A.C. 27.12.2024
Appendix-27

Based on Undergraduate Curriculum Framework

UNIVERSITY OF DELHI

UNDERGRADUATE PROGRAMMES OF STUDY

STRUCTURE, COURSES & SYLLABI OF SEMESTER -VII and SEMESTER-VIII

Semester VII

Sl.No.	Content
1	**BSc. (Hons.) Computer Sciences** DISCIPLINE SPECIFIC CORE (DSC) (1) Compiler Design
2	**DISCIPLINE SPECIFIC ELECTIVES (DSE)** (1) Digital Image Processing (2) Advanced Algorithms (3) Reinforcement Learning (4) Cyber Forensics
3	**GENERIC ELECTIVES (G.E.)** (1) Computer Networks (2) Internet Technologies: Mobile App Design and Development (3) Machine Learning (4) Cloud Computing (5) Ethical Hacking (6) Design and Analysis of Algorithms
4	**BSc. (Prog.) with Computer Sciences** **DISCIPLINE SPECIFIC CORE (DSC)** (1) Design and Analysis of Algorithms
5	**BA (Prog.). with Computer Sciences as Major/Non-major discipline** **DISCIPLINE SPECIFIC CORE (DSC)** (1) Design and Analysis of Algorithms

Department of Computer Science

COURSES OFFERED BY DEPARTMENT OF COMPUTER SCIENCE

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

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

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC19 Compiler Design	**4**	**3**	**0**	**1**	Pass in Class XII	One course in any Programming Language

Learning Objectives

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

Learning Outcomes

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

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

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

Syllabus

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

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

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

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

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

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

References

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

Additional References

(i) V Raghvan, *Principles of Compiler Design*, TMH, 2010.

(ii) Santanu Chattopadhayay, *Compiler Design*, PHI, 2005.

Suggested Practical List

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

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

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

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

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

COMMON POOL OF DISCIPLINE ELECTIVE COURSES (DSE) COURSES

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Data Analysis and Visualization

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7a: Digital Image Processing	**4**	**3**	**0**	**1**	Pass in Class XII	One course in any Programming Language

Course Objective

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

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

Course Learning Outcomes

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

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

Syllabus

Unit 1 Introduction: Digital Image Fundamentals, Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

Unit 2 Spatial Domain Filtering: Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters-Low pass filters, Order Statistics filters; Sharpening Spatial Filters- Laplacian filter.

Unit 3 Filtering in Frequency Domain: The Discrete Fourier Transformation (DFT), Frequency Domain Filtering:-Ideal and Butterworth Low pass and High pass filters

Unit 4 Image Degradation/Restoration Process: Noise models, Noise Restoration Filters

Unit 5 Image Compression: Fundamentals of Image Compression, Huffman Coding, Run Length Coding

Unit 6 Morphological Image Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

Unit 7 Image Segmentation: Point, Line and Edge Detection, Thresholding.

References

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

Additional References

(i) Castleman, K. R. *Digital Image Processing*, 1st edition, Pearson education,,2007.
(ii) Gonzalez, R. C., Woods, R. E., & Eddins, S. *Digital Image Processing using*

MATLAB, Pearson education Inc., 2004.
(iii) Jain, A. K. *Fundamentals of Digital Image Processing*, 1st edition, Prentice Hall of India,1988.

Suggested Practical List

The Practicals are to be conducted using Python.
Become familiar with basic Python libraries for Image Processing like OpenCV, Scikit-Image etc.

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

0. To write and execute image processing programs using point processing method:

   a. Obtain Negative image
   b. Obtain Flip image
   c. Thresholding
   d. Contrast stretching

0. To write and execute programs for image arithmetic operations:

   a. Addition of two images
   b. Subtract one image from other image
   c. Calculate mean value of image
   d. Different Brightness by changing mean value

0. To write and execute programs for image logical operations:
   a. AND operation between two images
   b. OR operation between two images
   c. Calculate intersection of two images
   d. Water Marking using X-OR operation
   e. NOT operation (Negative image)

