCIENCE

[UG Programme for Bachelor in Computer Science (Honours)]

DISCIPLINE SPECIFIC CORE COURSE – 16 (DSC-16) : Artificial Intelligence

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		
DSC-16 Artificial Intelligence	4	3	0	1	Pass in Class XII	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

The course objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

Learning outcomes

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

- identify problems that are amenable to solutions by specific AI methods.
- state the utility of different types of AI agents.
- apply different informed search techniques for solving real world problems.
- use knowledge representation techniques for AI systems.

SYLLABUS OF DSC-16

Unit 1 6 Hours

Introduction: Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent

approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

Unit 2 12 Hours

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

Unit 3 16 Hours

Knowledge Representation: Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

Unit 4 8 Hours

Understanding Natural Languages: Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented transition nets.

Unit 5 3 Hours

AI The Present and the Future: Symbolic AI, Data-driven AI and Machine Learning, Introduction to Machine Learning and Deep Learning based AI, some applications of symbolic and data driven AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

Essential/recommended readings

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4th edition, 2020..
2. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th edition, 2012.
3. Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
4. Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5th edition, Springer, 2003.

Additional references

1. Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010

Practical List :

Practical exercises such as

1. Write a program in Prolog to implement TowerOfHanoi(N) where N represents the number of disks.
2. Write a program to implement the Hill climbing search algorithm in Prolog.
3. Write a program to implement the Best first search algorithm in Prolog.
4. Write a program to implement A* search algorithm in Prolog.
5. Write a program to implement the min-max search algorithm in Prolog.

6. Write a program to solve the Water-Jug Problem in Prolog.
7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Prolog.
8. Write a Prolog program to implement the family tree and demonstrate the family relationship.
9. Write a Prolog program to implement knowledge representation using frames with appropriate examples.
10. Write a Prolog program to implement `conc(L1, L2, L3)` where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement `reverse(L, R)` where List L is original and List R is reversed list.
12. Write a Prolog program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a Prolog program to recognize context free grammar $a^n b^n$.

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 CORE COURSE– 18 (DSC-18): 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		
DSC-18 Introduction to Parallel Programming	4	3	0	1	Pass in Class XII	DSC-02 Computer Systems Architecture DSC04 /A course in C/C++ at plus 2 level, DSC-07 Data Structures with C++, DSC-08 Operating Systems

Learning Objectives

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).

Learning outcomes

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

- State the need of Parallel algorithms
- Describe architectures for parallel and distributed systems.
- Develop elementary parallel algorithms for shared memory models.
- Develop elementary parallel algorithms for distributed memory models.

SYLLABUS OF DSC-18

Unit 1 (6 Hours)

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 (15 Hours)

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, Recursive task spawning and pitfalls

Unit 3 (15 Hours)

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, MPI Collectives (MPI Collectives and MPI broadcast, MPI Gathering and scattering collectives, MPI reduction and Alltoall collectives, MPI collectives design), Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

Unit 4 (9 Hours)

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

Essential/recommended readings

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

No additional references mentioned

Suggested Practical List :

Practical exercises such as

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

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.

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 COURSE: 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	DSC03 Mathematics for Computing - I, DSC17 Machine Learning

Learning 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 OF DSE

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, Understanding and visualizing CNNs, 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, Bidirectional LSTM

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: 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	Any Programming Language at plus 2 level or above

Learning 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 this course, the student will be able to:

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

SYLLABUS OF DSE

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.

Unit 2 (6 hours)

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

Unit 3 (6 hours)

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

Unit 4 (6 hours)

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

Unit 5 (6 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 (6 hours)

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

Unit 7 (6 hours)

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

Unit 8 (5 Hours)

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

Essential/recommended readings

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:
 - a. <https://www.sans.org/cyberaces/>
 - b. <https://skillsforall.com/>
 - c. <https://www.hackingloops.com/ethical-hacking/>

Suggested Practical List

Practical exercises such as

Perform the following activities.

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

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

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
		Lecture	Tutorial	Practical/ Practice		
Cloud Computing	4	3	0	1	Pass in Class XII	NIL

Learning Objectives

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

Learning outcomes

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

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

SYLLABUS OF DSE

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.

Essential/recommended readings

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:

Practical exercises such as

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

COMMON POOL OF GENERIC ELECTIVES (GE) COURSES

GENERIC ELECTIVES (GE-6a): 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
		Lecture	Tutorial	Practical/ Practice		
GE6a Computer Networks	4	3	0	1	Pass in Class XII	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

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

On 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 OF GE-6a

Unit 1

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.

Unit 2

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

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, data link layer switching, point-to-point protocol.

Unit 4

Network layer: Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Introduction to Network Layer protocol- (ARP, IPV4, ICMP).

Unit 5

Introduction to Transport and Application Layer: Introduction to 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).

Essential/recommended readings

1. Tanenbaum, A.S. & Wetherall, 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 Publication, 2015.
2. (ii) Stallings, W., *Data and Computer Communications*, 10th edition, Pearson education India, 2017.

