ENGINEERING ECONOMICS
ENCE 356

Lecture : 3 — **Year : III**
Tutorial : 1 — **Part : II**
Practical : 0

Course Objectives:
The objective of this course is to provide concept of economic principles and the economic environment at the project, firm, societal, and national levels, enabling them to analyze cause-and-effect relationships. The course aims to equip students with the ability to apply economic theories and tools for project selection, equipment replacement, property valuation, and price variation. By the end of the course, students will be able to evaluate alternatives and make informed, economically sound decisions in engineering and business contexts.

1 Introduction (2 hours)

1.1 Micro, macro and engineering economics (History, fundamental principle and application)
1.2 Terminology related to engineering economics
1.3 Economic decision and role of engineers in decision making
1.4 Cash flow and cash flow diagram

2 Market Economics (3 hours)

2.1 Market, demand, supply and relationship
2.2 Elasticity, application of elasticity and government policies
2.3 Externality and market inefficiency
2.4 Market failure and firm behavior

3 Cost (8 hours)

3.1 Cost classification
- 3.1.1 Total, average, fixed, variable, and marginal costs
- 3.1.2 Direct, indirect, and standard costs
- 3.1.3 Cash versus book cost, manufacturing and non-manufacturing cost
- 3.1.4 Sunk cost, opportunity cost, element of cost, life-cycle cost

3.2 Cost estimation and control

4 Time Value of Money (6 hours)

4.1 Money (Type, functions and time value of money)
4.2 Simple and compound interests (Nominal, effective and continuous compounding)

4.3 Economic equivalence
4.4 Cash flow types (Single, uniform, linear gradient, geometric gradient and irregular)

5 Methods of Economic Analysis (12 hours)

5.1 Capital budgeting
5.2 Minimum attractive rate of return (MARR)
5.3 Economic analysis of single and multiple projects
  5.3.1 Payback period (Simple and discounted)
  5.3.2 Equivalent worth (Net present, annual, future and capitalized)
  5.3.3 Rate of return (Internal and external)
  5.3.4 Public Sector economic analysis (Benefit cost analysis)
  5.3.5 Financial and economic analysis
5.4 Weighted average cost of capital (WACC)
5.5 Repeatability assumption and co-Terminated assumptions
5.6 Multiple investment project alternatives (Dependent, independent and contingent)

6 Replacement Analysis (5 hours)

6.1 Replacement strategies (Asset life and selection of challengers over defenders)
6.2 Economic service life of an asset
6.3 Replacement strategy for asset (Project with finite and infinite planning horizon)

7 Risk Analysis (5 hours)

7.1 Origin of risk in projects
7.2 Risk analysis of projects (Sensitivity, breakeven and scenario analyses)
7.3 Decision tree

8 Depreciation and Taxes (5 hours)

8.1 Concept and terminology
8.2 Depreciation calculation (Straight line, declining balance, sinking fund, sum of the year digit and modified accelerated cost recovery methods)
8.3 Tax and corporate income tax
8.4 Economic analysis (After-tax cash flow)

9 Measurement of Nation Income (5 hours)

9.1 Gross domestic product (Components, real and nominal gross domestic product)
9.2 Unemployment (Measurement, job search, minimum wage law)

9.3 Inflation
- 9.3.1 Causes, effects and measurement of inflation
- 9.3.2 Constant and current cash flow
- 9.3.3 Equivalence calculation under inflation
- 9.3.4 Inflation controlling measures

9.4 Real and nominal exchange rates, fiscal budget and monetary policy

9.5 Financial statement

Tutorial **(15 hours)**

1. Cash flow diagram construction, apply and analyze it for different projects
2. Problems related to market equilibrium, elasticity, and government policies
3. Exercises on calculating interests, and analyze cash flows with economic equivalence
4. Application of capital budgeting techniques for given cash flows
5. Selection of best replacement strategies and also, apply risk analysis for real case
6. Computation of after-Tax cash flow for economic analysis
7. Calculation and analysis of gross domestic product, performing equivalence calculations under different inflation scenarios, analyze real and nominal exchange rates
8. Calculation of price variation and price escalation
9. Summarize the fiscal budget

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Mark distribution*
1	2	3
2	3	5
3	2	3
4	6	8
5	12	16
6	5	6
7	5	6
8	5	8
9	5	5
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Mankiw, N. G. (2017). Principles of economics (8th ed.). Cengage Learning.
2. Park C.S. (2016). Contemporary Engineering Economics: Prentice Hall, Inc.
3. McConnell, C. R., Brue, S. L., Flynn, S. M. (2020). Economics: Principles, problems, and policies (22nd ed.). McGraw-Hill Education.

ARTIFICIAL INTELLIGENCE
ENCT 351

Lecture	**: 3**	**Year : III**
Tutorial	**: 1**	**Part : II**
Practical	**: 3**	

Course Objectives:

The objective of this course is to provide students with a foundation in Artificial Intelligence (AI), covering intelligent agents, search techniques, knowledge representation, machine learning, and AI ethics. It aims to equip students with both theoretical understanding and practical skills to apply AI techniques to real-world problems, while also developing awareness of the ethical and societal implications of AI systems.

