गोंय विद्यापीठ
ताळगांव पठार,
गोंय - ४०३ २०६
फोन : + ९१ - ८६६९६०९०४८

Goa University
Taleigao Plateau,Goa - 403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

(Accredited by NAAC with Grade A+)

GU/Acad –PG/BoS -NEP/2026-27/221 Date: 17.07.2026

CIRCULAR

Ref. No.: GU/Acad –PG/BoS -NEP/2024/643 dated 11.11.2024,
GU/Acad –PG/BoS -NEP/2025-26/274dated 30.07.2025

In continuation to the above referred Circulars, the syllabus for Semester V & VI of the **Bachelor of Engineering in Information Technology** Programme as approved by the Academic Council in its meeting held on 20th February 2026 is attached.

The Dean, Faculty of Engineering and Principals of affiliated Colleges offering the **Bachelor of Engineering in Information Technology** Programme are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN V LAWANDE
Digitally signed by ASHWIN V LAWANDE
Date: 2026.07.17 16:30:55 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,

1. The Dean, Faculty of Engineering, Goa University.
2. The Principals of Affiliated Engineering Colleges.

Copy to:

1. Chairperson, BoS in Information Technology Engineering, Goa University.
2. Controller of Examinations, Goa University.
3. Assistant Registrar Examinations (Technical and Allied), Goa University
4. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

INFORMATION TECHNOLOGY (AY 2024-25)

SEMESTER - V							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**ITH-300**	Database Systems	2	0	0	2
		ITH-301	Database Systems Lab	0	0	2	2
2	**Major**	**ITH-302**	Fundamentals of Artificial Intelligence	3	0	0	3
		ITH-303	Fundamentals of Artificial Intelligence Lab	0	0	1	1
3	**Major**	**ITH-304**	Software Engineering	3	0	0	3
		ITH-305	Software Engineering Lab	0	0	1	1
4	**Major**	**ITH-306**	Theory of Computation	3	0	0	3
		ITH-307	Theory of Computation Lab	0	0	1	1
5	**Prof. Electives**	**ITH-321**	Cloud Infrastructure and Services	3	0	0	3
		ITH-322	Cloud Infrastructure and Services Lab	0	0	1	1
		OR					
		ITH-323	Mobile Computing	3	0	0	3
		ITH-324	Mobile Computing Lab	0	0	1	1
		OR					
		ITH-325	Parallel Computing	3	0	0	3
		ITH-326	Parallel Computing Lab	0	0	1	1
TOTAL				**14**	**0**	**6**	**20**

Approved by: Academic Council on 24th April, 12th & 13th May 2026

SEMESTER - VI							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**ITH-308**	Web Technology	2	0	0	2
		ITH-309	Web Technology Lab	0	0	2	2
2	**Major**	**ITH-310**	Augmented and Virtual Reality	2	0	0	2
		ITH-311	Augmented and Virtual Reality Lab	0	0	2	2
3	**Major**	**ITH-312**	Principles of Compilers	3	0	0	3
		ITH-313	Principles of Compilers Lab	0	0	1	1
4	**Major**	**ITH-314**	Software Test Automation and Quality Assurance	3	0	0	3
		ITH-315	Software Test Automation and Quality Assurance Lab	0	0	1	1
5	**Prof. Electives**	**ITH-327**	Internet of Things and Blockchain Technology	4	0	0	4
		OR					
		ITH-328	Machine Learning Techniques	4	0	0	4
		OR					
		ITH-329	Multimedia Technology	4	0	0	4
			TOTAL	**14**	**0**	**6**	**20**

Semester V

Major Courses

Name of the Programme : B.E. in Information Technology
Course Code : ITH-300
Title of the Course : Database Systems
Number of Credits : 2
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic knowledge of programming, data structures, and algorithms	
Course Objectives	This course will enable students: 1. To provide a fundamental understanding of database concepts, data models, and the architecture of database systems. 2. To develop the ability to design ER/EER models and convert them into relational schemas. 3. To impart knowledge of relational algebra, SQL query writing, and schema refinement concepts. 4. To familiarize students with transaction processing, query optimization techniques, and emerging NoSQL technologies.	
Contents		**No of Hours**
Unit-1	**Introduction:** Characteristics of Database approach, advantages of using the DBMS approach, Three schema architecture, Data Models, Database Users **Entity–Relationship Model:** Entity-Relationship diagram, Attributes and Entity sets, Relationships and Relationship sets, Reduction to relational schema, Extended-ER features. **Relational Model:** Relational model concepts, Constraints and relational Database schema	**8**
Unit -2	**Relational Algebra** – Selection and projection, Set operations, renaming, Joins, Division, Aggregate functions, and Grouping. **SQL Query Language** – SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, Set Operations, Aggregate Functions – GROUPBY – HAVING, Nested Sub-queries, Views, Triggers.	**8**
Unit - 3	**Introduction to Schema Refinement:** Informal design guidelines for relational schemas, Functional dependencies, Normal forms: 1NF, 2NF, 3NF, BCNF. Properties of Decompositions: Lossless Join & Dependency Preservation	**7**

	Query Processing & Optimization: Measures of Query Cost, Selection operation, Join operation. Overview of Optimization, Heuristics in Optimization	
Unit - 4	**Introduction to Transaction Processing:** Transaction and system concepts, desirable properties of a transaction **Concurrency Control Techniques:** Two-phase locking technique for concurrency control, concurrency control based on timestamp ordering **No SQL:** Introduction to NoSQL, Types of NoSQL, and Advantages of NoSQL.	**7**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References / Readings	**Text Books** 1. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2017, ISBN-978-9332582705. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill, 2021, ISBN-978-9390727506. **Reference Books** 1. Raghu Ramkrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition McGraw-Hill, 2014, ISBN- 978-8131769591. 2. C.J. Date, A. Kannan, S. Swami Nadhan, An Introduction to Database systems, 8th Edition, Pearson, 2006, ISBN- 978-8177585568. 3. Pramod J. Sadalage, Martin Fowler, NOSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st Edition, Pearson, 2012, ISBN-978-0-321-82662-6.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain the fundamental concepts and terminology related to databases systems CO 2. Construct ER/EER diagrams and convert them into relational schemas with appropriate constraints. CO 3. Write relational algebra expressions and formulate SQL queries, including joins, subqueries, triggers, and constraints to manage and retrieve data. CO 4. Evaluate database schemas and refine them using normalization	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-301**
Title of the Course : **Database Systems Lab**
Number of Credits : **2**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic knowledge of the Relational Database Management System	
Course Objectives	The course will enable students to: 1. To develop the ability to design robust database models using Entity–Relationship diagrams. 2. To build proficiency in SQL operations 3. To apply advanced SQL concepts 4. To integrate theoretical and practical skills in a mini-project.	
Contents	**List of programs**	**No of Hours**
	RDBMS 1. Creating an Entity-Relationship Diagram using online tools. 2. Implementation of DDL commands in SQL. 3. Implementation of DML commands in SQL. 4. Basic SQL Queries. 5. Implementation of different types of constraints. 6. Implementation of different types of functions with suitable examples. 7. Implementation of different types of Joins. 8. Implementation of Nested Queries and Correlated Queries. 9. Implementation of Group by, Order by, and having clause. 10. Implementation of Stored Procedures 11. Implementation of Triggers **NoSQL** 12. Creating and Manipulating Collections / Tables 13. CRUD Operations: Insert, Update, Delete 14. Querying Documents 15. Indexing 16. Schema Design in NoSQL 17. Develop a mini project based on either NoSQL or RDBMS concepts	**60**
Pedagogy	The course adopts a hands-on, practice-oriented approach through interactive demonstrations and lab-based learning. Students engage in problem-solving exercises and a mini-project to apply database concepts	

	in real-world scenarios.
References / Readings	**Text Books** 1. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2017, ISBN-978-9332582705. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill, 2021, ISBN-978-9390727506. **Reference Books** 1. Raghu Ramkrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition McGraw-Hill, 2014, ISBN- 978-8131769591. 2. C.J. Date, A. Kannan, S. Swami Nadhan, An Introduction to Database systems, 8th Edition, Pearson, 2006, ISBN- 978-8177585568. 3. Pramod J. Sadalage, Martin Fowler, NOSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st Edition, Pearson, 2012, ISBN-978-0-321-82662-6.
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Design ER diagrams and create corresponding database schemas. CO 2. Apply SQL commands (DDL, DML, constraints, functions, joins) to manage relational data. CO 3. Analyse and execute advanced SQL queries including nested queries, grouping, and triggers. CO 4. Develop a mini-project demonstrating complete database design and implementation.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-302**
Title of the Course : **Fundamentals of Artificial Intelligence**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic knowledge of mathematics and programming	
Course Objectives	This course will enable students: 1. Understand the fundamental concepts, scope, and applications of Artificial Intelligence and intelligent agents. 2. Learn different problem-solving methods, including uninformed and informed search strategies. 3. Understand and utilize knowledge representation techniques and reasoning mechanisms. 4. Gain foundational knowledge of machine learning paradigms, neural networks, and their real-world applications.	
Contents		**No of Hours**
Unit 1	**Introduction to artificial intelligence:** Definition and scope of AI, applications in the real world, Intelligent Agent Structure of agents, Types of agents, Simple Reflex, Goal-based, Learning Agents, Task environments, properties. **Problem solving and search strategies:** Problem formulation and State space**,** Uninformed Search: BFS, DFS, ID-DFS **Informed Search:** Heuristic Functions, Local Search algorithm, Hill Climbing, Best first search, A* algorithm. **Game playing:** Min-max algorithm, Alpha-Beta pruning.	**12**
Unit 2	**Knowledge representation:** Predicate logic, unification, Resolution, forward / backward chaining. **Knowledge Representation Approaches:** Semantic Networks, conceptual graphs, frames. **Reasoning:** Reasoning Types - Deductive, Inductive, Introduction to a rule-based expert system	**10**
Unit 3	**Planning:** Defining Classical Planning Problems using PDDL, Forward and Backward State Space Search, Partial Order Planning, Goal Stack Planning. **Uncertain knowledge and reasoning:** Acting under uncertainty, Basic Probability Notations, Fully Joint Distribution, Independence, Baye's Rule and its uses, Reasoning using Bayesian Network.	**11**