Suggested Practical List :

Practical exercises such as

Introduce students to CISCO Packet Tracer 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 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-6c): Artificial Intelligence

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		
GE6c: Artificial Intelligence	4	3	0	1	Pass in Class XII	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

The course objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

Learning outcomes

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

- identify problems that are amenable to solutions by specific AI methods.
- state the utility of different types of AI agents.
- apply different informed search techniques for solving real world problems.
- use knowledge representation techniques for AI systems.

SYLLABUS OF GE-6c

Unit 1 6 Hours

Introduction: Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

Unit 2 12 Hours

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

Unit 3 16 Hours

Knowledge Representation: Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

Unit 4 8 Hours

Understanding Natural Languages: Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented transition nets.

Unit 5 3 Hours

AI The Present and the Future: Symbolic AI, Data-driven AI and Machine Learning, Introduction to Machine Learning and Deep Learning based AI, some applications of symbolic and data driven AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

Essential/recommended readings

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4th edition, 2020..
2. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th edition, 2012.
3. Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
4. Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5th edition, Springer, 2003.

Additional references

1. Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010

Practical List :

Practical exercises such as

14. Write a program in Prolog to implement TowerOfHanoi(N) where N represents the number of disks.
15. Write a program to implement the Hill climbing search algorithm in Prolog.
16. Write a program to implement the Best first search algorithm in Prolog.
17. Write a program to implement A* search algorithm in Prolog.
18. Write a program to implement the min-max search algorithm in Prolog.
19. Write a program to solve the Water-Jug Problem in Prolog.
20. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Prolog.
21. Write a Prolog program to implement the family tree and demonstrate the family relationship.
22. Write a Prolog program to implement knowledge representation using frames with appropriate examples.
23. Write a Prolog program to implement conc(L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
24. Write a Prolog program to implement reverse(L, R) where List L is original and List R is reversed list.
25. Write a Prolog program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
26. Write a Prolog program to recognize context free grammar $a^n b^n$.

(Computer Science Courses for Undergraduate Programme of study with Computer Science discipline as one of the three Core Disciplines)

DISCIPLINE SPECIFIC CORE COURSE (DSC06): 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
		Lecture	Tutorial	Practical/ Practice		
DSC06: Computer Networks	4	3	0	1	Pass in Class XII	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

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 OF DSC06

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.

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, data link layer switching, point-to-point protocol.

Unit 4 (8 hours)

Network layer: Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Network Layer protocol- (ARP, IPV4, ICMP).

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).

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.

Suggested Practical List :

Practical exercises such as

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 router.
9. To perform remote desktop sharing within LAN connection.

Computer Science Major

DISCIPLINE SPECIFIC CORE COURSE (DSC06): 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
		Lecture	Tutorial	Practical/ Practice		
DSC06 Computer Networks	4	3	0	1	Pass in Class XII	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

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 OF DSC06

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.

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, data link layer switching, point-to-point protocol.

Unit 4 (8 hours)

Network layer: Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Network Layer protocol- (ARP, IPV4, ICMP).

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).

Essential/recommended readings

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

Additional References

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

Suggested Practical List :

Practical exercises such as

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 router.
9. To perform remote desktop sharing within LAN connection.

DISCIPLINE COURSE (A6): 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		
A6: Deep Learning	4	3	0	1	Pass in Class XII	A5 Machine Learning

Learning 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 OF A6

Unit 1 (15 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 trade-off, Regularization, output units: linear, softmax, hidden units: tanh, RELU

Unit 2 (15 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 (15 hours)

Convolution neural networks (CNNs): Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Understanding and visualizing CNNs, Using a pre-trained convnet

References

1. Heaton, J., Deep Learning and Neural Networks, 1st edition, Heaton Research Inc., 2015.
2. Francois Chollet, *Deep Learning with python*, 2nd edition, Meaning Publications Co, 2021.

Additional References

- (i) Ian Goodfellow, Yodhua Bengio and Aaron Courville, *Deep Learning*, MIT Press Book, 2016.
- (ii) Bunduma, N., *Fundamentals of Deep Learning*, 1st edition, O'reilly Books, 2017.

Suggested Practical List

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.

Computer Science Courses for Undergraduate Programme of study with Computer Science discipline as one of the two Core Disciplines: non-Major
(For e.g. courses for B.A. Programmes with Computer Science as Non-major discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC06): 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
		Lecture	Tutorial	Practical/ Practice		
DSC06 Computer Networks	4	3	0	1	Pass in Class XII	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

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 OF DSC06

Unit 1

Review of: 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.

Unit 2

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

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, data link layer switching, point-to-point protocol.

Unit 4

Network layer: Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Network Layer protocol- (ARP, IPV4, ICMP).

Unit 5

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).

Essential/recommended readings

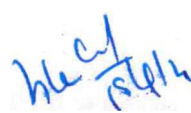
1. Tanenbaum, A.S. & Wethrall, D.J. *Computer Networks*, 5th edition. Pearson Education, 2012
2. Forouzan, B. A. *Data Communication and Networking*, 4th edition. McGraw-Hill Education, 2017

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.

Suggested Practical List :

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm.
6. Simulate and implement the Dijkstra algorithm for shortest-path routing.


REGISTRAR