0. To write and execute a program for histogram calculation and equalization:

   a. Using inbuilt function
   b. Without using inbuilt function

0. To write and execute a program performing the following geometric transformations on an image:
   a. Translation
   b. Scaling

c. Rotation
d. Shrinking
e. Zooming

0. To understand various image noise models and to write programs for:

a. Image restoration
b. Remove Salt and Pepper Noise
c. Minimize Gaussian noise
d. Median filter and Weiner filter

0. Write and execute programs to remove noise from images using spatial filtering.

a. Understand 1-D and 2-D convolution process
b. Use 3x3 Mask for low pass filter and high pass filter

0. Write and execute programs for image frequency domain filtering.

a. Apply FFT on given image
b. Perform low pass and high pass filtering in frequency domain
c. Apply IFFT to reconstruct image

0. Write and execute a program for edge detection using different edge detection mask.

0. Write and execute a program for image morphological operations erosion and dilation

DISCIPLINE SPECIFIC ELECTIVE COURSE: Advanced Algorithms

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7b: Advanced Algorithms	**4**	**3**	**1**	**0**	Pass in Class XII	Design and Analysis of Algorithms

Course Objective

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

Learning Outcomes

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

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

Syllabus

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

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

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

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

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

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

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

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

References

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

Additional References

(i) Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4th edition, Prentice Hall of India, 2022.

(ii) Williamson, D. P., Shmoys, D. B. *The Design of Approximation Algorithms*, 1st edition, Cambridge University Press, 2011.

Tutorials

Tutorials based on Theory

DISCIPLINE SPECIFIC ELECTIVE COURSE: Reinforcement Learning

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

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

Learning Objectives

The objectives of this course are:

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

Learning Outcomes

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

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

Syllabus

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

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

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

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

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

References

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

Additional References

(i) Phil Winder *Reinforcement Learning: Industrial Applications of Intelligent Agents*, O'Reilly Media, 2020.

(ii) Alexander Zai, Brandon Brown *Deep Reinforcement Learning in Action*, 1st edition, Manning Publications, 2020.

Suggested Practical List

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

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

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

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Cyber Forensics

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7d: Cyber Forensics	**4**	**3**	**1**	**0**	Pass in Class XII	**NIL**

Learning Objectives:

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

Course Learning Outcomes:

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

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

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

Unit 3 – Evidence Recovery: Introduction to Deleted File Recovery, Formatted Partition Recovery, Data Recovery Tools, Data Recovery Procedures and Ethics, Complete time line

analysis of computer files based on file creation, File modification and file access, Recover Internet Usage Data, Recover Swap Files/Temporary Files/Cache Files, Introduction to Encase Forensic Edition, Forensic Tool Kit (FTK), Use computer forensics software tools to cross validate findings in computer evidence.

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

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

References:

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

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

Additional References:

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

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

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

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

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

Suggested Practicals

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

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.

5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image

6. Using System internal tools for network tracking and process monitoring do the following:

a. Monitor live processes
b. Capture RAM
c. Capture TCP/UDP packets
d. Monitor Hard disk
e. Monitor Virtual Memory
f. Monitor Cache Memory

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-7a): 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		
GE7a: Computer Networks	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

The course objectives of this paper are to:

• Understand the concepts behind computer networks and data communication.

• Learn the different types of networks, network topologies and their characteristics.

• Learn the working of protocols used at various layers.

• Understand the utility of different networking devices.

Learning Outcomes

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

• differentiate between various types of computer networks and their topologies.

• understand the difference between the OSI and TCP/IP protocol suit.

• distinguish between different types of network devices and their functions.

• design/implement data link and network layer protocols in a simulated networking environment.

Syllabus

Unit 1 **(8 hours)**

Introduction:

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

Unit 2 **(9 hours)**

Physical Layer:

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

Unit 3 **(10 hours)**

Data Link and MAC Layer:

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

Unit 4 **(8 hours)**

Network layer:

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

Unit 5 **(10 hours)**

Transport and Application Layer:

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

Essential/recommended readings

1. Tanenbaum, A.S. & Wethrall, D.J.. Computer Networks, 5th edition, Pearson Education, 2012.

2. Forouzan, B. A.. Data Communication and Networking, 4th edition, McGraw-Hill Education, 2017.

Additional References

1. Comer, D. E.. Computer Networks and Internet, 6th edition, Pearson education, 2015.

2. Stallings, W., Data and Computer Communications, 10th edition, Pearson education India, 2017.

Practicals.

