Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: CIT110
Course Title: Cloud of Things

Programme: B. Tech	**L:** 4 **T:** 0 **P:** 4	**Credits:** 6
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 60(L)+60(P)=120 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 15%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Core Course		

Prerequisites (if any): CIT102
Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Elucidate the fundamental concepts, growth trends, architecture, and key characteristics of the Internet of Things (IoT).
2	Analyze IoT communication protocols, transport technologies, and addressing and routing mechanisms in IoT networks.
3	Compare and evaluate IoT hardware platforms and develop basic IoT system architectures using microcontrollers and single-board computers.
4	Apply software tools such as Python programming and wireless configuration techniques for IoT device implementation.
5	Evaluate cloud computing concepts, virtualization, and IoT analytics platforms (e.g., ThingSpeak) in real-world applications.
6	Analyze privacy, security, and vulnerability issues in IoT systems and propose appropriate mitigation strategies.

Contents

Part-A

Unit-1 Fundamentals of Internet of Things (IoT) 8(L) hrs

Definition and scope of the Internet of Things, evolution and growth trends of IoT – a statistical perspective, application domains of IoT, key characteristics of IoT systems, concept of "Things" in IoT, IoT architecture and stack, enabling technologies, challenges in IoT, and IoT deployment levels.

Unit–2 IoT Communication and Networking Protocols 10(L) hrs

Overview of IoT communication protocols, messaging protocols such as MQTT, CoAP, and AMQP, transport and connectivity technologies including Li-Fi, Bluetooth Low Energy (BLE), Zigbee, and Wi-Fi, Internet Protocol concepts with emphasis on IPv6 and 6LoWPAN, addressing and routing mechanisms in IoT networks, protocol stack comparison, and protocol selection criteria for IoT applications.

Unit–3 IoT Hardware Platforms, Sensors and Microcontroller Interfacing 10(L)hrs

Fundamentals of IoT hardware platforms, overview of sensors and sensor interfacing techniques, classification of sensors, web-based sensor control, overview of microcontrollers, Arduino, Raspberry Pi and Node MCU platforms, Architecture of Raspberry Pi, operating system installation

and configuration, remote access techniques, Python fundamentals for IoT, GPIO interfacing, and wireless connectivity configuration for IoT devices

Part-B

Unit-4 Cloud & Virtualization **10(L) hrs**

Introduction and evolution of cloud computing paradigms, Evolution of computing (Cluster vs. Grid vs. Distributed) Virtualization concepts, Hypervisors, Virtual Machine (VM), Containerization overview, Cloud Computing definition and characteristics, Service Models (IaaS, PaaS, SaaS), Deployment Models (Public, Private, Hybrid, Community). Transitioning from Cloud to Edge and Fog computing.

Unit 5 Cloud Platforms and Services **12(L)hrs**

Cloud-based IoT analytics platforms (e.g., ThingSpeak) including channel and field creation, data upload, real-time visualization dashboards, and cloud-based analytics; IoT cloud architecture with IoT hubs, cloud gateways, and publish/subscribe (Pub/Sub) model; device management with provisioning, device twins/shadows, and OTA updates; cloud data processing through message routing, telemetry handling, serverless computing, and IoT storage models; API-based data retrieval, system integration, and trigger alerts with automation.

Unit–6 Privacy, Security and Vulnerability Management **10(L)hrs**

Security requirements, threat analysis, use and misuse cases, security tomography, layered attacker models, identity management and establishment, access control mechanisms, message integrity, non-repudiation, availability, security protocols, and management aspects of cyber security.

Laboratory Work

Experiment No.	Experiment Title
1	Familiarization with Arduino/ Raspberry Pi and perform necessary software installation
2	To write and execute a basic Arduino sketch using setup () and loop () functions.
3	To perform serial communication using Arduino and display output on the serial monitor.
4	To interface an LED with Arduino and implement LED blinking patterns.
5	To interface a DHT sensor with NodeMCU/Arduino and display temperature and humidity data.
6	To interface an MQ-2 gas sensor with NodeMCU/Arduino and detect the presence of gases.
7	To interface an ultrasonic sensor with NodeMCU/Arduino for distance measurement.
8	To configure and verify wireless (Wi-Fi) connectivity on Raspberry Pi.
9	To interface a servo motor with NodeMCU/Arduino and control its angle using PWM.
10	To configure remote access to Raspberry Pi using SSH (PuTTY/Terminal).
11	To Configure an IoT edge device to securely publish real-time sensor telemetry to a local or cloud-hosted using MQTT broker
12	To Provision an edge device on a cloud-based IoT analytics platform to transmit telemetry data for real-time dashboard visualization,

13	To Develop a serverless logic flow using a visual programming tool to capture incoming MQTT telemetry
14	To Implement Over-The-Air (OTA) firmware update capabilities via a cloud management platform to remotely upgrade the microcontroller code
15	To Configure message routing rules within an IoT cloud platform to automatically forward incoming sensor telemetry into cloud database

Mini Project: Students are required to prepare a project based on thecourse contents in a group of five students. The students should use Arduino/ Raspberry Pi, LEDs, IR, LDR, DHT11 sensor, PIR Sensors, PI Camera, OLED, Micro USB Cables, Jumper Wires, Mouse and Keyboard, HDMI to VGA cables, Monitor, Python in the design and implementation of the project. The group of students must submit a project report of 8 to 10 pages (approx.) and the team will have to demonstrate the project as well as have to give a presentation of the same.

Suggested projects:

- Smart Home Automation using NodeMCU and Relay Module
- Environmental Monitoring System using DHT and MQ Sensors
- Smart Health Monitoring using Pulse Sensor
- Smart Agriculture Monitoring with IoT-based data logging
- Intrusion Detection System using Ultrasonic Sensor and NodeMCU

Text Books

1. Wiley Editorial Team, “Internet of Things”, 3rd Edition, Wiley India, 2023.
2. Raj Kamal, “Internet of Things – Architecture and Design Principles”, 2nd Edition, McGraw-Hill Education, 2022.
3. S. Misra, A. Mukherjee, and A. Roy, “Introduction to IoT”, Cambridge University Press, 2020
4. Davis Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, and Jerome Henry, “IoT Fundamentals – Networking Technologies, Protocols, and Use Cases for the Internet of Things”, 5th Impression, CISCO Press, 2020.
5. Mayur Ramgir, “Internet of Things – Architecture, Implementation and Security”, 1st Impression, Pearson India, 2020.

Reference Books

1. Perspectives on Artificial Intelligence and Internet of Things for Sustainable Environment — Gwanggil Jeon, Bhisham Sharma, Nitish Katal (Elsevier, 2025)
2. S. Misra, C. Roy, and A. Mukherjee, “Introduction to Industrial Internet of Things and Industry 4.0”, CRC Press, 2020.
3. Rajesh Singh, Anita Gehlot, Lovi Raj Gupta, Bhupendra Singh, Mahendra Swain, “Internet of Things with Raspberry Pi and Arduino”, 1st Impression, CRC Press, 2019.
4. Ashwin Pajankar, “Internet of Things with Arduino and Bolt”, 2nd Impression, BPB Publications, 2018.
5. Arshdeep Bahga, Vijay Madisetti, “Internet of Things – A Hands-On Approach”, 3rd Impression, Universities Press, 2017.
6. Gaston C. Hillar, “Internet of Things with Python”, 2nd Impression, PACKT Open Source Press, 2016.
7. Pethuru Raj and Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases”, CRC Press, 2017.

Online Learning Materials

1. https://www.amazon.in/Introduction-IoT-Sudip-Misra/dp/1108959741 Accessed on: Dec 23, 2025
2. https://www.wileyindia.com/internet-of-things-3ed.html Accessed on: Dec 23, 2025
3. https://onlinecourses.nptel.ac.in/noc22_cs53/preview Accessed on: Dec 23, 2025
4. https://nptel.ac.in/courses/106/105/106105166/ Accessed on: May 21, 2025
5. https://www.coursera.org/specializations/internet-of-things Accessed on: May 21, 2025
6. https://www.youtube.com/watch?v=mZ24zRRzlsI&list=PLPN-43XehstOS_3mv9LgFWnVXQE-7PKbF Accessed on: May 21, 2025
7. https://www.udemy.com/topic/internet-of-things/ Accessed on: May 21, 2025
8. https://www.edx.org/micromasters/curtinx-internet-of-things-iot Accessed on: May 21, 2025
9. https://www.udacity.com/course/intel-edge-ai-for-iot-developers-nanodegree--nd131 Accessed on: May 21, 2025
10. https://www.pluralsight.com/courses/security-internet-of-things-big-picture Accessed on: May 21, 2025

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Cloud Computing	By Prof. Soumya Kanti Ghosh |	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc26_cs55/preview/
2	Cloud Computing and Distributed Systems	By Prof. Rajiv Misra	IIT Patna	https://onlinecourses.nptel.ac.in/noc26_cs29/preview/

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: CIT111

Course Title: Computer Organization and Architecture

Programme: B. Tech	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+ 30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Core Course		

Prerequisites (if any): ESIT101, CIT101, CIT104

Additional Material Allowed in ESE: Scientific Calculator

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Analyze and design instruction sets, addressing modes, and data representation in modern computers.
2	Understand the organization of CPU, ALU, data path, and control unit, and evaluate performance metrics.
3	Design and analyze pipelined processors, including hazard detection and mitigation strategies.
4	Explain memory hierarchy, caching techniques, and I/O interfacing mechanisms.
5	Apply knowledge of computer architecture to evaluate system performance and suggest improvements.
6	Simulate and test basic computer components using software/hardware tools.

Contents

Part-A

Unit-1 Introduction to Computer Organization **5(L) hrs**

Basic structure of computers: Von Neumann and Harvard architectures, Instruction codes, Instruction Cycle, Interrupt Cycle, Performance metrics: clock rate, Cycles pe r Instruction (CPI), Million Instructions Per Second (MIPS), Mega Floating-Point Operations (MFLOPS), Amdahl's law. Machine instructions and instruction formats. Addressing modes: immediate, direct, indirect, register, indexed, relative.

Unit-2 Arithmetic and Logic Unit **6(L) hrs**

Design of ALU: arithmetic circuits, logic circuits, and fast adders (carry look-ahead), Programming Arithmetic and Logic Operations, Multiplication and division algorithms: Booth's algorithm, restoring and non-restoring division. Floating-point representation (IEEE 754 standard) and arithmetic.

Unit-3 CPU Design and Control Unit **6(L) hrs**

Datapath design: single-cycle and multi-cycle implementations. Control unit: hardwired and microprogrammed control. Introduction to RISC and CISC architectures.

Unit-4 Pipelining **6(L)hrs**

Basic pipeline concept and performance improvement, Types of pipeline: Arithmetic pipeline, Instruction pipeline, Pipeline hazards: structural, data, and control hazards. Hazard detection and resolution: forwarding, stalling, branch prediction. Case studies: MIPS pipeline.

Part-B

Unit-5 Memory System Organization **5(L)hrs**

Memory hierarchy: registers, cache, main memory, secondary storage. Cache memory: mapping techniques (direct, associative, set-associative), replacement policies.

Unit-6 Input/Output Organization **6(L)hrs**

I/O interfaces: programmed I/O, interrupt-driven I/O, Direct Memory Access. I/O processors and channels. Buses: types, arbitration, and timing. External storage devices: magnetic disks, SSDs, optical storage.

Unit-7 Advanced Architecture/Parallel Processing **6(L) hrs**

Introduction to parallel processing: Single Instruction Multiple Data (SIMD), Multiple Instruction Multiple Data (MIMD). Multi-core processors and multiprocessor systems. Introduction to Graphics Processing Unit, GPU architecture (overview), Applications of GPU computing.

Unit-8 Performance Evaluation and Emerging Trends **5(L)hrs**

Benchmarking, Types of benchmarking, performance analysis, Performance metrics, Introduction to quantum computing architecture (qualitative), Heterogeneous computing architecture, Trends in computer architecture: AI accelerators, edge computing.

