



JECRCTM
UNIVERSITY
BUILD YOUR WORLD



Introduction to this document

This document covers the tentative program structure & syllabus for B.Tech. CSE Specialisation in Software Product Engineering.

Quick links

Here's a quick index for the reader of this document to jump to relevant sections, if you would like to.

[Introduction to this document](#)

[Quick links](#)

[Semester-wise focus of the program](#)

[Program structure](#)

[Syllabus](#)

[Semester 1](#)

[Computing Systems & Development Tools](#)

[Computational Thinking & Programming Fundamentals](#)

[Critical Thinking](#)

[Web Interfaces and Frontend Architecture](#)

[Professional Communication & AI Collaboration](#)

[Breadth of Computer Science – 1](#)

[Foundations of Generative AI](#)

[Semester 2](#)

[Learning How to Learn](#)

[Discrete Mathematics](#)

[Data Engineering for Product Development](#)

[Applied Full Stack Web Development](#)

[Indian Constitution](#)

[Capstone Project](#)

[Breadth of Computer Science – 2](#)

[Professional Skills for the Workplace](#)

[Semester 3](#)

[Environmental Sciences](#)
[Personal Leadership & Self-Management](#)
[Internship / Project – 1](#)
[Data Structures and Algorithms](#)
[Object-Oriented Programming](#)

[Semester 4](#)

[Database Management Systems](#)
[Organizational Behavior](#)
[Operating Systems](#)
[Machine Learning & Statistical Modeling](#)
[Internship / Project – 2](#)

[Semester 5](#)

[Universal Human Values](#)
[Computer Organization & Architecture](#)
[Design & Analysis of Algorithms](#)
[DevOps Foundations](#)
[Internship / Project – 3](#)

[Semester 6](#)

[Essence of Indian Knowledge Tradition](#)
[Computer Networks](#)
[System Design](#)
[Internship / Project – 4](#)
[Professional Elective #1](#)

[Semester 7](#)

[Business Thinking & Entrepreneurship](#)
[Professional Elective #2](#)
[Professional Elective #3](#)
[Internship / Project – 5](#)

[Semester 8](#)

[Open Elective](#)
[Internship / Project – 6](#)

Semester-wise focus of the program

Sem 1–2: Foundations for AI-Native Engineers

Programming fundamentals, web basics, databases, full stack, CS breadth, communication, and learning science.

Students learn to think computationally and collaborate with AI tools from early semesters. They get to build smart capstone projects by the end of semester 2.

Sem 3–4: Core Engineering + Applied AI

DBMS, OOP, OS, DSA, full-stack systems, and practical ML foundations, integrated with real-world projects.

Students build systems that incorporate AI components rather than treating AI as a separate subject.

Sem 5–6: Systems Thinking in an AI World

Computer architecture, networks, system design, DevOps, cloud, and specialization electives.

Students understand how scalable, reliable infrastructure supports AI workloads and data-intensive systems.

Sem 7–8: Advanced Engineering & AI-at-Scale

Cloud-native systems, distributed systems, MLOps/serverless, business context, and production-grade internships/projects.

Students operate as AI-native engineers capable of deploying, monitoring, and evolving intelligent systems in real environments.

Syllabus

Semester 1

Computing Systems & Development Tools

Introduction

This course introduces students to the foundational tools and environments required for modern software development. It builds fluency in Linux systems, version control workflows, and development environment configuration. Students move from basic command usage to confidently operating within real-world engineering setups. The focus is on precision, reproducibility, and disciplined development habits.

Course Outcomes

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

1. Explain the structure and organization of a Unix-based operating environment and file system.
2. Apply Linux terminal commands to navigate directories, manipulate files, manage permissions, and inspect processes.
3. Use Git for version control including staging, committing, branching, merging, and resolving conflicts.

4. Configure and operate a development environment using modern editors integrated with a Linux system.
5. Modify and execute an existing codebase by identifying configuration points and implementing controlled changes.

Syllabus

Unit 1: Linux Environment and File Systems

Linux architecture overview, terminal navigation, directory hierarchy, file manipulation commands, file permissions and ownership, process basics, command pipelines, grep and text search utilities.

Unit 2: Development Environment Setup

Editor configuration in Linux, VS Code integration with terminal, extensions and debugging tools, environment variables, package management basics, reproducible development setups.

Unit 3: Version Control with Git

Version control concepts, local repositories, staging and commits, branching strategies, merging and resolving conflicts, remote repositories, collaborative workflows, commit hygiene and documentation practices.

Unit 4: Working with Existing Codebases

Cloning repositories, dependency installation, project structure exploration, configuration management, log inspection, searching within codebases using grep and related tools, running applications locally.

Unit 5: Controlled Code Modification and Execution

Identifying configuration variables, modifying source files safely, validating changes through execution, debugging terminal errors, documenting changes, maintaining clean commit history.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. The Linux Command Line; William E. Shotts Jr.; No Starch Press; 2019
3. Pro Git; Scott Chacon and Ben Straub; Apress; 2014

Reference Books

1. How Linux Works; Brian Ward; No Starch Press; 2021
2. Version Control with Git; Jon Loeliger and Matthew McCullough; O'Reilly Media; 2012

Computational Thinking & Programming Fundamentals

Introduction

This course bridges the gap between abstract logic and executable code. Designed for students with no prior programming experience, it prioritizes the ability to read, trace, and debug logic over rote memorization of syntax. Students learn to decompose real-world problems into structured flowcharts and translate them into Python. The course emphasizes "Code Comprehension"—the skill of predicting how a machine processes data step-by-step.

Course Outcomes

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

1. Explain the principles of computational thinking and decompose complex problems into stepwise algorithms.
2. Represent logical solutions visually using flowcharts and strictly defined pseudocode.
3. Apply Python fundamentals (variables, lists, dictionaries, conditionals, loops) to model data and logic.
4. Visualize and trace the execution state of a program (memory and variables) through every step of a loop or function.
5. Diagnose errors, debug broken scripts, and verify the logic of code (including AI-generated snippets) for correctness.

Syllabus

Unit 1: Computational Thinking and Logical Structures

Problem decomposition, mapping real-world processes to algorithms, flowchart symbols and logic flow, pseudocode conventions, Logic Visualization (Input → Process → Output), and creating trace tables to track variable states manually.

Unit 2: Python Fundamentals

Python syntax as a logical language, Variables as data containers, primitive data types (Integers, Strings, Booleans), Lists and Dictionaries for structuring data, indexing and slicing, and basic Input/Output operations.

Unit 3: Control Flow and Iteration

Conditional logic (if/else), nested decision making, Loops (for, while) and loop control (break, continue), iteration over data sequences, and preventing infinite loops.

Unit 4: Functions and Modularity

Function definition and invocation, passing arguments and returning values, understanding Scope (Local vs. Global), Writing clean/reusable logic, and reading multi-function scripts to understand dependencies.

Unit 5: Debugging, Refactoring & AI Verification

Common errors (Syntax vs. Runtime vs. Logical), The Art of Debugging (interpreting error messages), using print-debugging and breakpoints, refactoring messy code for readability, and verifying code logic (explaining line-by-line what a given script does).

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Python Crash Course; Eric Matthes; No Starch Press; 2019
3. Automate the Boring Stuff with Python; Al Sweigart; No Starch Press; 2020

Reference Books

1. Think Python; Allen B. Downey; O'Reilly Media; 2015
2. Introduction to Computation and Programming Using Python; John V. Guttag; MIT Press; 2021

Critical Thinking

Introduction

This course develops disciplined reasoning, structured analysis, and awareness of cognitive biases that influence judgment. Students learn how decisions are shaped by mental shortcuts, assumptions, and social influences. The course equips learners to evaluate arguments, identify flawed reasoning, and communicate well-supported conclusions. The emphasis is on clarity of thought in academic, professional, and digital environments.

Course Outcomes

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

1. Explain the distinction between intuitive and analytical modes of thinking.
2. Identify common cognitive biases and heuristics in reasoning and decision-making contexts.
3. Apply structured reasoning frameworks such as the scientific method to evaluate claims.
4. Analyze media content and arguments for logical consistency and evidence quality.
5. Construct well-reasoned arguments supported by evidence while avoiding common reasoning errors.

Syllabus

Unit 1: Foundations of Human Reasoning

Intuitive and analytical thinking systems, cognitive ease, prediction mechanisms of the brain, structured problem-solving, scientific method fundamentals.

Unit 2: Cognitive Biases and Heuristics

Anchoring effect, availability heuristic, representativeness bias, confirmation bias, sunk cost fallacy, commitment bias, loss aversion.

Unit 3: Judgment and Decision-Making

Conjunction fallacy, survivorship bias, fundamental attribution error, framing effects, hindsight bias, illusion of validity, expert intuition versus structured analysis.

Unit 4: Media Literacy and Information Evaluation

Misinformation and disinformation, persuasion techniques, narrative framing, statistical misinterpretation, evaluating credibility of sources, structured critique exercises.

Unit 5: Applied Critical Thinking and Argumentation

Prospect theory fundamentals, endowment effect, structured debate, constructing evidence-backed arguments, reflective analysis exercises, simulated case evaluations.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Thinking, Fast and Slow; Daniel Kahneman; Penguin; 2012
3. The Art of Thinking Clearly; Rolf Dobelli; HarperCollins; 2013

Reference Books

1. Predictably Irrational; Dan Ariely; HarperCollins; 2010
2. Factfulness; Hans Rosling; Sceptre; 2018

Web Interfaces and Frontend Architecture

Introduction

This course introduces students to the architecture of modern web interfaces through a spiral, hands-on model. Students transition from visual structure (HTML/CSS) to strict logic using JavaScript/ TypeScript. The course prioritizes "Code Comprehension"—teaching students to read and debug code before writing complex logic from scratch—making it accessible to non-CS backgrounds. By the end, students understand how the DOM, styling, and typed logic interact as a coherent system.

Course Outcomes

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

1. Construct semantically structured web pages using HTML5 and style them using modern CSS (Flexbox/Grid).
2. Read, trace, and debug TypeScript code to understand execution flow and logic.

3. Apply JavaScript/ TypeScript fundamentals to write type-safe, error-free frontend logic.
4. Develop interactive interfaces by manipulating the DOM and handling events.
5. Build a library of reusable UI components based on props and state principles.

Syllabus

Unit 1: The Web Page as a Living System

Semantic HTML5 structure, document flow, block vs. inline elements, CSS fundamentals, the Box Model, linking scripts to HTML, script loading strategies, and using Browser Developer Tools to inspect elements and styles.

Unit 2: Layout Systems and Responsive Interfaces

CSS positioning strategies, Flexbox fundamentals (alignment, direction), CSS Grid basics, responsive design principles, media queries, relative units (rem/em/vh), and building responsive navigation bars and layouts.

Unit 3: Code Comprehension and JavaScript/ TypeScript Fundamentals

Transitioning from Logic to Code: Introduction to JavaScript/ TypeScript. The Art of Reading Code: Tracing execution flow, understanding variable scope, and predicting outputs. Debugging: Reading stack traces, handling Type Errors, using the VS Code Debugger. Fundamentals: Basic types, interfaces, functions, and control flow.