1 Introduction (4 hours)

1.1 Definition, foundation, history of AI
1.2 Importance of knowledge and learning
1.3 Cognition and learning (Neuroscience)
1.4 Human intelligence and machine intelligence
1.5 AI tree: Branches and interdisciplinary nature
1.6 Intelligent agents and types
1.7 Agentic AI

2 Problem Solving and Search (9 hours)

2.1 Formal problem definition: States, actions, transitions, well-defined problems
2.2 Constraint satisfaction problems: Node consistency, path consistency, backtracking
2.3 Search algorithms, strategies and evaluations
2.4 Uninformed: BFS, DFS, iterative deepening
2.5 Informed Search: Best first search, greedy search, A* algorithm
2.6 Adversarial search: Minimax algorithm, alpha-beta pruning
2.7 Local search and optimization: Hill climbing, simulated annealing
2.8 Evolutionary optimization: Genetic algorithm

3 Knowledge Representation and Probabilistic Reasoning (7 hours)

3.1 Knowledge-based agent
3.2 Knowledge representation techniques and issues in representation
3.3 Propositional and predicate logic
3.4 Semantic networks, frames and knowledge graph
3.5 Review of Bayes' Theorem and probabilistic reasoning

3.6 Fuzzy logic: Membership functions, fuzzy inference systems

4 Machine Learning Fundamentals (10 hours)

4.1 Foundations and four pillars of machine learning
4.2 Review of mathematics for machine learning
4.3 Continuation optimization; Unconstraint optimization, constraint optimization, convex optimization
4.4 Review of machine learning pipeline
  4.4.1 Model development: Generative versus discriminative
  4.4.2 Learning algorithm
  4.4.3 Capacity, overfitting and underfitting
  4.4.4 Hyperparameters and validation sets
  4.4.5 Estimators, bias and variance
  4.4.6 Review of MLE and MAP and cross entropy
4.5 Supervised learning algorithm: Decision tree and support vector machine
4.6 Unsupervised learning algorithm: t-SNE
4.7 Semi supervised and reinforcement learning
4.8 Model evaluation: Energy-based indicators

5 Neural Networks and Deep Learning Algorithms (8 hours)

5.1 Neural networks: Structures, activation functions and universal approximation theorem
5.2 Perceptron, multilayer perceptron and backpropagation
5.3 Introduction to deep learning
5.4 Concepts on recurrent and generative neural networks

6 AI Applications (5 hours)

6.1 Expert systems: Characteristics, architecture, development and various applications
6.2 NLP: Level of analysis, challenges, modern approaches and applications
6.3 Robotics and computer vision: Fundamental, components and applications
6.4 Sustainable AI systems

7 Emerging Trends (2 hours)

7.1 Sequence to sequence models
7.2 Federated learning
7.3 Edge AI
7.4 Ethics and AI: Responsible AI

Tutorial (15 hours)

1. Design a simple agent model (Vacuum cleaner, chess player)
2. Formulate a route-finding problem using states, actions and transitions
3. Solve the problem with uninformed strategies

4. Apply the A* algorithm with the given heuristic values
5. Solve the minimax tree and demonstrate alpha–beta pruning
6. Solve a CSP like map coloring or mini-Sudoku
7. Perform one iteration of the genetic algorithm (Selection, crossover, mutation)
8. Represent facts using semantic networks and frames
9. Solving problems related to Bayes' theorem and Fuzzy Logic
10. Understanding machine learning basics through model training, cross-validation and performance evaluation
11. Calculation related to decision tree and support vector machine
12. Model Evaluation: Build a confusion matrix; Calculate sensitivity, specificity, precision, recall and f1-score
13. Solve perceptron weight updates for a small dataset
14. Derive forward and backward propagation numerically for a 2-layer NN
15. Case study: Identify ethical risks in real-world AI use-responsible AI in Nepal's context

Practical **(45 hours)**

1. Intelligent agents and problem formulation
2. Uninformed search techniques
3. Informed (Heuristic) search techniques
4. Adversarial search and game playing
5. Constraint satisfaction problems (CSP)
6. Evolutionary computation (Genetic algorithms)
7. Knowledge representation (Logic, semantic networks, frames)
8. Probabilistic and fuzzy reasoning
9. Machine learning pipeline and data preprocessing
10. Supervised and unsupervised learning
11. Neural networks and deep learning basics
12. Mini project, AI applications and ethics

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapters	Hours	Marks distribution*
1, 6 and 7	11	14
2	9	12
3	7	10
4	10	14
5	8	10
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Russell, S., Norvig, P. (2021). Artificial intelligence: A modern approach. Pearson.
2. Rich, E., Knight, K., Nair, S. B. (2009). Artificial intelligence. McGraw-Hill Education.
3. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
4. Deisenroth, M. P., Faisal, A. A., Ong, C. S. (2020). Mathematics for machine learning. Cambridge University Press.
5. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT Press.

SOFTWARE ENGINEERING
ENCT 352

Lecture	**: 3**	**Year : III**
Tutorial	**: 1**	**Part : II**
Practical	**: 3/2**	

Course Objectives:

The objective of this course is to provide students with a foundation in software engineering, covering software characteristics, principles, process models, requirements engineering, system and object-oriented modeling, architectural and design principles, coding standards, testing and quality assurance. Students will gain both theoretical understanding and practical experience in applying software engineering concepts to a project development, including modern practices such as CI/CD, containerization and AI-assisted development.

