SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE100704

Course / Subject Name: Block chain Technology

w. e. f. Academic Year:	2023-24
Semester:	6th
Category of the Course:	Open Elective – 3

Prerequisite:	Cryptography
Rationale:	Blockchain helps to manage and secure data as a record with their digital relationships in adecentralized manner. With blockchain technology, there is no need for a central authority and two parties can securely conduct communication with one another at a faster pace. Blockchain makes the Transaction transparent by keeping the chain of records.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Explain different types of Blockchain	Understand
2	Describe Consensus algorithms for decentralized network architecture	Apply
3	Explain the support of Blockchain technology in a cryptocurrency or Digital currency.	Apply
4	Use the Blockchain technology for given application	Apply

CO PO Mapping													
Course Outcome (CO)	**Description**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	Explain different types of Blockchain	3	2	-	-	2	-	-	-	-	1	-	1
CO2	Describe consensus algorithms for decentralized network architecture	3	2	2	-	2	-	-	-	-	1	-	1
CO3	Explain the support of Blockchain technology in cryptocurrency/digital currency	3	2	2	1	2	-	-	-	-	1	-	2

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE100704

Course / Subject Name: Block chain Technology

CO PO Mapping													
Course Outcome (CO)	**Description**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO4	Use Blockchain technology for a given application	3	3	3	2	3	1	-	-	2	2	2	3

CO PSO Mapping				
Topic	**Course Outcome (CO)**	**PSO1 (Domain Knowledge)**	**PSO2 (Problem Analysis & Design)**	**PSO3 (Modern Tools & Application)**
1. Explain different types of Blockchain	CO1: Understand public, private, consortium, and hybrid blockchain types	✓	✓	
2. Describe Consensus algorithms for decentralized network architecture	CO2: Analyze PoW, PoS, DPoS, PBFT, etc. in decentralized networks	✓	✓	
3. Explain the support of Blockchain in cryptocurrency or digital currency	CO3: Demonstrate the role of blockchain in digital/cryptocurrency systems	✓		✓
4. Use Blockchain technology for a given application	CO4: Apply blockchain solutions in use cases like supply chain, healthcare, etc.		✓	✓

Teaching and Examination Scheme:

Hours per Week			**Course Credits**	**Examination Marks**				
				Theory Marks		**Practical Marks**		
L	**T**	**P**		**E**	**M**	**I**	**V**	**150**
2	**0**	**2**	**3**	**60**	**40**	**20**	**30**	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE100704

Course / Subject Name: Block chain Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Fundamentals of Blockchain: Introduction, Origin of Blockchain, Blockchain solution, Components of Blockchain, Block in Blockchain, The Technology Blockchain Types and 1 Consensus Mechanism: Introduction, Decentralization and Distribution, Types Hours 06 of Blockchain, Consensus Protocol Cryptocurrency – Bitcoin, Altcoin, Token: Introduction, Bitcoin and the cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrency, Cryptocurrency usage	6	21
2.	Public Blockchain System: Introduction, Public Blockchain, Popular public Blockchain, The Bitcoin 2 Blockchain, Ethereum Blockchain Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart contracts, Types 07 of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry	7	25
3.	Private Blockchain System: 3 4 Introduction, Key characteristics of Private Blockchain, Private Blockchain Examples, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, Byzantine Fault	5	17
4.	Consortium Blockchain Introduction, Key characteristics of Consortium Blockchain, Hyperledger platform,Overview of Ripple, Overview of Corda	5	17
5.	Applications of Blockchain: Introduction, Blockchain in Education, Blockchain in Healthcare	3	10
6.	Limitations and Challenges of Blockchain 03 Blockchain Implementation – Limitation, Challenges	2	1
	Total	28	100

SWAMINARAYAN UNIVERSITY

Program Name:

EngineeringLevel: UG

Course / Subject Code: BE100704

Course / Subject Name: Block chain Technology

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Blockchain Technology By Chandramouli Subramanian, Asha George, Abhilash K A and Meena Karthikeyan , Universities Press Publication
Blockchain Blueprint for a New Economy, By Melanie Swan,O'Reilly Publication
Blockchain For Dummies By Tiana Laurence, Wiley Publication

Suggested Course Practical List:

1) Ethereum – With respect to Ethereum, carry out following :
• Install Geth Client & Configure Ethereum Nodes
• Manage Accounts and Account states.
• Enable Mining and checking balance in Ether
• Setting up Metamask and Testing Fund Transfer – work with Ethereum Ecosystem
Add Parameters to Cryptocurrency
• Check Balance Before Transfer
• Adding Transfer Event for Logging
2) Hyperledger - With respect to Hyperledger, carry out following :
• Installing Hyperledger Fabric(latest version)
• Build Network with Network configuration
• Hyperledger Fabric Demo
3) Corda - With respect to Corda, carry out following :
• Corda – Setting up the Environment
• Building the Corda Network

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE100704

Course / Subject Name: Block chain Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

List of Laboratory/Learning Resources Required:

Introduction to Blockchain and Cryptography
Setting up a Private Ethereum Blockchain
Smart Contract Development using Solidity
Deploying Smart Contracts on Testnet
Blockchain Wallets and Transactions
Consensus Mechanism Demonstration (PoW, PoS)
Hyperledger Fabric Setup and Chaincode Deployment
Interacting with Blockchain via Web3.js
Case Study: Cryptocurrency Implementation

Suggested Project List:
Project: Study and Comparison of Blockchain Types
Project: Consensus Mechanisms in Blockchain
Project: Blockchain in Cryptocurrency and Digital Currencies
Project: Blockchain-based Application Development

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103701

Course / Subject Name: Modern Compiler Design

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	Professional Core

Prerequisite:	• Data Structures & Algorithm • Languages • Memory Management • Discrete Mathematics • Operating Systems & System Programming
Rationale:	• Seamless experiences across devices • Scalable cloud infrastructure • Smart embedded systems (like those in cars and appliances) • Many of the productivity-enhancing features in modern programming environments

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Understand the basic concepts; ability to apply automata theory and knowledge on formal languages.	Understand
2	Ability to identify and select suitable parsing strategies for a compiler for various cases. Knowledge in alternative methods (top-down or bottom-up, etc).	Apply
3	Understand backend of compiler: intermediate code, Code optimization Techniques and Error Recovery mechanisms	Apply
4	Understand issues of run time environments and scheduling for instruction level parallelism.	Understand

CO PO Mapping		
Course Outcome (CO)	**Mapped Program Outcomes (POs)**	**Rationale**
CO1: Understand the basic concepts; ability to apply automata theory and knowledge on formal languages.	PO1, PO2	PO1 (Engineering Knowledge) for understanding theory; PO2 (Problem Analysis) for applying automata to solve formal language problems.
CO2: Ability to identify and select suitable parsing strategies for a compiler for various cases.	PO1, PO2, PO4	PO1 for applying knowledge of parsing techniques; PO2 for analyzing requirements and selecting the best strategy; PO4 (Investigation) for experimenting with methods.

CO PO Mapping		
Course Outcome (CO)	**Mapped Program Outcomes (POs)**	**Rationale**
CO3: Knowledge in alternative methods (top-down or bottom-up, etc).	PO1, PO2	PO1 for understanding the concepts; PO2 for selecting appropriate techniques in practical compiler construction problems.
CO4: Understand backend of compiler: intermediate code, code optimization techniques, and error recovery.	PO1, PO2, PO3	PO1 for intermediate code generation and optimization knowledge; PO2 for analyzing and applying optimization methods; PO3 (Design) for integrating error recovery.
CO5: Understand issues of run-time environments and scheduling for instruction-level parallelism.		

CO PSO Mapping			
Course Outcome (CO)	**PSO1**	**PSO2**	**PSO3**
CO1: Understand the basic concepts and apply automata theory and formal languages.	✓	✓	
CO2: Identify and select suitable parsing strategies for a compiler in various cases using top-down, bottom-up techniques.	✓	✓	
CO3: Understand the backend of the compiler: intermediate code generation, code optimization techniques, and error recovery mechanisms.	✓	✓	✓
CO4: Understand issues of runtime environments and instruction-level parallelism scheduling.		✓	✓

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			**Total Credits L+T+ (PR/2)**	**Assessment Pattern and Marks**				**Total Marks**
				Theory		**Tutorial / Practical**		
L	**T**	**PR**	**C**	(E)	(M)	(I)	(V)	
3	0	2	4	60	40	20	30	150