Unit 4: DOM Interaction and Asynchronous Logic

Selecting elements with TypeScript, event listeners (clicks, inputs), modifying DOM content dynamically, form handling, introduction to Asynchronous programming (Promises, Async/Await), and fetching data from public APIs using typed interfaces.

Unit 5: Building a Component Library

Component thinking: Breaking UIs into reusable blocks (Cards, Buttons, Modals). Passing data via function arguments (Props), managing local UI state (e.g., toggles, counters), code organization standards, and deploying a multi-page interactive portfolio.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. HTML and CSS: Design and Build Websites; Jon Duckett; Wiley; 2011
3. Learning TypeScript; Josh Goldberg; O'Reilly Media; 2022

Reference Books

1. You Don't Know JS Yet; Kyle Simpson; O'Reilly Media; 2020

2. Don't Make Me Think; Steve Krug; New Riders; 2014

Professional Communication & AI Collaboration

Introduction

This course develops professional communication capability for engineers operating in AI-augmented workplaces. It focuses on clarity, structured thinking, precision in writing, and effective collaboration with both humans and intelligent systems. Students learn to communicate technical ideas clearly, draft structured documents, use AI responsibly as a co-pilot, and participate confidently in professional discussions. The emphasis is on disciplined communication as an engineering skill.

Course Outcomes

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

1. Organize ideas into structured written documents suited for professional and technical contexts.
2. Apply clarity principles to eliminate ambiguity, verbosity, and cognitive overload in communication.
3. Draft analytical and persuasive technical write-ups for workplace scenarios.
4. Collaborate effectively with AI systems to enhance productivity while maintaining accuracy and integrity.
5. Deliver structured oral presentations and participate constructively in professional discussions.

Syllabus

Unit 1: Foundations of Professional Communication

Communication as structured thinking, audience awareness, curse of knowledge, clarity and brevity principles, answering to the point, eliminating ambiguity, logical structuring of ideas, introduction to analytical writing, tone and professional standards in workplace communication.

Unit 2: Analytical and Technical Writing

Writing executive summaries, technical explanations, structured problem statements, design trade-off articulation, writing decision memos, Amazon-style pre-read documents, organizing long-form content, coherence and flow, editing for clarity and precision.

Unit 3: AI as a Communication Co-Pilot

Principles of AI collaboration, prompting techniques, role-context-task-output framing, refining AI outputs critically, fact verification and hallucination awareness, iterative drafting using AI, ethical considerations in AI-assisted communication, responsible usage boundaries.

Unit 4: Professional Speaking and Structured Dialogue

Public speaking fundamentals, presentation structuring, slide discipline and narrative coherence, answering questions concisely, professional meeting participation, structured argument presentation, handling disagreement constructively, giving and receiving feedback.

Unit 5: Workplace Communication Scenarios

Resume clarity and defensibility, portfolio articulation, behavioural response structuring, communication in team settings, writing status updates, drafting professional emails, collaborative documentation practices, peer review exercises, simulation of workplace communication challenges.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. On Writing Well; William Zinsser; Harper Perennial; 2006
3. HBR Guide to Better Business Writing; Bryan A. Garner; Harvard Business Review Press; 2013

Reference Books

1. Thinking, Fast and Slow; Daniel Kahneman; Penguin; 2011
2. Made to Stick; Chip Heath and Dan Heath; Random House; 2007

Breadth of Computer Science – 1

Introduction

This course introduces students to the foundational principles that govern computing systems, from logic circuits to low-level program execution. It builds a structural understanding of how hardware and software interact, moving beyond surface-level programming to the mechanics of computation. The focus is conceptual clarity and system-level thinking rather than advanced optimization. By the end of the course, learners understand how instructions flow from logic to memory to execution.

Course Outcomes

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

1. Explain how Boolean logic forms the basis of digital computation.
2. Interpret number representations and arithmetic operations used in computer systems.
3. Describe how memory, registers, and instruction cycles enable program execution.
4. Analyze the structure of simple machine-level instructions and control flow mechanisms.
5. Relate low-level program execution concepts to higher-level programming abstractions.

Syllabus

Unit 1: Digital Logic Foundations

Boolean algebra fundamentals, logic gates and truth tables, Boolean expressions and simplification, combinational circuit concepts, multi-bit representations, introduction to hardware simulation principles, conceptual construction of basic arithmetic circuits.

Unit 2: Number Systems and Arithmetic Circuits

Binary, decimal and hexadecimal representations, signed and unsigned integers, two's complement representation, binary addition and subtraction, overflow concepts, adder circuits, introduction to arithmetic logic units, representation of data in memory.

Unit 3: Memory and Sequential Logic

Sequential circuits and flip-flops, registers and counters, hierarchical memory organization, RAM structure and addressing, instruction storage and retrieval, memory access patterns, conceptual understanding of stack and heap memory.

Unit 4: Instruction Execution and Machine Language

Machine language structure, instruction formats and encoding, addressing modes, control flow through branching, instruction execution cycle, interaction between CPU and memory, conceptual instruction decoding and execution tracing.

Unit 5: Assembly-Level Thinking and Hardware-Software Interface

Assembly language structure, symbolic representation of instructions, mapping high-level constructs to low-level operations, pointer concepts, basic input-output interaction models, translation of simple logic into assembly-level representations, conceptual design of simple computing systems.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Code: The Hidden Language of Computer Hardware and Software; Charles Petzold; Microsoft Press; 2022
3. Computer Organization and Design; David A. Patterson and John L. Hennessy; Morgan Kaufmann; 2017

Reference Books

1. Digital Design and Computer Architecture; David Harris and Sarah Harris; Morgan Kaufmann; 2012
2. Structured Computer Organization; Andrew S. Tanenbaum; Pearson; 2016

Foundations of Generative AI

Introduction

This course introduces the foundational concepts behind modern Generative AI systems, including large language models and transformer architectures. It focuses on conceptual understanding of how these systems are trained, how they generate responses, and how engineers can interact with them responsibly and effectively. Students explore core ideas such as tokens, embeddings, attention mechanisms, hallucination, and retrieval-augmented generation. Small implementation exercises reinforce conceptual clarity without moving into advanced model training.

Course Outcomes

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

1. Explain the core components and workflow of large language models.
2. Describe transformer-based attention mechanisms at a conceptual level.
3. Apply structured prompting techniques to guide AI-generated outputs.
4. Analyze limitations such as hallucination, bias, and context constraints in generative systems.
5. Implement simple API-based integrations to build AI-assisted applications.

Syllabus

Unit 1: Foundations of Modern AI Systems

Evolution of artificial intelligence, supervised and unsupervised learning overview, neural networks basics, emergence of large language models, tokens and embeddings, probabilistic text generation, model training vs inference, temperature and sampling concepts.

Unit 2: Transformers and Attention Mechanisms

Sequence modeling challenges, self-attention mechanism, multi-head attention overview, encoder-decoder architecture, contextual embeddings, positional encoding, conceptual overview of the transformer architecture, impact of attention-based models on NLP systems.

Unit 3: Prompting and AI Interaction

Principles of effective prompting, structured prompting approaches, role-context-task-output framing, iterative refinement, managing ambiguity, controlling response style and depth, prompt chaining concepts, context window management strategies.

Unit 4: Limitations, Bias, and Responsible AI Usage

Hallucination phenomena, bias in training data, sycophancy behavior, evaluation challenges, verification strategies, human-in-the-loop approaches, ethical and regulatory considerations, responsible AI deployment principles.

Unit 5: Applied Generative AI Engineering

Using AI APIs for text generation, structured output formatting, building simple

retrieval-augmented generation systems, embeddings and vector similarity search concepts, integration with external data sources, evaluation of generated responses, logging and monitoring AI outputs.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Artificial Intelligence: A Modern Approach; Stuart Russell and Peter Norvig; Pearson; 2020
3. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow; Aurélien Géron; O'Reilly Media; 2022

Reference Books

1. Deep Learning; Ian Goodfellow, Yoshua Bengio, Aaron Courville; MIT Press; 2016
2. Pattern Recognition and Machine Learning; Christopher M. Bishop; Springer; 2006

Semester 2

Learning How to Learn

Introduction

This course equips students with scientifically grounded learning strategies required for sustained success in an intensive engineering program. It integrates cognitive science, behavioral design, productivity systems, and AI-assisted learning methods. Students learn how memory works, how focus is built, and how to design their own learning systems. The emphasis is on deliberate practice, self-regulation, and long-term retention rather than short-term performance.

Course Outcomes

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

1. Explain core cognitive principles that influence memory, attention, and learning effectiveness.
2. Apply structured learning techniques such as spaced repetition, interleaving, and active recall.
3. Design a personalized learning workflow integrating note-taking, revision, and reflection systems.

4. Analyze personal productivity patterns and implement strategies to improve focus and consistency.
5. Use AI tools responsibly to enhance comprehension, revision, and skill acquisition.

Syllabus

Unit 1: The Science of Learning and Memory

Neuroplasticity basics, working memory and long-term memory, cognitive load theory, growth mindset principles, learning myths versus evidence-based strategies, desirable difficulty, Dunning-Kruger effect, metacognition and self-assessment.

Unit 2: Focus, Time, and Behavioral Systems

Goal setting frameworks, habit formation principles, procrastination psychology, deep work concepts, attention residue, digital distraction management, time-blocking systems, productivity blueprints, resilience in learning setbacks.

Unit 3: Cognitive Learning Strategies

Active recall, spaced repetition, interleaving, chunking, mental models, first principles thinking, mind mapping, structured note-taking systems, comprehension techniques for technical reading.

Unit 4: Complex Thinking and Retention Techniques

Mnemonics, visualization methods, memory palace concepts, abstraction and pattern recognition, applying mental models to engineering concepts, reflection journaling, feedback loops in skill development.

Unit 5: AI-Augmented Learning Systems

Using AI for explanation refinement, generating practice questions, AI for spaced revision, AI-assisted summarization and concept mapping, critical evaluation of AI outputs, designing a personal AI-enabled learning toolkit.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Make It Stick: The Science of Successful Learning; Peter C. Brown, Henry L. Roediger III, Mark A. McDaniel; Harvard University Press; 2014
3. Deep Work; Cal Newport; Grand Central Publishing; 2016

Reference Books

1. Thinking, Fast and Slow; Daniel Kahneman; Penguin; 2012
2. Atomic Habits; James Clear; Random House; 2018

Discrete Mathematics

Introduction

This course introduces the mathematical foundations essential for computer science, including logic, proofs, combinatorics, sets, relations, graphs, and probability. Students learn to reason formally, analyse discrete structures, and apply mathematical techniques to computational problems. The course builds problem-solving skills that support future subjects such as algorithms, data structures, and machine learning.

Course Outcomes

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

1. Explain fundamental concepts of logic and construct simple mathematical proofs.
2. Apply counting techniques, permutations, combinations, and recurrence relations to discrete problems.
3. Use set theory, relations, and functions to represent and analyse structured data.
4. Interpret basic graph and tree concepts relevant to computing applications.
5. Apply elementary probability concepts to analyse random events and simple stochastic processes.

Syllabus

Unit 1: Basic Logic and Proof Techniques

Overview of discrete mathematics, propositional and predicate logic, proof methods and strategies, mathematical induction and strong induction, disproof techniques, applications of formal reasoning in computing.