Unit 4	**Types of learning:** Rote learning, learning from observation, Inductive learning, example-based learning, explanation-based learning, learning by Decision tree. **Introduction to machine learning:** Types of machine learning - Supervised, Unsupervised, Reinforcement **Fundamentals of Neural Networks:** Architecture, Benefits of Neural Networks, and supervised learning algorithm. Applications of machine learning	**12**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Textbooks:** 1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2022, 978-1-292-40117-1. 2. Dr. Nayim Sepay & Dr. Sunita Roy, An Introduction to Artificial Intelligence, 4th Edition, Techno World Publication, 2024, ISBN: 978-81-19777-82-2. **Reference Books:** 1. Pradeep Singh, Tapan K Gandhi, Balasubramanian Raman, Artificial intelligence and machine learning, McGraw-Hill, 2025, ISBN: 9789364440820. 2. Reema Thareja, Artificial intelligence: beyond classical AI, Pearson Education, 2023, ISBN: 978-9356069329. 3. Elaine Rich, Kevin Knight, Shivashankar B Nair, Artificial Intelligence, 4th Edition, MedTech Science Press, 2024, ISBN-13 : 978-8196544577.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain the fundamental concepts, scope, and applications of AI, including intelligent agents, agent architectures, and task environments. CO 2. Apply classical AI problem-solving techniques and game-playing strategies. CO 3. Analyse knowledge representation schemes and reasoning methods for constructing intelligent systems. CO 4. Implement basic machine learning techniques by understanding learning paradigms, and evaluate AI applications with ethical, social, and technological factors.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-303**
Title of the Course : **Fundamentals of Artificial Intelligence Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	The course will enable students to: 1. Provide hands-on experience in designing and implementing intelligent agents and classical search algorithms. 2. Build competency in game-playing techniques, knowledge representation models, and reasoning methods. 3. Utilize Python and Prolog tools to develop expert systems, machine learning models, and rule-based inference solutions. 4. Apply automated planning and related tools to solve real-world AI problems.	
Contents	**List of programs**	**No of Hours**
	1. Implement a simple reflex-based vacuum-cleaner agent that operates in an N-cell environment 2. Implement the Best-First Search algorithm for a given graph and visualize the order of node expansion. 3. Implement the A* search algorithm to compute the shortest path on a weighted graph using an admissible heuristic. 4. Develop a Tic-Tac-Toe game where the computer uses the Minimax An algorithm (with optional alpha-beta pruning) to make optimal moves. 5. Implement a frame-based knowledge representation system for a chosen domain 6. Implement rule-based inference in Prolog using forward and Backward chaining mechanisms. 7. Develop a simple expert system in Prolog that diagnoses diseases based on symptoms 8. Implement the ID3 algorithm to build a decision tree for a given dataset 9. Implement a PDDL-based planning system using the python-pddl library and solve a domain-specific planning problem. 10. Implement the perceptron learning algorithm to train a model that classifies the outputs of AND and OR logic gates.	**30**

Pedagogy	Integration of instructional learning, constructive thinking, inquiry-based, Collaborative, experiential, and problem-solving approaches.
References/ Readings	**Text books** 1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson,2022, 978-1-292-40117-1 2. Elaine Rich, Kevin Knight, Shivashankar B Nair, Artificial Intelligence,4th Edition, MedTech Science Press, 2024, ISBN-10 819654457X **Reference Books** 1. Dan Fitzpatrick, Amanda Fox & Brad Mathew Weinstein, The AI Classroom: The Ultimate Guide to Artificial Intelligence in Education, Teacher Goals Publication, 2023, ISBN 978-1-959419-11-2. 2. Ivan Bratko, Prolog Programming for Artificial Intelligence, 4th edition, Pearson, 2023, ISBN 978-0-321-41746-6. 3. Elian Greystone, Mastering AI with Python: From Machine Learning Basics to Building Real World AI Applications,3rd edition, 2025, ISBN 979-829214233
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Understand and construct intelligent agent models and basic AI problems Representations. CO 2. Apply search and game-playing techniques to solve AI problems. CO 3. Develop knowledge-based systems using logic programming and reasoning. CO 4. Implement simple machine-learning algorithms and assess their performance.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-304**
Title of the Course : **Software Engineering**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic knowledge of software development fundamentals.	
Course Objectives	This course will enable students: 1. Understand core software engineering principles, processes, life-cycle models, and team structures. 2. Apply requirement modelling, design concepts, estimation techniques, and basic testing methods. 3. Create UML models using standard diagramming techniques and tools. 4. Analyse project planning, quality processes, risks, and project closure activities.	
Contents		**No of Hours**
Unit -1	**Introduction to Software Engineering: Scope of software engineering-** Historical aspects, Economic aspects, Maintenance aspects, Requirements, Analysis, and Design Aspects, Team Development aspects. **The Software Process:** user requirement phase, specification phase, design phase, implementation phase, integration phase, maintenance phase, Capability Maturity Models. **Software Life Cycle Models:** Build and Fix Model, Waterfall, Rapid Prototyping Model, Open-Source Life-Cycle Model, Synchronize and Stabilize Model, Spiral Model, Comparison of Life-Cycle Models. **Teams:** Team Organization, Democratic Team Approach, Classical Chief Programmer Team Approach, Synchronize and Stabilize Teams.	**10**
Unit - 2	**Requirement Modelling:** Data Modelling Concepts, Flow-Oriented Modeling -Creating a Data Flow Model. **From Modules to Objects:** Coincidental Cohesion, Communicational Cohesion, Functional Cohesion. Coupling-Content Coupling, Control Coupling, and Data Coupling. **Planning and Estimating:** Estimating Duration and Cost - Metrics for the Size of a Product, Function point analysis, Intermediate COCOMO, COCOMO II. **Testing:** Quality Issues, Non-Execution-Based Testing-Walkthroughs, Inspections, What Should Be Tested: Utility, Reliability, Robustness, Performance, Correctness. Introduction to	**10**

	White Box Testing and Black Box Testing.	
Unit - 3	**Object modelling using UML:** UML overview, nature and purpose of models, Static view, Use Case view, Activity view, Interaction view, sequence diagram. **Sample Tool**: Argo UML, an open-source tool.	**10**
Unit - 4	**Project planning:** process database, process capability baseline, Process planning: process tailoring, requirements change management. **Quality Planning:** quality concepts, quality process planning, defect prevention planning. **Risk management**: concepts of risk management activities, risk assessment. **Project execution:** review process**,** Project monitoring and control: project tracking & milestone analysis. **Project closure:** The role of closure analysis, closure analysis report.	**11**
Pedagogy	Interactive and inquiry-based teaching with problem-solving activities, tool demonstrations, and short case studies.	
References/ Readings	**Text Books** 1. Stephen R.Schah, Object-Oriented and Classical Software Engineering; 8th Edition,2011, ISBN-13 978-0073376189. 2. Pankaj Jalote, Software Project Management in Practice, PEA, 2002, ISBN-13 978-0201737219. **Reference Books** 1. James Rumbaugh, Ivar Jacobson, Grady Booch; The Unified Modeling Language Reference Manual, 2nd Edition, Pearson Education, ISBN-13 9788177581614 2. Roger S. Pressman, Software Engineering – A practitioner's approach; McGraw-Hill,7th Edition, ISBN-13 978-0073375977. 3. J. Rumbaugh et al; Object Oriented Modelling & Design,2nd Edition, 2005, ISBN-13 978-8131711064.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain the scope of software engineering, software process phases, and major life-cycle models, and team organizational approaches. CO 2. Apply requirement modelling techniques, analyse cohesion–coupling in design, perform basic software estimation, and differentiate major software testing approaches. CO 3. Create UML models such as static, use-case, activity, and sequence diagrams using appropriate modelling tools. CO 4. Analyze software project plans, quality processes, risk management activities, and demonstrate project monitoring and closure techniques.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH 305**
Title of the Course : **Software Engineering Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic Programming Skills (Python/Java/C++) and basic knowledge of Software Development Lifecycle.	
Course Objectives	The course will enable students to: 1. To develop the ability to document and analyze software requirements using structured and object-oriented techniques. 2. To provide hands-on experience in using professional tools to create essential UML models for system design. 3. To enable learners to apply fundamental project planning and tracking techniques, including creating scheduling artifacts and monitoring project progress. 4. To enable learners to integrate and implement the designed specifications by translating models (such as Class and Sequence Diagrams) into a working software prototype.	
Content	**List of Programs/Experiments**	**No of Hours**
	1. Project Initialization: Defining Problem Scope, Functional, and Non-Functional Requirements. 2. Create a Software Requirement Specification (SRS) document as per IEEE. 3. Creating WBS, Gantt, and tracking Gantt chart using any IT PM Tool 4. Developing Entity-Relationship Diagrams (ERD) and Data Flow Diagrams (DFD) 5. Creating Detailed Use Case Diagrams and Specifications. 6. Designing the Static View using Detailed Class Diagrams. 7. Representing Object Lifecycle using State Machine Diagrams (State Diagrams). 8. Representing System Logic using Activity Diagrams. 9. Object Interactions using Sequence Diagrams. 10. Project Implementation	**30**
Pedagogy	Hands-on implementation of design and analysis artifacts using specialized modelling tools (like ArgoUML, draw.io, or equivalent), and practical application of project planning and tracking techniques.	
References/ Readings	**Text Books** 1. Stephen R.Schah, Object-Oriented and Classical Software Engineering; 8th Edition,2011, ISBN-13 978-0073376189.	