1 Introduction (4 hours)

1.1 Software and software engineering
1.2 Nature and characteristics of software
1.3 Software application domains
1.4 Legacy software
1.5 Software crisis
1.6 Software myths
1.7 Software engineering practice: Essence of practice, general principles

2 The Software Process (8 hours)

2.1 Process framework and umbrella activities
2.2 Traditional (Plan-driven) process models
  2.2.1 Waterfall model and its extensions
  2.2.2 Incremental process model
  2.2.3 Evolutionary process models (Prototyping, spiral)
2.3 Agile and adaptive process models
  2.3.1 Agile manifesto and the 12 principles
  2.3.2 Agile versus plan-driven development
  2.3.3 Scrum framework (Roles, artifacts, ceremonies)
  2.3.4 Extreme programming (XP) practices
  2.3.5 Lean software development
2.4 Model selection considerations

3 Software Requirements Engineering (6 hours)

3.1 Requirement engineering process

- 3.2 SRS (Structure, characteristics, users)
- 3.3 Functional and non-functional requirements
- 3.4 Gathering requirements using use cases modeling and scenarios
- 3.5 Agile requirements engineering
  - 3.5.1 User stories and acceptance criteria
  - 3.5.2 Product backlog creation and prioritization
  - 3.5.3 Story mapping basics
  - 3.5.4 Continuous requirements refinement (Backlog grooming)

4 Architectural Design (3 hours)

- 4.1 Introductions and importance
- 4.2 Architectural design principles
- 4.3 Taxonomy of architectural styles
- 4.4 Modular design, cohesion and coupling

5 System Modeling (9 hours)

- 5.1 System modeling
  - 5.1.1 Need for system modeling
  - 5.1.2 Role of abstractions in managing complexity
- 5.2 Process modeling using DFD
- 5.3 Scenario-based analysis
  - 5.3.1 Concept of scenarios
  - 5.3.2 Use-case descriptions and diagrams
- 5.4 Behavioral and structural modeling
  - 5.4.1 Activity diagrams
  - 5.4.2 Class-based modeling

6 Coding and Testing (5 hours)

- 6.1 Coding standards and guidelines
- 6.2 Code review
  - 6.2.1 Code walkthrough
  - 6.2.2 Code inspection
  - 6.2.3 Cleanroom technique
- 6.3 Software testing fundamentals
  - 6.3.1 Verification and validation
  - 6.3.2 Unit testing
  - 6.3.3 Integration testing
  - 6.3.4 System testing
  - 6.3.5 Acceptance testing
- 6.4 Black-box and white-box testing approach
- 6.5 Agile testing practices
  - 6.5.1 Test-driven development (TDD)
  - 6.5.2 Continuous testing

6.5.3 Automated unit testing basics
6.5.4 Refactoring for code quality

7 Software Quality, Assurance, Maintenance (4 hours)

7.1 Quality concepts
7.2 Quality attributes
7.3 Reviews, inspections and QA concepts
7.4 ISO standards, CMMI levels
7.5 Software maintenance and its types
7.6 Maintenance effort and lifecycle considerations

8 Software Configuration Management (3 hours)

8.1 Software configuration management
8.2 Version control and continuous integration
8.3 Change management process
8.4 Branching strategies (Git-flow basics)

9 Recent Trends (3 hours)

9.1 DevOps and continuous deployment pipelines
9.2 Continuous integration/continuous delivery (CI/CD) in modern projects
9.3 Cloud-native, microservices architectures and infrastructure as code (IaC)
9.4 AI-assisted software development
9.5 Low-code/no-code, green software engineering

Tutorial (15 hours)

1. Analyze the project domain, identify challenges, and discuss software myths
2. Select and justify an appropriate software process model
3. Prepare preliminary SRS, including functional and non-functional requirements
4. Elicit and model requirements using use cases (Actors, scenarios, system boundary, diagram, and descriptions)
5. Develop behavioral models (Scenario identification and activity diagrams)
6. Develop a structural model (Class diagram)
7. Propose an architectural style and apply modular design principles
8. Prepare user stories and product backlog; Conduct a mock sprint planning session
9. Define coding standards and perform basic code review/refactoring
10. Design black-box and white-box test cases
11. Apply version control using git (Commit, branch, merge, conflict resolution)
12. Plan a basic CI/CD workflow and compile the final documentation of all artifacts

Practical **(22.5 hours)**

1. Preparation and organization of software requirements specification (SRS) using standard templates
2. Construction of UML diagrams (Use case, activity, class, and sequence) using modeling tools
3. Design of system architecture and component structure using CASE tools
4. Implementation of key system modules following coding standards and best practices
5. Execution of testing procedures (Unit and integration) and documentation of results
6. Application of version control (Git), CI/CD workflow setup, and containerization using docker
7. Term project

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	4	5
2	8	11
3	6	8
4	3	4
5	9	12
6	5	7
7	4	5
8	3	4
9	3	4
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Pressman, R. S., Maxim, B. R. (2019). Software engineering: A practitioner's approach. McGraw-Hill Education.
2. Mall, R. (2018). Fundamentals of software engineering. PHI Learning.
3. Sommerville, I. (2016). Software engineering. Pearson.
4. Martin, R. C. (2009). Clean code: A handbook of agile software craftsmanship. Pearson.
5. Schwaber, K., Beedle, M. (2002). Agile software development with Scrum. Prentice Hall.