Laboratory Work

Tools Recommended: Simulators: MARS (MIPS Assembler and Runtime Simulator), CPUSim, Logisim, QEMU.
Hardware (optional): FPGA boards (e.g., Basys 3) for RTL design.
Software: C/C++ for performance simulation, Python for analysis scripts**.**

Experiment No.	Experiment Title	Tool/Platform
1	Simulate arithmetic operations using ALU design in Logisim.	Logisim
2	Implement and analyze different addressing modes using MARS.	MARS Simulator
3	Design and simulate a single-cycle CPU datapath.	Logisim / CPUSim
4	Implement and optimize a pipelined processor with hazard handling.	MARS / Custom Simulator
5	Analyze cache performance using C/C++ simulation (varied block size, mapping).	C/C++
6	Simulate I/O operations and interrupt handling using an emulator.	QEMU / Custom Simulator
7	Implement interrupt-driven I/O simulation.	C / Assembly
8	Benchmark analysis of different CPU architectures using performance counters.	Linux perf / Python

Mini Project: Students will work in groups to design and simulate a basic CPU with pipeline support, preparing a report and presentation.

Text Books

1) William Stallings, Computer Organization and Architecture: Designing for Performance, 11th Edition, Pearson, 2025.

2) David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 6th Edition, Morgan Kaufmann, 2021.
3) M. Morris Mano, Computer System Architecture, 3rd Edition, Pearson, 2017.

Reference Books

1) John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann, 2017.
2) Andrew S. Tanenbaum, Structured Computer Organization, 6th Edition, Pearson, 2012.
3) Randal E. Bryant and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, 3rd Edition, Pearson, 2015.

Online Learning Materials

1. https://www.geeksforgeeks.org/computer-organization-architecture/computer-organization-and-architecture-tutorials/
   Accessed on January 05, 2026
2. https://vardhaman.org/wp-content/uploads/2021/03/CO.pdf
   Accessed on January 05, 2026
3. https://www.mbit.edu.in/wp-content/uploads/2020/05/computer-systems-Architecture.pdf
   Accessed on January 05, 2026
4. https://medium.com/@raufpokemon00/mastering-computer-organization-architecture- and-assembly-language-the-ultimate-beginners-guide-d50e944bc91
   Accessed on January 05, 2026
5. https://www.youtube.com/playlist?list=PLxCzCOWd7aiHMonh3G6QNKq53C6oNXGrX
   Accessed on January 05, 2026
6. https://www.youtube.com/watch?v=Ol8D69VKX2k
   Accessed on January 05, 2026

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Computer Architecture	Prof. Smruti Ranjan Sarangi	IIT Delhi	https://nptel.ac.in/courses/106102157/
2	Computer Architecture and Organization	Prof. Indranil Sengupta, Prof. Kamalika Datta	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc22_cs88/preview/
3	Computer Organization and Architecture	Prof. Kamakoti	IIT Madras	https://nptel.ac.in/courses/106106166
4	Computer Organization and Architecture: A	Prof. Santhosh biswasProf. Jatindra kumar	IIT Guwahati	https://nptel.ac.in/courses/106103180

	Pedagogical Aspect	deka Prof. Arnab Sarkar		
5	Digital Computer Organizatio n	Prof. P.K. Biswas	IIT Kharagpur	https://nptel.ac.in/courses/117105078/
6	Advanced Computer Architecture	Prof. John Jose	IIT Guwahati	https://nptel.ac.in/courses/106103206/
7	Computer Architecture	Prof. David Wentzlaff	Princeton University, New Jersey, United States	https://www.coursera.org/learn/comparch/
8	Computatio n Structures	Prof. Chris Terman	MIT, USA	https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/

Virtual Lab Links (Simulation-Based)
MARS MIPS Simulator: https://dpetersanderson.github.io
CPUSim: https://www.cs.colby.edu/djskrien/CPUSim
Logisim: http://www.cburch.com/logisim

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: CIT112

Course Title: Design and Analysis of Algorithms

Programme: B.Tech.	**L:** 3 **T:** 1 **P:** 2	**Credits:** 5
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L) + 15(L) + 30(P) = 90 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Core Course		

Prerequisites (if any): CIT101

Additional Material Allowed in ESE: Nil

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Analyze and evaluate algorithmic time and space complexities using asymptotic notations to justify the selection of efficient algorithms for large-scale problems.
2	Design and optimize algorithms using divide and conquer and greedy techniques to solve real-world computational problems.
3	Formulate and evaluate solutions using dynamic programming, backtracking, and branch and bound techniques for solving optimization and constraint satisfaction problems.
4	Analyze and implement advanced graph and network flow algorithms to solve shortest path, minimum spanning tree, and maximum flow problems in practical systems.
5	Evaluate computational intractability by classifying problems into P, NP, NP-hard and NP-complete and examine the significance of approximation algorithms for solving real-world NP-complete problems.
6	Compare and justify the use of string matching, randomized, and parallel algorithms based on efficiency, scalability, and reliability requirements.

Contents

Part-A

Unit-1 Introduction **4(L) hrs**

Algorithm and its importance, Mathematical foundations– Growth functions, Complexity analysis of algorithms, Asymptotic notations: Big Oh, Omega and Theta; Calculation and plotting of graph of upper bound, lower bound and tight bound of a function.

Unit-2 Divide and Conquer **5(L) hrs**

Basic algorithmic-method of Divide and Conquer, application of divide and conquer design technique on Binary Search, finding maximum and minimum values in a given array using divide and conquer and comparing its performance with the straightforward method, calculation of time complexities of the mergesort and quicksort algorithms involving divide and conquer technique.

Unit-3 Greedy Algorithms **5(L) hrs**

General Greedy method, using greedy algorithm to solve fractional Knapsack problem, solving minimum-cost spanning tree problem using Prim's and Kruskal's algorithm, solving Travelling

salesperson problem using greedy algorithmic design approach, Huffman Coding, Dijkstra's algorithm for solving single source shortest path problem, limitations of Dijkstra's algorithm.

Unit-4 Dynamic Programming **6(L) hrs**

Introduction to dynamic programming, principle of optimality, overcoming the limitations of Dijkstra's algorithm by dynamic programming approach, Bellman and Ford algorithm for solving single-source shortest path problem, solving all-pairs shortest path problem using Floyd-Warshall algorithm, applying dynamic programming for solving multistage graph problem using forward and backward approaches, solving 0/1 Knapsack problem by dynamic programming

Unit-5 Branch and Bound **3(L) hrs**

General method of Branch and bound, state space tree, live nodes, dead nodes, Expanding nodes (E-nodes), bounding functions, First in First Out (FIFO) branch and bound, Last In First Out (LIFO) branch and bound, Least-Cost (LC) Branch and Bound, solving Travelling Salesperson Problem using Branch and Bound

Part-B

Unit-6: Backtracking **4(L) hrs**

General backtracking algorithm, Application of backtracking to solve: n-Queens' problem, Sum of subsets, Graph coloring, Hamiltonian cycles and 0/1 Knapsack problem.

Unit-7 Network Flows **4(L) hrs**

The maximum s-t flow problem in capacitated networks, Ford Fulkerson Algorithm for maximum flow problem, Max-flow min-cut theorem.

Unit-8 Parallel and Randomized Algorithms **5(L) hrs**

Data and control parallelism, embedding of problem graphs into processor graphs, parallel algorithms for matrix multiplication. ssGeneral method of Randomized algorithms, Randomized Quick Sort Algorithm, Karger's Minimum Cut Algorithm

Unit-9 NP-completeness and Approximation Algorithms **6(L) hrs**

Decision problems, optimization problems, polynomial time algorithms, non-polynomial time algorithms, deterministic algorithms, non-deterministic algorithms, Conjunctive Normal Form (CNF), Disjunctive Normal Form (DNF), satisfiability problem, Problem classes P, NP, NP-hard and NP-complete, Cook's Theorem, NP-complete problems: Clique Problem, Vertex Cover, Need of approximation algorithms, Absolute approximation, f(n)-approximation and Epsilon(ε)-approximation

Unit-10 String Matching Algorithms **3(L) hrs**

Brute Force algorithm, Rabin-Karp algorithm, Knuth Morris Pratt (KMP) algorithm, Boyer-Moore algorithm using looking-glass heuristic, bad-character rule and good-suffix rule, Compare the performance of its variations for average and worst-cases.

Laboratory Work

Exp. No.	Experiment Title
1.	**Algorithm Selection for Scalable E-Commerce Search:** Investigate the problem of searching products in a rapidly scaling e-commerce platform: a) Analyse how algorithmic time complexity impacts system performance as data volume increases from thousands to millions of records. b) Choose appropriate search algorithms among linear and binary search techniques for different stages of system growth and real-time operational requirements.

2.	**Divide and Conquer vs Brute Force in Smart City Analytics:** Examine the challenge of identifying extreme sensor values collected from a smart city infrastructure: a) Determine the maximum and minimum values by using a straightforward brute-force method as well as divide-and-conquer approach. b) Evaluate the suitability for large-scale, real-time data analytics systems.
3.	**Sorting Strategy Optimization in Banking Systems:** Explore the problem of sorting transaction records in a banking environment under varying data distributions: a) Select algorithms among merge sort and quick sort based on reliability and performance constraints in mission-critical financial systems.
4.	**Greedy Decision-Making in Cloud Resource Allocation:** Model cloud resource allocation as a fractional knapsack problem where limited computational resources must be assigned to maximize revenue: a) Analyse how greedy strategies yield optimal solutions in this context. b) Justify the constraints under which greedy decision-making remains effective for real-world cloud service providers.
5.	**Network Cost Optimization for Campus Infrastructure:** Design a minimum-cost communication network for a university campus: a) Apply Prim's and Kruskal's algorithms. b) Compare their performance for sparse and dense graph conditions so that the most efficient algorithm could be chosen based on realistic deployment constraints.
6.	**Data Compression using Huffman Coding:** Examine the use of Huffman coding for data compression in multimedia streaming platforms: a) Compare Huffman coding with the fixed-length encoding schemes. b) Evaluate the trade-off between compression ratio and computational overhead in order to justify real-world applicability.
7.	**Shortest Path Reliability in Navigation Systems:** Study the route optimization in GPS-based navigation systems: a) Use Dijkstra's algorithm and study its limitations in the presence of negative edge weights. b) Examine the consequences of incorrect routing decisions c) Justify the need for alternative shortest-path algorithms in dynamic traffic environments.
8.	**Financial Network Optimization with Bellman–Ford Algorithm:** Model a financial transaction network containing penalties and rebates using weighted graphs with possible negative edges: a) Analyse how the Bellman–Ford algorithm detects negative cycles. b) Compare its correctness and performance with Dijkstra's algorithm.
9.	**Greedy vs Dynamic Programming in Logistics Optimization:** Study the cargo loading problem faced by logistics companies under strict weight constraints and to consider it as the 0/1 knapsack problem:

	a) Apply Greedy and dynamic programming approaches for this logistics problem. b) Recommend the suitable approach for different operational priorities**.**
10.	**Branch and Bound Strategies for Route Optimization:** Investigate the application of branch and bound technique to the Travelling Salesperson Problem for route optimization: a) Analyze the growth of the state-space tree b) Evaluate the FIFO, LIFO, and Least-Cost strategies.
11.	**Constraint Satisfaction in Academic Scheduling:** Examine the problem of generating a university examination timetable subjected to multiple constraints like time slot, room capacity, availability of proctor etc.: a) Apply backtracking technique for the constraint satisfaction of such an academic scheduling.
12.	**Bandwidth Optimization Using Network Flow Models:** Optimize the data flow in an Internet Service Provider network during peak demand periods: a) Model using the Ford–Fulkerson algorithm. b) Apply max-flow min-cut theorem to identify the network's maximum data capacity and bottlenecks during peak demand periods.
13.	**Reliability Analysis Using Randomized Algorithms:** Explore network vulnerability assessment using randomized algorithms: a) Analyze the probabilistic guarantees of Karger's minimum cut algorithm. b) Compare the deterministic and randomized approaches. c) Justify the applicability of randomized algorithms to large-scale systems.
14.	**Approximation Strategies for NP-Complete Problems:** **S**olve the NP-complete problem of Vertex Cover problem in social network analysis a) Use approximation algorithms. b) Analyze the infeasibility of exact solution. c) Evaluate the approximation ratios and design the feasible approximation strategies for real-world applications.
15.	**Efficient String Processing for Plagiarism Detection Systems:** Examine the problem of detecting textual similarity in plagiarism detection systems used by academic institutions: a) Evaluate the Brute Force, Rabin–Karp, Knuth–Morris–Pratt (KMP) and Boyer–Moore algorithms under different text and pattern characteristics. b) Justify the most suitable string-matching strategy for real-world plagiarism detection systems handling large volumes of text data.