Unit 2: Counting and Recurrence Relations

Basic counting principles, permutations and combinations, binomial theorem and Pascal's triangle, recurrence relations, solving linear recurrences, recurrence applications in algorithm analysis such as Towers of Hanoi and Merge Sort.

Unit 3: Sets, Relations and Functions

Set notation and operations, types of sets, relations and their properties, partial orders, equivalence relations and partitions, functions and function composition, types of functions, applications in data modelling.

Unit 4: Discrete Structures: Graphs and Trees

Introduction to graphs, graph representations and properties, traversals and connectivity, introduction to trees, tree traversals, minimum spanning trees, Prim's and Kruskal's algorithms, paths and circuits, shortest-path concepts, introduction to asymptotic behaviour in discrete structures.

Unit 5: Probability and Random Processes

Basic probability concepts, conditional probability and Bayes' theorem, expectation and variance, standard deviation and deviations analysis, probability distributions, introduction to random variables, simple random walks and probabilistic reasoning in computing.

Text Books

1. Discrete Mathematics and Its Applications – Kenneth H. Rosen, McGraw-Hill, 8th Edition, 2019
2. Discrete Mathematics with Applications – Susanna S. Epp, Cengage Learning, 4th Edition, 2011

Reference Books

1. Concrete Mathematics – Ronald L. Graham, Donald E. Knuth, Oren Patashnik, Addison-Wesley, 2nd Edition, 1994
2. Elements of Discrete Mathematics – C.L. Liu, McGraw-Hill, 2nd Edition, 1985

Data Engineering for Product Development

Introduction

This course introduces students to structured data modeling and relational database engineering for production applications in the AI era. It bridges conceptual database design with practical implementation using PostgreSQL. Students learn to translate problem statements into normalized schemas, write complex queries, and understand how modern databases act as both transactional stores and long-term memory for AI systems (via Vector embeddings). The focus is on production-grade database thinking rather than theoretical research.

Course Outcomes

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

1. Translate real-world problem statements into structured ER models and normalized schemas.

2. Write SQL queries using joins, aggregations, and window functions to extract business insights.
3. Implement PostgreSQL advanced features, including JSONB for semi-structured data and pgvector for AI embeddings.
4. Integrate a relational database with a backend application using secure connection patterns.
5. Analyze query performance using execution plans and apply indexing strategies for optimization.

Syllabus

Unit 1: Data Modeling and ER Design

Entity-relationship modeling, translating use cases into schema structures, normalization principles (1NF to 3NF), primary and foreign keys, constraints, relational integrity, designing schemas for SaaS products.

Unit 2: SQL and Relational Querying

SELECT queries, filtering and ordering, multi-table joins and subqueries, aggregation functions, grouping and HAVING clauses, window functions basics, view creation, and transactional consistency principles.

Unit 3: PostgreSQL, JSONB & Vector Storage

Postgres architecture overview, handling semi-structured data with JSONB, indexing strategies (B-Tree, GIN), introduction to Vector Databases, using pgvector to store and query semantic embeddings, constraints, and triggers.

Unit 4: Application Integration

Connecting Node/TypeScript applications to Postgres, ORM concepts (Prisma/Drizzle), database migrations, seeding data for development, connection pooling strategies, and handling database errors in application logic.

Unit 5: Performance and Deployment

Query optimization principles, analyzing EXPLAIN ANALYZE execution plans, N+1 query problems, backup and restore strategies, deploying a managed database (e.g., Neon/Supabase), and production readiness checklists.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. SQL in 10 Minutes, Sams Teach Yourself; Ben Forta; Sams Publishing; 2020

3. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly Media; 2017

Reference Books

1. Database System Concepts; Abraham Silberschatz, Henry F. Korth, S. Sudarshan; McGraw Hill; 2019
2. PostgreSQL: Up and Running; Regina Obe and Leo Hsu; O'Reilly Media; 2017

Applied Full Stack Web Development

Introduction

This course develops the capability to design, build, secure, and deploy full-stack web applications using modern JavaScript/TypeScript frameworks. Students work on a structured boilerplate codebase to simulate industry "ramp-up," tracing data flow across layers. The course emphasizes architectural thinking, application security, environment management, and the integration of backend logic with frontend state.

Course Outcomes

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

1. Explain full-stack application architecture and trace end-to-end data flow across client, server, and database layers.
2. Develop modular frontend components with structured state management and routing using Next.js.
3. Implement backend API routes and business logic within a unified full-stack framework.
4. Secure applications using environment variable management, authentication, and input validation.
5. Deploy a functional, secure CRUD web application to a production cloud environment.

Syllabus

Unit 1: Full Stack Architecture and Data Flow

Client-server architecture, request-response lifecycle, RESTful principles, monorepo structure, tracing data flow from database to API to UI, reading and navigating an existing boilerplate repository, understanding separation of concerns, and debugging full-stack applications.

Unit 2: Frontend Architecture and State Management

React and Next.js fundamentals, component composition and hierarchy design, props and state lifecycle, client-side routing, controlled forms and validation, global state management (Context/Zustand), handling asynchronous UI updates, and skeleton loading patterns.

Unit 3: Backend API and Business Logic

API route creation, request validation (Zod), middleware concepts, structured error handling, CRUD design patterns, database interaction through ORM abstraction, authentication flow (NextAuth/Clerk), and role-based access control.

Unit 4: Database Integration and Application Security

Environment variable management (.env) and secrets security, connection pooling basics, preventing SQL injection via ORMs, secure session management, handling concurrent writes, and distinguishing between build-time and run-time configurations.

Unit 5: Deployment, Scalability and Production Readiness

Build pipelines, environment setup for deployment (Vercel/Render), containerization overview, logging strategies, rate limiting concepts, caching intuition, post-deployment validation, and performance auditing (Lighthouse).

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Learning React; Alex Banks and Eve Porcello; O'Reilly Media; 2020
3. Node.js Design Patterns; Mario Casciaro and Luciano Mammino; Packt Publishing; 2020

Reference Books

1. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly Media; 2017
2. Web Application Security; Andrew Hoffman; O'Reilly Media; 2020

Indian Constitution

Introduction

This course introduces the constitutional framework of India with a focus on its relevance to modern professionals. It builds awareness of rights, duties, governance structures, and the relationship between law, society, and technology. The emphasis is on constitutional literacy and responsible citizenship in a digital era.

Course Outcomes

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

1. Explain the foundational philosophy and structure of the Indian Constitution.
2. Interpret Fundamental Rights, Duties, and Directive Principles in contemporary contexts.
3. Describe the functioning of legislative, executive, and judicial institutions.

4. Examine constitutional principles in relation to technology and digital governance.
5. Apply constitutional values to case-based discussions involving public policy and engineering ethics.

Syllabus

Unit 1: Constitutional Foundations and Governance Structure

Historical background of the Constitution, Preamble and constitutional philosophy, Fundamental Rights and Duties, Directive Principles of State Policy, structure of Union and State governments, separation of powers, federalism, emergency provisions, amendment procedures, constitutional bodies and regulatory institutions.

Unit 2: Constitution in the Digital and Professional Context

Rule of law and accountability, digital privacy and data protection principles, intellectual property rights in India, freedom of speech and online platforms, constitutional values in technology design, ethical and legal responsibilities of engineers, case discussions on emerging technology governance.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Introduction to the Constitution of India; D. D. Basu; LexisNexis; 2015
3. Indian Constitutional Law; M. P. Jain; LexisNexis; 2014

Reference Books

1. The Constitution of India; P. M. Bakshi; Universal Law Publishing; 2021
2. Constitutional Law of India; V. N. Shukla; Eastern Book Company; 2019

Capstone Project

Introduction

This course enables learners to design, build, and deploy an end-to-end software system addressing a real user problem. The emphasis is on ownership, system thinking, integration of multiple technologies, and production deployment. Students work independently or in small teams under mentor review to ship a functional smart product.

Course Outcomes

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

1. Formulate a problem statement and design a structured software solution.
2. Architect a full-stack system integrating frontend, backend, and database layers.
3. Incorporate AI-driven or data-driven features into an application workflow.
4. Implement secure authentication, validation, and deployment practices.
5. Present and defend technical decisions with documented artefacts and live demonstrations.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Clean Architecture; Robert C. Martin; Pearson; 2017
3. The Lean Startup; Eric Ries; Crown Business; 2011

Reference Books

1. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly Media; 2017
2. Building Microservices; Sam Newman; O'Reilly Media; 2021

Breadth of Computer Science – 2

Introduction

This course builds conceptual intuition about how the modern internet and connected systems function. It focuses on mental models rather than implementation, helping students understand how computers communicate, how the web works, and how large-scale systems remain reliable and secure. The course lays the theoretical foundation for later subjects such as Computer Networks, DevOps, and Distributed Systems.

Course Outcomes

1. Explain how computers communicate over networks using client-server and packet-based models.
2. Interpret the lifecycle of a web request from browser to server and back.
3. Analyze layered system architectures and the role of abstraction in large-scale systems.
4. Examine virtualization and cloud computing concepts at a conceptual level.
5. Evaluate reliability and security mechanisms used in modern internet systems.

Syllabus

Unit 1: Foundations of Computer Networking

Evolution from standalone systems to networked systems, client-server and peer-to-peer

models, packets and switching concepts, IP addressing fundamentals, latency and bandwidth intuition, reliability challenges in communication, introduction to network topology concepts.

Unit 2: How the Web Works

Domain Name System concepts, structure of URLs, HTTP request-response lifecycle, statelessness and sessions, cookies and basic authentication concepts, browser-server interaction, content delivery basics, tracing a web request end-to-end.

Unit 3: Abstraction and System Architecture

Concept of abstraction in computing, layered architectural thinking, high-level overview of protocol layering, APIs as abstraction boundaries, separation of concerns, interaction between application, network, and hardware layers, introduction to distributed thinking at an intuitive level.

Unit 4: Virtualization and Cloud Foundations

Need for virtualization, virtual machines and hypervisors, containers at a conceptual level, infrastructure as a service model, scalability fundamentals, horizontal vs vertical scaling intuition, cloud service models overview, multi-tenant systems.

Unit 5: Reliability, Security, and Trust in Distributed Systems

Redundancy and fault tolerance basics, load balancing intuition, high availability concepts, encryption fundamentals and HTTPS overview, trust models on the internet, basic cybersecurity awareness for system design, case studies of system outages and failure scenarios.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Computer Networking: A Top-Down Approach; James F. Kurose and Keith W. Ross; Pearson; 2017
3. Computer Systems: A Programmer's Perspective; Randal E. Bryant and David R. O'Hallaron; Pearson; 2015

Reference Books

1. Computer Networks; Andrew S. Tanenbaum and David J. Wetherall; Pearson; 2011
2. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly Media; 2017

Professional Skills for the Workplace

Introduction

This course prepares learners for the professional expectations of modern engineering workplaces. It focuses on communication clarity, collaboration, accountability, and ethical conduct. The emphasis is on behavioral readiness for internships and early career roles.

Course Outcomes

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

1. Demonstrate structured and concise professional communication.
2. Participate effectively in team-based discussions and collaborative workflows.
3. Apply workplace ethics and accountability principles in simulated scenarios.
4. Manage feedback, conflict, and professional disagreements constructively.
5. Present ideas clearly in interview and workplace settings.