SIMULATION AND MODELING

ENCT 353

Lecture	**: 3**	**Year : III**
Tutorial	**: 1**	**Part : II**
Practical	**: 3/2**	

Course Objectives:

The objective of this course is to develop knowledge of modeling and simulation techniques for discrete and continuous systems. It emphasizes the development and analysis of simulation models, generation and testing of random numbers and variables, and application of simulation methods to evaluate the performance of stochastic systems.

1 Introduction to Simulation **(4 hours)**

1.1 System and system environment concepts
1.2 Continuous and discrete systems
1.3 Types of models
1.4 Model development life cycle
1.5 Simulation and steps in simulation study
1.6 Advantages and disadvantages of simulation
1.7 Monte-Carlo simulation
1.8 Discrete-event system simulation

2 Physical and Mathematical Models **(4 hours)**

2.1 Differential and partial differential equations
2.2 Static physical model
2.3 Dynamic physical model
2.4 Static mathematical models
2.5 Dynamic mathematical models

3 Simulation of Continuous System **(5 hours)**

3.1 Continuous system models
3.2 Analog computer
3.3 Analog methods
3.4 Hybrid simulation
3.5 Digital-analog simulators
3.6 Continuous system simulation languages (CSSLs)
3.7 Feedback systems

4 Simulation of Queuing System **(6 hours)**

4.1 Elements of queuing system

4.2 Characteristics of queuing systems
4.3 Model of queuing system
4.4 Types of queuing system
4.5 Queuing notation (Kendall's notation)
4.6 Measurement of system performance
4.7 Network of queues
4.8 Applications of queuing system

5 Markov Chains **(3 hours)**

5.1 Key features of Markov chains
5.2 Markov process with examples
5.3 Applications of Markov chains

6 Random Number **(10 hours)**

6.1 Properties of random numbers
6.2 Generation of pseudo-random numbers
6.3 Random number generation: Linear and arithmetic congruential methods
6.4 Tests for random numbers
 6.4.1 Kolmogorov-Smirnov test
 6.4.2 Chi-Square (χ2) test
 6.4.3 Gap test
 6.4.4 Poker's method
 6.4.5 Testing for auto correlation
6.5 Generating discrete distribution
6.6 Inversion, rejection, composition and convolution

7 Verification and Validation of Simulation Models **(3 hours)**

7.1 Verification and validation
7.2 Verification of simulation models
7.3 Calibration and validation of models
7.4 Naylor and finger validation process
7.5 Validation: Errors

8 Analysis of simulation output **(4 hours)**

8.1 Confidence intervals and hypothesis testing
8.2 Estimation methods
8.3 Simulation run statistics
8.4 Replication of runs
8.5 Elimination of initial bias

9 Simulation software **(3 hours)**

9.1 Simulation in Java

9.2 Simulation in GPSS
9.3 Simulation in Python
9.4 Other simulation software

10 Simulation of Computer Systems (3 hours)

10.1 Simulation tools
10.2 High level computer: System simulation
10.3 CPU simulation
10.4 Memory simulation
10.5 Simulation of computer networks

Tutorial (15 hours)

1. Poisson process
2. Monte Carlo simulation
3. Queuing system and Markov chains
4. Different methods of random number generation
5. Kolmogorov-Smirnov test, Chi-square test, gap-test, poker and autocorrelation test
6. Modeling continuous and discrete systems
7. Simulation output analysis

Practical (22.5 hours)

1. Simulation of the R-C amplifier circuit and mass spring damper system
2. Generation of random number
3. Chi-square goodness-of-fit test and Kolmogorov-Smirnov test
4. Simulation of queuing system
5. Simulation of Markov chain

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapters	Hours	Marks distribution*
1, 2	8	10
3	5	7
4	6	8
5	3	5
6	10	13
7	3	4
8	4	5
9, 10	6	8
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Banks, J. (2005). Discrete-event system simulation. Pearson.
2. Gordon, G. (1978). System simulation (Latest Edition). Prentice Hall.
3. Law, A. M., Kelton, W. D. (2007). Simulation modeling and analysis. McGraw-Hill Education.
4. Rubinstein, R. Y., Melamed, B. (1998). Modern simulation and modeling (Latest Edition). John Wiley & Sons.
5. Singh, V. P. (2009). System modeling and simulation. New Age International Publishers.

MINOR PROJECT
ENCT 354

Credit :1 **Year: III** **Part: II**

Course Objectives:
The objective of this course is to enable students to carry out a small-scale engineering research- or product-based project in the field of computer engineering, thereby developing practical skills and hands-on experience in system design, implementation, and testing

General Procedures
Under the supervision of an assigned supervisor, students shall be organized into groups of 3–4 members. Each group shall select a project topic related to computer engineering, with emphasis on areas such as artificial intelligence and machine learning, image processing and computer vision, networks and cybersecurity, audio and natural language processing, electronic devices and embedded systems, big data technologies, Internet of Things (IoT), and web and mobile application development. Where appropriate, students may conduct field visits or collect real-world datasets for system design, implementation, and analysis. Each group shall systematically complete the project through stages including topic selection, planning, development, testing, documentation, and final presentation. Continuous guidance, monitoring, and feedback shall be provided by the supervisor throughout the duration of the course to ensure effective project development and attainment of learning outcomes.