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

1. To Study basic network command and Network configuration commands.

2. To study and perform PC to PC communication.

3. To create Star topology using Hub and Switch.

4. To create Bus, Ring, Tree, Hybrid, Mesh topologies.

5. Perform an initial Switch configuration.

6. Perform an initial Router configuration.

7. To implement Client Server Network.

8. To implement connection between devices using a router.

9. To perform remote desktop sharing within LAN connection.

GENERIC ELECTIVES (GE-7b): Internet Technologies: Mobile 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		
GE7b: Internet Technologies: Mobile App Design and Development	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

- To comprehend Android and iOS mobile operating systems, their architecture, and development environments.
- To understand the basics of creating user interfaces in Android, including layouts, views, and UI components, and explore the iOS technology stack for developing intuitive iOS interfaces.
- To learn to use Android Data and Storage APIs, manage data with SQLite, share data between applications using Content Providers, and utilise various Android APIs for multimedia, networking, web, telephony, and location-based services. Understand the Core Data framework in iOS for data persistence and perform CRUD operations.
- To introduce Swift programming language, its data types, variables, control flow, operators, collections, functions, classes, structures, inheritance, and enumerations.
- To develop skills to handle user interactions in both Android and iOS, including controls, gesture recognisers, touch events, location-based services, and integration with Google Maps and iOS sensors.

Learning Outcomes

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

- Demonstrate proficiency in Android and iOS development, including installing and configuring development environments.
- Create effective user interfaces using Android UI components and iOS StoryBoard, applying event listeners, animations, and notifications.
- Manage application data efficiently using Android Data and Storage APIs, SQLite, Content Providers, multimedia, networking, and location-based services. Implement the Core Data framework in iOS for data persistence and CRUD operations.
- Code in Swift, utilising data types, control flow, collections, functions, classes, structures, inheritance, and closures.
- Enhance user interaction in mobile applications through gesture recognition, touch events, location services, and integration of Google Maps and iOS sensors.

Syllabus

Unit 1 Android Systems: Introduction to Mobile devices and applications, Open Handset Alliance (OHA), Overview of Android OS and architecture, installing Android Studio. Introduction to Android application components, Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions Activities and intents: understanding activity and its life cycle, Types of intents, intent filter, context, data sharing using intent

Unit 2 Android User Interface: Basic Android UI, layouts, view and view attributes, buttons, and controls. UI events and event listeners, animations, notifications, progress dialog, Action bar, toolbar, menus and pop-ups, Tab based UI, Fragment, Types of Fragment, Fragment Lifecycle, communication between fragment and activity

Unit 3 Android Storage and APIs: Android storage: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers
Android APIs: Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location-based services

Unit 4 iOS Technology Stack: Introduction to iOS technology stack: iOS architecture, Storyboard, features of Xcode, components of iOS SDK. Introduction to Swift: data types, variables, control flow and operators, Collections and functions in Swift, classes and structures, inheritance, closure, and. enumerations

Unit 5 User interactions: Controls, gesture organizers, touching views, Core Location and Mapkit, using Google Maps in iOS, and sensors in iOS. Data persistence: Core Data framework for storing persistent data, CRUD operations.

Note: Kotlin will be used for the implementation.

References

1. Meier Reto and Ian Lake, *Professional Android*, 4th edition, Wrox, 2018.
2. Craig Grummitt, *iOS Development with Swift,* Manning publications.
3. Rick Boyer, *Android 9 Development Cookbook*, Packt Publishing Limited, 2018.

Suggested Practical List

1. Set up Android Studio on their computers, create a new project, and run a simple "Hello World" application on an emulator.

2. Create an Android app with a LinearLayout containing a TextView, EditText, and Button. Set up a click listener for the Button to display a Toast message with the text entered in the EditText.

3. Create an Android app that allows users to add, view, and delete notes. Use SQLite to store the notes and display them in a ListView.

4. Set up Xcode on Macs/Mac emulators, create a new project, and run a simple "Hello World" application on an iOS simulator.

5. Create an iOS app with a UILabel, UITextField, and UIButton. Set up an action for the UIButton to update the UILabel with the text entered in the UITextField.

6. Create an iOS app that allows users to add, view, and delete contacts. Store the contacts in Core Data and display them in a UITableView.