Mini Project: Students have to do a project assigned to them from the course contents in a group. They must submit a project report and give a presentation of the same.

Text Books

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", 2nd Edition, Universities Press (India) / Galgotia Publications, 2018
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 4th Edition, The MIT Press, 2022

3. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 3rd Edition, Pearson Education, 2017

Reference Books

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Fourth Edition, Pearson, 2014.
2. Steven S. Skiena, "The Algorithm Design Manual", 3rd Edition, Springer, 2020.
3. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022.

Online Learning Materials

1. "Design and Analysis of Algorithms by MIT Open Course Ware", https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/ Accessed on December 10, 2025
2. "Design and Analysis of Algorithms by Indian Institute of Science, Bangalore", https://www.youtube.com/watch?v=iBhJCze0_Rk Accessed on December 12, 2025
3. "CS60007 Algorithm Design and Analysis by IIT Kharagpur", https://cse.iitkgp.ac.in/~palash/Courses/2018AlgoDesignAnalysis/DAA2018.html Accessed on December 13, 2025
4. "Digital notes on Design and Analysis of Algorithms by Malla Reddy College of Engineering and Technology, Hyderabad, Telangana", https://mrcet.com/downloads/digital_notes/IT/Design%20and%20Analysis%20Algorithms.pdf Accessed on December 15, 2025
5. "Design and Analysis of Algorithms by Prof. Abhiram Ranade ,Department of Computer Science, IIT Bombay", http://www.infocobuild.com/education/audio-video-courses/computer-science/design-and-analysis-of-algorithms-iit-bombay.html Accessed on December 17, 2025

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1.	Design and Analysis of Algorithms	Prof. Madhavan Mukund	Chennai Mathematical Institute	https://nptel.ac.in/courses/106106131
2.	Design and Analysis of Algorithms	Prof. Abhiram G Ranade, Prof. Ajit A Diwan, Prof. Sundar Viswanathan	IIT Bombay	https://nptel.ac.in/courses/106101060
3.	Fundamental Algorithms: Design and Analysis	Prof. Sourav Mukhopadhyay	IIT Kharagpur	https://nptel.ac.in/courses/106105157

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: CIT113
Course Title: Machine Learning

Programme: B. Tech.	**L:** 3 **T:** 1 **P:** 2	**Credits:** 5
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L) + 15(T) + 30(P) = 90hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 20%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Core Course		

Prerequisites (if any): ESC105, BSIT101, CIT108
Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Apply regression and logistic regression models and evaluate them using appropriate regression metrics.
2	Implement classification algorithms and analyze their performance using standard evaluation measures.
3	Analyze and implement Support Vector Machines and Decision Trees for classification and regression tasks.
4	Apply unsupervised learning methods, including clustering and Gaussian mixture models, to discover patterns and detect anomalies in datasets.
5	Design ensemble learning models using bagging, boosting, and stacking techniques to improve predictive accuracy and model robustness.
6	Apply dimensionality reduction techniques, such as PCA and manifold learning, to enhance the computational efficiency and interpretability of machine learning models.

Contents

Part-A

Unit–1 Regression **7 (L) hrs**

Linear Regression - (Normal Equation, Computational Complexity), Gradient Descent – (Batch Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent), Polynomial Regression, Learning Curves, Regularized Linear Models – (Ridge Regression, Lasso Regression, Elastic Net, Early Stopping), Regression Metrics - (Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared Score, Root Mean Squared Error (RMSE)).

Unit –2 Classification Algorithms **8 (L) hrs**

Binary Classification, Multiclass Classification, Error Analysis, Multilabel Classification, Multioutput Classification, Classification Algorithms: Logistic Regression – (Probability Estimation, Cost Function, Decision Boundaries, Softmax Regression), K-Nearest Neighbors (KNN), Naïve Bayes; Performance Measures: Accuracy, Cross-Validation, Confusion Matrix, Precision, Recall; Precision/Recall Tradeoff, Receiver Operating Characteristic ROC) Curve.

Unit –3 Support Vector Machines (SVM) and Decision Trees **8 (L) hrs**

Linear SVM classification: soft margin classification; Non-Linear SVM Classification: Polynomial, Gaussian, RBF (Radial Basis Function) kernels; Computational Complexity, SVM regression, Decision Tree: Training and Visualization, Making Predictions, Class Probabilities, CART Training Algorithm, Gini Impurity & Entropy; Regularization Techniques, Hyperparameters, Hyperparameter Tuning.

Part-B

Unit –4 Unsupervised Learning **8 (L) hrs**

Clustering: K-Means, K-Means Limitations, K-Modes Clustering for Categorical Data, Clustering for Image Segmentation, Clustering for Preprocessing, Clustering for Semi-Supervised Learning, DBSCAN; Gaussian Mixtures: Anomaly Detection, Selecting Number of Clusters; Curse of Dimensionality, Dimensionality Reduction Approaches: Projection, Manifold Learning; Principal Component Analysis (PCA).

Unit –5 Ensemble Learning **6(L) hrs**

Voting Classifiers, Bagging and Pasting, Random Patches and Random Subspaces, Random Forest: Extra-Trees, Feature Importance; Boosting: AdaBoost, Gradient Boosting, XGBoost; Stacking: Base and Meta Learners, Comparison with Bagging and Boosting.

Unit –6 Prompt Engineering **8(L) hrs**

Introduction, Role in Generative AI and modern ML systems, Types of prompts: Zero-shot, One-shot, Few-shot, Instruction-based and role-based prompts; Prompt design techniques: Clarity, context, constraints, Output structuring; Prompt templates, Chain-of-thought prompting, Controlling tone and bias, Applications: Chatbots, text generation, code generation, AI-assisted data analysis.

Laboratory Work

1. Implement linear regression using a real-world dataset and evaluate model performance.
2. Apply batch, stochastic, and mini-batch gradient descent for regression and compare convergence.
3. Perform polynomial regression and analyze underfitting and overfitting using learning curves.
4. Implement ridge and lasso regression and analyze the effect of regularization.
5. Implement logistic regression for binary classification and evaluate decision boundaries.
6. Perform KNN classification on the MNIST dataset and evaluate accuracy using the confusion matrix, precision, and recall.
7. Analyze Naïve Bayes classifier performance using the ROC curve and the precision–recall tradeoff.
8. Implement linear and nonlinear SVM using polynomial and RBF kernels and compare results.
9. Train and visualize a decision tree using the CART algorithm and compare Gini impurity and entropy.
10. Implement Random Forest and evaluate performance and feature importance.
11. Implement a Boosting technique and evaluate model performance.
12. Implement K-Means clustering and analyze clustering for preprocessing or image segmentation.
13. Apply PCA for dimensionality reduction and analyze explained variance.
14. Implement Prompt engineering concepts using AI tools.

Mini Project

End-to-end Machine Learning application development, data preprocessing, and feature engineering for real-world datasets, model selection, and evaluation, in particular domains such as healthcare /agriculture /finance /cybersecurity /image processing /natural language processing /recommendation systems /smart cities /IoT. The student has to select one domain, select an appropriate dataset, perform data preprocessing, implement suitable machine learning models, evaluate their performance, and present the results in the form of a report and demonstration.

Text Books:

1. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, 3rd Edition, 2022.
2. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, 3rd Edition. 2020
3. Tom M. Mitchell, "Machine Learning", McGraw-Hill Education. 2018
4. Berryman, John, and Albert Ziegler, "Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications," O'Reilly Media, 2024.

Suggested Reading:

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 1st Edition, 2006.
2. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 1st Edition, 2016
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning", Springer, 2009.
5. Phoenix, James, and Mike Taylor, "Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs," O'Reilly Media, 2024.

Online Courses and Video Lectures: SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Machine Learning for Engineering and Science Applications	Prof. Suresh Sundaram	IISc Bangalore	https://onlinecourses.nptel.ac.in/noc20_cs62/preview
2.	Introduction to Machine Learning	Prof. Balaraman Ravindran	IIT Madras	https://onlinecourses.nptel.ac.in/noc23_cs18/preview
3.	Machine Learning Using Python Programming	Prof. Satyajit Das	Institute of Science, Banaras Hindu University	https://onlinecourses.swayam2.ac.in/ini25_cs02/preview

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: ESIT102
Course Title: Design Thinking

Programme: B.Tech.	**L: 0 T:** 0 **P:** 2	**Credits:** 1
Semester: 5	**Theory/ Integrated/ Practical:** Practical	**Teaching Hours:** 30(P) = 30 hrs
Total Max. Marks: 50	**Continuous Assessment (CA) Marks:** 50	**End Semester Examination (ESE) Marks:** NIL
Minimum Percentage of Numerical / Design / Programming Problems in ESE: Nil		
Duration of End Semester Examination (ESE): NA		
Course Type: Engineering Science/ Program Specific Course		

Prerequisites (if any): CIT101, LCIT103, CIT107
Additional Material Allowed in ESE: Nil

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Analyze real-world problems by using various tools and techniques of design thinking.
2	Evaluate the role of cognitive processes in effective problem-solving and design decisions.
3	Design customer-centric solution by using interactive design process.
4	Use scenario as a tool for human-centered design thinking.
5	Create and refine prototypes through testing and evaluation techniques.
6	Develop innovative, sustainable and universally accessible design solutions for diverse users by taking into considerations the ethical issues.

Contents

Part-A

Unit-1 Basics of Design Thinking for Online Applications and Websites 3(L) hrs

Definition of Design Thinking, Need for Design Thinking, Objective of Design Thinking, 3 P's of Design Thinking (People, Process and Place), Models of Design Thinking, Double-Diamond Design Thinking, Stanford 5-Phase Design Thinking Process - Empathize, Define, Ideate, Prototype and Test, Aligning the stages of design thinking process with Kolb's Experiential Learning Theory.

Activity: Students analyze any one computer-based system such as an online learning platform, digital library, e-governance portal, or mobile service application and identify user-centric problems using the design thinking process.

Unit-2 Design Thinking Tools and Techniques for IT-based Software System 3(L) hrs

Personas, Visualization, Stakeholder Mapping, Journey Mapping, Mind Mapping, Star Bursting, Divergent Thinking, Convergent Thinking, Ethnography, Brainstorming, Storytelling, Role Playing, User Interviews, The Six Thinking Hats: White Hat (Information), Red Hat (Emotions), Black Hat (Caution), Yellow Hat (Optimism), Green Hat (Creativity), Blue Hat (Process).

Activity: Students analyze a selected IT-based software system and create a detailed hat-wise analysis report by applying the six thinking hats technique. On the basis of the report suggest ideas based on divergent thinking.

Unit-3 Cognitive Foundations of the Design Thinking Process for an Application Software **3(L) hrs**

Human Memory – sensory, short-term and long-term memory, Reasoning – deductive, inductive and abductive reasoning, Problem solving, Gestalt Theory, Problem-space theory, Analogy in problem-solving, Human errors and mental models, Emotion and individual differences, using Cognitive Psychology for deriving models and evaluation techniques for a human-centered system.

Activity: Students conduct user interviews and observations for any one selected system such as a healthcare appointment system, online banking application, or e-commerce website, and prepare empathy maps highlighting mental models, cognitive load and user expectations while using software applications.

Unit-4 Designing of IT-based Software Products involving Human Interaction 3(L) hrs

The Interaction Design Process, Designing customer-centric solution by understanding users and context: user research methods, user characteristics and personas, task and activity analysis, context of use, user needs and requirements elicitation, observation and empathy techniques, evaluating and enhancing customer experience.