1 Project Selection and Planning

1.1 Selection of minor project based on theory/practical subject studied or as per the research or industry needs
1.2 Problem identification and scope definition
1.3 Approval of project by the department

2 Literature Review and Proposal

2.1 Basic literature review and background study, study of existing systems in case of product based project
2.2 Preparation of brief project proposal including objective and methodology
2.3 Selection of software tools, technologies, and resources (Programming languages, frameworks, databases)

3 Design, Development and Project Management

3.1 Design and development (Software/hardware/hybrid system)

3.2 Application of programming skills, algorithms and engineering tools
3.3 Integration of concepts from different subjects (Computer programming, object oriented programming, software engineering, artificial intelligence, compiler design, big data technology, database management system, computer organization and architecture, computer network, electronic devices)
3.4 Applications of project management skills such as work division, time management, leadership

4 Implementation and Testing

4.1 Implementation and deployment of the project
4.2 Testing and validation of the developed system (Unit testing, integration testing, and user testing)
4.3 Performance evaluation and improvement

Timeline of the defense

i. Proposal submission and defense: A group of students shall submit the proposal in the prescribed format to the concerned department within two weeks after the commencement of the sixth semester
ii. Mid-term defense: The mid-term defense of the minor project shall be conducted one month prior to the final defense
iii. Final defense: The final defense shall be conducted two weeks prior to the board examination

Evaluation

The final evaluation of the minor project shall be based on assessments from the supervisor, the proposal defense, mid-term defense, final defense, and the internal examiner. The respective weightage and total marks are presented in the following table:

Supervisor	**Defense**			**Internal Examiner**	**Total Marks**
	Proposal	**Mid-Term**	**Final**		
25	5	5	5	10	50

NETWORK AND SYSTEMS PROGRAMMING
ENCT 386

Lecture	**: 3**	**Year : III**
Tutorial	**: 2**	**Part : II**
Practical	**: 1**	

Course Objectives:
The objective of this course is to provide knowledge of networking and system-level programming concepts. It focuses on socket programming, process and thread management, and the use of system calls for efficient resource handling. Upon completion, students will be able to develop concurrent, scalable, and high-performance network applications suitable for real-world computing environments.

1 System Programming Fundamentals (4 hours)

1.1 Overview of system programming and its role in networking
1.2 User space versus kernel space
1.3 System calls and standard libraries
1.4 File descriptors and low-level I/O
1.5 Linux/Unix system architecture overview

2 Process Management and IPC (7 hours)

2.1 Process creation and control (Fork, exec, wait)
2.2 Process states and scheduling concepts
2.3 Signals and signal handling
2.4 Inter-process communication: Pipes and FIFOs, message queues, shared memory, semaphores

3 Threads and Concurrent Programming (7 hours)

3.1 Thread models and lifecycle
3.2 POSIX threads (Pthreads)
3.3 Thread synchronization: Mutexes, condition variables
3.4 Deadlocks, race conditions, and avoidance
3.5 Multithreading versus multiprocessing (Performance considerations)

4 Socket Programming and Network APIs (10 hours)

4.1 Socket interface and API
4.2 TCP socket programming (Client-server model)
4.3 UDP socket programming
4.4 Blocking versus non-blocking sockets
4.5 I/O multiplexing (Select, poll basics)

4.6 Error handling and debugging techniques

5 Advanced Network Programming (7 hours)

5.1 Concurrent server design: Iterative versus concurrent servers, multi-process and multi-threaded servers
5.2 Event-driven programming: Select, poll, epoll
5.3 High-performance server design principles
5.4 Concepts of remote procedure call (RPC)
5.5 Introduction to RESTful APIs and microservices
5.6 Secure communication basics (SSL/TLS overview)

6 System-Level I/O and Performance Optimization (5 hours)

6.1 Advanced file I/O (Read/write optimization)
6.2 Non-blocking I/O and asynchronous I/O concepts
6.3 Memory-mapped files (Mmap)
6.4 File locking and concurrency
6.5 Performance tuning and profiling basics

7 Network Application Development and Case Studies (5 hours)

7.1 Design of scalable client-server systems
7.2 Development of network applications: Chat servers, file transfer systems
7.3 Debugging and monitoring tools (Netstat, ss, tcpdump, wireshark)
7.4 Case studies of real-world systems (Web servers, distributed apps)

Tutorial (30 hours)

1. Implement file operations using low-level system calls (Open, read, write, close) and compare with standard I/O
2. Demonstrate file descriptor manipulation using dup()/dup2() for I/O redirection
3. Write a program using fork() and exec() to create and replace processes
4. Implement IPC using pipes and shared memory, comparing their behavior
5. Create multiple threads using POSIX threads (pthreads) and observe execution behavior
6. Solve the producer-consumer problem using mutexes and condition variables
7. Develop a TCP client-server application (Echo service)
8. Implement a UDP communication program and compare with TCP behavior.
9. Build a multi-client concurrent server using threads or processes
10. Implement I/O multiplexing using select() or poll() for handling multiple clients
11. Demonstrate memory-mapped file (mmap) usage and compare with standard file I/O
12. Implement file locking mechanisms to handle concurrent file access
13. Develop a multi-client chat application using sockets
14. Capture and analyze traffic of your application using tools like Wireshark/tcpdump