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103701

Course / Subject Name: Modern Compiler Design

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Overview of the Compiler and its Structure**: Language processor, Applications of language processors, Definition-Structure-Working of compiler, the science of building compilers, Basic understanding of interpreter and assembler. Difference between interpreter and compiler. Compilation of source code into target language, Cousins of compiler, Types of compiler	3	10
2.	**Lexical Analysis:** The Role of the Lexical Analyzer, Specification of Tokens, Recognition of Tokens, Input Buffering, elementary scanner design and its implementation (Lex), Applying concepts of Finite Automata for recognition of tokens.	5	10
3.	**Syntax Analysis:** Understanding Parser and CFG(Context Free Grammars), Left Recursion and Left Factoring of grammar Top Down and Bottom up Parsing Algorithms, Operator-Precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators, Automatic Generation of Parsers. Syntax-Directed Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S-Attributed Definitions, L-Attributed Definitions, syntax directed definitions and translation schemes	11	20
4.	**Error Recovery** Error Detection & Recovery, Ad-Hoc and Systematic Methods	4	10
5.	**Intermediate-Code Generation:** Variants of Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Syntax Directed Translation Mechanisms, Attributed Mechanisms And Attributed Definition.	5	15
6.	**Run-Time Environments:** Source Language Issues, Storage Organization. Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management	4	10
7.	**Code Generation and Optimization:** Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Machine dependent optimization, Machine independent optimization Error detection of recovery	6	15

8.	**Instruction-Level Parallelism:** Processor Architectures, Code-Scheduling Constraints, Basic-Block Scheduling, Pass structure of assembler	4	10
	Total	**42**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Modern Compiler Design - Dick Grune, Henri E. Bal, Jacob, Langendoen, WILEY India
2. Compiler Construction -Waite W.N. And Goos G., Springer Verlag
3. Compiler Construction-Principles And Practices - D.M.Dhamdhere, Mcmillian
4. Principles of Compiler Design, V. Raghavan, McGrawHill

(b) Open source software and website:

- GeeksforGeeks Compiler Design Tutorial
- TutorialsPoint Compiler Design
- NPTEL Compiler Design Course
- GetVM's Compiler Design Collection
- LibHunt Compiler Design Projects

Suggested Course Practical List:

1. Finding "First" set
   Input: The string consists of grammar symbols.
   Output: The First set for a given string.
   Explanation:
   The student has to assume a typical grammar. The program when run will ask for the string to be entered. The program will find the First set of the given string.
2. Generate 3-tuple intermediate code for given infix expression
3. Extract Predecessor and Successor from given Control Flow Graph
4. Introduction to YACC and generate Calculator Program
5. Finding "Follow" set
   Input: The string consists of grammar symbols.
   Output: The Follow set for a given string.
   Explanation:

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103701

Course / Subject Name: Modern Compiler Design

List of Laboratory/Learning Resources Required:

	Tool	Use Case
1.	**ANTLR**	Build parsers and interpreters easily.
2.	**LLVM**	Explore backend code generation and optimization.
3.	**Flex & Bison**	Classic tools for lexical and syntax analysis.
4.	**Loxcraft**	Rust-based implementation of the Lox language.
5.	**EXPL NITC**	Practical guide to building your own compiler from scratch.

Suggested Project List:

1. **Custom Language with LLVM Backend**: Design a new language and compile it to machine code using LLVM.
2. **Just-In-Time (JIT) Compiler**: Use LLVM or similar frameworks to compile code at runtime.
3. **Garbage Collector Implementation**: Add memory management to your language runtime.
4. **Cross-Platform Compiler**: Build a compiler that targets multiple architectures (e.g., x86, ARM).
5. **Mini Compiler for a Toy Language**: Implement a compiler for a simple language like PL/0 or Lox.
6. **Intermediate Code Generator**: Translate source code into three-address code or LLVM IR.
7. **Semantic Analyzer**: Add type checking and scope resolution to your parser.
8. **Code Optimizer**: Implement constant folding, dead code elimination, and loop unrolling.

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103702

Course / Subject Name: Artificial Intelligence with PROLOG

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	Professional Core

Prerequisite:	• Artificial Intelligence Basics • Mathematical Background • Programming Skills
Rationale:	PROLOG in Artificial Intelligence (AI) lies in its unique ability to model logic, relationships, and inference in a way that closely mirrors human reasoning.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Understand the search technique procedures applied to real world problems	Understand
2	Understand and use various types of logic and knowledge representation schemes.	Understand
3	Understand various Game Playing techniques and apply them in programs.	Understand
4	Gain knowledge in AI Applications and advances in Artificial Intelligence	Analyze
5	Use Prolog Programming language using predicate logic	Apply