Activity: Students analyze an existing app/website (ATM interface, healthcare app) and identify usability issues based on human interaction principles and suggest improvements.

Unit-5 Scenario-based Human-Centered Design Thinking for the Software Applications **3(L) hrs**

Introduction to Scenarios in Design Thinking, Types of Scenarios for human-centered design, Elements of a Scenario in Design Thinking, Scenario Development Process, Scenario Representation Methods, Using Scenarios for Ideation, Prototyping and Testing, Mapping Scenarios to requirements and user needs, Iterative-refinement based on feedback.

Activity: Students develop a user persona (age, needs and limitations), write a real-life usage scenario and map the user journey and identify moments of frustration and opportunity. Design the software application based on this scenario-based design thinking, user-needs and perform iterative-refinement based on feedback.

Part-B

Unit-6 Prototyping, Testing and Evaluation of the Design of the Software Solution 4(L) hrs

Types of prototyping, user involvement in prototyping, qualitative and quantitative user-feedback of the design, analysis and interpretation of feedback, feedback-driven iteration, user-testing and usability-testing, context-based and scenario-driven testing, functional and performance testing, identifying errors and design issues, redesigning and re-creating, goals of evaluation, choosing and evaluation method, evaluation through experts, evaluation through user participation, analysis of evaluation results, design validation and improvement.

Activity: Students create paper or cardboard prototypes of their software solution for a practical application like online appointment system, online shopping system, library management, etc. Peer groups test the prototype and provide feedback. Students iterate and document changes based on feedback.

Unit-7 Innovative and Universal Design of the Software Systems 4(L) hrs

SCAMPER (Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Rearrange), Co-creation and Collaborative Innovation of the Design, Universal Design Principles, multi-modal interaction, designing for diversity: designing for user with disabilities: visual impairment, hearing impairment, physical impairment, speech impairment, Dyslexia, Autism; designing for different age groups: older people, children; designing for cultural and regional differences.

Activity: Students incorporate universal design principles in any one software solution (education, healthcare, governance, or utility domain) and present the final prototype along with design and evaluation documentation.

Unit-8 Ethics in Design Thinking of the Application Software Systems **3(L) hrs**

Introduction to Ethics, Importance of Ethics in Design Thinking, Values and Morals, Ways of Ethical Thinking, Ethical Issues, Causes of Unethical Practices, Sources of Ethics, Qualities needed by Ethical Design Thinkers, Universal Principles of Ethics, Ethical Design Test.

Activity: Students evaluate the selected software solution such as a data-driven application, cloud-based system, or AI-enabled service with respect to ethical issues.

Unit-9 Design Thinking of the Software System for Sustainability **4(L) hrs**

Sustainability and sustainable futures, Three Pillars of Sustainability: Environmental Pillar (Planet), Social Pillar (People) and Economic Pillar (Profit), Sustainable Development Goals (SDGs), Design Thinking for achieving Sustainable Development Goals, Circular Economy aids Sustainability.

Activity: Students analyze an existing software system with respect to sustainability issues and apply the design thinking process to develop a sustainable solution that minimizes environmental and resource impact.

Note: More activities may be conducted other than the activities mentioned above as per the discretion of the course coordinator.

Final Report Submission: Student must submit a detailed report of all the activities performed during the course.

Text Books

1. Sanjay Kumar Magoo and Rakesh Gupta, “Design Thinking”, McGraw Hill, 2025.
2. E. Balaguruswamy, “Design Thinking: A Beginner’s Perspective”, McGraw Hill Education India, 2024.
3. Amith Sheelvanth Nataraj, “The Design Thinking Code: Unlock Innovation, Growth and Market Domination”, Adhyyan Books, 2025.
4. Christian Müller-Roterberg ,“Design Thinking for Dummies”, Wiley, 2020

Reference Books

1. Tim Brown, “Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation”, HarperCollins, 2019
2. Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale, “Human-Computer Interaction”, Pearson, Third Edition, 2017
3. Abbie Griffin, Michael Luchs, Scott Swan, “Design Thinking: New Product Development Essentials”, Wiley, 2015
4. Don Normam, “The Design of Everyday Thing”, MIT Press, 2013.

Online Learning Materials

1. “Design Thinking”, https://www.interaction-design.org/literature/topics/design-thinking?srsltid=AfmBOorlk1q6PDrLPOK7wWlw3iJen1uuTZRDdq_mpMmqTXV-XJydlIK1

Accessed on January 10, 2026

2.“An Introduction to Design Thinking Process Guide”
https://web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf

Accessed on January 12, 2026

3.“The 5 Stages in the Design Thinking Process”,
https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process?srsltid=AfmBOorVSMm06FHdGZ0CzcPzdXoIwtm9o8VTJw-X5y7Xx5Q_NFUJ4pDz

Accessed on January 20, 2026

4."Understanding Design Thinking", https://www.tandfonline.com/doi/full/10.1080/07370024.2014.919212
Accessed on January 22, 2026

5. "How to Create Scenarios For Design Thinking", https://www.innovationtraining.org/how-to-create-scenarios-for-design-thinking/
Accessed on January 25, 2026
6. "Innovation and Design Thinking", https://vtu.ac.in/pdf/sm/idt.pdf Accessed on January 26, 2026
7. "Universal Design", https://www.interaction-design.org/literature/topics/universal-design?srsltid=AfmBOoqsOtVdRJyZPLQ3JWehPNFAunFh8EO6ZjnHoaAe2mL7aBbu3Zkc
Accessed on January 28, 2026

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1.	Design Thinking - A Primer	Prof. Ashwin Mahalingam, Prof. Bala Ramadurai	IIT Madras	https://nptel.ac.in/courses/110106124
2.	Product Engineering and Design Thinking	Prof. Pranab K Dan, Prof. Prabir Sarkar	IIT Kharagpur, IIT Ropar	https://nptel.ac.in/courses/112105316
3.	Design, Technology and Innovation	Prof. B.K. Chakravarthy	IIT Bombay	https://nptel.ac.in/courses/107101088
4.	Augmenting Design Thinking with Human-Computer Interaction	Prof. Sonal Atreya	IIT Roorkee	https://nptel.ac.in/courses/107107580

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Theory of Computation	Prof. Subrahmanyam Kalyanasundaram	IIT Hyderabad	https://onlinecourses.nptel.ac.in/noc24_cs49/preview
2	Theory of Computation	Prof. Raghunath Tewari	IIT Kanpur	https://onlinecourses.nptel.ac.in/noc21_cs83/preview
3	Compiler Design	Prof. Santanu Chattopadhyay	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc21_cs07/preview
4	Theory of Automata, Formal Languages and Computation	Prof. Kamala Krithivasan	IIT Madras	https://nptel.ac.in/courses/106106049/

5	Principles of Compiler Design	Prof. Y.N. Srikanth	IISc Bangalore	https://nptel.ac.in/courses/106108113/
6	Theory of Computation	Prof. Michael Sipser	MIT, Cambridge, USA	https://ocw.mit.edu/courses/18-404j-theory-of-computation-fall-2020/resources/introduction-finite-automata-regular-expressions/
7	Automata Theory	Prof. Jeff Ullman	Stanford University, CA, USA	https://online.stanford.edu/courses/soe-ycsautomata-automata-theory

Tools & Platforms

Java Formal Languages and Automata Package (JFLAP) for automata simulation
Online GATE preparation portals for practice problems.

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT101
Course Title: Artificial Intelligence

Programme: B.Tech.	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+ 30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): EIT101, ESC105, BSIT201

Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Explain the fundamental concepts of Artificial Intelligence, intelligent agents, and problem-solving techniques.
2	Apply uninformed and informed search algorithms to formulate and solve AI problems.
3	Model knowledge using propositional and first-order logic and perform automated reasoning.
4	Analyze decision-making under uncertainty using probabilistic reasoning and Bayesian models.
5	Evaluate modern AI techniques, including optimization, Edge AI, AutoML, and explainable AI for real-world deployment.
6	Assess ethical, societal, and sustainability implications in the design of AI systems.

Contents

Part-A

Unit-1 Foundations of Intelligent Agents **8(L) hrs**

AI Basics: State-of-the-art and the concept of Rationality; Agent Architectures: Simple reflex, model-based, goal-driven, and utility-driven agents; Environments: Nature of environments (PEAS) and Bounded Rationality.

Problem Solving and Heuristic Search: Problem Reduction; Means-End Analysis, and Stochastic Search (Simulated Annealing, Genetic Algorithms); Constraint Satisfaction (CSPs): Constraint propagation, Backtracking search, and Local search for CSPs.

Unit-2 Adversarial Search and Game Theory **6(L) hrs**

Optimal Decisions: Game trees, Minimax algorithm, and Alpha-Beta Pruning; Strategic Play: Branch and Bound, Iterative Deepening, and Imperfect Information games; Decision Theory: Basics of utility theory and sequential decision problems.

Unit-3 Knowledge Representation and Formal Logic **8(L) hrs**

Logic Systems: Propositional and First-order Predicate logic; Natural Deduction, and Resolution; Reasoning: Forward vs. Backward chaining, ISA relationships, and computable functions; Structures: Semantic networks, Frames, and Procedural vs. Declarative knowledge; Planning: Goal Stack Planning, Partial Order Planning, and Hierarchical Planning (Blocks World).

Part B

Unit-4 Probabilistic Reasoning and Uncertainty 8(L) hrs

Quantifying Uncertainty: Probability basics, Bayes' Rule, and Independence; Probabilistic Models: Bayesian Networks (Semantics and Inference) and Dempster-Shafer theory; Fuzzy Systems: Fuzzy sets, Fuzzy logic, and decision-theoretic expert systems.

Unit-5 Efficient AI and Sustainable Deployment 8(L) hrs

Model Optimization: Model Pruning, Quantization, and Knowledge Distillation; Deployment Frameworks: Overview of TensorFlow, PyTorch, TensorFlow Lite for Edge AI; Sustainable AI: Green data management, reducing storage footprints, and energy-efficient model training.

Unit-6 Future Frontiers and Ethics 7(L) hrs

Explainable AI (XAI): Popular XAI Methods, XAI Evaluation Metrics, Applications; Edge AI: Architecture, software frameworks, model optimization; AutoML: Definition and need for AutoML; AutoML Frameworks and Tools: Google AutoML, Auto-sklearn, H2O AutoML, TPOT, Microsoft Azure AutoML; TinyML: Introduction, TinyML vs Edge AI.

Laboratory Work

Practical No.	Practical Title	Objective	Description / Task	Examples
1	To describe an intelligent agent using the PEAS (Performance, Environment, Actuators, Sensors) framework.	To understand task environments using the PEAS framework	Identify Performance, Environment, Actujators, and Sensors for real-world agents like vacuum cleaner and traffic signal system	Smart home system, ATM machine, delivery robot
2	To design and implement a simple reflex agent based on condition–action rules.	To design an agent that acts based only on current percept	Implement condition–action rules (e.g., if dirty → clean, else → idle)	Light automation, fire alarm system, automatic door
3	To design a goal-based intelligent agent that selects actions to achieve a specified goal.	To implement goal-oriented decision making	Design an agent that moves from a start state to a goal state on a grid or map	Maze solver, robot navigation,shortest path finder
4	To implement the Best-First Search algorithm using a heuristic function to find an optimal solution path.	To apply heuristic-based search techniques	Solve the 8-puzzle problem using misplaced tiles heuristic	15-puzzle, route optimization, pathfinding problems
5	To solve a problem using the Means–Ends Analysis approach by reducing the difference	To solve problems by reducing differences	Solve the Water Jug problem step-by-step using	Tower of Hanoi, block world problem

	between the current state and the goal state.	between states	means–ends analysis	
6	To implement the Minimax algorithm for decision-making in two-player adversarial games.	To understand adversarial search strategies	Implement Tic-Tac-Toe AI using minimax algorithm	Connect-4, simple chess endgames
7	To represent knowledge using propositional logic and logical connectives.	To represent knowledge using logical statements	Convert real-life statements into propositional logic and verify validity using truth tables	Weather prediction, access control rules
8	To implement a forward chaining inference system for reasoning with facts and rules.	To implement rule-based reasoning	Design a forward chaining system using IF-THEN rules to infer conclusions	Medical diagnosis, expert systems

Mini Projects: Students can choose one of the case studies, such as "Intelligent Course Recommendation System," as a mini project:

Title	Problem Statement	AI Concepts / Techniques Used	Student Learning & Extensions
Intelligent Course Recommendation System	Students face difficulty choosing suitable electives due to lack of guidance and eligibility awareness	Goal-based agent, propositional logic, forward chaining, heuristic ranking	Add fuzzy interest levels, build chatbot recommender, compare rule-based vs ML

Textbooks

1. Russell, S. and Norvig, P. "Artificial Intelligence: A Modern Approach", Pearson, 4th Edition, 2021.
2. Poole, D. and Mackworth, A. "Artificial Intelligence: Foundations of Computational Agents", Cambridge University Press, 2nd Edition, 2017.