15. Design and implement a scalable network application (chat/file server) incorporating: Concurrency (threads/processes), socket programming, basic performance optimization

Practical **(15 hours)**

1. System calls and file operations in Linux
2. Process creation using fork/exec and process control
3. IPC using pipes, shared memory, and message queues
4. Signal handling programs
5. Thread creation and synchronization using Pthreads
6. TCP client-server socket programming
7. UDP-based communication programs
8. Multi-client concurrent server (thread/process-based)
9. I/O multiplexing using select/poll
10. Packet capture and analysis using Wireshark
11. Mini-project: Design and implement a scalable network application

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	4	5
2	7	9
3	7	9
4	10	14
5	7	9
6	5	7
7	5	7
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Forouzan, B. A. (2017). Data communications and networking. McGraw-Hill Education.
2. Kurose, J. F., Ross, K. W. (2021). Computer networking: A top-down approach. Pearson.
3. Stevens, W. R. (2013). Advanced programming in the UNIX environment. Addison-Wesley.
4. Stevens, W. R. (2003). UNIX network programming. Addison-Wesley.
5. Kernighan, B. W., Pike, R. (1984). The UNIX programming environment (Latest Edition). Prentice Hall.

Next Generation Networking with IPv6

ENCT 387

Lecture : 3 | **Year : III**
Tutorial : 2 | **Part : II**
Practical : 1

Course Objectives:

The objective of this course is to develop a solid understanding of internet protocols with a focus on IPv6 and next-generation networking concepts. It aims to equip students with knowledge of IPv6 addressing, routing, security and transition mechanisms from IPv4 to IPv6. The course also emphasizes practical skills in configuring, analyzing and deploying IPv6-based networks and services. Ultimately, it prepares students to design and manage modern network infrastructures in real-world environments.

1 Fundamentals of Internet and Networking Protocols (3 hours)

1.1 Evolution of the Internet and networking paradigms
1.2 OSI versus TCP/IP model
1.3 IPv4 addressing overview
1.4 IPv4 addressing review and limitations
1.5 Need and importance of next generating IP addressing

2 Next Generation Internet Protocol (12 hours)

2.1 Introduction to IPv6
   2.1.1 History and need for IPv6
   2.1.2 IPv6 header format
   2.1.3 Features of IPv6
   2.1.4 IPv6 addressing format and types
   2.1.5 Extension headers
2.2 ICMPv6
   2.2.1 Features
   2.2.2 General message format
   2.2.3 ICMP error and informational message types
   2.2.4 Neighbor discovery
   2.2.5 Path MTU discovery

3 Security and Quality of Service in IPv6 (6 hours)

3.1 Network threats and vulnerabilities in IPv6
3.2 Security techniques
3.3 IPSEC framework
3.4 Basic firewall and filtering concepts

3.5 Zero trust networking
3.6 QoS concept in IPv6
3.7 QoS in multimedia and real-time applications

4 Routing with IPv6 **(6 hours)**

4.1 Routing in the internet and CIDR
4.2 Unicast and multicast routing concepts
4.3 Unidirectional link routing
4.4 Routing protocols: RIPng, OSPFv3 and BGP (Basic concepts for IPv6)
4.5 Multicast routing (PIM-SM overview)

5 IPv4/IPv6 Transition Mechanisms **(6 hours)**

5.1 Tunneling: Automatic tunneling; Configured tunneling
5.2 Dual stack approach
5.3 Translation mechanisms
5.4 Migration strategies for enterprises and ISPs
5.5 Case studies of global IPv6 transition

6 IPv6 Deployment and Network Design **(5 hours)**

6.1 IPv6 deployment challenges and risks
6.2 Enterprise and ISP deployment strategies
6.3 IPv6 addressing plan design
6.4 IPv6 DNS (AAAA records)
6.5 IPv6 enabled proxy, web and mail servers

7 Advanced Networking Applications **(7 hours)**

7.1 Fundamentals of SDN
7.2 Introduction to IBN and NDN
7.3 IPv6 as the foundations of 5G network
7.4 IPv6 integration SDN/IBN/NDN

Tutorial **(30 hours)**

1. Compare OSI vs TCP/IP models through layered packet flow analysis
2. Analyze limitations of IPv4 addressing and justify the need for IPv6 using numerical examples
3. Construct and analyze an IPv6 packet, including header fields and extension headers
4. Perform IPv6 addressing and subnetting exercises (Global, link-local, multicast)
5. Implement and observe neighbor discovery protocol and ICMPv6 message types
6. Demonstrate basic IPsec configuration concepts
7. Analyze QoS mechanisms in IPv6, especially for real-time/multimedia traffic

8. Configure and test IPv6 unicast routing (OSPFv3 basics) in a simulated environment
9. Comparison of IPv6 multicast routing (PIM-SM concept) with unicast routing
10. Implement dual stack configuration in a lab setup
11. Demonstrate tunneling mechanisms and analyze packet flow
12. Design an IPv6 addressing plan for an enterprise network
13. Configure and test IPv6 DNS (AAAA records) and basic web service
14. Analyze the role of IPv6 in SDN/5G/NDN architectures through a case study
15. Design and simulate a complete IPv6-enabled network, including: Addressing plan, routing (OSPFv3 or static), transition mechanism (Dual stack/tunneling), basic service deployment (DNS/Web)