Program Outcomes	
PO Code	**Program Outcome Description**
PO1	Engineering Knowledge
PO2	Problem Analysis
PO3	Design/Development of Solutions
PO4	Conduct Investigations of Complex Problems
PO5	Modern Tool Usage
PO6	The Engineer and Society
PO7	Environment and Sustainability
PO8	Ethics
PO9	Individual and Team Work
PO10	Communication
PO11	Project Management and Finance
PO12	Life-long Learning

CO PO Mapping						
CO \ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6–PO12 (optional or low impact)**
CO1	3	3	2	2	2	–
CO2	3	3	2	2	2	–
CO3	3	2	3	2	2	–
CO4	3	2	2	1	2	–
CO5	3	3	2	2	3	–

CO PSO Mapping			
Course Outcomes (COs)	**PSO1**	**PSO2**	**PSO3**
1. Understand the search technique procedures applied to real world problems	✓	✓	✓
2. Understand and use various types of logic and knowledge representation schemes	✓	✓	
3. Understand various Game Playing techniques and apply them in programs	✓	✓	
4. Gain knowledge in AI Applications and advances in Artificial Intelligence	✓	✓	✓
5. Use Prolog Programming language using predicate logic	✓	✓	

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	(E)	(M)	(I)	(V)	
3	0	2	4	60	40	20	30	150

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103702

Course / Subject Name: Artificial Intelligence with PROLOG

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction :** The AI Problems, The Underlying Assumption, AI techniques, The Level of The Model, Criteria For Success	2	14
2.	**Problems, State Space Search & Heuristic Search Techniques:** Defining The Problems As A State Space Search, Production Systems, Production Characteristics, Production System Characteristics and Issues in the Design of Search Programs, Generate-And-Test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis.	6	22
3.	**Knowledge Representation:** Representations And Mappings, Approaches To Knowledge Representation, Representation Simple Facts In Logic, Representing Instance And Isa Relationships, Computable Functions and Predicates, Resolution, Procedural versus Declarative Knowledge, Logic Programming, Forward versus Backward Reasoning	7	10
4.	**Probabilistic Reasoning:** Probability And Bays' Theorem, Certainty Factors And Rule-Base Systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic	2	10
5.	**Game Playing:** Overview, MiniMax Search Procedure, Alpha-Beta Cut-offs, Refinements, Iterative deepening.	3	10
6.	**Natural Language Processing:** Introduction, Syntactic Processing, Semantic Analysis, Discourse And Pragmatic Processing, Spell Checking	2	10
7.	**Connectionist Models:** Introduction: Hopfield Network, Learning In Neural Network, Application Of Neural Networks, Recurrent Networks, Distributed Representations, Connectionist AI And Symbolic AI.	3	10
8.	**Expert Systems:** Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition	3	10
9.	**Genetic Algorithms:** A Peek into the Biological World, Genetic Algorithms (GAs), Significance of the Genetic Operators, Termination Parameters.	4	10
10.	**Introduction to Prolog:** Introduction, Converting English to Prolog Facts and Rules, Goals, Prolog Terminology, Variables, Control Structures, Arithmetic Operators, Matching in Prolog, Backtracking, Cuts, Recursion, Lists	2	10
	Total	**34**	**100**

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103702

Course / Subject Name: Artificial Intelligence with PROLOG

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. "Artificial Intelligence" -By Elaine Rich And Kevin Knight (2nd Edition) Tata Mcgraw-Hill
2. "Artificial Intelligence: A Modern Approach" -By Stuart Russel, Peter Norvig, PHI
3. "Introduction to Prolog Programming" -By Carl Townsend.
4. "PROLOG Programming For Artificial Intelligence" -By Ivan Bratko(Addison-Wesley)
5. "Programming with PROLOG" –By Klocksin and Mellish.

(b) Open source software and website:

1. Prolog | An Introduction – GeeksforGeeks
2. Prolog Programming Features – Engineering Notes
3. Introduction to Prolog – Built In

Suggested Course Practical List:

1. Write a program to implement Tic-Tac-Toe game problem.
2. Write a program to implement BFS (for 8 puzzle problem or Water Jug problem or any AI search problem) .
3. Write a program to implement DFS (for 8 puzzle problem or Water Jug problem or any AI search problem)
4. Write a program to implement Single Player Game (Using any Heuristic Function)
5. Write a program to Implement A* Algorithm.
6. Write a program to implement mini-max algorithm for any game development.
7. Assume given a set of facts of the form father(name1,name2) (name1 is the father of name2). Define a predicate brother(X,Y) which holds iff X and Y are brothers.
8. Define a predicate cousin(X,Y) which holds iff X and Y are cousins.
   Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.
   Define a predicate descendent(X,Y) which holds iff X is a descendent of Y.
   Consider the following genealogical tree:
   father(a,b).
   father(a,c).
   father(b,d).
   father(b,e).
   father(c,f).