7. Create an Android app with two activities. The first activity should have a Button that, when clicked, navigates to the second activity using explicit intent. The second activity should display a message.

8. Create an Android app that uses the Google Maps API to display a map with the user's current location marked.

9. Create an iOS app that recognises and responds to tap and swipe gestures on a UIView. For example, you could change the view's colour on a tap and move it on a swipe.

10. Create an Android app that fetches and displays a list of items from a public API (e.g., JSONPlaceholder) using Retrofit, Volley, or any other networking APIs.

GENERIC ELECTIVES (GE-7c): Machine 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		
GE 7c: Machine Learning	**4**	**3**	**0**	**1**	Pass in Class XII	Programming using Python/ Object Oriented Programming using Python

Course 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

Unit 1 **(5 Hours)**

Introduction:

Basic definitions and concepts, key elements, supervised and unsupervised learning, applications of ML.

Unit 2 **(8 Hours)**

Preprocessing:

Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis), class balancing, outlier detection and removal.

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 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 R2 score for regression models • Accuracy, TP, TN, FP, TN, 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

GENERIC ELECTIVES (GE-7d): Cloud Computing

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
GE 7d: Cloud Computing	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective:

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

Course Learning Outcomes:

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

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

Syllabus:

Unit 1: Overview of Computing Paradigm
Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing
Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

Unit 3: Cloud Computing Architecture

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

Unit 4: Case Studies

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

Unit 5: Cloud Computing Management

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

Unit 6: Cloud Computing Security

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

References:

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

Additional References:

1. *Cloud Computing Bible,* Barrie Sosinsky*, Wiley-India,* 2010

0. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski*, Wile,* 2011

0. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam*, Springer,* 2012

0. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing,* Ronald L. Krutz, Russell Dean Vines*, Wiley-India,* 2010

Suggested Practical List:

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

GENERIC ELECTIVES (GE-7e): Ethical Hacking

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
GE 7e: Ethical Hacking	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

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

Learning outcomes

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

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

Syllabus

Unit 1 **(4 Hours)**

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

Unit 2 **(6 hours)**

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

Unit 3 **(8 hours)**

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

Unit 4 **(8 hours)**

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

Unit 5 **(8 hours)**

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

Unit 6 **(8 hours)**

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

Unit 7 **(8 hours)**

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

Unit 8 **(4 Hours)**

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

References

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

Additional References

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

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

https://skillsforall.com/

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

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

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

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

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

GENERIC ELECTIVES (GE-7f): Design and Analysis of Algorithms

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
GE 7f: Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	Data Structures

Course Objectives

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

Learning Outcomes

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

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(4 hours)**

Hash Tables Hash Functions, Collision resolution schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.

2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using the Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.
7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.

11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

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

(For e.g. courses for B.Sc. Programme with Computer Science as discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC07): Design and Analysis of Algorithms

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC07: Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	Data Structures

Course Objective

The course is designed to develop an understanding of different algorithm design techniques and use them for problem-solving. The course shall also enable the students to verify the correctness of algorithms and analyze their time complexity.

Learning Outcomes

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

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(4 hours)**

Hash Tables Hash Functions, Collision resolution schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.

2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using the Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.
7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.

10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

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/Non-major discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC07): Design and Analysis of Algorithms

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC07: Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

Learning Outcomes

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

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(4 hours)**

Hash Tables Hash Functions, Collision resolution schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.

2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.

7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim’s algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

Semester VIII

Sl.No.	Content
1	**BSc. (Hons.) Computer Sciences** **DISCIPLINE SPECIFIC CORE (DSC)** (1) Information Security
2	**DISCIPLINE SPECIFIC ELECTIVES (DSE)** (1) Information and Image Retrieval (2) Natural Language Processing (3) Blockchain and Its Applications (4) Distributed Algorithms (5) Cloud Computing
3	**GENERIC ELECTIVES (G.E.)** (1) Information Security (2) Digital Marketing and Social Media Analytics (3) Introduction to Parallel programming (4) Cyber Forensics
4	**BSc. (Prog.) with Computer Sciences** **DISCIPLINE SPECIFIC CORE (DSC)** (1) Information Security
5	**BA (Prog.). with Computer Sciences as Major/Non-major discipline** **DISCIPLINE SPECIFIC CORE (DSC)** (1) Design and Analysis of Algorithms