Practical **(15 hours)**

1. Enabling and configuring IPv6 in Windows/Linux
2. IPv6 header and packet analysis using Wireshark
3. Neighbor Discovery and ICMPv6 analysis
4. IPv6 addressing and subnetting practice
5. Unicast routing using OSPFv3
6. Multicast routing using PIM-SM
7. IPv6 DNS, web, and proxy server configuration
8. Case Study: Design and deploy an IPv6-based network

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapters	Hours	Marks distribution*
2	12	16
3	6	8
4	6	8
5	6	8
6	5	8
1 and 7	10	12
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Davies, J. (2008). Understanding IPv6. Microsoft Press.
2. Hagen, S. (2014). IPv6 essentials. O'Reilly Media.
3. Thomas, S. A. (1996). IPng and the TCP/IP protocols. John Wiley & Sons.
4. Hersent, O., Gurle, D., & Petit, J.-P. (2000). IP telephony: Deploying VoIP protocols. Addison-Wesley.

ANALYSIS OF ALGORITHMS
ENCT 388

Lecture	**: 3**	**Year : III**
Tutorial	**: 2**	**Part : II**
Practical	**: 1**	

Course Objectives:

The objective of this course is to provide students with a strong foundation in the analysis of algorithm efficiency and computational complexity. It aims to develop understanding of key algorithm design paradigms, including divide-and-conquer, greedy methods, dynamic programming, and backtracking, and covers fundamental concepts of NP-completeness and approximation algorithms.

1 Introduction to Algorithm Analysis (5 hours)

1.1 Algorithm and its properties, RAM model, time and space complexity, detailed analysis of algorithms, Concept of Aggregate Analysis
1.2 Asymptotic notations (Big-O, Big-Ω and Big-Ө), their geometrical interpretations and examples
1.3 Concept of best case, average case and worst case performance of an algorithm
1.4 Modeling algorithms by recurrence relation
1.5 Solving recurrence relation for evaluating computational complexity
  1.5.1 Recursion tree method
  1.5.2 Substitution method
  1.5.3 Using masters theorem

2 Iterative and Numeric Algorithms (8 hours)

2.1 Algorithm for GCD and Fibonacci number
2.2 Sequential search
2.3 Review of bubble sort, selection sort, and insertion sort algorithms
2.4 Number theoretic notations
2.5 Euclid's and Extended Euclid's algorithms
2.6 Solving modular linear equations using Chinese remainder theorem
2.7 Fermat's theorem
2.8 Miller-Rabin randomized primility test and algorithm

3 Divide and Conquer Algorithms (8 hours)

3.1 Binary search, min max finding algorithm
3.2 Analysis of sorting algorithms
  3.2.1 Merge sort

- 3.2.2 Heap sort
- 3.2.3 Quick sort
- 3.2.4 Randomized quick sort

3.3 Order statistics

- 3.3.1 Selection in expected linear time
- 3.3.2 Selection in worst case linear time

4 Greedy Algorithms (7 hours)

4.1 Basic concepts
4.2 Fractional knapsack problem
4.3 Job sequencing with deadlines
4.4 Analysis of minimum spanning trees related algorithms
4.5 Analysis of single source shortest path algorithm

5 Dynamic Programming (8 hours)

5.1 Basic concepts
5.2 All pair shortest path algorithm
5.3 Travelling salesperson problem
5.4 String editing
5.5 0/1 knapsack problem using dynamic programming
5.6 Matrix chain multiplication
5.7 Flow shop scheduling

6 Backtracking Techniques (4 hours)

6.1 Basic concepts
6.2 The N-Queen problem
6.3 Sum of subsets
6.4 Graph coloring
6.5 Hamiltonian cycles
6.6 0/1 knapsack problem using backtracking approach

7 NP-Hard and NP-Complete Problems (5 hours)

7.1 Basic concepts
7.2 Cook's theorem
7.3 NP-Hard graph problems
7.4 NP-Hard scheduling problems
7.5 NP-Hard code generation problems
7.6 Simplified NP-hard problems
7.7 Approximation algorithms: ε- approximation, polynomial time approximation scheme, probabilistically good algorithms
7.8 Vertex cover problem, subset sum problem

Tutorial **(30 hours)**

1. Algorithm analysis
2. Iterative and numeric algorithms
3. Divide and conquer algorithms
4. Greedy algorithms
5. Dynamic programming
6. Backtracking techniques
7. NP-Hard and NP-complete problems

Practical **(15 hours)**

1. Implementation and complexity analysis of iterative, numeric and recursive algorithms
2. Implementation and complexity analysis of greedy algorithms
3. Implementation and complexity analysis of algorithms involving divide and conquer strategy
4. Implementation and complexity analysis of algorithms based on dynamic programming
5. Implementation and complexity analysis of algorithms using backtracking concept

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	5	6
2	8	10
3	8	11
4	7	9
5	8	12
6	4	5
7	5	7
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Horowitz, E., Sahni, S., Rajasekaran, S. (2007). Fundamentals of computer algorithms. Universities Press.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2022). Introduction to algorithms. MIT Press.
3. Kleinberg, J., Tardos, É. (2006). Algorithm design. Pearson.
4. Skiena, S. S. (2020). The algorithm design manual. Springer.
5. Dasgupta, S., Papadimitriou, C. H., Vazirani, U. V. (2006). Algorithms. McGraw-Hill Education.