Reference Books

1. Rich, E., Knight, K., and Nair, S. B. Artificial Intelligence, McGraw-Hill, 3rd Edition, 2009.
2. Kaushik, S. Artificial Intelligence, Cengage Learning India, 2011.
3. Trivedi, M. C. A Classical Approach to Artificial Intelligence, Khanna Publishing, 2020.
4. Goodfellow, I., Bengio, Y., & Courville, A., "Deep Learning", MIT Press, 2016.

E-Books and online learning material

1. Barr, A., & Feigenbaum, E. A. (Eds.). The Handbook of Artificial Intelligence. Stanford University. https://stacks.stanford.edu/file/druid:qn160ck3308/qn160ck3308.pdf

Online Courses

1. Prof. Deepak Khemani, IIT Madras, Artificial Intelligence: Knowledge Representation and Reasoning, https://onlinecourses.nptel.ac.in/noc26_cs63/preview

2. Prof. Anupam Basu, Prof. S. Sarkar, IIT Kharagpur, Artificial Intelligence, https://nptel.ac.in/courses/106105077
3. Coursera – Artificial Intelligence Courses, https://www.coursera.org/courses?query=artificial%20intelligence

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT102
Course Title: Advanced Computer Networks

Programme: B. Tech	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): ESIT101, ESC103, CIT102
Additional Material Allowed in ESE: NIL
Course Outcomes (COs) - After completing this course, students will be able to:

CO#	Course Outcomes
CO1	Analyze the evolution and architectural limitations of traditional TCP/IP networks and explain the design principles of high-performance and large-scale network architectures.
CO2	Formulate and analyze congestion control problems using mathematical, utility-based, and optimization-based models to understand network stability and fairness.
CO3	Analyze Enhanced IGRP and Open Shortest Path First routing protocols.
CO4	Understand unicast and multicast routing protocols and their application in wireless network technologies.
CO5	Apply multimedia wireless networking concepts to analyze streaming audio, video, and Voice over IP (VoIP) systems.
CO6	Compare ad hoc networks with cellular networks

Contents

Part-A

Unit-1 High-Performance Network Architectures **8(L) hrs**

Evolution of Internet architecture and design goals, limitations of traditional TCP/IP networks in high-speed and large-scale environments, principles of high-performance network design, modern router and switch architectures, separation and interaction of control plane and data plane, fast packet forwarding mechanisms, packet buffering and memory management strategies, flow-based networking concepts, scalability challenges, reliability mechanisms, fault tolerance techniques in large-scale and backbone networks.

Unit-2 Advanced Congestion Control and Traffic Engineering 8(L) hrs

Mathematical foundations of congestion control, utility-based and optimization-based congestion models, window-based versus rate-based congestion control, modern congestion control algorithms including BBR and PCC, active queue management techniques, ECN and in-network feedback mechanisms, flow scheduling and fairness, traffic engineering in large-scale networks, centralized and software-driven traffic engineering, WAN and data-center traffic optimization.

Unit-3 Unicast and Multicast Routing Protocols 7(L)hrs

Concept of unicast and multicast communication, Classification of Unicast and multicast Routing Protocols, Distance vector Routing and Link state Routing, Hybrid Routing, Unicast Routing applications, Scalability Challenges in Unicast and Multicast Routing.

Part-B

Unit-4 Network Routing 8(L)hrs

Overview of Router, Static and Dynamic Routing, Introduction to Classless Routing, Distance Vector Routing Protocols, Router Administrative Configurations, Router Interfaces, Viewing, Saving and Erasing Configurations, Routing Information Protocol, Configuration of EIGRP (Enhanced IGRP) and OSPF (Open Shortest Path First).

Unit-5 Wireless Network Technologies 7(L)hrs

Wireless Network Communication- 3G, 4G, 5G. Multimedia Wireless Networks – Streaming Audio and Video, Voice Over Internet Protocol (VoIP), Protocols – Real time Transport Protocol (RTP), Real-Time Streaming Protocol (RTSP).

Unit-6 Ad hoc Networks 7(L) hrs

Features, Advantages and Applications, Adhoc versus Cellular networks, Network Architecture, Protocols: MAC and Routing Protocols, Technologies, Applications of Mobile Ad hoc Networks.

Laboratory Work

Experiment No.	Experiment Title
1	Installing Packet Tracer and Wireshark**.**
2	Analyze network packets to detect performance bottlenecks using any suitable tool e.g. Wireshark.
3	Analyze forwarding performance differences between TCAM-based lookup and software-based lookup using emulation tools.
4	Basic Congestion Observation Using Ping and Iperf
5	Compare performance of window-based and rate-based congestion control in high bandwidth-delay scenarios.
6	Study of Unicast and Multicast Routing Protocols
7	Study of router hardware components and router operating system commands.
8	Study of 3G, 4G and 5G Wireless Technologies and Streaming Audio and Video over Wireless Network.
9	Configuration and analysis of Voice over Internet Protocol (VoIP) communication over a wireless network
10	Configuring Ad hoc Network.

Mini Project: - Student has to do a project assigned from course contents in a group of two or three students. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same. Note: It is

recommended that mini project allocation to students be done within two-three weeks of the start of the semester.

Text Books

1. Forouzan Behrouz A. Data Communication and Networking 5E McGraw Hill Education (India), New Delhi, 2005, ISBN-13:978-1-25 906475-3
2. Computer Networks & Internets – Douglas E. Comer (Latest Edition)
3. Comer Douglas E. Internetworking with TCP/IP, Volume I, Fourth Edition. Prentice Hall of India Private Limited, New Delhi- 110001 ISBN-81-203 2065-4
4. Forouzan Behrouz A. TCP/IP Protocol Suite Tata McGraw-Hill Edition, Ne Nelson, B., Phillips, A., & Steuart, C. (2014). Guide to computer forensics and investigations. Cengage Learning
5. Computer Networks – Andrew S. Tanenbaum & David Wetherall (Latest Edition)

Course Link (NPTEL)

https://onlinecourses.nptel.ac.in/noc26_cs60/preview
https://www.nptelprep.in/courses/106106243
http://nptel.ac.in/content/syllabus_pdf/106105081.pdf

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT103
Course Title: Advanced Web Technologies

Programme: B.Tech	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): ESC103, LCIT103, CIT106
Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Apply Bootstrap framework concepts to design responsive, user-friendly web interfaces using containers, utilities, components, and JavaScript add-ons.
2	Analyze the MVC architectural pattern and explain its role in structuring scalable, maintainable, and performance-oriented web applications.
3	Design and develop Laravel-based web applications using routing, controllers, Blade templates, migrations, and Eloquent ORM to perform basic CRUD operations.
4	Implement middleware, validation, session management, authentication, and file handling in Laravel while adhering to security and ethical coding practices.
5	Develop RESTful APIs using Laravel to enable JSON-based data exchange and simple CRUD services for web and mobile applications.
6	Work effectively individually or in teams to document, present, and manage a mini web project using Bootstrap and Laravel while demonstrating communication and project management skills.

Contents

Part-A

Unit-1 Bootstrap Framework for Responsive Web Design **10 (L)hrs**

Bootstrap Concept and its File Structure, Bootstrap Utilities and Containers- Flexbox utilities, Spacing utilities, Shadow utilities. Components in Bootstrap, Accordions, Toast, Offcanvas. Responsive utilities and layouts, visibility, Icons, Typography and its basic Elements. The JavaScript Add-ons: Custom data attributes, Basic modal window, Tabs, Collapse, Tooltips and popovers, Dropdown Menu, Carousels.

Unit-2 MVC Approach for Web Applications **12(L) hrs**

Need for MVC in Web Applications, MVC Architecture: Model, View, Controller, MVC Framework, Design Patterns, Foundation- Autoloading, Exceptions, Type Methods, Base Class, Configuration of MVC, Caching: Performance Bottlenecks, Routing, Templates, Overview of Object-Relational Mapping (ORM).

Part-B

Unit-3 Laravel Fundamentals **11(L) hrs**

Laravel Framework, Installation & Environment Setup, Laravel Project Structure, Artisan CLI: Creating controllers, Creating models, Routing: Basic routes, Resource routes, Controllers: Basic controllers, Resource controllers, Blade Templates: Layouts, Sections, Components, Eloquent ORM, Database Migrations, Seeders and CRUD Operations, One-to-One and One-to-Many Relationships.

Unit–4 Laravel Middleware, Security & APIs **12(L) hrs**

Laravel Middleware: Concept and usage, Request lifecycle, Form Handling: Request validation, Error messages, Session management, Flash messages, File upload and storage, Authentication & Authorization: Login and Registration using Laravel Breeze/UI, RESTful API Development: API routes, JSON responses, Simple CRUD API.

Laboratory Work

Experiment No.	Experiment Title
1	Design a responsive webpage using Bootstrap by implementing containers, grid system, utilities, and typography.
2	Develop responsive user interface components such as accordions, toasts, offcanvas elements, media objects, and responsive tables using Bootstrap.
3	Create responsive web forms using Bootstrap and implement client-side and server-side validation techniques.
4	Implement interactive web features using Bootstrap JavaScript components such as modal windows, tabs, collapse, tooltips, popovers, dropdown menus, and carousels.
5	Install and configure the Laravel framework and explore its directory structure and MVC architecture.
6	Implement routing and controllers in Laravel and integrate dynamic views using Blade templates.
7	Perform database operations using Eloquent ORM and database migrations, including the creation of basic model relationships.
8	Develop form handling functionality in Laravel with validation, user input processing, redirects, session management, and flash messages.
9	Implement middleware, basic authentication, and develop a simple RESTful API in Laravel that returns JSON responses for CRUD operations, API Testing using Postman or Insomnia.
10	Implement application security and performance optimization in a Laravel web application by configuring environment variables, enabling caching, handling exceptions, and applying basic authorization policies.

Mini Project: The instructor will provide a list of suggested Mini-projects from the course content to students to opt for. The group of 2-3 students is allowed to submit a project report and the team will have to demonstrate their project along with a presentation.

Note: It is recommended that mini project allocation to students be done within two-three weeks of the start of the semester. This is only the suggested list of Practicals. Instructors may also frame additional Practical's relevant to the course contents (if required).