Department of Computer Science

COURSES OFFERED BY DEPARTMENT OF COMPUTER SCIENCE

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

DISCIPLINE SPECIFIC CORE COURSE -20 (DSC-20) : Information Security

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC20 Information Security	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

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

Learning Outcomes

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

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

Syllabus

Unit 1 **(3 hours)**

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

Unit 2 **(6 hours)**

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

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

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

Unit 5 **(8 hours)**

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

Unit 6 **(6 hours)**

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

Unit 7 **(7 hours)**

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

References

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

Additional References

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

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

Suggested Practical List

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

COMMON POOL OF DISCIPLINE ELECTIVE COURSES (DSE) COURSES

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Information and Image

Credit distribution, Eligibility and Pre-requisites of the Course

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

Course Objective

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

Course Learning Outcomes

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

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

Syllabus

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

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

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

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

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

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

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

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

References

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

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

Suggested Practical List

To be implemented in Python

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

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

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

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	60	8
	Positive	22	10

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Natural Language Processing

Credit distribution, Eligibility and Pre-requisites of the Course

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

Course Objective

The objectives of this course are:

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

Course Learning Outcomes

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

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

Syllabus

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

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

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

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

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

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

References

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

Additional Reference

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

Suggested Practical List

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

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

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

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

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Blockchain and its Applications

Credit distribution, Eligibility and Pre-requisites of the Course

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

Course Objective

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

Course Learning Outcomes

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

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

Syllabus

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

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

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

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

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

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

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

References

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

Additional Reference

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

Suggested Practical List

Use any programming language to implement the following:

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

DISCIPLINE SPECIFIC ELECTIVE COURSE: Distributed Algorithms

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE8d: Distributed Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	Data Structures, Design and Analysis of Algorithms

Course Objective

The course introduces the students to distributed algorithms in synchronous and asynchronous network models. The course would give the students hands-on practice to write programs for distributed algorithms using Remote Procedure Call (RPC) or Message Passing Interface (MPI)

Course Learning Outcomes

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

1. Describe Network Models for distributed Algorithms
2. Develop elementary synchronous distributed algorithms
3. Develop elementary asynchronous distributed algorithms

Syllabus

Unit 1 System Model/Network Models: Synchronous Network Model, Asynchronous System Model, Asynchronous Network Model

Unit 2 Synchronous Network Algorithms: Distributed problems in Synchronous Networks such as Leader Election in a Synchronous Ring. Algorithms in General Synchronous Networks (for example Leader Election in a General Network, Breadth-First Search, Maximal Independent Set etc). Problems of reaching consensus in a distributed network namely, distributed consensus with link failures coordinated Attack Problem (Deterministic Version and Randomized Version) and distributed consensus with link failures(Stopping failures, Introduction to Byzantine Failures). More Consensus Problems such as the k-Agreement etc.

Unit 3 Asynchronous Network Algorithms: Basic Asynchronous Network Algorithms such as Leader Election in a Ring, Leader Election in an Arbitrary Network etc. Logical Time Asynchronous Networks, Adding Logical Time to Asynchronous Algorithms, Applications such as Banking System etc. Basics of Network Resource Allocation (mutual Exclusion, resource allocation etc) and Basics of Asynchronous Networks with Process Failures such as k-Agreement etc.

References

1. Lynch, N. *Distributed Algorithms*, Morgan Kaufmann Publishers, Inc.,1996.
2. M. van Steen, A. S. Tanenbaum, *Distributed Systems, CreateSpace Independent* Publishing Platform, 2017.

Additional References

(i) Garg, V. *Elements of Distributed Computing*, Wiley, 2014.