Say which answers, and in which order, are generated by your definitions for the following queries in Prolog:

?- brother(X,Y).

?- cousin(X,Y).

?- grandson(X,Y).

?- descendent(X,Y).

9. Write a program to solve Tower of Hanoi problem using Prolog.
10. Write a program to solve N-Queens problem using Prolog.
11. Write a program to solve 8 puzzle problem using Prolog.
12. Write a program to solve travelling salesman problem using Prolog.

List of Laboratory/Learning Resources Required:

1. Installing and using **SWI-PROLOG** or **GNU PROLOG**
2. Running and debugging PROLOG programs in an interpreter.
3. r/Prolog on Reddit – Ask questions, share projects, and get feedback.
4. SWI-Prolog Official Docs – In-depth documentation and tutorials.
5. Study.com: Prolog in AI – Overview of PROLOG's role in AI with examples.

Suggested Project List:

1. **Medical Diagnosis Expert System**: Use symptoms and rules to suggest possible diseases.
2. **Animal Identification System**: Ask yes/no questions to identify an animal based on features.
3. **Travel Planner**: Use graph traversal to find optimal routes between cities.
4. **Natural Language Parser**: Parse simple English sentences into grammatical structures.
5. **8-Puzzle Solver**: Implement A* or iterative deepening search to solve the sliding puzzle.
6. **Theorem Prover**: Build a basic logic theorem prover using resolution and unification.
7. **Planning Agent**: Solve STRIPS-style planning problems (e.g., robot navigation).
8. **Game AI (Tic-Tac-Toe or Connect Four)**: Implement a minimax-based AI opponent.
9. **Legal Reasoning System**: Encode legal rules and simulate case-based reasoning.
10. **Recipe Recommender**: Suggest recipes based on available ingredients using backward chaining.
11. **Smart Home Rules Engine**: Model automation rules like "If it's after 6 PM and someone is home, turn on the lights."

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: EngineeringLevel: UG

Course / Subject Code: BE103703

Course / Subject Name: Cloud Computing

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	Professional Elective-4

Prerequisite:	Fundamentals of Distributed Computing
Rationale:	This course aims students to understand the hardware, software concepts and architecture of cloud computing. Students realize the importance of Cloud Virtualization, Abstractions and Enabling Technologies.

CO PSO Mapping		
Course Outcome (CO)	**Mapped PSOs**	**Justification**
CO1: Compare the strengths and limitations of cloud computing.	PSO1, PSO3	Requires analytical understanding of computing trends and implications including societal and environmental impact.
CO2: Identify the architecture, infrastructure and delivery models of cloud computing.	PSO1	Involves understanding technical structures and components relevant to cloud solutions.
CO3: Apply suitable virtualization concepts.	PSO1, PSO2	Practical use of virtual machines and environments ties to real-world software/system design.
CO4: Choose the appropriate cloud provider, programming models and approaches.	PSO1, PSO2	Involves decision-making and design for cloud-based applications using real-world platforms.
CO5: Address the core issues of cloud computing such as security, privacy and interoperability.	PSO1, PSO3	Engages understanding of secure system design and ethical implications of cloud technologies.

SWAMINARAYAN UNIVERSITY

Program Name: EngineeringLevel: UG

Course / Subject Code: BE103703

Course / Subject Name: Cloud Computing

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Compare the strengths and limitations of cloud computing	Understand
2	Identify the architecture, infrastructure and delivery models of cloud computing	Apply
3	Apply suitable virtualization concept.	Apply
4	Choose the appropriate cloud player, Programming models and approach	Apply
5	Address the core issues of cloud computing such as security, privacy and interoperability	Understand