Syllabus

Unit 1: Professional Communication Foundations

Clarity and brevity in communication, structured email writing, business writing fundamentals, avoiding ambiguity, curse of knowledge concept, answering to the point, professional tone and etiquette.

Unit 2: Collaboration and Agile Practices

Introduction to Agile and Scrum roles, stand-ups and sprint reviews, collaborative documentation practices, structured feedback cycles, meeting preparation using written briefs, peer evaluation frameworks.

Unit 3: Interview and Career Readiness

Resume structuring principles, portfolio presentation strategies, behavioral interview frameworks, STAR method basics, mock interview simulations, articulation of technical decisions.

Unit 4: Workplace Conduct and Ethics

Professional accountability, workplace boundaries, managing conflict respectfully, ethical dilemmas in corporate settings, professional responsibility and ownership.

Unit 5: Personal Effectiveness and Growth

Time management basics, goal setting, adaptability in dynamic environments, reflective practice, professional identity building.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. HBR Guide to Better Business Writing; Bryan A. Garner; Harvard Business Review Press; 2013
3. Crucial Conversations; Kerry Patterson et al.; McGraw-Hill; 2012

Reference Books

1. The Pyramid Principle; Barbara Minto; Pearson; 2016
2. Mindset; Carol S. Dweck; Ballantine Books; 2016

Semester 3

Environmental Sciences

Introduction

This course introduces environmental systems, pollution, and sustainability from a scientific and engineering perspective. It helps students understand how technological growth impacts natural systems and how engineers can design responsible solutions. The focus is on systems thinking, environmental risk awareness, and sustainable practices in a technology-driven world.

Course Outcomes

1. Explain the structure and stability of major environmental systems and the causes of ecological imbalance.
2. Analyze different forms of pollution and their long-term environmental and health impacts.
3. Interpret waste and wastewater management systems from an engineering perspective.
4. Examine environmental risks associated with industrial and biomedical waste.
5. Evaluate sustainable practices and environmental responsibility in technology and engineering contexts.

Syllabus

Unit 1: Environmental Systems and Pollution Dynamics

Ecosystem stability and biogeochemical cycles, anthropogenic disruptions and climate change, air pollution sources and particulate matter, water pollution and toxic contaminants, persistent pollutants and bioaccumulation, invisible pollution challenges including microplastics and e-waste, environmental risk assessment in urban and industrial systems.

Unit 2: Waste Management, Engineering Solutions and Sustainability

Sewer systems and wastewater flow processes, wastewater treatment plant stages and

engineering principles, solid waste management lifecycle, biomedical and hazardous waste handling, sustainability frameworks in cities and industries, environmental compliance and regulatory awareness, role of engineers in sustainable design and pollution mitigation.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Environmental Studies; Anubha Kaushik and C.P. Kaushik; New Age International; 2018

Reference Books

1. Environmental Science; G. Tyler Miller and Scott Spoolman; Cengage Learning; 2019
2. Introduction to Environmental Engineering and Science; Gilbert M. Masters and Wendell P. Ela; Pearson; 2014

Personal Leadership & Self-Management

Introduction

This course develops structured approaches to personal discipline, execution consistency, and professional self-management. Students learn how to design practical systems for focus, productivity, decision-making, and accountability in high-performance environments. Emphasis is placed on building sustainable habits and operating effectively in an AI-enabled knowledge economy where execution, not information access, differentiates professionals.

Course Outcomes

1. Explain foundational principles of habit formation, discipline systems, and identity-based goal setting.
2. Apply structured techniques for time management, task prioritisation, and execution planning.
3. Analyze personal productivity patterns and design systems to improve focus and consistency.
4. Demonstrate professional accountability and ownership in simulated workplace scenarios.
5. Develop a structured personal performance framework aligned with long-term academic and career goals.

Syllabus

Unit 1: Foundations of Personal Discipline and Habit Systems

Identity-based habits, systems versus goals, habit stacking, keystone habits, the two-minute rule, “eat the frog” principle, building execution rituals, tracking consistency, overcoming procrastination using behavioural triggers.

Unit 2: Time Management and Execution Architecture

Task decomposition, prioritisation frameworks, calendar-based planning, time blocking, weekly reviews, lead and lag indicators, avoiding context switching, managing distractions in digital environments, building a personal execution dashboard.

Unit 3: Cognitive Load and Knowledge Management

Brain limitations and cognitive load theory, externalising memory systems, structured note-taking methods, active recall practices, interleaving and spaced repetition for skill mastery, reflective journaling, decision logs, avoiding information overload in the AI era.

Unit 4: Energy, Focus, and Performance Sustainability

Deep work principles, shallow versus meaningful work, environment design for productivity, energy cycles and rest discipline, managing stress under deadlines, anti-burnout routines, resilience and self-regulation techniques.

Unit 5: Ownership, Accountability, and Professional Credibility

Circle of control framework, radical responsibility, proactive communication, reliability standards, meeting discipline, feedback integration, ethical decision-making in teams, building a professional reputation through consistent behaviour.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Atomic Habits; James Clear; Random House; 2018
3. Deep Work; Cal Newport; Piatkus / Little, Brown; 2016

Reference Books

1. Getting Things Done; David Allen; Penguin Books; 2015
2. Mindset; Carol S. Dweck; Random House; 2017

Internship / Project – 1

Introduction

This course marks the transition from structured academic learning to real-world contribution. Students engage in internships, open-source development, simulated industry sprints, or AI-driven product building. The focus is on applying technical knowledge in live environments under real constraints.

Course Outcomes

1. Apply academic knowledge to real-world engineering tasks.
2. Demonstrate professional collaboration within team-based environments.
3. Analyze project requirements and translate them into structured execution plans.

4. Implement functional deliverables under real-world constraints and deadlines.
5. Evaluate personal performance and professional growth through reflective documentation.

Data Structures and Algorithms

Introduction

This course develops structured problem-solving skills using fundamental data structures and algorithmic techniques. It emphasizes complexity analysis, recursion, pattern-based problem solving, and efficient data handling. The goal is to build strong computational reasoning required for scalable systems and advanced computing.

Course Outcomes

1. Analyze time and space complexity of algorithms using asymptotic notation.
2. Apply recursion and sorting techniques to solve structured computational problems.
3. Implement core linear data structures and analyze their performance trade-offs.
4. Design solutions using trees, hash maps, and graph traversal strategies.
5. Evaluate algorithmic correctness and efficiency in real-world problem scenarios.

Syllabus

Unit 1: Foundations and Complexity Analysis

Mathematical foundations for algorithms, asymptotic notation, time and space complexity, revisiting arrays and strings, sieve techniques, computational trade-offs.

Unit 2: Recursion and Algorithm Patterns

Recursion fundamentals, divide-and-conquer strategies, merge sort and quick sort, sorting optimization, two-pointer technique, sliding window pattern, problem pattern recognition.

Unit 3: Linear Data Structures

Stacks, queues and deques, linked lists, pointer manipulation, cycle detection, infix-postfix-prefix conversions, practical applications of linear structures.

Unit 4: Trees and Hash Maps

Binary trees, traversal techniques, depth computation, tree transformations, hash maps and hashing strategies, designing efficient lookup structures.

Unit 5: Binary Search Trees and Graphs

BST properties and validation, graph representation methods, breadth-first and depth-first traversal, cycle detection in graphs, connectivity problems, dependency resolution models.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Introduction to Algorithms; Thomas H. Cormen et al.; MIT Press; 2009

Reference Books

1. Data Structures and Algorithms Made Easy; Narasimha Karumanchi; CareerMonk Publications; 2016
2. Algorithm Design; Jon Kleinberg and Éva Tardos; Pearson; 2013

Object-Oriented Programming

Introduction

This course introduces object-oriented thinking as a structured approach to designing scalable and maintainable software systems. Students learn how to model real-world domains using classes, objects, interfaces, and design principles. The emphasis is on clean architecture, modularity, and disciplined software construction.

Course Outcomes

1. Apply object-oriented principles to model real-world problems using classes and objects.
2. Implement abstraction, encapsulation, inheritance, and polymorphism in structured programs.
3. Analyze object relationships and system structure using UML class diagrams.
4. Apply SOLID principles to improve modularity and maintainability of software systems.
5. Design small-scale service-oriented architectures using object-oriented design techniques.

Syllabus

Unit 1: Foundations of Object-Oriented Thinking

Programming paradigms overview, procedural vs object-oriented design, Java programming fundamentals, control structures and collections review, classes and objects, member variables and methods, object lifecycle, arrays of objects, domain modelling basics.

Unit 2: Constructors, Encapsulation and Abstraction

Constructors and constructor overloading, static members, access modifiers, data hiding strategies, abstraction using abstract classes and interfaces, interface contracts, modelling constraints through object design.

Unit 3: Inheritance and Polymorphism

Single and multilevel inheritance, interfaces and multiple inheritance through interfaces, method

overriding and dynamic binding, method overloading, composition vs inheritance, UML modelling of hierarchies.

Unit 4: Advanced Object Design

Cohesion and coupling, dependency relationships, introduction to design patterns, layered architecture concepts, refactoring techniques, testable object design.

Unit 5: SOLID Principles and Modular Systems

Single Responsibility Principle, Open-Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, Dependency Inversion Principle, separation of concerns, service layer abstraction, modular system design case study.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Head First Java; Kathy Sierra, Bert Bates; O'Reilly Media; 2005
3. Clean Code; Robert C. Martin; Prentice Hall; 2008

Reference Books

1. Effective Java; Joshua Bloch; Addison-Wesley; 2018
2. Design Patterns: Elements of Reusable Object-Oriented Software; Erich Gamma et al.; Addison-Wesley; 1994

Semester 4

Database Management Systems

Introduction

This course builds strong theoretical and practical foundations in relational database systems. It covers data modeling, relational algebra, SQL, normalization, transactions, indexing, and database security. The emphasis is on data reliability, integrity, and performance in production-grade systems.

Course Outcomes

1. Explain the architecture and components of database management systems.
2. Design relational schemas using ER modeling and normalization principles.
3. Apply relational algebra and SQL to query and manipulate structured data.
4. Analyze transaction management, indexing, and query optimization techniques.
5. Evaluate database security, concurrency control, and recovery mechanisms.

Syllabus

Unit 1: Database Foundations and Architecture

DBMS concepts and purpose, three-schema architecture, data independence, database users and roles, historical evolution of databases, data models overview, relational model foundations.

Unit 2: Relational Modeling and Algebra

Entity-relationship modeling, keys and constraints, converting ER diagrams to relational schemas, Codd's rules, relational algebra operations, selection and projection, joins, aggregation, solved algebraic expressions.

Unit 3: SQL and Data Manipulation

DDL and DML commands, creating databases and tables, insertion and retrieval, filtering and sorting, aggregate functions, subqueries, joins, views, working with dates, real-world SQL case exercises.

Unit 4: Transactions, Indexing and Optimization

ACID properties, transaction control language, concurrency control basics, locking concepts, indexing using B-Trees, query optimization techniques, performance case studies.