	2. Pankaj Jalote, Software Project Management in Practice, PEA, 2002, ISBN-13 978-0201737219. **Reference Books** 1. James Rumbaugh, Ivar Jacobson, Grady Booch; The Unified Modeling Language Reference Manual, 2nd Edition, Pearson Education, ISBN-13 9788177581614 2. Roger S. Pressman, Software Engineering – A practitioner's approach; McGraw-Hill,7th Edition, ISBN-13 978-0073375977. 3. J. Rumbaugh et al; Object Oriented Modelling & Design,2nd Edition, 2005, ISBN-13 978-8131711064.
Course Outcomes	After going through this course, the student will be able to: CO 1. Analyze a problem statement to document a structured and detailed Software Requirements Specifications (SRS). CO 2. Apply structured design techniques (DFD, ERD) CO 3. Create comprehensive UML Object-Oriented models (Use Case, Class, Activity, Sequence Diagrams) using professional modelling tools. CO 4. Demonstrate fundamental project planning and tracking skills, including creating WBS, Gantt charts, and monitoring project milestones.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-306**
Title of the Course : **Theory of Computation**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic knowledge of mathematics	
Course Objectives	This course will enable students: 1. Understand the fundamental concepts of formal languages, grammars, and automata as the foundation of computation. 2. Design and construct computational models 3. Analyse the languages using formal methods like the Pumping Lemma, grammar transformations, and automata minimization. 4. Explore the limits of computation through Turing machines, decidability, reducibility, and computational complexity.	
Contents		**No of Hours**
Unit -1	**Regular Languages and Finite Automata:** Basic terminology (Language, Alphabet, String), Regular Languages and Regular Expressions, Deterministic Finite Automata (DFA), Pumping Lemma for Regular Languages, Moore and Mealy Machines. **Nondeterminism and Kleene's Theorem:** Nondeterministic Finite Automata (NFA), Nondeterministic Finite Automata with Λ-transitions (ε-NFA), Kleene's Theorem, Minimization of Finite Automata.	**12**
Unit- 2	**Context–Free Languages:** Definitions & Examples, Regular Grammars, Derivation Trees and Ambiguity, Simplification of CFGs (Elimination of Null & Unit Productions), Chomsky Normal Form, Greibach Normal Form, Pumping Lemma for Context–Free Languages. **Pushdown Automata:** Definition, Deterministic Pushdown Automaton, PDA Corresponding to a given CFG, A CFG corresponding to a given PDA.	**12**
Unit -3	**Turing Machine:** Computing a Partial function with a Turing machine, Variations of Turing Machine, Nondeterministic Turing Machine, Church- Turing thesis.	**11**
Unit -4	**Recursively Enumerable languages:** Recursively enumerable and recursive, Unrestricted Grammars, Context-Sensitive Language, and Chomsky Hierarchy **Unsolvable Problems:** Reducing one problem to another: Halting	**10**

	problem, Rice's Theorem.	
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings	**Textbooks:** 1. John C. Martin, Introduction to languages and the theory of computation, Tata McGraw-Hill, 4th edition,2010, ISBN 978-0074637227 2. K.L.P Mishra, Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI, 3rd edition,2008, ISBN 978-8120329683 **Reference Books:** 1. Kamala Krithivasan, Rama R., Introduction to Formal Languages, Automata Theory and Computation, 3rd edition, Pearson Education India, 2018, ISBN: 9788131723562. 2. Adesh K. Pandey, Theory of Automata and Computation, S.K. Kataria & Sons, 2nd edition,2013, ISBN 978-9350141571 3. S.R. Jena, S.K. Swain, Theory of computation and application: Automata, Formal languages, Computational Complexity, Laxmi Publications Private Limited,2nd edition, 2020, ISBN 978-9386202154	
Course Outcomes	Upon completion of the course, students will be able to CO 1. Explain the fundamental concepts of formal languages, grammars, and automata as the theoretical foundation of computer science. CO 2. Identify the capabilities and limitations of regular and context-free language models. CO 3. Construct deterministic and nondeterministic models for various computational problems. CO 4. Analyse and evaluate the computational power of complex computational models based on real-world problems.	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-307**
Title of the Course : **Theory of Computation Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course	Prior theoretical knowledge of formal languages and grammars, DFA / NFA / ε-NFA, context-free grammars (CFG), pushdown automata (PDA), basics of Turing machines, Chomsky hierarchy (Types 0–3 languages), and fundamentals of programming (C / C++ / Java / Python).	
Course Objectives	The course will enable students to: 1. Provide hands-on experience in implementing and simulating automata models such as DFA, NFA, ε–NFA, and PDA. 2. Develop practical understanding of context-free grammars, language generation, and grammar transformation techniques. 3. Enable students to construct and test Turing Machines for computational problem solving and language recognition. 4. Strengthen analytical thinking and problem-solving abilities using simulation tools and programming logic related to formal language theory.	
Contents	**List of Programs/Experiments**	**No of Hours**
	1. Implement a program to simulate a Deterministic Finite Automaton (DFA) for string validation over {a, b}. 2. Design and simulate a DFA to recognize binary strings with an even number of 1's. 3. Design and simulate NFA and ε–NFA and convert them into equivalent DFA. 4. Implement minimization of finite automata using the state equivalence method. 5. Design Moore and Mealy Machines for input-output sequence processing. 6. Create Context-Free Grammar (CFG) and generate valid strings using leftmost & rightmost derivations. 7. Implement Pushdown Automaton (PDA) for balanced parentheses or palindromes. 8. Verify Chomsky Normal Form (CNF) conversion for a given CFG. 9. Design a Turing Machine to compute unary addition (e.g., 111 + 11). 10. Construct a Turing Machine to check whether a string is a palindrome over {a, b}.	**30**
Pedagogy	The lab follows interactive, inquiry-based, and tool-assisted learning	

	through hands-on simulations of automata and Turing Machines using software tools like JFLAP. It emphasizes problem-solving, collaborative learning, critical thinking
References/ Readings	**Textbooks:** 1. John C. Martin, Introduction to languages and the theory of computation, Tata McGraw-Hill, 4th edition,2010, ISBN 978-0074637227 2. K.L.P Mishra, Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI, 3rd edition,2008, ISBN 978-8120329683 **Reference Books:** 1. Kamala Krithivasan, Rama R., Introduction to Formal Languages, Automata Theory and Computation, 3rd edition, Pearson Education India, 2018, ISBN: 9788131723562. 2. Adesh K. Pandey, Theory of Automata and Computation, S.K. Kataria & Sons, 2nd edition,2013, ISBN 978-9350141571 3. S.R. Jena, S.K. Swain, Theory of computation and application: Automata, Formal languages, Computational Complexity, Laxmi Publications Private Limited,2nd edition, 2020, ISBN 978-9386202154
Course Outcomes	Upon completion of the course, students will be able to CO 1. Design and simulate finite automata and regular expressions for recognizing regular languages. CO 2. Construct and test context-free grammars (CFGs) and implement basic parsing or string generation techniques. CO 3. Apply algorithms for language recognition such as NFA-to-DFA conversion, DFA minimization, and grammar transformations. CO 4. Develop and analyze simple Turing Machine simulations to understand computability and problem-solving using formal models.

Professional Electives

Name of the Programme : B.E. in Information Technology
Course Code : ITH-321
Title of the Course : Cloud infrastructure & services
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic Computer Systems Knowledge, Virtualization Basics, and Fundamentals of Distributed Systems	
Course Objectives	This course will enable students: 1. To provide foundational knowledge of cloud computing concepts, service models, deployment models, and cloud architecture. 2. To develop an understanding of cloud infrastructure technologies, including virtualization, hypervisors, cloud storage, and distributed data center architectures. 3. To enable students to work with major cloud platforms (AWS, Azure, GCP) and understand compute, storage, and database services along with Hadoop and MapReduce. 4. To strengthen student capability in cloud security, performance, management, and exposure to emerging cloud trends.	
Contents		**No of Hours**
Unit 1	**Introduction to Cloud Computing:** Definition of Cloud Computing, Evolution and roots of Cloud Computing, Characteristics of Cloud, Advantages and Disadvantages of Cloud. **Service Models**: IaaS, PaaS, SaaS: Features, Examples, advantages and disadvantages of different models, Deployment Models: Public, Private, Hybrid, Community Clouds, advantages and disadvantages of deployment models, Cloud Reference Architecture (NIST), Benefits and Challenges of Cloud Adoption, Applications of Cloud.	**11**
Unit 2	**Cloud Infrastructure** Virtualization: Basics, Benefits, Types of virtualization, Advantages and Disadvantages of Virtualization, Hypervisors (Type 1 & Type 2), Protection rings, Approaches to virtualization: Full, Para, and Hardware Assisted. AWS Architecture, Simple Storage Service (S3), Working of S3, **Data Centres**: Basics, Components, and Types. HBase Architecture, Components, Advantages, and Disadvantages of HBase. Cross Data Centre Replication, Types of cross data centre replication.	**12**

Unit 3	**Cloud Services and Platforms** Introduction to Major Cloud Providers: AWS, Azure, Google Cloud**.** Managing Compute Resources: AWS EC2/Azure VM/GCP Compute Engine, Google Cloud Platform, Benefits of GCP **Database Services:** RDS, features, and Benefits. **Map Reduce**: Architecture, Components, Applications, Map Reduce examples, Hadoop Basics, types of schedulers in Hadoop, fault Tolerance in Hadoop.	**10**
Unit 4	**Cloud Security, Management & Emerging Technologies** **Cloud Security:** Challenges and risks, components of cloud security architecture, principles, threats, Security controls: types of controls, Data security: types, strategies, Application security: types. Strategies: Virtual machine security types, strategies, and benefits. Availability, Security, Scalability, and Reliability in the cloud. Cloud performance: benefits, metrics, and cloud performance strategies. **Emerging Cloud Trends:** Edge Computing, Multi-Cloud & Hybrid Cloud, AI in Cloud Operations, Green Cloud Computing.	**12**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Text books** 1. Thomas Erl (Author), Eric Barceló Monroy, Cloud Computing: Concepts, Technology, Security & Architecture, 2nd Edition, Pearson Education, 2024, 978-8196943219 2. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2021, ISBN: 9788126541256. **Reference Books** 1. A. Srinivasan, J Suresh, Cloud Computing: A Practical Approach for Learning and Implementation, Pearson Education, 2014, ISBN: 978-8131776513 2. Thomas Erl, Zaigham Mahmood, Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson Education, 2023, ISBN: 978-0133387513. 3. Michael Wittig, Andreas Wittig, Amazon Web Services in Action (3rd Edition), Manning Publications, 2023, ISBN: 978-1617292880	
Course Outcomes	After completing this course, students will be able to: CO 1. Explain the principles, characteristics, and models of cloud computing and identify the benefits and challenges of cloud adoption. CO 2. Demonstrate understanding of virtualization technologies, hypervisors, and cloud storage mechanisms, and data center	

	architectures. CO 3. Use major cloud platforms to configure compute resources, storage services, and database solutions, and describe Hadoop and MapReduce functioning. CO 4. Evaluate cloud security requirements and performance metrics, and analyze emerging cloud technologies and their real-world applications.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-322**
Title of the Course : **Cloud Infrastructure and Services Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic knowledge of computer networks, operating systems, and programming.	
Course Objectives	The course will enable students to: 1. Understand and configure different cloud service models and deployment models. 2. Demonstrate virtualization, hypervisors, VM creation, snapshots, and resource allocation. 3. Configure and operate Hadoop, MapReduce, and HBase environments for distributed processing. 4. Implement cloud security, IAM policies, monitoring, replication, and performance strategies.	
Contents	**List of programs**	**No of Hours**
	1. Study about the Services models (IaaS, PaaS, SaaS) with real time example 2. Installation & use of Virtualization software (VirtualBox / VMware) VM creation. 3. Deploying different OS on OnWorks 4. Deployment of different applications on the Google cloud platform Or AWS 5. AWS S3: Bucket creation, versioning, policies, lifecycle rules 6. A case study on Hadoop, MapReduce, and HDFS 7. Install a Hadoop single-node cluster and run simple applications like word count. 8. Deployment and Secure Connectivity of a Managed Cloud Database (Relational/NoSQL) with a Cloud Compute Instance 9. Configure IAM Users/Roles with Least-Privilege Access Policies using qwiklabs or AWS 10. Mini Project: Building and Analysing a website using Google Sites, Google Search Console, and Google Analytics.	**30**
Pedagogy	Integration of instructional learning, constructive thinking, inquiry-based, collaborative, experiential, and problem-solving approaches.	
References / Readings	**Text Books** 1. Thomas Erl, Zaigham Mahmood, Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson, 2013, ISBN: 9780133387520	