Text Books

1. Hussain, Frahaan and Kameron Hussain. Mastering Bootstrap 5: From Basics to Expert Projects. Sonar Publishing, 2023.
2. Ben Frain, Responsive Web Design with HTML5 and CSS, 4th ed. O'Reilly Media, 2024.
3. Chris Pitt, Pro PHP 8 MVC: Model View Controller Architecture-Driven Application Development, 2nd ed. Apress, 2021.
4. M. Stauffer, Laravel: Up & Running: A Framework for Building Modern PHP Apps, 3rd ed. O'Reilly Media, 2023.
5. Sanjib Sinha, Beginning Laravel, 2nd ed. Apress, 2019

Reference Books

1. E. G. Lumsden, Laravel 12 Essentials: A Developer's Guide to Building, Testing, and Deploying Modern PHP Web Applications, Large Print ed., 2 Nov. 2025.
2. D. Jain, Ultimate Laravel for Modern Web Development. New Delhi, India: Orange Education Pvt. Ltd., 2024.
3. Silvio Moreto, Matt Lambert, Jason Marah & Benjamin Jakobus, Bootstrap 4 – Responsive Web Design. Packt Publishing, 2017.
4. Arda Kilicdagi, Laravel Design Patterns and Best Practices. Packt Publishing, 2023
5. Martin Bean,"Laravel 5 Essentials", Packt Publishing, 2015.
6. Christopher John Pecoraro, Mastering Laravel. Independently published, 2023

E-Books and Online Learning Material

1. Learn Laravel in 30 Days https://laracasts.com/series/30-days-to-learn-laravel-11 Accessed on Jan 7, 2026
2. Official Bootstrap Documentation https://getbootstrap.com/docs/5.0/getting-started/introduction/ Accessed on Jan 7, 2026
3. Official Laravel Documentation https://laravel.com/docs Accessed on Jan 7, 2026
4. Learning Laravel eBook[pdf] https://riptutorial.com/ebook/laravel Accessed on Jan 7, 2026
5. JackVo, Learning Laravel: The Easiest Way free PDF, 2016 https://www.computer-pdf.com/webprogramming/php/541-tutorial-learning-laravel-the-easiest-way.html Accessed on Jan 7, 2026
6. https://www.youtube.com/watch?v=b9g4_8nAsdA&list=PLK5U0tyd34tCrlOvIfeVO6YzW_z-f0jxk
7. https://youtu.be/b9g4_8nAsdA?si=Yr8e3Y5szvuTr6iX Accessed on Jan 7, 2026
8. https://www.youtube.com/playlist?list=PLcN9Ke6gpdsVogUdX9Ng-X_DVj7SDPjUe Accessed on Jan 7, 2026
9. https://www.youtube.com/playlist?list=PLWiQT7FWaG1jG8RQq1dFiSE7O-g9HQuYE Accessed on Jan 7, 2026

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Web Technologies	Dr. S. Sasirekha	NITTTR, Chennai	https://onlinecourses.swayam2.ac.in/ntr25_ed123/preview

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT104

Course Title: Blockchain Architecture and its Applications

Programme: B.Tech	**L:**3 **T:** 0 **P:**2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+30(P) =75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): CIT101, CIT102, CIT108

Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
CO1	Apply cryptography concepts like hashing, keys, and digital signatures to secure blockchain data.
CO2	Analyze blockchain data structures, transactions, and UTXO/account models to maintain ledger integrity.
CO3	Implement consensus mechanisms such as PoW, PoS, and fork resolution to achieve distributed agreement.
CO4	Design and execute cryptocurrency transactions and ERC-20 tokens to understand digital currency operations.
CO5	Develop and deploy smart contracts on Ethereum and Hyperledger Fabric for decentralized applications.
CO6	Evaluate blockchain applications, security, scalability, and privacy challenges to propose effective solutions.

Contents

Part-A

Unit-1 Cryptographic Foundations of Blockchain 8(L)hrs

Cryptography, Types: Symmetric and Asymmetric cryptography, Comparison, Hash functions working of hash functions, Digital signatures, working of digital signatures, Public Key cryptography, components and working of Public Key Cryptography.

Unit-2 Blockchain Data Structures & Internals 8(L)hrs

Overview of blockchain structure, Block components: header, transactions, timestamp, nonce, Transaction structure and verification process, Linking of blocks using hashes, UTXO model vs account-based model, Forks, chain reorganizations, and chain selection rules

Unit-3 Consensus Mechanisms in Blockchain 7(L) hrs

Need for consensus, Role of consensus in blockchain security, Byzantine Generals Problem, Types: Proof of Work (PoW), Proof of Stake (PoS), Proof of Capacity (PoC), Proof of Authority (PoA), Practical Byzantine Fault Tolerance (PBFT).

Part-B

Unit-4 Cryptocurrencies **8(L) hrs**

Bitcoin architecture, Design of Bitcoin Blockchain, Steps of Bitcoin Transactions, Mining process, Cryptocurrency wallets, Types of Wallets, Benefits, Challenges in choosing wallets, UTXO model: Working and its importance, concept and Importance of Tokens

Unit-5 Smart Contract Development **8(L)hrs**

Ethereum: Working, Types, Features, Ethereum Virtual Machine (EVM), Smart contracts: concepts & lifecycle, Features, Types, Working of Smart Contract, Solidity basics, Introduction to Hyperledger Fabric, working and workflow

Unit-6 Blockchain Applications and challenges **6(L) hrs**

Cryptocurrency & digital payments, Blockchain in Finance services, Supply Chain Management, Healthcare and governance, Blockchain and IoT, Security issues in blockchain systems, Scalability challenges, Privacy issues in blockchain networks.

Laboratory Work

Experiment No	Experiment Title
1	Implement SHA-256 hash function for data integrity verification
2	Generate and verify digital signatures using asymmetric cryptography
3	Simulate a basic blockchain with block creation and validation
4	Demonstrate UTXO-based transaction model for cryptocurrency systems
5	Implement Proof-of-Work mining using nonce and hash difficulty
6	Analyze blockchain fork scenarios and resolve using longest-chain rule
7	Simulate Bitcoin wallet creation and transaction execution process
8	Deploy a simple smart contract on test network
9	Develop and deploy Solidity smart contracts using Remix IDE
10	Design and deploy basic chaincode in Hyperledger Fabric network

Tools: Python (3.x), Remix IDE

Mini Project: Student has to do a project assigned from course contents in a group of students. They must submit a project report and give a presentation of the same.

Textbooks

1. Imran Bashir, “Mastering Blockchain”, 3rd Edition, Packt Publishing, 2023.
2. Andreas M. Antonopoulos, “Mastering Bitcoin: Programming the Open Blockchain”, 2nd Edition, O’Reilly Media, 2017.
3. Kirankalyan Kulkarni, “Essentials of Bitcoin and Blockchain”, Packt Publishing
4. Daniel Drescher, “Blockchain Basics: A Non-Technical Introduction in 25 Steps”, Apress, 2017.

Reference Books

1. Vikram Narayana and Nishith Pathak, “Blockchain Enabled Applications”, Apress, 2019.
2. Qin Zhan and Shouhuai Xie, “Blockchain Security and Privacy”, Springer, 2022.

Online Learning Materials

1. https://www.coursera.org/learn/blockchain-basics
2. https://www.udemy.com/course/blockchain-build-secure-and-deploy-real-world-applications/

Supplementary SWAYAM Course

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Blockchain and its Applications	Prof. Sandip Chakraborty, Prof. Shamik Sural	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc26_cs34/preview
2	Blockchain Architecture Design and Use Cases	Prof. Sandip Chakraborty & Prof. Praveen Jayachandran	IIT Madras	https://nptel.ac.in/courses/106105184

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT105
Course Title: Advance Java

Program: B.Tech	**L: 3 T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/ Practical/ Integrated:** Theory	**Teaching Hours =** 45(L) + 30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Internal Assessment (CIA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 20%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Additional Material Allowed in ESE: NIL
Prerequisites: CIT107

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Understand and develop GUI-based applications using AWT, Swing and modern UI concepts.
2	Apply file handling techniques and advanced I/O operations including streams For data processing.
3	Analyze and utilize Java Collections Framework and Stream API to efficiently manage, process, and manipulate data in real-world scenarios.
4	Evaluate different approaches for developing web applications using Servlets, JSP with respect to performance and scalability.
5	Design and develop robust and modular Java applications using advanced features such as Generics, Lambda Expressions, Functional Interfaces, Annotations, Reflection, and modern Java constructs (Records, Optional, switch expressions).

Contents

Part-A

Unit- 1 Graphics Programming using AWT, Swing and Layout Manager 7(L) hrs

Introduction to Graphics Programming, The Graphics Class, Drawing Arcs and Polygons, Line Graphs, Drawing Bar Charts, Using Control Loops in Applets, Introduction to AWT Package, Window Fundamentals, Closing an AWT Window or Frame, Working with Applets and Fonts, Layout Managers, Handling Events on AWT Components, Introduction to Swing Package, Components and Containers, AWT versus Swing, GUI Applications (Bouncing Balls, Student Management System)

Unit -2 Managing Input/Output Files 6(L) hrs

Introduction, Concept of Streams, Stream Classes, Byte Stream Classes, Character Stream Classes, Using Streams, Other Useful I/O Classes, Using the File Class, Input/Output Exceptions, Creation of Files, Reading/Writing Characters, Reading/Writing Bytes, Handling Primitive Data Types, Concatenating and Buffering Files, Random Access Files, Interactive Input and Output, Other Stream Classes, Introduction to Scanner Class

Unit- 3 Java Collections Framework 6(L) hrs

Collections overview, Overview of Collection Interfaces, Collection Classes and Algorithms,

accessing a Collection via Iterator, Storing user defined classes in Collections, Practical Implementation Examples

Unit -4 Java Servlets and Java Server Pages 5(L) hrs

Overview, Evolution of N-Tier Architecture, Server types, Life cycle of Servlet, Servlet API, Reading Servlet Parameters, Difference between two servers- Application and Web, Overview to Java Naming and Directory Interface (JNDI), Overview of JSP with examples.

Part-B

Unit -5 Enumerations, Autoboxing, Assertions and Annotations 5(L) hrs

Enumeration Fundamentals, Type Wrappers, Autoboxing and methods, Assertions: Introduction, Rules, Implementation, Preconditions & Postconditions, Design by Contract, Basics of Annotations, Single-Member Annotations, The Built-In Annotations, Repeating Annotations.

Unit -6 I/O, Try- with Resources and Other Topics 5(L) hrs

I/O Basics, Reading Console Input (Characters and Strings), Writing Console Output, The Print Writer Class, Reading and Writing Files, automatically closing a file, Static Import

Unit -7 Generics and Lamada Expressions 6(L) hrs

Generic Overview, Generic class with two type Parameters, Bounded Types, Using Wildcard Arguments, creating a Generic Method, Generic Interfaces, Generic class hierarchies, Lamada Expression fundamentals, Block Lamada Expressions, Passing Lamanda Expressions as Arguments, Lamada Expressions and Exceptions, Basics of Module.

Unit -8 Switch Expressions and Other Recently Added Features 5(L) hrs

Pattern Matching with instance of, Enhancements to switch, introducing switch Expression and yield statement, introducing arrow in case statement, Test block Fundamentals, Understanding Leading Wide space, Sealed Classes and Interfaces.

LIST OF PRACTICALS

Unit 1: Graphics Programming using AWT & Swing

Practical 1: Drawing Basic Shapes using Graphics Class

Aim: To remember and understand the fundamentals of Java Graphics, apply drawing methods to create shapes like lines, rectangles, arcs, and polygons, analyze coordinate systems, and design a graphical interface using AWT/Swing components.

Description: Students will create a GUI application to draw multiple shapes using the Graphics class. They will explore coordinate systems, colors, and rendering logic.

Practical 2: Develop a Bar Chart / Line Graph Application

Aim: To understand data visualization, apply loops and graphics methods to plot charts, analyze graphical scaling, and create dynamic chart-based applications.

Description: Create a program that takes input values and displays them as a bar chart or line graph using AWT/Swing.

Practical 3: Event Handling in GUI Applications

Aim: To understand event-driven programming, apply event listeners, analyze user interactions, and develop responsive GUI systems.

Description: Build a GUI form (e.g., student registration) and implement event handling for buttons, text fields, etc.

Practical 4: Mini Project – Bouncing Ball Simulation

Aim: To apply animation concepts, analyze motion logic, and create an interactive GUI-based simulation.

Description: Develop a bouncing ball animation using threads and repaint methods.

Unit 2: File Handling and Streams

Practical 5: File Creation and Character Stream Operations

Aim: To understand file handling, apply character streams, analyze file content processing, and develop file-based applications.

Description: Create, read, and write text files using File Reader and File Writer.

Practical 6: Byte Stream Handling

Aim: To differentiate between byte and character streams, apply byte stream classes, and analyze binary data operations.

Description: Read/write binary files (e.g., images or serialized data).

Practical 7: Random Access File Implementation

Aim: To understand random file access, apply file pointer operations, and design efficient file manipulation systems.