Unit 5: Security, Normalization and Scalability

Functional dependencies, normalization forms (1NF to 5NF and BCNF), multivalued dependencies, role-based access control, database security fundamentals, backup and recovery, database scaling concepts.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Database System Concepts; Abraham Silberschatz, Henry Korth, S. Sudarshan; McGraw Hill; 2019

Reference Books

1. Fundamentals of Database Systems; Ramez Elmasri and Shamkant Navathe; Pearson; 2016
2. SQL Performance Explained; Markus Winand; Markus Winand Publications; 2014

Organizational Behavior

Introduction

This course examines how individuals and teams function within modern technology organizations. It equips students with structured understanding of motivation, leadership,

communication, culture, and decision-making in professional environments. The focus is on preparing students to operate effectively within complex organizational systems.

Course Outcomes

1. Explain individual behaviour and motivation theories in workplace contexts.
2. Analyze team dynamics and conflict situations in collaborative environments.
3. Apply leadership and communication strategies in structured work settings.
4. Examine organizational structures and their influence on performance and culture.
5. Evaluate ethical and professional responsibilities in organizational contexts.

Syllabus

Unit 1: Individual Behaviour and Motivation

Personality frameworks, perception and biases at work, motivation theories, emotional intelligence, accountability and ownership, professional identity formation.

Unit 2: Team Dynamics and Collaboration

Stages of team development, psychological safety, power and influence, conflict resolution strategies, feedback systems, stakeholder communication.

Unit 3: Leadership and Influence

Leadership styles, situational leadership, influence without authority, delegation practices, trust and credibility, managing performance.

Unit 4: Organizational Structure and Culture

Organizational design models, formal and informal systems, incentive structures, governance and compliance, remote and hybrid team structures.

Unit 5: Ethics and Professional Conduct

Workplace ethics, integrity under pressure, decision-making in ambiguous environments, professional boundaries, long-term career responsibility.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Organizational Behavior; Stephen P. Robbins, Timothy A. Judge; Pearson; 2019
3. Leaders Eat Last; Simon Sinek; Portfolio; 2014

Reference Books

1. Drive; Daniel H. Pink; Riverhead Books; 2011
2. The Five Dysfunctions of a Team; Patrick Lencioni; Jossey-Bass; 2002

Operating Systems

Introduction

This course develops a structured understanding of how operating systems manage computation, processes, memory, and storage. Students build conceptual clarity about concurrency, scheduling, and resource allocation. The emphasis is on analytical understanding supported by problem-solving.

Course Outcomes

1. Explain the structure and functions of operating systems.
2. Analyze process management and CPU scheduling mechanisms.
3. Apply synchronization techniques to solve coordination problems.
4. Examine memory management and virtual memory strategies.
5. Evaluate storage management and system-level resource allocation policies.

Syllabus

Unit 1: Operating System Fundamentals

Operating system functions, system calls, process lifecycle, context switching, interprocess communication, shared memory, message passing, remote procedure calls.

Unit 2: Threads and Scheduling

Threads and multithreading models, CPU scheduling algorithms, FCFS, SJF, priority scheduling, round robin scheduling, scheduling metrics, starvation and fairness.

Unit 3: Process Synchronization and Deadlocks

Critical section problem, hardware synchronization primitives, semaphores, monitors, classical synchronization problems, deadlock conditions, prevention and avoidance strategies.

Unit 4: Memory Management

Logical and physical memory, partitioning techniques, paging, segmentation, virtual memory concepts, page replacement algorithms, fragmentation.

Unit 5: Storage and Virtualization

File systems and allocation methods, crash recovery mechanisms, free space management, RAID concepts, virtual machines overview, system resource monitoring.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Operating System Concepts; Abraham Silberschatz, Peter B. Galvin, Greg Gagne; Wiley; 2018
3. Modern Operating Systems; Andrew S. Tanenbaum; Pearson; 2015

Reference Books

1. Understanding the Linux Kernel; Daniel P. Bovet, Marco Cesati; O'Reilly Media; 2005
2. Operating Systems: Three Easy Pieces; Remzi H. Arpaci-Dusseau; Arpaci-Dusseau Books; 2018

Machine Learning & Statistical Modeling

Introduction

This course introduces foundational machine learning concepts and statistical modelling techniques used in modern data-driven systems. Students learn how models are constructed, evaluated, and interpreted. The focus is on conceptual clarity and applied understanding rather than advanced mathematical derivation.

Course Outcomes

1. Explain foundational concepts in statistical learning and model building.
2. Apply regression and classification techniques to structured datasets.
3. Analyze model performance using appropriate evaluation metrics.
4. Examine unsupervised learning techniques and feature engineering methods.
5. Evaluate ethical considerations and limitations in machine learning systems.

Syllabus

Unit 1: Statistical Foundations

Probability basics, distributions, sampling concepts, mean and variance, bias-variance trade-off, hypothesis testing fundamentals.

Unit 2: Regression Models

Linear regression, logistic regression, loss functions, training workflow, regularization concepts, overfitting and underfitting.

Unit 3: Classification and Model Evaluation

Confusion matrix, precision and recall, ROC curves, cross-validation, train-test split strategies, model comparison techniques.

Unit 4: Unsupervised Learning

Clustering intuition, k-means algorithm, dimensionality reduction basics, feature engineering principles.

Unit 5: Responsible Machine Learning

Model interpretability, fairness and bias, ethical AI principles, deployment overview, applied case studies.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Introduction to Machine Learning with Python; Andreas C. Müller, Sarah Guido; O'Reilly Media; 2016
3. Practical Statistics for Data Scientists; Peter Bruce, Andrew Bruce; O'Reilly Media; 2020

Reference Books

1. Pattern Recognition and Machine Learning; Christopher M. Bishop; Springer; 2006
2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow; Aurélien Géron; O'Reilly Media; 2019

Internship / Project – 2

Introduction

This course strengthens the transition from structured learning to accountable contribution. Students engage in real-world engineering through internships, open-source contributions, or simulated industry projects. The emphasis is on collaboration, ownership, delivery discipline, and professional execution.

Course Outcomes

1. Apply technical knowledge to real-world engineering tasks.
2. Collaborate effectively within teams to deliver defined milestones.
3. Demonstrate professional communication and documentation standards.
4. Analyze requirements and implement structured technical solutions.
5. Evaluate project outcomes and reflect on performance improvements.

Semester 5

Universal Human Values

Introduction

This course examines human values in personal life, professional environments, and digital society. It connects ethical reasoning, workplace responsibility, social harmony, and sustainability with real-world case studies. Students engage in structured reflection, ethical dilemmas, and applied discussions to build clarity of thought and action.

Course Outcomes

1. Explain the role of universal human values in personal and professional life.
2. Analyze ethical dilemmas in technology, workplace, and societal contexts.
3. Apply value-based reasoning to real-world case studies and decision scenarios.
4. Examine the relationship between individual conduct, societal harmony, and sustainability.
5. Evaluate professional responsibility in digital and organizational environments.

Syllabus

Unit 1: Understanding Self and Values

Concept of universal human values, self-awareness and decision-making, clarity of purpose, happiness and success, harmony within the individual, reflective practices.

Unit 2: Relationships and Workplace Ethics

Trust and respect in professional relationships, accountability and ownership, integrity under pressure, digital professionalism, ethical communication, conflict resolution case studies.

Unit 3: Society and Technology Responsibility

Constitutional values and civic responsibility, social media ethics, misinformation and digital conduct, technology and social impact, inclusive design principles.

Unit 4: Work, Contribution and Leadership

Meaning of contribution, dignity of labour, responsibility in collaborative environments, ethical leadership, long-term reputation vs short-term gain, corporate case studies.

Unit 5: Sustainability and Global Responsibility

Environmental responsibility, sustainable development goals, ethical consumption, technology and sustainability trade-offs, responsible innovation.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Human Values and Professional Ethics; R. R. Gaur, R. Sangal, G. P. Bagaria; Excel Books; 2010
3. Justice: What's the Right Thing to Do?; Michael J. Sandel; Farrar, Straus and Giroux; 2009

Reference Books

1. The Difficulty of Being Good; Gurcharan Das; Penguin India; 2009
2. The Righteous Mind; Jonathan Haidt; Vintage; 2012

Computer Organization & Architecture

Introduction

This course develops a structured understanding of computer hardware organization and architectural design. Students analyze instruction execution, performance optimization, pipelining, memory hierarchy, and I/O systems. The focus is on performance reasoning and architectural trade-offs.

Course Outcomes

1. Explain system models and instruction set architecture principles.
2. Analyze instruction execution and control unit design mechanisms.
3. Examine pipelining techniques and hazard resolution strategies.
4. Apply number representation and arithmetic logic concepts in processor design.
5. Evaluate memory hierarchy and I/O organization for performance optimization.

Syllabus

Unit 1: System Organization and ISA

Von Neumann and Harvard models, Flynn's classification, RISC vs CISC design philosophies, instruction set architecture, registers and bus organization.

Unit 2: Instruction Handling and Control

Instruction cycle, instruction formats and addressing modes, control unit design, hardwired vs microprogrammed control, CPU performance metrics, Amdahl's Law.

Unit 3: Pipelining and Performance

Pipeline stages, structural hazards, data hazards, control hazards, hazard resolution techniques, pipeline optimization case studies.

Unit 4: Execution Unit Design

Number systems and 2's complement, floating point representation and IEEE 754, adder circuits, carry look-ahead adder, multiplication algorithms, arithmetic logic unit design.

Unit 5: Memory and I/O Organization

Cache memory mapping techniques, replacement policies, write policies, main memory organization, DMA, I/O techniques, system bus structure, performance modelling.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Computer Organization and Design; David A. Patterson, John L. Hennessy; Morgan Kaufmann; 2017
3. Computer Organization; Carl Hamacher, Zvonko Vranesic, Safwat Zaky; McGraw Hill; 2011

Reference Books

1. Structured Computer Organization; Andrew S. Tanenbaum; Pearson; 2016
2. Computer Architecture: A Quantitative Approach; John L. Hennessy, David A. Patterson; Morgan Kaufmann; 2019

Design & Analysis of Algorithms

Introduction

This course develops analytical understanding of algorithm design paradigms and computational complexity. Students learn structured approaches to solving optimization and decision problems using divide-and-conquer, greedy strategies, backtracking, and dynamic programming.

Course Outcomes

1. Analyze algorithm complexity using asymptotic notation.
2. Apply divide-and-conquer and greedy strategies to optimization problems.
3. Implement backtracking and string matching techniques for combinatorial problems.
4. Apply dynamic programming principles to structured optimization challenges.
5. Evaluate algorithmic trade-offs across different design paradigms.

Syllabus

Unit 1: Complexity and Recursion

Time and space complexity analysis, recurrence relations, recursion review, graph fundamentals, strongly connected components.

Unit 2: Divide and Conquer

Binary search, quick sort analysis, Strassen's matrix multiplication, skyline problem, peak element, optimal substructure identification.

Unit 3: Greedy Algorithms

Fractional knapsack, job sequencing with deadlines, minimum spanning trees, Prim's algorithm, Kruskal's algorithm, Dijkstra's algorithm.

Unit 4: Backtracking and String Matching

Hamiltonian cycle, graph coloring, N-Queens problem, Sudoku solver, subset sum problem, KMP algorithm, Rabin-Karp algorithm.