	2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill, 2022, ISBN: 978-0124114548 **Reference Books** 1. Amazon Web Services, AWS Certified Solutions Architect Official Study Guide, Wiley, 2016, ISBN: 9781119138556 2. Arshdeep Bahga, Vijay Madisetti, Cloud Computing: A Hands-On Approach, Universities Press, 2023, ISBN: 978-1494435141 3. Hadoop: The Definitive Guide (4th Edition), O'Reilly Media, 2021, ISBN: 978-1491901632
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Demonstrate knowledge of cloud platforms, service models, deployment models, and cloud infrastructure. CO 2. Configure and manage virtualized environments using virtualization tools and understand hypervisor types. CO 3. Illustrate Hadoop, MapReduce, and HBase installation, configuration, and data management. CO 4. Implement cloud security concepts, including IAM users/roles, access policies, secure storage, monitoring, and performance optimization on cloud platforms.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-323**
Title of the Course : **Mobile Computing**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basics of Communication Technologies	
Course Objectives	**This course will enable students:** 1. To understand the basic concepts of mobile computing and wireless networking 2. To be familiar with the MAC, IP, and Transport layer protocols and 3. To learn and understand Mobile Ad-Hoc networks. 4. To Know Operating systems used in Mobile computing and also mobile commerce.	
Contents		**No of Hours**
Unit -1	**Introduction to Mobile Computing and Wireless Networking** Basics of Mobile Computing, Mobile Computing vs. Wireless Networking, Mobile Computing Applications, Characteristics of Mobile Computing, Structure of Mobile Computing Application, Cellular Mobile Communication -Generations of Cellular Communication Technologies, Global System for Mobile Communications (GSM) -GSM Services, System Architecture of GSM, GSM Security **MAC Protocols** Properties Required of MAC Protocols, Wireless MAC Protocols: Some Issues-The Hidden and Exposed Terminal Problems in an Infrastructure-less Network, A Taxonomy of MAC Protocols, Fixed Assignment Schemes- Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Random Assignment Schemes -ALOHA Scheme, The CSMA Scheme, Reservation-based Schemes-MACA, The 802.11 MAC Standard- Infrastructure-based Mode.	**12**
Unit - 2	**Mobile Internet Protocol** Mobile IP, Packet Delivery, Overview of Mobile IP, Desirable Features of Mobile IP, Key Mechanisms Used in Mobile IP, Route Optimization, Dynamic Host Configuration Protocol (DHCP) -Significance of Dynamic Host Configuration Protocol **Mobile Transport Layer** Overview of TCP/IP, Terminologies of TCP/IP, Architecture of TCP/IP, An Overview of the Operation of TCP, Application Layer	**11**

	Protocols of TCP, TCP/IP versus ISO/OSI Model, Adaptation of TCP Window, Improvement in TCP Performance-Traditional Networks, Popular TCP Congestion Control Algorithms, TCP in Mobile Networks.	
Unit - 3	**Mobile Ad Hoc Networks** A Few Basic Concepts -How is an Ad Hoc Network Set Up without the Infrastructure Support? Why is Routing in a MANET a Complex Task? Battery Basics, Characteristics of Mobile Ad Hoc Networks (MANETs)- MANET Operational Constraints, Applications of MANETs, MANET Design Issues – Routing, Essentials of Traditional Routing Protocols -Link State Protocols (LSP), Distance Vector (DV) Protocols. **Routing in MANETs** A Few Basic Concepts - Routing in MANETs vs. Routing in Traditional Networks, A Classification of Unicast MANET Routing Protocols, Features of MANET Routing Protocols, Popular MANET Routing Protocols - Destination-Sequenced Distance-Vector Routing Protocol, Dynamic Source Routing (DSR) Protocol, Ad Hoc On-demand Distance Vector (AODV), Zone Routing Protocol, Multicast Routing Protocols for MANET, Vehicular Ad Hoc Networks (VANETs), MANET vs VANET, Security Issues in a MANET, Attacks on Ad Hoc Networks, Security Attack Countermeasures.	**11**
Unit - 4	**Operating Systems for Mobile Computing** A Few Basic Concepts, Special Constraints and Requirements of Mobile O/S - Special Constraints, Special Service Requirements, A Survey of Commercial Mobile Operating Systems - Windows Mobile, Palm OS, Symbian OS, iOS, Android, Blackberry Operating System, A Comparative Study of Mobile OSs. **Mobile Commerce** Applications of M-Commerce -Business-to-Consumer (B2C) Applications, Business-to-Business (B2B) Applications, Structure of Mobile Commerce, Pros and Cons of M-Commerce, Mobile Payment Systems -Mobile Payment Schemes, Desirable Properties of a Mobile Payment System, Mobile Payments Solutions, Process of Mobile Payment, Security Issues.	**11**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Text Books :** 1. Prasant Kumar Pattnaik, Rajib Mall, Fundamentals of Mobile Computing, Second Edition, PHI Learning, 2016, ISBN-13: 978-81-203-5181-3. 2. Jochen Schiller, Mobile Communications, Second Edition, Pearson	

	Education, 2013, ISBN-13: 978-81-317-2426-2 **Reference Books:** 1. Raj Kamla, Mobile Computing, Second Edition, Oxford University Press, 2013, ISBN -13: 978-0-19-806891-4 2. Asoke K Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing – Technology, Applications and Service Creation, Second Edition, Tata McGraw-Hill Education Private Limited, 2010, ISBN-13: 978-0-07-014457-6 3. Prashanta Kumar Patra, Sanjit Kumar Dash, Mobile Computing, Second Edition, Scitech Publications (India) Pvt. Ltd., ISBN -13: 978-81-8371-457-0
Course Outcomes	Upon completion of the course, students will be able to: CO 1. List key components and application areas of Mobile Computing CO 2. Explain the features of Mobile IP and Mobile Transport Layer CO 3. Analyze Mobile Ad Hoc Network and its internal Characteristics. CO 4. Study Different Operating systems used in Mobile Computing.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-324**
Title of the Course : **Mobile Computing Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

<table>
<tr><td>Pre-requisites for the Course:</td><td colspan="2">Basics of Communication Technologies</td></tr>
<tr><td>Course Objectives</td><td colspan="2">The course will enable students to:
1. To provide hands-on experience with wired and wireless network architectures and protocols.
2. To familiarize students with mobile network simulation tools (OPNET, NS2).
3. To understand and implement wireless communication techniques such as CDMA and frequency reuse.
4. To enable students to develop Android applications related to messaging, email, mapping, and connectivity.</td></tr>
<tr><td rowspan="2">Contents</td><td>List of programs</td><td>No of Hours</td></tr>
<tr><td>1. Study and Analysis of wired and wireless networks
2. Study of the OPNET tool for modeling and simulation of different cellular standards.
3. To implement a mobile network using open source software like NS2.
4. Implement Code Division Multiple Access (CDMA)
5. To write a programme to implement the concept of frequency reuse when given the size of the geographical area.
6. Develop an application to send SMS.
7. Write a program to send an email.
8. Write a program to display a map of the given location/position using a map view.
9. Study and Analysis of Bluetooth.
10. Study of Mobile IP</td><td>30</td></tr>
<tr><td>Pedagogy</td><td colspan="2">Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.</td></tr>
<tr><td>References/ Readings</td><td colspan="2">Text Books
1. Prasant Kumar Pattnaik, Rajib Mall, Fundamentals of Mobile Computing, Second Edition, PHI Learning, 2016, ISBN-13: 978-81-203-5181-3.
2. William Stallings, Wireless Communications & Networks, Pearson</td></tr>
</table>

	Education. ISBN: 9788131720936. **Reference Books** 1. Jochen Schiller, Mobile Communications, Second Edition, Pearson Education, 2013, ISBN-13: 978-81-317-2426-2 2. Raj Kamla, Mobile Computing, Second Edition, Oxford University Press, 2013, ISBN -13: 978-0-19-806891-4 3. Asoke K Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing – Technology, Applications and Service Creation, Second Edition, Tata McGraw Hill Education Private Limited, 2010, ISBN-13: 978-0-07-014457-6
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Analyze and compare wired and wireless networks based on architecture, performance, and mobility. CO 2. Use OPNET and NS2 tools to model, simulate, and evaluate mobile/wireless networks. CO 3. Implement wireless communication techniques like CDMA and frequency reuse using programming. CO 4. Develop and demonstrate basic Android applications to send SMS, email, and display maps using API.