Description: Develop an application to store and retrieve records randomly.

Practical 8: Buffered and Scanner Class Usage

Aim: To understand buffering concepts, apply efficient I/O techniques, and analyze performance improvements.

Description: Compare buffered vs unbuffered file operations and implement Scanner-based input.

Unit 3: Java Collections Framework

Practical 9: Implementation of List, Set, and Map

Aim: To understand collection hierarchy, apply different data structures, analyze performance differences, and evaluate appropriate usage.

Description: Implement ArrayList, HashSet, and HashMap with real-world data.

Practical 10: Iterator and Custom Objects in Collections

Aim: To understand iteration mechanisms, apply iterators, analyze traversal methods, and store user-defined objects.

Description: Store student objects in collections and traverse using Iterator.

Unit 4: Servlets and JSP

Practical 11: Basic Servlet Development

Aim: To understand web architecture, apply servlet lifecycle methods, and develop server-side applications.

Description: Create a servlet to handle HTTP requests and responses.

Practical 12: Form Handling using Servlet

Aim: To apply request handling, analyze form data processing, and develop dynamic web applications.

Description: Build a form and process user input using Servlet.

Practical 13: JSP-Based Dynamic Page

Aim: To understand JSP concepts, apply scripting elements, and create dynamic web content.

Description: Develop a JSP page for displaying dynamic data.

Unit 5: Enumerations, Assertions & Annotations

Practical 14: Enumeration and Wrapper Classes

Aim: To understand enums and wrappers, apply them in applications, and analyze type safety benefits.

Description: Implement enum-based program (e.g., days, roles).

Practical 15: Assertion and Design by Contract

Aim: To understand assertions, apply preconditions and postconditions, and evaluate program correctness.

Description: Use assertions in programs to validate logic.

Practical 16: Custom Annotations

Aim: To understand annotations, apply custom annotations, and analyze metadata usage.

Description: Create and use custom annotations in Java.

Unit 6: Advanced I/O and Try-With-Resources

Practical 17: Try-With-Resources Implementation

Aim: To understand resource management, apply automatic closing, and analyze exception safety.

Description: Rewrite file handling programs using try-with-resources.

Practical 18: Console I/O using PrintWriter

Aim: To apply console I/O operations, analyze output formatting, and develop efficient output systems.

Description: Implement formatted console output using PrintWriter.

Unit 7: Generics and Lambda Expressions

Practical 19: Generic Class and Method

Aim: To understand generics, apply type parameters, and analyze type safety.

Description: Create generic classes and methods.

Practical 20: Lambda Expressions Implementation

Aim: To understand functional programming, apply lambda expressions, and analyze concise coding techniques.

Description: Implement sorting, filtering using lambda expressions.

Practical 21: Functional Interfaces

Aim: To understand functional interfaces, apply lambda integration, and create reusable logic.

Description: Use predefined interfaces like Predicate, Consumer.

Unit 8: Modern Java Features

Practical 22: Switch Expression and Pattern Matching

Aim: To understand modern Java enhancements, apply switch expressions, and analyze code readability improvements.

Description: Rewrite traditional switch using modern syntax.

Practical 23: Sealed Classes Implementation

Aim: To understand restricted inheritance, apply sealed classes, and analyze controlled hierarchy design.

Description: Design class hierarchy using sealed classes.

Implement sorting, filtering using lambda expressions.

Mini Project

Practical 24: Full Stack Java Application (GUI + File + Collections + Lambda)

Aim: To integrate all learned concepts, analyze system design, evaluate implementation choices, and create a complete Java application.

Description: Develop a **Student Management System / Inventory System** using:

- GUI (Swing)
- File handling
- Collections
- Lambda expressions

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT106
Subject Name: Software Modelling and Analysis

Programme: B. Tech	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: NIL		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): CIT108
Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Identify and explain contemporary software life cycle processes, activities, and work products.
2	Elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.
3	Demonstrate formal correctness of simple procedure.
4	Implement sequential software systems based on formal models.
5	Verify attributes of formal models.
6	Describe the costs and benefits of formal methods.

Contents

Part- A

Unit-1 Software Engineering Fundamentals **12(L) hrs**

Overview of .NET platform and its evolution, .NET architecture including Common Language Runtime (CLR), Common Type System (CTS), and Common Language Specification (CLS), concept of managed and unmanaged code, Just-In-Time (JIT) compiler, assemblies and metadata, execution process of .NET applications, comparison of WinForms and WPF (Windows Presentation Foundation), installation and working of Visual Studio IDE, project templates, debugging environment.

Unit-2 Modeling foundations **10(L) hrs**

Modeling principles (e.g., decomposition, abstraction, generalization, projection/views, and use of formal approaches), Preconditions, post conditions, invariants, and design by contract, Introduction to mathematical models and formal notation.

Part- B

Unit-3 Types of models **13(L) hrs**

Information modeling (e.g., entity-relationship modeling and class diagrams), Behavioral modeling

(e.g., state diagrams, use case analysis, interaction diagrams, failure modes and effects analysis, and fault tree analysis), Architectural modeling (e.g., architectural patterns and component diagrams), Domain modeling (e.g., domain engineering approaches) , Enterprise modeling (e.g., business processes, organizations, goals, and workflow), Modeling embedded systems (e.g., real-time schedule analysis, and interface protocols.

Unit-4 Analysis fundamentals **10(L) hrs**

Analyzing form (e.g., completeness, consistency, and robustness), Analyzing correctness (e.g., static analysis, simulation, and model checking), Analyzing dependability (e.g., failure mode analysis and fault trees), Formal analysis (e.g., theorem proving).

Tutorial hours will be used for practice sessions for design/numerical problems/programming/case- studies etc. (as the case may be).

Laboratory Work

Experiment No.	Experiment Title
1	Identify a software system that needs to be developed and prepare a software requirement specification for that system.
2	Identify use cases and develop the use-case model for the system.
3	Identify the classes and develop the class diagrams for the system.
4	Find the interactions between the objects and represent those using UML sequence and collaboration diagrams.
5	Draw the relevant state chart diagrams and activity diagrams for the system.
6	Implement the system as per the detailed design by using an object-oriented programming language.
7	Test all the scenario is developed as per the use-case diagram.
8	Perform failure modes and effect analysis and fault-tree analysis of the system.
9	Check the model and improve its reusability and maintainability.
10	Implement the modified system and test it for various scenarios.

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same.

Note: It is recommended that mini project allocation to students be done within two-three weeks of the start of the semester. This is only the suggested list of Practical's. Instructor may also frame additional Practical's relevant to the course contents (if required).

Text Books:

1. Roger S. Pressman, Bruce R. Maxim, *"Software Engineering: A Practitioner's Approach"*, 9th Edition, McGraw Hill Education, 2019.
2. Ian Sommerville, *"Software Engineering"*, 10th Edition, Pearson India, 2018.
3. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, *"Object-Oriented Analysis and Design with Applications"*, 3rd Edition, Pearson India, 2015.
4. James Rumbaugh, Ivar Jacobson, Grady Booch, *"The Unified Modeling Language Reference Manual"*, 2nd Edition, Pearson, 2004.
5. Rajesh Gupta, Paul Le Guernic, Sandeep Kumar Shukla, Jean-Pierre Talpin, *"Formal Methods and Models for System Design: A System Level Perspective"*, Springer, 2004.

Reference Books:

1. Mehdi Khosrow-Pour, *"Systems and Software Development, Modeling, and Analysis: New Perspectives and Methodologies"*, IGI Global, 2014.
2. Donald W. Boyd, *"Systems Analysis and Modeling: A Macro to Micro Approach with Multidisciplinary Applications"*, Academic Press, 2001.
3. Hassan Gomaa, *"Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures"*, Cambridge University Press, 2011.
4. Ned Kock, *"Systems Analysis & Design Fundamentals: A Business Process Redesign Approach"*, SAGE Publications, 2006.
5. Bernd Bruegge, Allen H. Dutoit, *"Object-Oriented Software Engineering Using UML, Patterns, and Java"*, 3rd Edition, Pearson, 2010.

Course Link (NPTEL)

1. https://onlinecourses.nptel.ac.in/noc25_cs108/preview?utm_source=chatgpt.com
2. https://onlinecourses-archive.nptel.ac.in/noc17_cs25/?utm_source=chatgpt.com
3. https://swayam.gov.in/explorer?searchText=software+engineering&utm_source=chatgpt.com

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Subject Code: EIT107

Course Title: Business Intelligence and Data Analytics

Programme: B. Tech	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L)+ 30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 20%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): ESC103, ESC105**,** BSIT101, CIT105, CIT108

Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Explain the concepts, evolution, and role of Business Intelligence and Data Warehousing in organizational decision-making.
2	Analyze business information requirements and identify appropriate data marts and enterprise data warehouse structures.
3	Design dimensional models (star, snowflake, fact constellation) to support business reporting and OLAP analysis.
4	Interpret data mining outputs using association, classification, and clustering techniques for business decision support.
5	Develop BI performance frameworks using KPIs, dashboards, and Balanced Scorecards for organizational performance evaluation
6	Assess ethical, legal, social, and governance issues associated with Business Intelligence systems in real-world scenarios.

Contents

Part-A

Unit-1 Business Intelligence Fundamentals 8 (L) hrs

Concept and evolution of Business Intelligence (BI), Business Intelligence vs Management Information Systems (MIS), Business Intelligence vs Decision Support Systems (DSS), Business Intelligence vs Data Science vs Big Data (only conceptual boundaries), Role of BI in strategic, tactical and operational decision making, BI value chain and information flow in organizations, BI maturity models, BI in functional areas: finance, marketing, Human Resources (HR), operations (decision perspective only).

Unit-2 Enterprise Data Warehousing for BI 8 (L) hrs

Concept of enterprise data warehouse, Data warehouses vs operational databases, Characteristics of data warehouses: subject-oriented, integrated, time-variant, non-volatile; Data granularity and aggregation levels, Data marts: dependent vs independent data marts; Top-down vs bottom-up BI architecture, Enterprise integration of data warehousing with Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Knowledge Management (KM) systems, Metadata management and its role in BI.

Unit-3Dimensional Modeling for Business Reporting 7 (L) hrs

Dimensional modeling concepts for BI reporting, Facts, dimensions and measures, Slowly changing dimensions (Type-1, Type-2, Type-3 – conceptual), Star schema, snowflake schema, fact constellation schema, Schema comparison from a business reporting perspective, Design considerations for management dashboards and reports.

Part-B

Unit-4 Business-Oriented Data Mining Concepts 7 (L) hrs

Purpose of data mining in business intelligence, Data mining vs data science vs analytics (conceptual), Knowledge Discovery in Databases (KDD) – managerial overview, Association rule mining for market basket analysis: Support, confidence, lift, Business interpretation of rules, Apriori algorithm (conceptual understanding only), Classification and clustering – interpretation of results for business decisions.

Unit-5 BI Architecture, Governance & Performance Management 7 (L) hrs

Business Intelligence system architecture, BI user roles: executives, managers, analysts, Key Performance Indicators (KPIs) and performance scorecards, Balanced Scorecard framework, BI governance models, BI project justification and cost–benefit analysis, Security, privacy and access control in BI systems, Limitations of traditional BI systems.

Unit-6 BI Applications, Case Studies & Decision Support 8 (L) hrs

Decision Support Systems (DSS) and Executive Information Systems (EIS). Collaborative decision making in organizations. Marketing intelligence: Customer profiling, Sales performance analysis; Operations & logistics intelligence: Supply chain visibility, Inventory and demand planning (conceptual); Industry case studies: Customer retention strategies, Cross-selling and up-selling decisions, Operational efficiency improvement; Ethical, legal and social issues in business intelligence, Future scope of BI in organizations (non-technical trends).