CO PO Mapping												
CO/PO	PO1 Engg. Knowledge	PO2Problem Analysis	PO3Design/Development	PO4Investigations	PO5Modern Tools	PO6Engineer & Society	PO7Environment & Sustainability	PO8Ethics	PO9Individual/Team Work	PO10Communication	PO11Project Management	PO12 Life-long Learning
CO1	3	2	1	-	2	-	-	-	-	-	-	2
CO2	3	2	2	1	2	-	-	-	-	-	-	2
CO3	3	3	2	2	3	-	-	-	-	-	-	2
CO4	3	3	3	2	3	-	-	-	2	1	2	3
CO5	3	2	2	2	2	3	2	2	2	-	-	3

Teaching and Examination Scheme:

Hours per Week			Course Credits	Examination Marks				
				Theory Marks		Practical Marks		
L	T	P		E	M	I	V	150
3	0	0	4	60	40	20	30	

SWAMINARAYAN UNIVERSITY

Program Name: EngineeringLevel: UG

Course / Subject Code: BE103703

Course / Subject Name: Cloud Computing

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Introduction: Cloud Computing, Layers and Types of Clouds, Cloud Infrastructure Management, Challenges and Applications. Virtualization: Virtualization of 4 Computing, Storage and Resources. Cloud Services: Introduction to Cloud Services IaaS, PaaS and SaaS	4	10
2.	Software as a Service (SaaS): Evolution of SaaS, Challenges of SaaS Paradigm, SaaS Integration Services, SaaS Integration of Products and Platforms. Infrastructure As a Services (IaaS): Introduction, Background & Related Work, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and 8 Migration in Action. Platform As a service (PaaS): Integration of Private and Public Cloud, Technologies and Tools for Cloud Computing , Resource Provisioning services	8	20
3.	Abstraction and Virtualization: Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines Provisioning and Manageability Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context, Virtualization of CPU, Memory, I/O Devices, Virtual Clusters and Resource 8management, Virtualization for Data Center Automation	8	20
4.	Cloud Infrastructure and Cloud Resource Management: Architectural Design of Compute and Storage Clouds, Layered Cloud Architecture Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources. Administrating the Clouds, Cloud Management Products, Emerging Cloud Man	8	20
5.	Security: Security Overview, Cloud Security Challenges and Risks, Software-as-a Service Security, Cloud computing security architecture: Architectural Considerations, General Issues Securing the Cloud, Securing Data, Data Security, Application Security, Virtual Machine Security, Identity and Presence, Identity Management and Access Control, Autonomic Security Establishing Trusted Cloud computing, Secure Execution Environments and Communications, , Identity Management and Access control Identity management, Access control, Autonomic Security Storage Area Networks, Disaster Recovery in Clouds.	7	15
6.	Cloud Middleware: OpenStack, Eucaluptus, Windows Azure, CloudSim, EyeOs, Aneka, Google App Engine	5	5
7	Cloud Based Case-Studies: Overview of Cloud services, Designing Solutions for the Cloud, Implement & Integrate Solutions, Emerging Markets and the Cloud, Tools for Building Private Cloud: IaaS using Eucalyptus, PaaS on IaaS - AppScale	5	5
	Total	45	100

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Books

Rajkumar Buyya, James Broberg, Andrzej M Goscinski, Cloud Computing: Principles and Paradigms, Wiley publication.
Toby Velte, Anthony Velte, Cloud Computing: A Practical Approach, McGraw-Hill Osborne Media.
George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Publication.
John Rhoton, Cloud Computing Explained: Implementation Handbook for Enterprises, Recursive Press.

(a) Open source software and website:

Software	Website	Description
OpenStack	https://www.openstack.org	Cloud operating system for building IaaS platforms.
CloudStack	https://cloudstack.apache.org	Apache project for deploying and managing large networks of virtual machines.
Eucalyptus	https://www.eucalyptus.cloud	Compatible with AWS APIs for private/hybrid clouds.
OpenNebula	https://opennebula.io	Simple but powerful cloud management platform.
Kubernetes	https://kubernetes.io	Container orchestration for managing cloud-native apps.

SWAMINARAYAN UNIVERSITY

Program Name: EngineeringLevel: UG

Course / Subject Code: BE103703

Course / Subject Name: Cloud Computing

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	Professional Elective-4

Prerequisite:	Computer Network
Rationale:	Wireless communication provides mobility, flexibility, convenience. Wireless communication devices are used in various areas including healthcare. Wireless communication has opened many areas for research also.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Understand the mobile and wireless network systems such as 2G/3G/4G	Understand
2	Understand GSM and GPRS	Understand
3	Implement various error coding techniques.	Apply
4	Differentiate between multiple access schemes and various Spread Spectrum techniques.	Understand
5	Understand the working with local area network, Bluetooth and Android APIs environment.	Understand