Unit 5: Dynamic Programming

Principles of dynamic programming, 0/1 knapsack, travelling salesperson problem, matrix chain multiplication, optimal binary search trees, Floyd-Warshall, Bellman-Ford.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Introduction to Algorithms; Thomas H. Cormen et al.; MIT Press; 2009
3. Algorithm Design; Jon Kleinberg, Éva Tardos; Pearson; 2013

Reference Books

1. The Algorithm Design Manual; Steven S. Skiena; Springer; 2020
2. Data Structures and Algorithm Analysis; Mark Allen Weiss; Pearson; 2014

DevOps Foundations

Introduction

This course introduces modern DevOps practices including automation, containerization, CI/CD, cloud deployment, and infrastructure as code. Students build reproducible systems and automate software delivery pipelines using industry-standard tools.

Course Outcomes

1. Apply version control and command-line automation techniques in collaborative environments.
2. Implement containerized applications using Docker and orchestration tools.
3. Design and configure CI/CD pipelines for automated build and deployment.
4. Deploy applications on cloud infrastructure using infrastructure-as-code principles.
5. Evaluate system reliability, scalability, and monitoring practices.

Syllabus

Unit 1: Command Line and Version Control

Linux terminal operations, bash scripting fundamentals, Git workflows, branching strategies, pull requests, collaborative repository management.

Unit 2: Containerization with Docker

Docker architecture, images and containers, Dockerfile creation, container lifecycle management, volumes and networking, Docker Compose.

Unit 3: CI/CD Automation

Continuous integration principles, Jenkins ecosystem, pipeline configuration, Jenkinsfile, credentials management, deployment automation.

Unit 4: Cloud Deployment

Cloud computing fundamentals, GCP infrastructure, virtual machines, cloud storage, container registry, Kubernetes introduction.

Unit 5: Infrastructure as Code and Orchestration

Terraform fundamentals, provisioning compute resources, Kubernetes architecture, deployment manifests, monitoring services, automated infrastructure management.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. The DevOps Handbook; Gene Kim et al.; IT Revolution Press; 2016
3. Docker Deep Dive; Nigel Poulton; Independently Published; 2023

Reference Books

1. Kubernetes Up & Running; Kelsey Hightower et al.; O'Reilly Media; 2022
2. Terraform: Up & Running; Yevgeniy Brikman; O'Reilly Media; 2022

Internship / Project – 3

Introduction

This course strengthens applied engineering capability through sustained industry engagement, open-source contribution, or high-impact simulated projects. Students are evaluated on execution quality, collaboration discipline, documentation standards, and measurable contribution.

Course Outcomes

1. Apply advanced technical knowledge in real-world engineering contexts.
2. Demonstrate ownership of deliverables in collaborative environments.
3. Analyze real-world requirements and translate them into technical solutions.
4. Evaluate performance feedback and iterate on professional output.
5. Produce documented artefacts that demonstrate engineering maturity.

Semester 6

Essence of Indian Knowledge Tradition

Introduction

This course introduces core ideas from Indian knowledge traditions and explores their relevance to modern technological life. It focuses on ethics, self-awareness, systems thinking, sustainability, and responsible action — helping students connect cultural wisdom with contemporary engineering practice.

Course Outcomes

1. Explain foundational concepts of Indian knowledge systems and their philosophical orientation.
2. Analyze principles of self, duty, and responsibility in personal and professional life.
3. Examine traditional approaches to sustainability and societal balance in relation to technology.
4. Apply ethical reasoning frameworks from Indian thought to contemporary engineering dilemmas.
5. Evaluate the relevance of indigenous knowledge systems in modern global and digital contexts.

Syllabus

Unit 1: Knowledge, Self, and Responsibility in a Technological Age

Concept of self and consciousness, Pramana and ways of knowing, Dharma as responsibility not dogma, individual and collective duty, ethics of action (Karma framework), decision-making through long-term consequence thinking, harmony between individual, society and nature, applying classical ethical reasoning to AI, automation and digital systems.

Unit 2: Systems Thinking, Sustainability and Civilizational Wisdom

Holistic worldview and interconnectedness, balance and restraint in resource usage, sustainable living principles in traditional knowledge, governance and social organization insights, conflict resolution frameworks, applying systems thinking to climate change, data economy and technological disruption, indigenous knowledge and global innovation dialogue.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Indian Knowledge Systems; B. Mahadevan, Vinayak Rajat Bhat, Nagendra Pavana R.; PHI Learning; 2022

Reference Books

1. Being Different; Rajiv Malhotra; HarperCollins India; 2011
2. The Difficulty of Being Good; Gurcharan Das; Penguin; 2012

Computer Networks

Introduction

This course develops a deep understanding of how modern communication networks function across layered architectures. Students examine protocol design, routing, congestion control, security, and real-world performance considerations, while implementing networked systems through socket programming and packet analysis.

Course Outcomes

1. Explain layered network models and the interaction between protocols across layers.
2. Analyze data transmission mechanisms including error handling, multiplexing, and congestion control.
3. Apply IP addressing and routing algorithms to structured network design problems.
4. Implement reliable communication systems using transport-layer protocols.
5. Evaluate network performance and security trade-offs in modern distributed environments.

Syllabus

Unit 1: Network Architecture and Protocol Models

Network types and topologies, switching techniques, hardware components, performance metrics including latency, throughput and jitter, internet structure and packet switching, OSI and TCP/IP models comparison, encapsulation and decapsulation, how web requests travel end-to-end.

Unit 2: Physical and Data Link Layer

Signal encoding and modulation techniques, transmission media, bandwidth and capacity, multiplexing methods, error detection and correction techniques, CRC and Hamming codes, MAC protocols, CSMA/CD and CSMA/CA, flow control strategies, switching fundamentals.

Unit 3: Network Layer

IPv4 and IPv6 structure, subnetting and CIDR, DHCP and NAT, ICMP operations, routing fundamentals, link-state and distance-vector routing algorithms, packet forwarding and fragmentation, router internals, congestion at network layer.

Unit 4: Transport Layer and Reliable Communication

Port addressing and multiplexing, UDP protocol mechanics, TCP connection management, sliding window protocol, congestion control algorithms, TCP flow control, reliability mechanisms, implementation of TCP and UDP clients and servers, chat application development.

Unit 5: Application Layer and Network Security

Client-server and peer-to-peer models, DNS deep dive, HTTP versions and request lifecycle, SMTP and IMAP basics, packet capture and traffic inspection, intrusion detection fundamentals, IDS signatures and rule configuration, introduction to network attack vectors and defense mechanisms.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Computer Networking: A Top-Down Approach; James F. Kurose, Keith W. Ross; Pearson; 2017

Reference Books

1. Data and Computer Communications; William Stallings; Pearson; 2013
2. TCP/IP Illustrated, Volume 1; W. Richard Stevens; Addison-Wesley; 2011

System Design

Introduction

This course trains students to architect scalable, reliable, and distributed systems capable of serving large-scale users. It develops structured thinking around trade-offs, bottlenecks, consistency models, and real-world infrastructure decisions.

Course Outcomes

1. Explain principles of scalable and distributed system architecture.
2. Analyze system trade-offs involving consistency, availability, and fault tolerance.
3. Design architectures incorporating caching, replication, load balancing, and asynchronous processing.
4. Evaluate system bottlenecks using performance and reliability metrics.
5. Develop structured architectural proposals for large-scale applications.

Syllabus

Unit 1: Foundations of Large-Scale System Architecture

Requirement gathering, capacity estimation, latency and throughput modeling, monolith vs microservices, stateless services, horizontal scaling, scalability bottlenecks.

Unit 2: Data Architecture and Consistency

Database scaling strategies, replication models, sharding techniques, CAP theorem, quorum systems, distributed caching, data partitioning strategies.

Unit 3: Reliability and Fault Tolerance

Redundancy strategies, health checks, circuit breakers, retry mechanisms, rate limiting, load balancing algorithms, graceful degradation patterns.

Unit 4: Asynchronous and Event-Driven Systems

Message queues, pub-sub systems, idempotency, distributed transactions, saga patterns, eventual consistency in practice.

Unit 5: Observability and Optimization

Logging strategies, monitoring metrics, tracing, performance benchmarking, system trade-offs, case studies of real-world system failures and redesigns.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly Media; 2017

Reference Books

1. System Design Interview; Alex Xu; Independently Published; 2020
2. Building Microservices; Sam Newman; O'Reilly Media; 2021

Internship / Project – 4

Introduction

In this course, students engage in advanced industry projects, high-impact internships, or production-grade simulated sprints. The focus shifts from task execution to system-level contribution, ownership, and measurable impact. Students demonstrate technical depth, professional maturity, and the ability to deliver deployable outcomes in real-world environments.

Course Outcomes

1. Apply advanced technical skills to solve real-world engineering problems in live or simulated industry settings.
2. Analyze system requirements and translate them into structured technical deliverables.
3. Demonstrate professional collaboration, communication, and ownership within a team environment.
4. Evaluate performance, scalability, and reliability aspects of implemented solutions.
5. Document and present project outcomes with technical clarity and measurable impact.

Professional Elective #1

Blockchain Technologies

Introduction

This course introduces blockchain as a distributed trust infrastructure and programmable transaction system. Students build conceptual clarity around decentralization, consensus, and smart contracts while understanding practical limitations. The focus is architectural intuition and real-world evaluation rather than heavy cryptography or deep protocol engineering.

Course Outcomes

1. Explain the architectural components of blockchain systems and decentralized networks.
2. Describe how transactions are validated and recorded using consensus mechanisms.
3. Apply basic smart contract concepts to model simple decentralized workflows.
4. Analyze common security and scalability challenges in blockchain platforms.
5. Evaluate real-world use cases and assess when blockchain is an appropriate solution.

Syllabus

Unit 1: Distributed Trust and Blockchain Fundamentals

Centralized vs decentralized architectures, peer-to-peer networks, trust models in distributed systems, structure of a block, hashing and immutability concept, public and private blockchains, permissioned vs permissionless networks, overview of Bitcoin and Ethereum ecosystems.

Unit 2: Consensus and Transaction Lifecycle

Need for consensus, Proof of Work and Proof of Stake intuition, validators and miners, transaction validation flow, mempool concept, forks and chain selection, digital signatures and wallets (conceptual overview), gas and transaction fees.

Unit 3: Smart Contracts and Tokens

Concept of programmable contracts, Ethereum Virtual Machine overview, structure of a simple smart contract, state and functions, tokens (ERC20 and NFTs – intuition), decentralized applications architecture, interaction through wallets.

Unit 4: Security, Limitations, and Governance

Common smart contract vulnerabilities (high-level), importance of auditing, scalability challenges, Layer 2 overview, governance models, DAOs, regulatory considerations, environmental concerns.

Unit 5: Blockchain Applications and Evaluation Framework

Use cases in finance, supply chain, identity, gaming, healthcare, evaluating hype vs utility, cost-benefit thinking, decision framework for when blockchain is appropriate, future directions of decentralized technologies.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Blockchain Basics: A Non-Technical Introduction in 25 Steps; Daniel Drescher; Apress; 2017

Reference Books

1. Mastering Ethereum; Andreas M. Antonopoulos, Gavin Wood; O'Reilly Media; 2018
2. Blockchain Revolution; Don Tapscott, Alex Tapscott; Portfolio; 2016

Cybersecurity Fundamentals, Exploits and Vulnerabilities

Introduction

This course builds practical security intuition for modern software systems. Students learn how breaches happen, why common vulnerabilities persist, and how engineers reduce risk through secure design, defensive coding, and basic operational controls. The course emphasizes responsible practice, real-world case analysis, and building secure-by-default habits for web, APIs, cloud, and endpoints.