AUDIO PROCESSING
ENCT 389

Lecture	**: 3**	**Year : III**
Tutorial	**: 2**	**Part : II**
Practical	**: 1**	

Course Objectives:
The objective of this course is to develop competency in analyzing and processing audio signals in time and frequency domains. It covers fundamentals of acoustics, digital audio representation, spectral analysis and audio effects, along with practical skills in feature extraction and Music Information Retrieval (MIR). Hands-on use of Digital Audio Workstations (DAWs), MIDI, and plugins enables application of audio signal processing techniques in multimedia, speech processing, music technology and emerging digital audio applications.

1 Fundamental of Sound and Audio Signal (7 hours)

1.1 Physical acoustic: Wave propagation, frequency, amplitude, phase
1.2 Psycho acoustic: Pitch, loudness, timbre, dynamics and intensity
1.3 Fundamental of digital audio
1.4 Audio file format: WAV, AIFF, FLAC, ALAC, MP3
1.5 Practical audio quality: Sample rates (44.1khz, 48khz, 96khz) and bit depths

2 Time Domain Audio Processing (8 hours)

2.1 Discrete time audio signal, basic signal operations, system in time domain
2.2 Convolution and impulse response
2.3 Analog and digital filters in audio systems
2.4 Temporal feature extraction

3 Spectral Representation (7 hours)

3.1 DFT and FFT: Derivation and computational complexity
3.2 Windowing function in audio
3.3 Short time Fourier transform: Frame size, hope size and overlap
3.4 Spectrograms: Reading and interpreting time-frequency representation,
3.5 Mel spectrogram: Reading and interpreting; Mel frequency representation
3.6 Constant Q transform and its relevance to music analysis

4 Frequency Domain and Non-Linear Effects (10 hours)

4.1 Sinusoidal model
4.2 Deterministic plus residual model
4.3 Deterministic plus stochastic model

4.4 Basis of non-linear effects
4.5 Dynamic range control: Envelop follow, compressor and limiting, expansion and gating, ADSR, modulation effect
4.6 Time segment process: Time scratching and pitch shifting

5 Music Information Retrieval (6 hours)

5.1 MIR overview: Types of music data, challenges, application
5.2 Spectral centroid and roll off
 5.2.1 Basics of spectral centroid and spectral roll
 5.2.2 Comparison between centroid and roll off
 5.2.3 Application (Timbre analysis, music/ speech classification)
5.3 Mel frequency cepstral coefficient (MFCC)
 5.3.1 Motivation for ceptral features
 5.3.2 Steps in MFCC
 5.3.3 Interpretation of coefficient
 5.3.4 Application (Speech recognition, music classification)
5.4 Chroma
 5.4.1 Concept of pitch classes (12- tone system) mapping
 5.4.2 Chromagram representation,
5.5 Onset and tempo and their applications

6 Audio system and DAW Integration (4 hours)

6.1 Digital audio workstation (DAW)
 6.1.1 Role in audio production
 6.1.2 History and evolution of DAW
6.2 DAW available in market, DAW interface, timeline and arrangement view
6.3 MIDI: Basics, MIDI message structure, VSTi basics,
6.4 Common plugin: Equalizer, compressor, reverb, delay

7 Recent Trend (3 hours)

7.1 Machine learning in audio
7.2 Audio foundation models for bioacoustic and healthcare
7.3 Spatial audio
7.4 Audio DSP ICs
7.5 Generative AI in music

Tutorial (30 hours)

1. Discuss lossless versus lossy compression in the context of audio
2. Derive the convolution sum for a linear time-invariant (LTI) system and explain its physical significance in audio signal processing.
3. Discuss different types of analog filter. Write an advantage of FIR and IIR filter in audio processing
4. Case study discussion
5. Explain the various steps involved in MFCC

6. State the Nyquist theorem and find the Nyquist rate for a signal with maximum frequency 8 kHz.
7. Explain the concept of a chromagram in music information retrieval
8. Explain binaural rendering using head-related transfer functions (HRTFs) and identify one application domain for dedicated audio DSP hardware
9. Music feature analyzer: Input (Audio file), output (Spectrogram, MFCC, tempo)
10. Mini DAW project: Record and edit audio using DAW, apply multiple effects, mix a short track
11. Music classifier: Genre basis classifier

Practical **(15 hours)**

1. Generate sinusoids; Downsample to observe aliasing
2. Measure file size, SNR, and spectral differences between WAV, MP3 (128k/320k), and FLAC
3. Temporal features extraction
4. Spectrogram and STFT
5. Mel spectrogram and CQT
6. MIR features: MFCC
7. DAW: Implement basic features
8. DAW integration and MIDI, VSTi implementation

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	7	9
2	8	11
3	7	9
4	10	13
5	6	8
6	4	6
7	3	4
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Christensen, M. G. (2019). Introduction to audio processing. Springer.
2. Smith, J. O. (2011). Spectral audio signal processing. W3K Publishing.
   Zölzer, U. (Ed.). (2011). DAFX: Digital audio effects. Wiley.
3. Müller, M. (2015). Fundamentals of music processing: Audio, analysis, algorithms, applications. Springer.
4. Lyons, R. G. (2011). Understanding digital signal processing. Pearson Education.