Suggested Practical List

1. Implement Leader Election in a Synchronous Ring.
2. Implement Leader Election in a General Network (Synchronous Network)
3. Implement Breadth-First Search (Synchronous Network)
4. Implement Maximal Independent Set (Synchronous Network)
5. Implement Leader Election in an Asynchronous Ring.
6. Implement Asynchronous Banking System

Optional

0. Implement distributed consensus with link failure (Synchronous Network)
0. Implement distributed consensus with Process failure (Synchronous Network)

DISCIPLINE SPECIFIC ELECTIVE COURSE: Cloud Computing

Credit distribution, Eligibility and Pre-requisites of the Course

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

Course Objective:

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

Course Learning Outcomes:

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

1. Knowledge of the fundamental concepts and principles of cloud computing, including virtualization, scalability, reliability, and security.
2. Ability to design, develop, and deploy cloud-based applications using popular cloud platforms and services.
3. Familiarity with cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. Visualize the economic, legal, and ethical implications of cloud computing, including issues related to data privacy, ownership, and security.

5. Ability to evaluate and select cloud-based solutions based on their technical, economic, and business requirements.
6. Understanding of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: Overview of Computing Paradigm
Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing

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

Unit 3: Cloud Computing Architecture

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

Unit 4: Case Studies

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

Unit 5: Cloud Computing Management

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

Unit 6: Cloud Computing Security

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

References:

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

Additional References:

1. *Cloud Computing Bible,* Barrie Sosinsky, *Wiley-India,* 2010

0. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, *Wile,* 2011

0. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam, *Springer,* 2012

0. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing,* Ronald L. Krutz, Russell Dean Vines, *Wiley-India,* 2010

Suggested Practical List:

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

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-8a): Information Security

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		
GE8a: Information Security	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

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

Learning Outcomes

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

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

Syllabus

Unit 1 **(3 hours)**

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

Unit 2 **(6 hours)**

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

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

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

Unit 5 **(8 hours)**

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

Unit 6 **(6 hours)**

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

Unit 7 **(7 hours)**

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

References

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

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications,* 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

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

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

Credit distribution, Eligibility and Pre-requisites of the Course

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

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

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

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

Syllabus

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

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

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

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

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

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

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

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

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

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

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

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

References:

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

GENERIC ELECTIVES (GE-8c): 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		
GE8c: Introduction to Parallel Programming	**4**	**3**	**0**	**1**	Pass in Class XII	Computer System Architecture/A course in C++at class XII/Data Structures, Operating Systems

Course Objective

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

Course Learning Outcomes

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

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

Syllabus

Unit 1

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

Unit 2

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

Unit 3

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

Unit 4

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

References

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

Additional references

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

Suggested Practical List

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

GENERIC ELECTIVES (GE-8d): Cyber Forensics

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		
GE8d: Cyber Forensics	**4**	**3**	**0**	**1**	Pass in Class XII	

Course Objective:

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

Course Learning Outcomes:

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

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

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

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

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

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

References:

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

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

Additional References:

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

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

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

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

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

Suggested Practicals

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

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

(Computer Science Courses for Undergraduate Programme of study with Computer Science discipline as one of the three Core Disciplines)
(For e.g. courses for B.Sc. Programme with Computer Science as discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC08): Information Security

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC08: Information Security	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

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

Learning Outcomes

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

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

Syllabus

Unit 1 **(3 hours)**

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

Unit 2 **(6 hours)**

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

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

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

Unit 5 **(8 hours)**

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

Unit 6 **(6 hours)**

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

Unit 7 **(7 hours)**

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

References

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

Additional References

9. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
10. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications,* 2nd edition, Pearson Education, 2004.
11. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
12. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
13. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
14. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
15. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.

16. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

12. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
13. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
14. Use nmap/zenmap to analyze a remote machine.
15. Use Burp proxy to capture and modify the message.
16. Implement caesar cipher substitution operation.
17. Implement monoalphabetic and polyalphabetic cipher substitution operation.
18. Implement playfair cipher substitution operation.
19. Implement hill cipher substitution operation.
20. Implement rail fence cipher transposition operation.
21. Implement row transposition cipher transposition operation.
22. Implement product cipher transposition operation.

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/Non-major discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC08): Information Security

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC08: Information Security	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objective

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures.

This course also touches upon the implications of security in cloud and Internet of Things (IoT).

Learning Outcomes

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

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

Syllabus

Unit 1 **(3 hours)**

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

Unit 2 **(6 hours)**

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

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

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

Unit 5 **(8 hours)**

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

Unit 6 **(6 hours)**

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

Unit 7 **(7 hours)**

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

References

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

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications,* 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

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