Course Outcomes

1. Explain core cybersecurity concepts, threat models, and risk assessment in engineering contexts.
2. Identify common vulnerabilities in systems and applications using structured security thinking
3. Apply secure design and coding practices to reduce exploitability in web and API systems.
4. Analyze security incidents and propose preventive and response controls.
5. Demonstrate responsible security practices using basic tooling, logging, and governance frameworks.

Syllabus

Unit 1: Security Foundations and Risk Thinking

CIA triad, authentication vs authorization, threat modeling basics, assets and attack surfaces, vulnerability vs exploit vs risk, basic risk assessment, security mindset for engineers,

responsible disclosure and ethics, security failures through real breach narratives, introduction to governance and compliance vocabulary.

Unit 2: Systems and Network Security Fundamentals

Network basics for security, common attack patterns on networks, firewalls and access control concepts, secure configuration principles, OS hardening basics, least privilege, secrets management fundamentals, logging and monitoring basics, endpoint security concepts, packet inspection intuition, introduction to IDS/IPS concepts.

Unit 3: Web, Application, and API Security

Web request/response lifecycle, session fundamentals, OWASP Top 10 overview, injection fundamentals including SQL injection concept and prevention, XSS concept and prevention, CSRF concept and prevention, authentication pitfalls, secure password storage intuition, token-based auth basics, API security practices, input validation and output encoding principles, rate limiting and abuse prevention.

Unit 4: Cryptography Essentials for Engineers

Hashing vs encryption, symmetric and asymmetric intuition, key management basics, TLS basics and why HTTPS matters, certificates and trust intuition, secure storage of secrets, integrity checks, common crypto mistakes engineers make, practical “when to use what” guidance.

Unit 5: Incident Response, Forensics, and Secure Engineering Practice

Security incident lifecycle, detection vs response, triage and containment basics, incident reporting and documentation, log analysis intuition, basic forensics concepts, post-incident learnings and prevention, secure SDLC overview, security checklists in reviews, lightweight security policies for teams, security automation mindset.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Cybersecurity Essentials; Charles J. Brooks, Christopher Grow, Philip Craig; Sybex; 2018

Reference Books

1. Web Application Security; Andrew Hoffman; O’Reilly Media; 2020
2. The Web Application Hacker’s Handbook; Dafydd Stuttard, Marcus Pinto; Wiley; 2011

Autonomous AI Systems & Agents

Introduction

This course introduces how modern AI systems move beyond “single prompt → single output” into autonomous, tool-using workflows. Students learn how agentic systems are designed, how they plan and act, how they use memory and retrieval, and how engineers keep them safe, reliable, and evaluable. The focus is on engineering patterns, real workflows, and measurable system behavior rather than AI theory.

Course Outcomes

1. Explain the components and lifecycle of an autonomous AI system and an agent workflow.
2. Apply prompting and tool-use patterns to build reliable multi-step AI task pipelines.
3. Design retrieval and memory mechanisms to improve groundedness and continuity in agents.
4. Evaluate agent performance using structured test cases, metrics, and failure analysis.
5. Apply safety, reliability, and governance controls to reduce misuse and unpredictable behavior.

Syllabus

Unit 1: Foundations of Agentic Systems

From chatbots to agents, agent loop concepts (plan, act, observe, reflect), tasks vs workflows, tool-using systems, constraints and guardrails, role of prompts and instructions, structured prompting techniques, decomposition and planning intuition, agent architectures at a high level.

Unit 2: Tool Use, Orchestration, and Workflow Design

Tool calling patterns, function/tool selection logic, workflow orchestration concepts, chaining and routing, multi-agent vs single-agent systems, delegation and handoffs, state tracking and traceability, building reliable pipelines for common tasks (summarize, extract, validate, execute), error handling and retries.

Unit 3: Memory and Retrieval for Agents

Why agents hallucinate, grounding and context management, RAG intuition, embeddings and vector search basics, chunking and retrieval strategies, short-term vs long-term memory, memory write policies, freshness and relevance, citations and evidence patterns, retrieval failure cases.

Unit 4: Evaluation and Reliability Engineering for Agents

What “good” looks like for agents, test suites for workflows, golden datasets, prompt regression testing, deterministic vs probabilistic outputs, structured evaluation rubrics, latency and cost tradeoffs, observability and traces, failure mode analysis, reliability patterns for production.

Unit 5: Safety, Security, and Responsible Deployment

Prompt injection and data exfiltration risks, tool misuse risks, sandboxing tool access, permissions and scopes, safety filters and policy controls, privacy considerations, logging and

audit trails, human-in-the-loop approvals, responsible deployment checklists, governance and accountability.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. AI Engineering: Building Applications with Foundation Models; Chip Huyen; O'Reilly Media; 2025

Reference Books

1. Designing Machine Learning Systems; Chip Huyen; O'Reilly Media; 2022
2. Building LLM Powered Applications; Valentina Alto; O'Reilly Media; 2024

Semester 7

Business Thinking & Entrepreneurship

Introduction

This course equips engineers with structured business thinking to convert technical ideas into viable ventures. It focuses on value creation, customer discovery, financial logic, and execution strategy. Students learn how technology, markets, and operations interact in startup and enterprise environments.

Course Outcomes

1. Explain core concepts of value creation, markets, and business models in technology contexts.
2. Analyze customer problems and identify structured opportunity spaces.
3. Develop a business model and basic financial logic for a technology venture.
4. Evaluate operational, regulatory, and scaling risks in entrepreneurial settings.
5. Formulate a structured go-to-market plan for a product or service.

Syllabus

Unit 1: Foundations of Business and Markets

Value creation, demand and supply intuition, market structures, pricing basics, customer segments, problem-solution fit, introduction to business models.

Unit 2: Opportunity Discovery and Validation

Customer interviews, competitor mapping, value proposition design, market sizing basics, lean experimentation, product-market fit signals.

Unit 3: Business Models and Financial Thinking

Revenue models, cost structures, unit economics, contribution margin, customer acquisition cost, lifetime value, subscription and platform models.

Unit 4: Operations, Risk and Governance

Operational workflows, hiring structures, regulatory considerations, risk assessment, ethical responsibilities in scaling, sustainability of ventures.

Unit 5: Execution and Go-to-Market Strategy

Distribution channels, branding basics, growth strategies, pitch design, funding overview, intrapreneurship in large organizations.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. The Lean Startup; Eric Ries; Crown Business; 2011
3. Zero to One; Peter Thiel; Crown Business; 2014

Reference Books

1. Business Model Generation; Alexander Osterwalder; Wiley; 2010
2. The Mom Test; Rob Fitzpatrick; CreateSpace; 2013

Professional Elective #2

Cloud Computing

Introduction

This course builds foundational and applied understanding of cloud infrastructure, virtualization, and scalable deployment. Students learn how modern applications leverage distributed cloud resources for reliability and elasticity. The focus is architectural clarity rather than vendor memorization.

Course Outcomes

1. Explain core concepts of cloud computing, virtualization, and service models.
2. Analyze infrastructure requirements and map them to appropriate cloud services.
3. Configure and deploy scalable applications in a cloud environment.

4. Evaluate cost, reliability, and security trade-offs in cloud architectures.
5. Design a cloud-based solution for a real-world application scenario.

Syllabus

Unit 1: Foundations of Cloud Computing

Service models IaaS PaaS SaaS, public private hybrid cloud, virtualization basics, hypervisors, containers vs virtual machines.

Unit 2: Cloud Infrastructure Components

Compute services, storage systems, object storage, networking in cloud, load balancing, auto-scaling concepts.

Unit 3: Deployment and Scalability

Infrastructure provisioning, container deployment, managed databases, monitoring and logging, horizontal and vertical scaling.

Unit 4: Security and Cost Management

Identity and access management, shared responsibility model, cost optimization strategies, availability zones and disaster recovery.

Unit 5: Cloud Architecture Case Studies

Designing scalable web applications, microservices in cloud, serverless integration overview, performance and reliability trade-offs.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Cloud Computing: Concepts, Technology & Architecture; Thomas Erl; Pearson; 2013
3. Architecting the Cloud; Michael J. Kavis; Wiley; 2014

Reference Books

1. Cloud Native Patterns; Cornelia Davis; Manning; 2019
2. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly; 2017

Distributed System Design

Introduction

This course develops conceptual clarity about distributed systems that power modern internet-scale applications. Students understand coordination, consistency, and fault tolerance

across distributed nodes. Emphasis is on architectural reasoning rather than implementation detail.

Course Outcomes

1. Explain fundamental principles of distributed systems and coordination.
2. Analyze consistency models and their trade-offs.
3. Interpret the CAP theorem and fault tolerance mechanisms.
4. Evaluate distributed design patterns for scalability and reliability.
5. Design a high-level architecture for a distributed application scenario.

Syllabus

Unit 1: Introduction to Distributed Systems

Characteristics of distributed systems, latency, replication, partition tolerance, distributed communication.

Unit 2: Consistency and Coordination

Strong and eventual consistency, consensus basics, leader election concepts, quorum mechanisms.

Unit 3: Fault Tolerance and Reliability

Failure models, replication strategies, heartbeat protocols, recovery mechanisms.

Unit 4: Scalability Patterns

Sharding, caching, load balancing, distributed transactions, microservices architecture.

Unit 5: Case Studies in Distributed Design

Distributed databases, message queues, distributed file systems, real-world system architecture analysis.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly; 2017
3. Distributed Systems; Maarten van Steen; CreateSpace; 2017

Reference Books

1. Site Reliability Engineering; Betsy Beyer et al.; O'Reilly; 2016
2. Distributed Systems Concepts and Design; Coulouris et al.; Pearson; 2011

High-Performance Computing

Introduction

This course introduces computational performance optimization, parallelism, and large-scale computation. Students understand how hardware and algorithms interact to achieve speed and efficiency. The focus is architectural and performance reasoning.

Course Outcomes

1. Explain principles of parallelism and performance measurement.
2. Analyze algorithm performance under parallel execution models.
3. Evaluate memory hierarchy impact on performance.
4. Apply optimization strategies to computational workloads.
5. Interpret performance metrics in high-performance environments.

Syllabus

Unit 1: Performance Fundamentals

Performance metrics, speedup and efficiency, Amdahl's Law, benchmarking basics.

Unit 2: Parallel Computing Models

Shared memory vs distributed memory, SIMD MIMD, thread-level parallelism.

Unit 3: Memory Hierarchy and Optimization

Cache behavior, memory latency, locality principles, vectorization concepts.

Unit 4: Parallel Algorithms

Parallel sorting, reduction operations, matrix multiplication, workload partitioning.

Unit 5: Applications of HPC

Scientific computing, AI workloads, simulation systems, cloud-based HPC.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Introduction to Parallel Computing; Ananth Grama et al.; Pearson; 2003
3. High Performance Computing; Charles Severance; O'Reilly; 2017

Reference Books

1. Parallel Programming in C with MPI and OpenMP; Michael Quinn; McGraw Hill; 2003
2. Computer Architecture: A Quantitative Approach; Hennessy and Patterson; Morgan Kaufmann; 2017

Professional Elective #3

MLOps and LLMOps Foundations

Introduction

This course introduces the operational lifecycle of machine learning and large language model systems. Students learn how models move from experimentation to production, and how reliability, monitoring, and governance are maintained. The emphasis is on engineering discipline, reproducibility, and scalable AI deployment.