Laboratory Work

Experiment No.	Experiment Title
1	Introduction to Business Intelligence tools and study of BI workflow and user roles.
2	Identification and analysis of enterprise data sources and mapping transactional data to analytical requirements.
3	Design of a Business Intelligence value chain by identifying data producers, data consumers, and decision-makers.
4	Conceptual design of an enterprise data warehouse by identifying subject areas, business processes, and data marts.
5	Identification of facts and dimensions and design of a Star Schema for a given business scenario.
6	Design and comparison of Snowflake Schema and Fact Constellation (Galaxy) Schema for analytical reporting.
7	Simulation and analysis of OLAP operations such as roll-up, drill-down, slice, dice, and pivot using spreadsheet or BI tools.
8	Identification and documentation of Key Performance Indicators (KPIs) and performance metrics for different business domains.
9	Design of a business intelligence dashboard layout highlighting KPIs, visual components, and decision support features
10	Design and implementation of a Balanced Scorecard aligned with organizational strategy and performance objectives.
11	Business interpretation of association rule mining by computing support, confidence, and lift for transactional data.

12	Case study analysis on BI governance, security, privacy, and access control issues with suggested improvements.
13	Cost–benefit analysis and justification of a Business Intelligence project including ROI evaluation.
14	Analysis of industry-based BI case studies focusing on customer retention, cross-selling, and operational efficiency.

Mini project: Student must design an end-to-end Business Intelligence solution, including data warehouse design, KPIs, dashboards, a governance model, and decision-making outcomes.

Text Books

1. Ramesh Sharda, Dursun Delen, and Efraim Turban, Business Intelligence and Analytics: Systems for Decision Support, 10th Edition, Pearson Education, 2015.
2. Alex Berson and Stephen J. Smith, Data Warehousing, Data Mining, and OLAP, McGraw-Hill Education, 1997.

Reference Books

1. Ralph Kimball and Margy Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition, Wiley India, 2013.
2. Thomas H. Davenport and Jeanne G. Harris, Competing on Analytics: The New Science of Winning, Harvard Business Review Press, 2017.
3. Alejandro Vaisman and Esteban Zimányi, Data Warehouse Systems: Design and Implementation, Springer Nature, 2022.

Online Courses and Video Lectures: SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Data Mining for Decision Making	Prof. Varun Dutt	IIT Mandi	https://onlinecourses.nptel.ac.in/noc26_cs14/preview/
2	Introduction to Data Warehousing and Data Mining	Dr. Vaibhav Panwar	Manipal University, Jaipur	https://onlinecourses.swayam2.ac.in/imb26_mg92/preview/
3	Business Intelligence & Analytics	Prof. Saji K Mathew	IIT Madras	https://onlinecourses.nptel.ac.in/noc26_cs64/preview/

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT108
Subject Name: Digital Image Processing

Programme: B. Tech.	**L:** 3 **T:** 0 **P:** 2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45 (L) +30 (P)= 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30 %		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites (if any): CIT105, CIT106
Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1	Apply techniques and principles of image formation, sampling, quantization, spatial and frequency domain which will allow them to investigate specific image processing techniques.
2	Identify, formulate, and design algorithms in the area of digital image processing.
3	Use the techniques, skills, and modern engineering tools such as MATLAB and OCTAVE for processing digital images.
4	Function on multi-disciplinary teams through experiments and projects related to imaging techniques.
5	Identify potential applications of image processing to advancement of knowledge in sciences and engineering with benefits in, e.g., policing, public safety, and social issues such as privacy.
6	Demonstrate a high level of self-directed learning ability, good oral and written communication skills on technical topics of digital image processing.

Contents

Part-A

Unit-1 Introduction **6(L)hrs**

Fundamental steps in digital image processing, Components of an image processing system, Applications of image processing, Sampling, Quantization.

Unit-2 Digital Image Processing Operations **8(L)hrs**

Sequential data modeling, Recurrent Neural Networks (RNNs), Backpropagation Through Time (BPTT), Challenges of Training Recurrent Network, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Use cases in text analytics and temporal data modeling.

Unit-3 Image Enhancement in Spatial Domain **7(L)hrs**

Image enhancement point operations: Linear and non-linear functions, Piecewise linear functions, Histogram processing. Spatial filtering - basics of filtering in the spatial domain, Smoothing linear and non-linear filters, sharpening filters.

Part-B

Unit-4 Image Enhancement in Frequency Domain **5(L)hrs**

Basics of filtering in the frequency domain, Image smoothing and sharpening using frequency domain filters.

Unit-5 Image Compression **8(L)hrs**

Image compression model, Compression measures, Compression algorithm and its types (Entropy, Predictive, Transform and layered coding), Types of redundancy (Coding, Inter-pixel, Psycho-visual and Chromatic), Lossless compression algorithms – Run-length, Huffman, Bit-plane, Lossy compression algorithms – Lossy predictive, Block transform coding.

Unit-6 Image Segmentation **7(L)hrs**

Classification of image segmentation algorithms, Point, Line and Edge detection, Global thresholding, Otsu's method, Region-based segmentation.

Unit-7 Color Image Processing **4(L)hrs**

Color Image-Processing Fundamentals, RGB Models, HSI Models, Relationship between different models.

Laboratory Work

Experiment No.	Experiment Title
1	Introduction about the software tool.
2	Understanding the basic data types and their conversion from one to another.
3	Understanding arrays and vectors.
4	Learning to build functions and scripts.
5	Implementation of various flow control and decision statements.
6	Implementation of various arithmetic, logical, and geometrical operations.
7	Implementation of various image enhancement techniques in the spatial domain.
8	Implementation of various image enhancement techniques in the frequency domain.
9	Implementation of various image compression techniques.
10	Implementation of various image segmentation techniques.
11	Implementation of various colour models and conversion of an image from one model to another

Mini Project: By using various concepts of image processing, students are required to prepare a project in a group of two to three students. The group of students must submit a project report of 8 to 10 pages.

Text Books:

1. R.C.Gonzalez and R.E.Woods, "Digital Image Processing", Pearson Education, 2013.
2. S. Sridhar, "Digital Image Processing", Oxford University Press, 2011.

Reference Books:

1. M.Sonka, V.Hlavac and Roger Boyle, "Image Processing, Analysis and Machine Vision", Thomas Learning, 2007.
2. K. R. Castleman, "Digital Signal Processing", Pearson Education, 2007.

Online Courses and Video Lectures: SWAYAM Course

1. https://onlinecourses.nptel.ac.in/noc25_ee126/preview?utm_source=chatgpt.com
2. https://www.getyoureducation.net/course/digital-image processing?utm_source=chatgpt.com

Guru Nanak Dev Engineering College, Ludhiana

An Autonomous College under UGC Act 1956

B.Tech. 3rd Year (Information Technology)

Course Code: EIT109
Course Title: Quantum Computing

Programme: B. Tech	**L:**3 **T:** 0 **P:**2	**Credits:** 4
Semester: 5	**Theory/Practical:** Theory	**Teaching Hours:** 45(L) + 30(P) = 75 hrs
Total Max. Marks: 150	**Continuous Assessment (CA) Marks:** 90	**End Semester Examination (ESE) Marks:** 60
Minimum Percentage of Numerical / Design / Programming Problems in ESE: 40%		
Duration of End Semester Examination (ESE): 3 hours		
Course Type: Elective Course		

Prerequisites: BSIT101, ESC105

Additional Material Allowed in ESE: NIL

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
CO1	Apply the fundamentals of quantum computing, qubits, and quantum mechanics concepts relevant to IT applications
CO2	Apply linear algebra and quantum mechanics principles to represent quantum states and operations
CO3	Design and simulate quantum circuits using basic and multi-qubit quantum gates
CO4	Implement and analyze quantum algorithms such as Deutsch–Jozsa, Grover, and Shor using Python/Qiskit
CO5	Analyze the effect of measurement, decoherence, and quantum noise in quantum circuits
CO6	Evaluate emerging applications of quantum technologies, including quantum cryptography, quantum communication, and quantum machine learning,

Contents

Part-A

Unit-1 Introduction to Quantum Computing 7(L)hrs

Need for quantum computing, Classical vs quantum computation, Qubits and quantum information, Applications and limitations of quantum computing, Overview of quantum technologies.

Unit-2 Mathematical & Quantum Foundations 8(L)hrs

Linear algebra basics for quantum computing, Complex vector spaces, Dirac (bra–ket) notation, Quantum states and postulates, Superposition and measurement.

Unit-3 Qubits, Entanglement & Quantum Gates 8(L)hrs

Single-qubit systems, Bloch sphere, multi-qubit systems, Quantum entanglement, Quantum gates: Pauli, Hadamard, Phase, Controlled gates (CNOT, Toffoli).

Part-B

Unit-4 Quantum Circuits & Algorithms 8(L)hrs

Quantum circuits and universality, Deutsch–Jozsa algorithm, Bernstein–Vazirani algorithm, Quantum Fourier Transform, Grover's search algorithm, Concept of Shor's algorithm.

Unit-5 Quantum Error Correction & Fault Tolerance 7(L)hrs

Quantum errors: bit-flip, phase-flip, Basic quantum error correction codes: Shor code, steane code, Fault-tolerant computation principles, Threshold theorem and error rates, Error mitigation techniques for near-term devices.

Unit-6 Applications & Emerging Trends 7(L)hrs

Quantum cryptography, Quantum communication, Introduction to Quantum machine learning, Optimization problems, Overview of quantum hardware: superconducting, trapped ions, photonics Challenges and future directions.

Laboratory Work

Exp. No.	Experiment Title
1	To install and configure the quantum programming environment using Python and
2	To represent and analyze single qubit states using a quantum simulator
3	To demonstrate the principle of superposition of a single qubit
4	To measure the process of quantum states and observe probabilistic outcomes
5	To simulate and analyze basic quantum gates
6	To implement and study multi-qubit quantum systems using quantum simulators
7	To create and verify quantum entanglement using the CNOT gate
8	To design, visualize, and simulate simple quantum circuits
9	To implement and analyze the Deutsch–Jozsa algorithm for a simple input case
10	To implement and study Grover's search algorithm for a small database

Tools: Python (3.x), QisKit

Mini Project: Student has to do a project assigned from course contents in a group of students. They must submit a project report and give a presentation of the same.

Textbooks

1. Quantum Computing: An Applied Approach – Jack D. Hidary, 2nd Edition, Springer Cham (2021).
2. Quantum Computing: A Gentle Introduction – Eleanor G. Rieffel & Wolfgang H. Polak, 1st Edition, MIT Press (2011**).**
3. Quantum Information Science – Manenti R., Motta M., 1 st Edition, Oxford University Press (2023).
4. Quantum Computing &Techniques, Dr. R. Chopra, Khanna Publishing House (2025).
5. Learn Quantum Computing with Python and IBM Quantum Experience – Robert Loredo, 1st Edition, Packt Publishing Limited (2020).

Reference Books

1. Quantum computation and quantum information – Nielsen M. A., and Chuang I. L., 10th Anniversary edition, Cambridge University Press (2010).
2. A Pathak, Elements of Quantum Computation and Quantum Communication, Boca Raton, CRC Press (2015).
3. Quantum Computing Explained, David McMahon, Wiley (2008).
4. Machines That Think: The Future of Quantum Computing, V. Dunjko and H. J. Briegel,Cambridge University Press, 2020.
5. Quantum Computing Mechanics, A. B. Bhattacharya, Khanna Publishing House (2025).

Online Learning Materials

1. https://cepqip.iitd.ac.in/post/program/certification-in-quantum-computing-and-machine-learning-batch-07
2. https://www.youtube.com/watch?v=2SPjEA-4lKk&list=PLuBwWyD3M82x9PfxeF7oxb0E122mQAWh6

Supplementary SWAYAM Course

Sr. No.	Course Name	Instructor	Host Institute	URL
1	Introduction to Quantum Computing: Quantum Algorithms and Qiskit	Prof. Prabha Mandayam Prof. Anupama Ray Prof. Sheshashayee Raghunathan	IBM and IITM	https://nptel.ac.in/courses/106106232
2	Quantum Computing	Prof. Debabrata Goswami	IIT Kanpur	https://nptel.ac.in/courses/104104082
3	Quantum Computing: Algorithms and Limitations Through the Query Model	Prof. Rajat Mittal	IIT Kanpur	https://onlinecourses.nptel.ac.in/noc26_cs06/preview