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-325**
Title of the Course : **Parallel Computing**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic programming in C/C++, data structures, operating systems concepts, and fundamental computer architecture.	
Course Objectives	This course will enable students: 1. Understand fundamental concepts, models, and performance metrics of parallel computing. 2. Differentiate between shared-memory and distributed-memory architectures and apply synchronization principles. 3. Develop parallel programs using OpenMP and MPI for real-world computing systems. 4. Explore GPU programming, CUDA, and modern parallel computing applications such as clusters, databases, and edge computing.	
Contents		**No of Hours**
Unit-1	**Introduction to Parallel Computing:** Need for parallelism, Instruction-level parallelism, Hardware multi-threading, Classification of Parallel Computers: Flynn's Taxonomy (SISD, SIMD, MISD, MIMD), Performance metrics: Speedup, Efficiency, Amdahl's Law, Gustafson's Law. **Physical Organization of parallel computer:** Architecture of an Ideal Parallel Computer, Interconnection Networks for Parallel Computers, Network Topologies.	**11**
Unit - 2	**Parallelism Concepts**: Shared memory v/s Distributed memory, Threads, Processes, Data parallelism, Task parallelism, Synchronization Principles, Race conditions & Deadlocks. **Communication model of parallel platforms:** Shared-Address-Space Platforms, Message-Passing Platforms. **Distributed Memory Programming Model – MPI** Introduction to MPI, MPI Execution Model: Processes and Distributed Memory, The trapezoidal rule in MPI, Dealing with I/O, Collective communication, MPI-derived data types, Basic MPI Programs, Advantages and Limitations of MPI.	**12**
Unit - 3	**Shared Memory Programming Model – OpenMP** Introduction to OpenMP, OpenMP Execution Model: Threads and Shared Memory, The trapezoidal rule, Scope of variables, The reduction clause, loop carried dependency, scheduling loops, Advantages and Limitations of OpenMP.	**11**

Unit - 4	**GPU programming with CUDA:** GPUs and GPGPU, GPU architectures, Heterogeneous computing, Threads, blocks, and grids, Nvidia compute capabilities and device architectures, Vector addition, Returning results from CUDA kernels, CUDA trapezoidal rule I, CUDA trapezoidal rule II: improving performance, CUDA trapezoidal rule III: blocks with more than one warp.	**11**
Pedagogy	A blend of lectures, programming (OpenMP, MPI), tutorials, and real-world case studies to reinforce conceptual understanding and practical skills.	
References/ Readings	**Text books** 1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. Introduction to Parallel Computing. 2nd ed. Addison Wesley, 2003, ISBN: 0-201-64865-2 2. Pacheco, Peter S., and Matthew Malensek. An Introduction to Parallel Programming. 2nd ed. Morgan Kaufmann, 2021. ISBN: 978-0-12-804605-0. **Reference books** 1. M J Quinn, Parallel Computing: Theory and Practice, Second Edition, Tata McGraw-Hill, 1993, ISBN-13: 978-0070512948. 2. David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors, AHands-on Approach, Second Edition, Elsevier, 2012, ISBN-13: 978-0124159921. 3. M J Quinn, Parallel Programming in C with MPI and OpenMP, Tata McGraw-Hill, 2003, ISBN-13: 978-0072822564.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain the need for parallel computing; classify architectures using Flynn's taxonomy, and compute performance metrics like speedup and efficiency. CO 2. Apply concepts of threads, processes, data parallelism, task parallelism, and synchronization to design correct parallel programs. CO 3. Develop and debug parallel programs using OpenMP for shared-memory systems and MPI for distributed-memory systems. CO 4. Analyse GPU architectures and real-world parallel computing applications, including clusters, databases, and edge computing.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-326**
Title of the Course : **Parallel Computing Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of marketing principles. Interest in digital technologies, social media, and online business models	
Course Objectives:	This course will enable students: 1. Understand fundamental concepts, models, and performance metrics of parallel computing. 2. Differentiate between shared-memory and distributed-memory architectures and apply synchronization principles. 3. Develop parallel programs using OpenMP and MPI for real-world computing systems. 4. Explore GPU programming, CUDA, and modern parallel computing applications such as clusters, databases, and edge computing.	
Contents	**List of programs**	**No of Hours**
	1. Measure speedup and efficiency of a CPU program using Amdahl's and Gustafson's laws. 2. Compute the sum of an array of 100M integers using C++ and OpenMP. Calculate speedup. 3. Compare SISD, SIMD, and MIMD performance using a matrix operation. 4. Write an OpenMP program to compute the sum of an array of 1 million integers using the reduction clause. 5. Find the Maximum Element of an Array. Implement using a critical section and then using reduction. Compare performance. 6. Demonstrate the difference between private, firstprivate, and shared using a simple vector initialization. 7. Implement the trapezoidal rule using MPI, where each process computes a partial integral over its subrange. 8. Count character frequencies of a text file using: MPI_Scatter, Local histograms, and MPI_Reduce with MPI_SUM 9. Distribute parts of an array using MPI_Scatter, multiply each element by 2, and gather results using MPI_Gather. 10. Create a case study comparing GPU and CPU throughput for large data operations.	**30**
Pedagogy:	A blend of hands-on programming (OpenMP, MPI), tutorials, and real-world case studies to reinforce conceptual understanding and practical	

	skills.
References/ Readings	**Text books** 1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. Introduction to Parallel Computing. 2nd ed. Addison Wesley, 2003, ISBN: 0-201-64865-2 2. Pacheco, Peter S., and Matthew Malensek. An Introduction to Parallel Programming. 2nd ed. Morgan Kaufmann, 2021. ISBN: 978-0-12-804605-0. **Reference books** 1. M J Quinn, Parallel Computing: Theory and Practice, Second Edition, Tata McGraw-Hill, 1993, ISBN-13: 978-0070512948. 2. David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors, AHands-on Approach, Second Edition, Elsevier, 2012, ISBN-13: 978-0124159921. 3. M J Quinn, Parallel Programming in C with MPI and OpenMP, Tata McGraw-Hill, 2003, ISBN-13: 978-0072822564.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain the need for parallel computing; classify architectures using Flynn's taxonomy, and compute performance metrics like speedup and efficiency. CO 2. Apply concepts of threads, processes, data parallelism, task parallelism, and synchronization to design correct parallel programs. CO 3. Develop and debug parallel programs using OpenMP for shared-memory systems and MPI for distributed-memory systems. CO 4. Analyse GPU architectures and real-world parallel computing applications, including clusters, databases, and edge computing.

Semester VI

Major Courses

Name of the Programme : B.E. in Information Technology
Course Code : ITH-308
Title of the Course : Web Technology
Number of Credits : 2
Effective from AY : 2024-25

Pre-requisites for the Course:	Basic Programming (Python/C/C++)	
Course Objectives	This course will enable students to: 1. To understand the fundamentals of web technology, client-server architecture, HTML5/CSS3/JavaScript, and responsive design principles. 2. To introduce advanced client-side scripting, frameworks like React, AJAX, and JSON handling for interactive web pages. 3. To develop server-side programming skills using Node.js/Express.js, including REST APIs, CRUD operations, sessions, cookies, and form processing. 4. To explore web application deployment, Git version control, hosting platforms, and basic security concepts like XSS, CSRF, HTTPS, and SSL.	
Contents		**No of Hours**
Unit - 1	**Introduction to Web Technology and Client-side Development:** Overview of Web Technology, History and Evolution, Internet and Web Architecture: HTTP/HTTPS, Client-Server Model, HTML5: Structure, Semantic Elements**,** CSS3: Styling, Box Model, Flexbox, Grid JavaScript Basics: Syntax, DOM Manipulation, Event Handling Responsive Web Design Principles.	**8**
Unit - 2	**Advanced Client-side Scripting and Frameworks:** JavaScript ES6+ Features (let/const, arrow functions, promises) Introduction to DOM API and AJAX, Client-side Framework Overview (ReactJS or Angular basic concepts), JSON, and Local Storage Basic Project: Building Interactive Web Pages.	**7**
Unit - 3	**Server-side Web Development** Introduction to Server-side Programming: Overview of Node.js Web Servers and Databases: Concepts of REST APIs, CRUD Operations Express.js Framework Basics: Session Management and Cookies	**8**

	File Handling and Form Processing	
Unit - 4	**Web Application Deployment and Security Basics** Web Hosting and Domain Basics Version Control with Git and GitHub (Basics) Introduction to Web Security Concepts: XSS, CSRF, Injection Attacks HTTPS and SSL Certificates, Deployment of Simple Web Applications on Platforms (Heroku, Netlify)	**7**
Pedagogy	Inquiry-Based Learning, Reflective, Integrative, Conceptual Learning.	
References/ Readings	**Text Books** 1. P.J. Deitel, H.M. Deitel, and Abbey Deitel, Internet and World Wide Web: How to Program, Pearson, 2000, ISBN 978-0130161437. 2. Robert W. Sebesta, Programming the World Wide Web, Pearson,4th Edition,2011, ISBN 978-8131764589. **Reference Books** 1. Jeffrey C Jackson, Web Technologies: A Computer Science Perspective, Pearson, 2006, ISBN 978-0131856035. 2. Achyut Godbole, Atul Kahate, Web Technologies: TCP/IP, Web/ Java Programming, and Cloud Computing, 3rd Edition, McGraw-Hill,2013, ISBN 978-9383286577. 3. Ethan Brown; Web Development with Node and Express; 2nd Edition; O'Reilly; 2019; ISBN-978-9352139446.	
Course Outcomes	After going through this course, the students will be able to: CO 1. Explain web technology fundamentals, client-server architecture, HTML5/CSS3/JavaScript, server-side programming, and deployment/security concepts CO 2. Analyse and evaluate client-side frameworks, server-side technologies like Node.js/Express, security vulnerabilities, and deployment strategies. CO 3. Apply web technologies to develop responsive front-end pages, server-side applications with CRUD operations/sessions, and deploy using Git/platforms. CO 4. Create full-stack dynamic web applications integrating front-end, back-end APIs, security measures, and live deployment to meet specific requirements.	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-309**
Title of the Course : **Web Technology Lab**
Number of Credits : **2**
Effective from AY : **2024-25**