Course Outcomes

1. Explain the end-to-end lifecycle of machine learning and LLM systems in production.
2. Analyze model versioning, experiment tracking, and reproducibility workflows.
3. Apply containerization and orchestration principles to AI deployment pipelines.
4. Evaluate monitoring, drift detection, and model governance strategies.
5. Design a structured MLOps pipeline for a production-ready AI application.

Syllabus

Unit 1: Foundations of MLOps and LLMOps

ML lifecycle stages, experimentation to deployment, challenges in production AI, reproducibility, model artifacts, data pipelines.

Unit 2: Versioning and Pipeline Automation

Data versioning, model version control, experiment tracking systems, CI/CD for ML, workflow orchestration concepts.

Unit 3: Deployment Architectures for AI Systems

Batch vs real-time inference, containerization for ML workloads, scalable serving, API endpoints for models, prompt pipelines for LLMs.

Unit 4: Monitoring and Reliability

Model drift, data drift, performance monitoring, observability metrics, logging strategies, rollback mechanisms.

Unit 5: Governance, Security and Responsible AI

Model audit trails, compliance considerations, bias evaluation, prompt injection risks, human-in-the-loop systems, operational safeguards.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Machine Learning Engineering; Andriy Burkov; True Positive Inc.; 2020
3. Designing Machine Learning Systems; Chip Huyen; O'Reilly; 2022

Reference Books

1. Building Machine Learning Powered Applications; Emmanuel Ameisen; O'Reilly; 2020
2. Reliable Machine Learning; Cathy Chen et al.; O'Reilly; 2022

Serverless Computing

Introduction

This course develops conceptual and practical understanding of event-driven and serverless architectures. Students learn how cloud-native applications are built using functions, managed services, and scalable backends without direct server management. The focus is architectural reasoning and deployment models.

Course Outcomes

1. Explain principles of serverless and event-driven architectures.
2. Analyze trade-offs between traditional server-based and serverless systems.
3. Implement function-based services integrated with managed cloud components.
4. Evaluate performance, cost, and scalability considerations in serverless systems.
5. Design a serverless architecture for a modern web application.

Syllabus

Unit 1: Foundations of Serverless Architecture

Event-driven systems, functions as a service, stateless execution, managed infrastructure concepts.

Unit 2: Cloud Functions and Managed Services

Function triggers, API gateways, storage triggers, managed databases, authentication services.

Unit 3: Scalability and Cost Models

Auto-scaling mechanisms, cold starts, pricing models, performance optimization techniques.

Unit 4: Security and Observability

Access control, environment variables and secrets, monitoring and logging, debugging distributed functions.

Unit 5: Designing Serverless Applications

Microservices with serverless, integration with AI services, real-world design case studies, architectural trade-offs.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Serverless Architectures on AWS; Peter Sbarski; Manning; 2017
3. Cloud Native Development Patterns and Best Practices; John Gilbert; Packt; 2021

Reference Books

1. Cloud Native Patterns; Cornelia Davis; Manning; 2019
2. Designing Data-Intensive Applications; Martin Kleppmann; O'Reilly; 2017

Business Intelligence & Data Storytelling

Introduction

This course develops the ability to translate data into structured insights for business decision-making. Students learn data modeling, visualization principles, dashboard design, and narrative communication. The emphasis is on clarity, interpretation, and actionable intelligence.

Course Outcomes

1. Explain core principles of business intelligence and data-driven decision-making.
2. Analyze datasets to identify trends, patterns, and anomalies.
3. Develop dashboards and visualizations aligned to business objectives.
4. Interpret statistical summaries to support managerial decisions.
5. Construct a structured data narrative to communicate insights effectively.

Syllabus

Unit 1: Foundations of Business Intelligence

BI systems overview, data sources, ETL concepts, metrics and KPIs, descriptive analytics.

Unit 2: Data Modeling for Decision Support

Fact and dimension tables, star schema, aggregation concepts, data cleaning fundamentals.

Unit 3: Visualization and Dashboard Design

Principles of visual perception, chart selection, dashboard layout, storytelling through visuals, avoiding misleading graphics.

Unit 4: Analytical Interpretation

Trend analysis, cohort analysis, forecasting basics, variance interpretation, scenario modeling.

Unit 5: Communicating Insights

Data narratives, executive summaries, stakeholder communication, ethics in data representation, case-based presentation exercises.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Storytelling with Data; Cole Nussbaumer Knaflic; Wiley; 2015
3. Data Smart; John Foreman; Wiley; 2013

Reference Books

1. The Big Book of Dashboards; Steve Wexler et al.; Wiley; 2017
2. Naked Statistics; Charles Wheelan; W. W. Norton; 2013

Internship / Project – 5

Introduction

This course represents advanced industry immersion where students operate at near-professional standards. Learners engage in internships, open-source contributions, startup projects, or high-intensity simulated industry sprints. Emphasis is on ownership, impact, system thinking, and measurable contribution.

Course Outcomes

1. Apply advanced technical knowledge to solve real-world engineering problems.
2. Analyze project requirements and translate them into structured implementation plans.
3. Demonstrate professional collaboration, communication, and documentation practices.
4. Evaluate system performance, scalability, and reliability in deployed solutions.
5. Produce a documented, deployable project demonstrating production-level readiness.

Semester 8

Open Elective

Creative Writing with AI for Digital Media

Introduction

This course develops creative expression for digital platforms using human imagination supported by AI tools. Students learn narrative structure, persuasive copywriting, storytelling formats, and ethical AI-assisted content creation. The emphasis is on originality, clarity, and responsible creative collaboration with AI systems.

Course Outcomes

1. Explain core principles of storytelling and digital narrative structure.
2. Apply structured writing techniques for blogs, scripts, marketing copy, and short-form media.
3. Use AI tools responsibly to draft, refine, and enhance creative content.
4. Evaluate tone, audience alignment, and persuasive effectiveness in digital writing.
5. Produce a portfolio of AI-assisted creative works suitable for professional publication.

Syllabus

Unit 1: Foundations of Storytelling

Narrative arc, character and conflict, emotional hooks, audience awareness, writing clarity and brevity.

Unit 2: Digital Formats and Platform Writing

Blog writing, social media storytelling, script writing basics, newsletter formats, short-form and long-form content strategies.

Unit 3: Writing with AI as a Creative Partner

Prompting techniques for ideation, drafting and rewriting with AI, iterative refinement, tone transformation, avoiding over-reliance on automation.

Unit 4: Persuasion and Content Strategy

Copywriting principles, call-to-action design, brand voice, storytelling for marketing, content performance metrics.

Unit 5: Ethics, Authenticity and Portfolio Development

AI attribution and originality, plagiarism risks, misinformation risks, content responsibility, curated digital writing portfolio.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Everybody Writes; Ann Handley; Wiley; 2014
3. Steal Like an Artist; Austin Kleon; Workman Publishing; 2012

Reference Books

1. Made to Stick; Chip Heath and Dan Heath; Random House; 2007
2. Building a StoryBrand; Donald Miller; HarperCollins; 2017

Legal Literacy for Engineers

Introduction

This course equips engineers with foundational legal awareness relevant to technology, innovation, and digital systems. Students understand contracts, intellectual property, compliance, and regulatory frameworks that influence modern engineering practice. The focus is practical literacy rather than legal theory.

Course Outcomes

1. Explain key legal principles relevant to engineers and technology professionals.
2. Interpret intellectual property frameworks applicable to software and digital products.
3. Analyze legal risks in contracts, data usage, and digital platforms.
4. Apply regulatory compliance principles in technology development scenarios.
5. Evaluate ethical and legal implications of emerging technologies.

Syllabus

Unit 1: Foundations of Law for Engineers

Structure of Indian legal system, civil and criminal law basics, contracts and enforceability, liability principles.

Unit 2: Intellectual Property and Innovation

Copyright, patents, trademarks, open-source licensing, software licensing models, infringement risks.

Unit 3: Data Protection and Cyber Regulations

Data privacy principles, Indian data protection framework, global standards overview, compliance obligations for digital platforms.

Unit 4: Technology Contracts and Employment Law

Non-disclosure agreements, employment contracts, service agreements, freelancing and consulting considerations.

Unit 5: Law, Ethics and Emerging Technologies

AI regulation discussions, platform accountability, content liability, risk mitigation frameworks for engineers.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Cyber Law and IT Protection; Harish Chander; PHI Learning; 2019
3. Intellectual Property Rights; V. K. Ahuja; LexisNexis; 2017

Reference Books

1. The Indian Contract Act; Bare Act Edition; Universal Law Publishing; Latest Edition
2. Information Technology Act with Rules; Bare Act Edition; Universal Law Publishing; Latest Edition

Economics, Politics and Society

Introduction

This course builds practical understanding of how economic systems, political institutions, and global dynamics shape individual lives and technological ecosystems. Students analyze markets, governance, public finance, and international power structures. The course emphasizes real-world interpretation over abstract theory.

Course Outcomes

1. Explain core economic principles influencing prices, inflation, and public finance.
2. Analyze monetary systems, banking structures, and economic policy decisions.
3. Interpret global economic disparities and trade dynamics.
4. Examine Indian governance structures and constitutional power distribution.
5. Evaluate international political systems and geopolitical strategies.

Syllabus

Unit 1: Foundations of Economics and Public Finance

Demand and supply, price formation, inflation and recession, government revenue and expenditure, taxation principles, fiscal deficit.

Unit 2: Money, Markets and Economic Systems

Role of central banks, monetary policy, banking profit models, stock markets, currency exchange, capitalism vs welfare states.

Unit 3: Globalisation and International Economics

Trade and comparative advantage, resource paradox, multinational corporations, economic impact of war, global inequality case studies.

Unit 4: Indian Political and Governance Structures

Fundamental rights and duties, federalism, division of powers, electoral systems, policy implementation at state and local levels.

Unit 5: International Politics and Geopolitics

Political systems comparison, global power structures, diplomacy and alliances, nuclear deterrence, crisis negotiation simulations.

Text Books

1. Kalvium Livebook – Structured Course Material (Internal Publication)
2. Basic Economics; Thomas Sowell; Basic Books; 2015
3. Indian Polity; M. Laxmikanth; McGraw Hill; 2020

Reference Books

1. The Undercover Economist; Tim Harford; Oxford University Press; 2012
2. Prisoners of Geography; Tim Marshall; Elliott & Thompson; 2015

Internship / Project – 6

Introduction

This course represents the final professional immersion before graduation. Students operate at near-industry standards, contributing to production systems, research initiatives, startups, or advanced simulated work environments. The focus is autonomy, impact, systems thinking, and measurable outcomes.

Course Outcomes

1. Apply advanced domain knowledge to solve complex real-world engineering problems.
2. Analyze ambiguous project requirements and translate them into structured execution plans.
3. Demonstrate professional accountability, communication, and collaboration in team environments.
4. Evaluate system scalability, performance, and reliability in deployed solutions.
5. Deliver a production-ready solution with comprehensive documentation and measurable impact.