CO PO Mapping												
CO → PO ↓	**PO1 (Engg Knowledge)**	**PO2 (Problem Analysis)**	**PO3 (Design/Development)**	**PO4 (Investigate)**	**PO5 (Modern Tools)**	**PO6 (Engineer & Society)**	**PO7 (Environment)**	**PO8 (Ethics)**	**PO9 (Teamwork)**	**PO10 (Communication)**	**PO11 (Project Mgmt)**	**PO12 (Lifelong Learning)**
CO1	3	2	1	1	2	1	1	1	1	1	1	2
CO2	3	2	1	1	2	1	1	1	1	1	1	2
CO3	3	3	2	2	2	1	1	1	2	1	2	3
CO4	3	2	2									

CO PSO Mapping				
CO No.	**Course Outcome Statement**	**Bloom's Level**	**Mapped PSO(s)**	**Justification**
CO1	Understand the mobile and wireless network systems such as 2G/3G/4G	Understand	PSO1, PSO2	Involves understanding of communication technologies (PSO1) and their practical implementation in wireless systems (PSO2).

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

CO PSO Mapping				
CO No.	**Course Outcome Statement**	**Bloom's Level**	**Mapped PSO(s)**	**Justification**
CO2	Understand GSM and GPRS	Understand	PSO1, PSO2	Understanding of cellular networks (GSM/GPRS) contributes to communication system knowledge (PSO1) and practical network design (PSO2).
CO3	Implement various error coding techniques	Apply	PSO1	Application of coding techniques directly supports error control in communication systems, relevant to design and testing (PSO1).

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	(E)	(M)	(I)	(V)	
3	0	0	3	60	40	00	00	100

Couse Contents			
Unit No.	**Content**	**No. of Hours**	**% of Weightage**
1.	**Introduction, Transmission Fundamentals:** Signals for Conveying Information, Analog and Digital Data Transmission, Channel Capacity, Transmission Media, Multiplexing Communication Networks: LANs, MANs and WANs, Switching Techniques, Circuit Switching, Packet Switching	7	14
2.	**Cellular Wireless Networks:** Principles of Cellular Networks, First-Generation Analog Second-Generation TDMA Second-Generation CDMA, Third-Generation Systems Antennas and Propagation: Antennas, Propagation Modes, Line-of-Sight Transmission, Fading in the Mobile Environment	10	22
3.	**Multiple access in Wireless System:** Multiple access scheme, frequency division multiple access, Time division multiple access, code division multiple access, space division multiple access, packet radio access, multiple access with collision avoidance. Global system for mobile communication: Global system for mobile communication, GSM architecture, GSM entities, call routing in GSM,	13	20

	PLMN interface, GSM addresses and identifiers, network aspects in GSM, GSM frequency allocation, authentication, and security. **General packet radio service (GPRS):** GPRS and packet data network, GPRS network architecture, GPRS network operation, data services in GPRS, Applications of GPRS, Billing and charging in GPRS. **Wireless System Operations and Standards:** Cordless Systems, Wireless Local Loop, WiMAX and IEEE 802.16 Broadband Wireless Access Standards Mobile IP and Wireless Application, Protocol		
4.	**Wi-Fi and the IEEE 802.11 Wireless LAN Standard:** IEEE 802.11 architecture and services, IEEE 802.11 Medium access control, IEEE 802.11 physical layer, Wi-Fi protected access.	4	10
5.	**Bluetooth:** Radio specification, baseband specification, link manager specification, logical link control and adaption protocol.	4	10
6.	Android APIs, Android Architecture, Application Framework, The Application components, The manifest file, downloading and installing Android, Exploring the Development Environment, Developing and Executing the first Android application, Working with Activities, The Linear Layout , The Relative Layout , The Scroll View Layout, The Table Layout, The Frame Layout, Using the Text View, Edit Text View, Button View, Radio Button, Checkbox, Image Button, Rating Bar , The options Menu, The Context Menu.	7	10
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

References/Suggested Learning Resources:

(a) Books:

1. Wireless Communications & Networks, Second Edition, William Stallings by Pearson.
2. Mobile Computing Technology, Applications, and service creation by Asoke K Telukder, Roopa R. Yavagal, TMH.
3. Wireless Communications, Principles and Practices by T. S. Rappaport, Pearson Education India.
4. Android Application Development Black Book by Pradeep Kothari, Dreamtech Press.
5. Wireless and Mobile Networks by Dr. Sunilkumar S. Manvi, Dr. Mahabaleshwar S.Kakkasageri, WILEY.
6. Wireless Networks by P. Nicopolitidis, M.S. Obaidat, G. I. Papadimitriou, A.S.Pomportsis by Wiley
7. Mobile Computing by Raj Kamal, Oxford.
8. Mobile Computing Theory and Practice-Kumkum Garg-Pearson.
9. Lauren Darcey and Shane Conder, Android Wireless Application Development, Pearson Education, 2nd ed. (2011).