<table>
<tr><td>Pre-requisites for the Course</td><td colspan="2">Basic knowledge of computers and programming fundamentals.</td></tr>
<tr><td>Course Objectives</td><td colspan="2">This course will enable students to:
1. To design and develop structured, responsive, and visually appealing web pages using HTML and CSS.
2. To implement client-side interactivity using JavaScript for validation, dynamic content, and user interface enhancements.
3. To develop dynamic web applications by integrating AJAX/Fetch API, server-side scripting, and database connectivity.
4. To deploy and test complete web applications using modern web development tools and platforms.</td></tr>
<tr><td rowspan="2">Contents</td><td>List of Programs</td><td>No of Hours</td></tr>
<tr><td>Create a Personal Portfolio Webpage.
1. Build a Multi-Page Website with Navigation.
2. Design a Responsive Layout Using CSS Flexbox/Grid.
3. Style a Webpage Using External CSS.
4. Perform Form Validation Using JavaScript.
5. Create a Dynamic Carousel.
6. Build an Interactive To-Do List Application
7. Develop a Simple Calculator Using JavaScript.
8. Implement Ajax to Fetch Data Without Reloading.
9. Create a three-tier web application using PHP, AJAX, and a database.
10. Create a basic server-side web application using Node.js with Express.js (or PHP with MySQL).
11. Build and Deploy a Mini Web Application</td><td>60</td></tr>
<tr><td>Pedagogy</td><td colspan="2">Hands-on, demonstration-based learning with progressive experiments, collaborative practice, continuous feedback, and project-based application using modern web development tools.</td></tr>
<tr><td>References/ Readings</td><td colspan="2">Text books
1. Kogent Learning Solutions Inc.; Web Technologies: HTML, Javascript, PHP, Java, JSP, ASP.NET, XML, and AJAX, Black Book, Dream tech Press,2015, ISBN:978-81-7722-997-4
2. Jon Duckett, JavaScript and jQuery: Interactive Front-End Web</td></tr>
</table>

	Development, Wiley,2014; ISBN-9781118531648 **Reference books** 1. Paul Deitel, Harvey Deitel, Abbey Deitel; Internet and World Wide Web. How to program: Fifth Edition, Pearson Education India, 2018, ISBN: 978-9352868599. 2. Achyut Godbole, Atul Kahate, Web Technologies: TCP/IP, Web/ Java Programming, and Cloud Computing, 3rd Edition, McGraw-Hill, 2013, ISBN 978-9383286577 3. Mridula Parihar, et. al. – ASP .NET Bible; Publisher: John Wiley & Sons (2002); ISBN: 978- 0764548161
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Students will be able to create well-structured, responsive, and styled web pages using HTML and CSS. CO 2. Students will be able to implement interactive client-side features using JavaScript and DOM manipulation. CO 3. Students will be able to develop dynamic web applications by integrating server-side scripts, APIs, and databases. CO 4. Students will be able to deploy, test, and demonstrate complete web applications on web hosting platforms.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-310**
Title of the Course : **Augmented and Virtual Reality**
Number of Credits : **2**
Effective from AY : **2024-25**

Pre-requisites for the Course	Basic Programming (Python/C/C++)	
Course Objectives	This course will enable students: 1. To understand the basic concepts of Virtual Reality (VR) and Augmented Reality (AR). 2. To know the input and output devices of VR and AR. 3. To understand the interaction techniques of VR and AR. 4. To discuss applications of AR and VR in various industries.	
Contents		**No of Hours**
Unit-1	**Introduction to Virtual Reality** Defining Virtual Reality, History of VR, Introduction to VR- Four Key Elements of Virtual Reality Experience, Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality. **Representing the Virtual World** Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR, and Haptic Representation in VR.	**7**
Unit - 2	**The Geometry of Virtual Worlds & The Physiology of Human Vision** Geometric Models, Changing Position and Orientation, Axis-Angle Representations of Rotation, Viewing Transformations, Chaining the Transformations, Human Eye, eye movements & implications for VR. **Visual Rendering, Perception, and Interactive Technique** **Rendering-**Visual Rendering Systems, Aural Rendering Systems, Haptic Rendering Systems **Interacting with the Virtual World-** Manipulating a Virtual World: Manipulation Methods-Direct user control, physical control, and virtual control. Navigating in a Virtual World: Wayfinding.	**8**
Unit - 3	**Augmented Reality (AR):** Taxonomy, Technology and Features of Augmented Reality, AR Vs VR, Challenges with AR, AR systems	**8**

	and functionality, Augmented Reality Methods, Visualization Techniques for Augmented Reality, Enhancing interactivity in AR Environments, Evaluating AR systems **Computer Vision for Augmented Reality** Computer Vision for Augmented Reality - Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Simultaneous Localization and Mapping, Outdoor Tracking	
Unit - 4	**AR Techniques- Marker-based & Markerless tracking. Marker-based approach-** Introduction to marker-based tracking, types of markers, marker camera pose and identification, visual tracking, and mathematical representation of matrix multiplication. Marker types- Template markers, 2D barcode markers, imperceptible markers. **Marker-less approach**- Localization-based augmentation, real-world examples. Tracking methods- Visual tracking, feature-based tracking, hybrid tracking, and initialization and recovery.	**7**
Pedagogy	Learner-centric teaching	
References/ Readings	**Text Books** 1. William R Sherman and Alan B Craig, "Understanding Virtual Reality, Interface, Application and Design",(The Morgan Kaufmann Series in Computer Graphics), Morgan Kaufmann Publishers, San Francisco, CA, 2002, ISBN 1558603530, 9781558603530. 2. Schmalstieg, Hollerer, "Augmented Reality: Principles & Practice" by Pearson Education India; First edition (12 October 2016), ISBN-10: 9332578494. **Reference Books** 1. Burdea, Grigore C and Philippe Coiffet, "Virtual Reality Technology", Wiley Interscience, India, 2003, ISBN 0471360899, 9780471360896 2. Alan B Craig, William R Sherman, Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann Publishers, 2009, ISBN: 978-0-12-374943-7. 3. Sanni Siltanen- Theory and applications of marker-based augmented reality. Julkaisija – Utgivare Publisher. 2012. ISBN 978-951-38-7449-0	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Understand the fundamental concepts of AR and VR and describe the input and output devices used. CO 2. Apply the various representations and techniques in AR and VR. CO 3. Analyze the methods required for hardware and software in AR and VR. CO 4. Design and implement the real-world examples of marker-based and marker less augmented reality applications.	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-311**
Title of the Course : **Augmented and Virtual Reality Lab**
Number of Credits : **2**
Effective from AY : **2024-25**

Pre-requisites for the Course	Fundamentals of computer graphics and basic programming knowledge.	
Course Objectives	The course will enable students to: 1. The fundamental tools, workflows, and development environments used in AR and VR application development using Unity. 2. Applying 3D modelling, lighting, texturing, and interaction techniques to build realistic and engaging AR/VR experiences. 3. Interactive AR applications using AR Foundation and mobile-based deployment workflows. 4. VR interactive techniques using the XR Interaction Toolkit and device simulation to prototype virtual environments.	
Contents	**List of programs**	**No of Hours**
	1. Installation of Unity and Visual Studio, setting up Unity for VR development, and understanding the documentation of the same. 2. Develop a scene in Unity that includes: a cube, a plane, and a sphere, and apply transformations on the 3 game objects. 3. Create and manipulate 3D primitive objects and apply camera projections to understand scene composition. 4. Import 3D models from Unity Asset Store and apply lighting, materials, and visual effects to enhance realism. 5. Model simple 3D objects using Blender and import them into Unity; apply textures, materials, and shaders. 6. Create an interactive AR application using AR Foundation with plane detection and virtual object placement. 7. Add audio, text, and particle effects to AR objects to create enhanced AR experiences. 8. Develop VR-enabled applications using Unity XR Interaction Toolkit with simulated controllers. 9. Create AR-based visualization applications such as human anatomy, architecture preview, or science models (DNA/RNA, solar system, etc.). 10. Implement AR image tracking using AR Foundation to display 3D models upon scanning predefined images.	**60**

	11. Develop an AR application using face tracking or body tracking to overlay virtual masks, filters, or accessories. 12. Build an AR navigation prototype using anchored waypoints and directional indicators for indoor guidance. 13. Create a VR environment with physics-based interactions such as grabbing, throwing, and colliding objects. 14. Design a VR user interface (VR canvas) with interactive buttons, sliders, and menus using XR Interaction Toolkit. 15. Develop a VR locomotion system (ray-based teleportation or joystick-based movement) without requiring a VR headset. 16. Implement object snapping, scaling, and rotation interactions in VR for assembly or puzzle-type tasks. 17. Build an AR shopping/preview application where users place and resize virtual furniture in real-world space. 18. Develop a VR simulation prototype with environmental effects such as fog, lighting transitions, and spatial audio. 19. Create a mixed-reality storytelling experience combining AR world elements and VR-style UI interactions. 20. Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects. 3D game objects can be created using Blender or using available 3D	
Pedagogy	Active Learning, Collaborative Learning, Inquiry-based Learning	
Instructions:	Any 15 experiments should be conducted.	
References/ Readings	**Text Books** 1. William R Sherman and Alan B Craig, "Understanding Virtual Reality, Interface, Application and Design",(The Morgan Kaufmann Series in Computer Graphics), Morgan Kaufmann Publishers, San Francisco, CA, 2002, ISBN 1558603530, 9781558603530. 2. Schmalstieg, Hollerer, "Augmented Reality: Principles & Practice" by Pearson Education India; First edition (12 October 2016), ISBN-10: 9332578494. **Reference Books** 1. Burdea, Grigore C and Philippe Coiffet, "Virtual Reality Technology", Wiley Interscience, India, 2003, ISBN 0471360899, 9780471360896 2. Alan B Craig, William R Sherman, Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann Publishers, 2009, ISBN: 978-0-12-374943-7. 3. Sanni Siltanen- Theory and applications of marker-based augmented reality. Julkaisija – Utgivare Publisher. 2012. ISBN 978-951-38-7449-0	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Build functioning AR applications capable of plane detection, object	

	placement, and basic interactive features. CO 2. Create VR prototypes using Unity's XR tools, demonstrating interaction techniques like grabbing, teleportation, and UI interaction. CO 3. Design and integrate 3D models, materials, lighting, audio, and effects to enhance the quality and realism of immersive experiences. CO 4. Deploy AR/VR applications on mobile and simulator-based platforms and evaluate their performance and usability.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-312**
Title of the Course : **Principles of Compilers**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course	Understanding of automata theory, data structures, programming (C/C++), and basic computer organization.	
Course Objectives	This course will enable students: 1. To provide foundational knowledge of the theory, algorithms, and tools involved in building compilers. 2. To explore the design and functioning of lexical, syntax, and semantic analysis for source programs. 3. To examine the development and use of syntax trees and symbol tables in compilation. 4. To gain a clear understanding of code generation methods and optimization strategies.	
Contents		**No of Hours**
Unit 1	**Language Processing & Compiler Tools** **Language Processing System** – Overview and role in program translation, Assemblers, Macro Processors, Linkers, Loaders, Debuggers, Text Editors, Compilers & Interpreters. Introduction to Language Translators, Phases of Compilation, Overview of each phase, Compiler-Writing Tools **The role of a lexical analyser**: Design of a lexical analyser. Implementation of a lexical analyzer. A Language for specifying lexical analysers. Study of the features and applications of LEX/FLEX tools.	**12**
Unit 2	**Overview of Context-free grammar:** Derivations and Parse trees, Ambiguity, left recursion, Left factoring. **Top-down parsing:** Recursive descent parsing and Predictive parsers. **Bottom-up parsing:** Shift-reduce parsers. Operator precedence parsers, LR parsers. **Study of YACC Tool:** Programming with YACC. Combining YACC and FLEX.	**12**
Unit 3	**Intermediate Code Generation:** Intermediate Language, Declarations, Assignment statements, Boolean expressions, Case statement, Procedure call.	**11**

	Run Time environments: Source language issues, Storage organization, Storage allocation strategies. **Symbol tables:** The content and Data structures for the Symbol Table. **Error detection and recovery:** Lexical phase errors, Syntactic phase errors, Semantic errors.	
Unit 4	**Code generation:** Issues in the design of a code Generator, Basic blocks and flow graphs, Next-use information, A simple Code generator, The DAG representation of Basic blocks, Peephole Optimization, Generating code from DAGs. **Code optimization:** The principal sources of optimization and Optimization of basic blocks.	**10**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Textbooks** 1. Aho and Ulman: Principles of Compiler Design; Publisher: Narosa Publishing House, Second Edition, 2002, ISBN: 81-85015-61-9. 2. Aho, Ulman, and Sethi; Compilers, Principles, techniques, and tools; Publisher: Pearson Education Inc, 1986, 2006 ISBN: 0-201-10088-6. **Reference Books** 1. Vinu V. Das: Compiler design with FLEX and YACC; PHI publication, 2007 ISBN: 978-81-203-3251-4. 2. Louden; Compiler Construction, Principles and Practice; Galgotia Publication, Second Edition, 1998, ISBN:0-534-93972-4. 3. Engineering a Compiler — Keith D. Cooper & Linda Torczon; Morgan Kaufmann Publishers, Second Edition, 2011, ISBN: 978-0120884780.	
Course Outcomes	Upon completion of the course, students will be able to CO 1. Explain the process by which compilers convert source programs into executable machine code. CO 2. Demonstrate the use of automated compiler construction tools such as LEX and YACC. CO 3. Describe parsing techniques and the fundamentals of basic code optimization. CO 4. Apply compiler design principles to develop and test a working compiler for a simple language.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-313**
Title of the Course : **Principles of Compilers Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives	The course will enable students to: 1. Understand the core principles and practical workflow involved in compiler design. 2. Utilize appropriate tools and techniques for implementing various phases of a compiler. 3. Apply different parsing strategies to construct syntactic analyzers in compiler development. 4. Analyze the translation process from a high-level programming language to machine-level code within a compiler.	
Contents	**List of programs**	**No of Hours**
	1. A program to detect the tokens from the user-defined expression. 2. A LEX program to find if the input is an integer, a real number, or a word. 3. A LEX program to insert line numbers into the given text. 4. A LEX program that represents a decimal number in hexadecimal numbers. 5. A LEX program to compute the average of a given set of numbers. 6. A YACC program to parse an expression for a given grammar. 7. A program that combines YACC and LEX. 8. A program to find FIRST and FOLLOW sets of a grammar. 9. A program to find LEADING and TRAILING sets of a grammar. 10. Implement a simple code generation algorithm.	**30**
Pedagogy	Integration of instructional learning, constructive thinking, inquiry-based,Collaborative, experimental, and problem-solving approaches.	
References/ Readings	**Text Books** 1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman; Compilers, Principles, Techniques and Tools; Second Edition; Pearson Education; 2007; ISBN: 978-81-317-2101-8.	

	2. Alfred V. Aho, Jeffrey D. Ullman; Principles of Compiler Design; Second Edition; Narosa publishing House; Re-print 2002; ISBN 978-81-85015-61-3. **Reference Books** 1. John R Levine, Tony Mason, Doug Brown; lex & yacc; Second edition; O'reilly, 1992, ISBN: 9781565920002. 2. Dhamdhere D.M, Compiler Construction Principles & Practice; Macmillan India Ltd., 2008; ISBN: 9780333904060. 3. Vinu V. Das; Compiler design using FLEX and YACC; PHI publication; 2007; ISBN: 978-81-203-3251-5.
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Understand the fundamental concepts and practical functioning of a compiler. CO 2. Apply LEX and YACC tools to design and implement lexical analyzers and parsers. CO 3. Develop programs to solve parsing problems using appropriate parsing techniques. CO 4. Generate machine-level code from intermediate representations using code generation.

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-314**
Title of the Course : **Software Test Automation and Quality Assurance**
Number of Credits : **3**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic Programming (Python/C/C++)	
Course Objectives	This course will enable students: 1. Introduce foundational concepts of software quality assurance, including testing methodologies, test levels, test types, and QA best practices. 2. Develop student skills in creating and executing automated tests using modern automation frameworks and tools to enhance testing efficiency. 3. Teach students how to design effective test plans and strategies that support defect prevention, continuous improvement, and high-quality software delivery. 4. Enable students to apply QA processes within real-world development environments, including version control, continuous integration, defect tracking, and reporting.	
Contents		**No of Hours**
Unit-1	**Basic Concepts and Preliminaries:** Role of Testing, Verification and Validation, Failure, Error, fault and Defect, Notion of Software Reliability, Objectives of Testing, What is a Test case? Expected Outcome, Concept of Complete Testing, Central Issue in Testing, Testing Activities, Test Levels, White-box and Black-box Testing, Monitoring and Measuring Test Execution, TestTeam Organization and Management. **Software Quality:** 5 views of SW Quality, McCall's Quality Factors and Criteria, ISO 9126 Quality Characteristics, ISO 9000:2000 Software Quality Standard.	**12**
Unit -2	**Unit Testing:** Concept of Unit Testing, Static & Dynamic Unit Testing, Defect Prevention,Mutation Testing, Debugging. **Control Flow Testing:** Basic Idea, Outline, Paths in a Control Flow Graph, Path SelectionCriteria, Generating Test Input. **Data Flow Testing:** General Idea, Data Flow Anomaly, Overview of Dynamic Data Flow Testing, Data Flow Graph & Terms, Data Flow Testing Criteria, Comparison of Testing Techniques.	**13**
Unit - 3	**Functional Testing:** Concepts of Howden, Complexity of applying Functional Testing, Pairwise Testing, Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Error	**10**

	Guessing. **System Integration Testing:** Concept, Different types of Interfaces & Interface Errors, Granularity of System Integration Testing, Techniques, Software Hardware Integration, Test Plan for System Integration, Off-the-shelf Component Integration.	
Unit - 4	**System Test Design:** Test Design Factors, Requirements Identification, Characteristics of Testable Requirements, Test Objective Identification, Modelling a Test Design Process & Results, Test Case Design Effectiveness. **System Test Planning and Automation:** Structure of a System Test Plan, Intro & Feature Description, Assumptions, Test Approach, Test Suite Structure, Test Environment, Test Execution Strategy, Test Effort Estimation, Scheduling & Test Milestones, System Test Automation, Evaluation & Selection of Test Automation Tools, Characteristics of Automated Test cases, Structure of an Automated Test case.	**10**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Text Books** 1. Kshirasagar Naik and Priyadarshi Tripathy; Software Testing and Quality Assurance: Theory and Practice; 1st edition, Wiley Publications, 2008, ISBN-10 0471789119. 2. William E. Perry; Effective Methods for Software Testing, Third Edition; Wiley Publications. 2006, ISBN-10: 0-7645-9837-6 **Reference Books** 1. Louise Tamares; Introducing Software testing; 1st edition, 2008, ISBN: 81-7808-678-6. 2. Mukesh Sharma, Software Testing 2020: Preparing for New Roles, 1st edition, CRC Press, 2020, ISBN-10, 1138468428. 3. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement; 1st Edition 2006, ISBN: 8126508051.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Understand software testing principles to plan, design, and execute effective manual and automated test cases for various types of applications. CO 2. Design the test cases based on various types and levels of software testing so as to address maximum software defects CO 3. Analyze and evaluate software quality metrics to identify defects, measure performance, and ensure compliance with quality standards. CO 4. Develop and implement automated test scripts using industry-standard tools to improve testing efficiency and reliability.	

Name of the Programme : **B.E in Information Technology**
Course Code : **ITH-315**
Title of the Course : **Software Test Automation and Quality Assurance Lab**
Number of Credits : **1**
Effective from AY : **2024-25**

<table>
<tr><td>Pre-requisites for the Course:</td><td colspan="2">Basic Programming (Python/C/C++)</td></tr>
<tr><td>Course Objectives</td><td colspan="2">This course will enable students to:
1. To develop the ability to design effective test cases using fundamental testing techniques.
2. To provide hands-on experience in performing unit, integration, system, and functional testing.
3. To enable learners to prepare essential testing documents and plans used in real software projects.
4. To introduce students to automation testing tools and techniques for improving testing efficiency.</td></tr>
<tr><td rowspan="2">Contents</td><td>List of Programs</td><td>No of Hours</td></tr>
<tr><td>1. Test case preparation.
2. Test plan creation.
3. Black-box testing.
4. Unit testing using PyTest.
5. Control Flow Graph & Cyclomatic Complexity.
6. Data Flow Testing.
7. Mutation Testing.
8. Functional Testing – Decision Tables.
9. Pairwise Testing.
10. Integration Testing.
11. System Test Design & RTM.
12. Automation Testing using Selenium.</td><td>30</td></tr>
<tr><td>Pedagogy</td><td colspan="2">Hands-on experiments involving test case design, unit/integration testing, functional/system testing, and automation to reinforce software testing concepts practically.</td></tr>
<tr><td>References/ Readings</td><td colspan="2">Text Books
1. Kshirasagar Naik and Priyadarshi Tripathy; Software Testing and Quality Assurance: Theory andPractice; 1st edition, Wiley Publications, 2008, ISBN-10 0471789119.
2. William E. Perry; Effective Methods for Software Testing, Third Edition; Wiley Publications. 2006, ISBN-10: 0-7645-9837-</td></tr>
</table>

	Reference Books 1. Louise Tamares, Introducing Software testing; 1st edition, 2008, ISBN: 81-7808-678-6. 2. Mukesh Sharma, Software Testing 2020: Preparing for New Roles, 1st edition, CRC Press, 2020, ISBN: 1138468428. 3. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement, 1st Edition,2006, ISBN: 81-265-0805-1
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Apply fundamental software testing techniques such as black-box, white-box, boundary value, equivalence class, and decision table testing to Design effective test cases. CO 2. Analyze software modules using control flow and data flow testing, mutation testing, and unit testing to identify defects and improve code quality. CO 3. Develop and execute test plans, system test designs, and requirements traceability matrices (RTM) to ensure software meets functional and non - functional requirements. CO 4. Implement automated testing using tools like Selenium or Cypress to improve testing efficiency, accuracy, and reporting.

Professional Electives

Name of the Programme : B.E in Information Technology
Course Code : ITH-327
Title of the Course : Internet of Things and Blockchain Technology
Number of Credits : 4
Effective from AY : 2024-25

Pre-requisites for the Course	Basic knowledge of computer networks and communication protocols and programming concepts using languages such as Python, Java, or C is required.	
Course Objectives	This course will enable students: 1. Understand the fundamentals, architecture, and enabling technologies of the Internet of Things (IoT). 2. Analyze IoT protocols, domain-specific applications, and system design methodologies. 3. Understand blockchain concepts, architecture, consensus mechanisms, and cryptographic foundations. 4. Explore the integration of blockchain with IoT and other emerging technologies.	
Contents		**No of Hours**
Unit-1	**Introduction and Concepts:** **Introduction to IoT** – Definition and Characteristics, Physical design of things in IoT, IoT protocols, Logical Design of IOT, IOT Enabling Technologies **Domain Specific IoTs** –Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Lifestyle.	**13**
Unit - 2	**IOT and M2M** **Introduction** -M2M, Difference between M2M and IoT, SDN and NFV for IoT **IOT System Management**– SNMP, NETCONF, YANG. **Developing the Internet Of Things** **IoT Methodology** - Purpose & Requirements specification, process specification, domain model specification, information model specification, service specification, IoT level specifications, Functional View Specification, Operational View Specification, Device & Component Integration, Application Development.	**15**
Unit - 3	**Blockchain Technology** **Basics of Blockchain:** Concept of Blockchain, History, Definition, Fundamentals, Characteristics, Consensus in Trust-Building Exercise, Public, Private, and Hybrid Blockchain, Distributed Ledger	**16**

	Technologies, DLT Decentralized Application And Databases, Architecture of Blockchain, Transactions, Chaining Blocks	
Unit - 4	**Foundations of Blockchain Technology:** Decentralized Systems, Hash Functions, Consensus, Blockchain Components, Cryptography, Smart Contracts, Bitcoins **Blockchain and Allied Technologies:** Blockchain and Cloud Computing, Blockchain and Artificial Intelligence, Blockchain and IOT, Blockchain and ML, Blockchain And Robotics Process Automation	**16**
Pedagogy	Demonstrations of IoT architectures and blockchain tools, Use of case studies to support real-world applications, Emphasis on critical thinking, troubleshooting, and system integration	
References/ Readings	**Text Books** 1. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-On Approach, 1st Edition, Universities Press, 2015, ISBN: 9780996025522. 2. Vijay Madisetti, Arshdeep Bahga, Blockchain Applications: A Hands-On Approach, 1st Edition, Vijay Madisetti & Arshdeep Bahga Publishers, 2017, ISBN: 9780996025560. **Reference Books** 1. D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Pearson Education Inc., 2019, ISBN: 9789386873743. 2. Don Tapscott, Alex Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies Is Changing the World, Portfolio/Penguin, 2018, ISBN: 978-0063025851. 3. K. Saini, Blockchain and IoT Integration, Auerbach Publications, 2021, ISBN: 978-1032114866.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain the fundamental concepts, architecture, protocols, and domain-specific applications of IoT. CO 2. Apply IoT development methodology for designing purposes, processes, information models, and services. CO 3. Analyse blockchain concepts, components, consensus mechanisms, and cryptographic foundations. CO 4. Evaluate the integration of blockchain with IoT, AI, Cloud, and other emerging technologies.	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-328**
Title of the Course : **Machine Learning Techniques**
Number of Credits : **4**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic Programming Skills, Knowledge of Linear Algebra (matrices, vectors), Basic Probability and Statistics	
Course Objectives	This course will enable students: 1. Understand key concepts, applications, and challenges of Machine Learning. 2. Prepare data through preprocessing, wrangling, and feature engineering. 3. Apply supervised, unsupervised, Bayesian, and ensemble learning techniques. 4. Evaluate ML models using appropriate metrics and interpretability tools.	
Contents		**No of Hours**
Unit-1	**Introduction to Machine Learning:** Human learning and its types, Machine Learning and its types – supervised, unsupervised, reinforcement learning, comparison, Applications of Machine Learning, Challenges/issues in Machine Learning, tools in machine learning. **Preparing to Model:** Basic data types, Exploring numerical and categorical data, Exploring relationships between variables, Data Quality and Remediation, Data pre-processing- Dimensionality reduction – Principal Component Analysis (PCA), The Curse of Dimensionality, Feature construction, Feature extraction, Feature selection.	**15**
Unit - 2	**Supervised Learning:** Classification: model, learning steps, k-Nearest Neighbour (KNN), Decision Tree, Support Vector Machines. Regression: Linear Regression, multiple linear regression, polynomial regression, Logistic Regression, Maximum Likelihood estimation (MLE).	**15**
Unit - 3	**Unsupervised Learning:** Clustering: types, K-Means, K-medoids, Hierarchical clustering, Density-based- DBSCAN. Finding patterns using the Association Rule Learning-Apriori algorithm. **Bayesian Learning:** Fundamentals of Bayes Theorem, Classification using Bayes model -- Naïve Bayes Classifier and its	**15**

	applications, Brute-force Bayes algorithm, Bayes Optimal Classifier. **Ensemble Learning:** Methods for constructing an Ensemble Classifier, Bagging, Boosting, and Random Forest.	
Unit - 4	**Modelling and Evaluation:** Selecting a Model, training a Model, Model Representation and Interpretability – Underfitting, Overfitting, Bias- Variance Trade-off, Evaluating Performance of a Model- classification, regression, and clustering, Model Testing, and performance metric **Case-study of ML applications:** Image recognition, speech recognition, Email spam filtering, Online fraud detection.	**15**
Pedagogy	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings	**Text Books:** 1. Saikat Dutt, Subramanian Chjandramouli, Amit Kumar Das, Machine Learning, Pearson, 1st edition, 2019, ISBN:978-93-530-6669-7. 2. S. Sridhar, M. Vijayalakshmi, Machine Learning, Oxford University Press, 1st edition, 2021, ISBN: 978-0-19-012727-5. **Reference Books:** 1. Christopher M. Bishop, Pattern Recognition & Machine Learning, Springer, 2006. ISBN-13: 978-0387310732. 2. Tom M. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc., Indian Edition 1997. ISBN-13: 9780070428072. 3. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction To Data Mining, Pearson, 2nd Edition, 2019, ISBN: 978-0-273-76922-4.	
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Explain fundamental concepts, types, and challenges of Machine Learning. CO 2. Perform data pre-processing, wrangling, and feature engineering for ML tasks. CO 3. Apply supervised, unsupervised, Bayesian, and ensemble learning algorithms. CO 4. Evaluate and compare ML models using appropriate performance metrics.	

Name of the Programme : **B.E. in Information Technology**
Course Code : **ITH-329**
Title of the Course : **Multimedia Technology**
Number of Credits : **4**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Nil	
Course Objectives	This course will enable students to: 1. Describe the fundamental concepts of multimedia technology 2. Understand and apply the principles of handling and processing static media elements 3. Explain the fundamentals of sound and video systems 4. Explore the principles and techniques of computer animation	
Contents		**No of Hours**
Unit-1	**Multimedia Overview:** Multimedia Fundamentals, Application Areas, Multimedia Hardware and software requirements, Multimedia standards, Multimedia technologies and platforms, Multimedia and Interactivity, Multimedia Content creation.	**15**
Unit - 2	**Text:** Text Fundamentals, Digital text representation, text editing and processing, Text Applications, Text file formats **Image:** Image Fundamentals, Image I/O, Image Editing and processing- Geometric transformations, Image Color Management-Color models, Chromaticity, Color spaces, Color Conversions, Image File Formats.	**15**
Unit - 3	**Audio:** Audio Fundamentals, Audio I/O, Audio Editing and processing, Audio Applications, Audio File Formats. **Video:** Video Fundamentals, Video I/O, Video Editing and Processing, image Processing tasks, Time-based, SMPTE Timecode, Video Color Spaces.	**15**
Unit - 4	**Animation:** Animation Fundamentals,2D Animation,3D Animation, Animation Applications, Legal and Ethical Considerations, Challenges and Emerging Trends, Animation File Formats	**15**
Pedagogy	Combines theoretical understanding with practical application of multimedia technology.	
References/ Readings	**Text Books** 1. Ranjan Parekh, Principles of Multimedia, the Tata McGraw Hill companies third Edition, 2025, ISBN 9781032642949.	

	2. Prabhat K. Andleigh & Kiran Thakrar, Multimedia Systems Design, Pearson Education India, 2015, ISBN-13: 978-9332549388. **Reference Books** 1. Tay Vaughan, Multimedia - Making It Work, McGraw-Hill Osborne, 2001, ISBN: 9780072195217. 2. Syad Mahbubur Rahman, Multimedia Technologies: Concepts, Methodologies, Tools, and Applications, 2008, ISBN-13: 978-1599049533. 3. Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, Fundamentals of Multimedia, Springer International Publishing, 2021, ISBN: 9783030621247.
Course Outcomes	Upon completion of the course, students will be able to: CO 1. Understand the core components and systems of multimedia. CO 2. Analyze and process static multimedia elements CO 3. Evaluate and manipulate dynamic multimedia components CO 4. Apply animation principles and techniques.