(b) Open source software and website:

1. GeeksforGeeks – Mobile Computing
2. TutorialsPoint – Mobile Computing
3. MobiIndia Blog
4. Back4App Blog
5. LibHunt – Mobile Development

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

List of Laboratory/Learning Resources Required:

1. Flutter
2. React Native
3. Ionic
4. Xamarin
5. Apache Cordova (PhoneGap)
6. Kivy
7. Back4App

Suggested Project List:

1. **Mobile-Based Disaster Management System**: Alerts, evacuation routes, and rescue coordination.
2. **Mobile Voting for Local Governance**: Secure, transparent voting with blockchain backend.
3. **Mobile-Based Agricultural Advisory System**: Weather updates, crop suggestions, and market prices.
4. **Mobile-Based Mental Health Assistant**: Chatbot and journaling features with mood tracking.
5. **Mobile Emergency Alert System**: Location-based SOS alerts with live tracking.
6. **Mobile Cloud File Manager**: Upload, download, and manage files via cloud APIs.
7. **Mobile Learning App with Offline Mode**: Syncs content when online, usable offline.
8. **IoT-Based Home Automation App**: Control lights, fans, and appliances via mobile.

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103704

Course / Subject Name: Mobile Computing Technology

Suggested Course Practical List:

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

w. e. f. Academic Year:	2025-26
Semester:	7th CE
Category of the Course:	Professional Elective-4

Prerequisite:	• **Data Structures**: Trees, graphs, hash tables, inverted indexes • **Algorithms**: Searching, sorting, clustering, ranking • **Programming Skills**: Proficiency in Python, Java, or C++ (Python is widely used in modern IR)
Rationale:	**Modern Information Retrieval (IR)** lies in its critical role in helping users efficiently access relevant information from vast and complex data sources.

Course Outcomes (COs)

CO Code	Course Outcome Description
CO1	Understand the theoretical basis behind the standard models of IR (Boolean, Vector-space, Probabilistic and Logical models)
CO2	Apply appropriate method of text classification or clustering
CO3	Use performance evaluation metric for IR
CO4	Understand the standard methods for Web indexing and retrieval

Assumed Program Outcomes (POs)

PO Code	Description (Abbreviated)
PO1	Engineering Knowledge
PO2	Problem Analysis
PO3	Design/Development of Solutions
PO4	Conduct Investigations
PO5	Modern Tool Usage
PO6	Engineer and Society
PO7	Environment and Sustainability
PO8	Ethics
PO9	Individual and Team Work
PO10	Communication
PO11	Project Management
PO12	Lifelong Learning

CO PO MAPPING						
CO \ PO	**PO 1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6–PO12**
CO1	3	2				
CO2	2	3	3	2	3	
CO3	2	3		3	3	
CO4	2	2			3	

Teaching and Examination Scheme:

Hours per Week			Course Credits	Examination Marks				
				Theory Marks		Practical Marks		
L	**T**	**P**		**E**	**M**	**I**	**V**	**100**
3	**0**	**0**	**3**	**60**	**40**	**00**	**00**	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Information Retrieval:** The nature of unstructured and semi-structured text. Inverted index and Boolean queries.	5	10
2.	**Text Indexing, Storage and Compression:** Text encoding: tokenization, stemming, stop words, phrases, index optimization. Index compression: lexicon compression and postings lists compression. Gap encoding, gamma codes, Zipf's Law. Index construction. Postings size estimation, merge sort, dynamic indexing, positional indexes, n-gram indexes, real-world issues.	7	10
3.	**Retrieval Models:** Boolean, vector space, TFIDF, Okapi, probabilistic, language modeling, latent semantic indexing. Vector space scoring. The cosine measure. Efficiency considerations. Document length normalization. Relevance feedback and query expansion. Rocchio.	7	10
4.	**Performance Evaluation:** Evaluating search engines. User happiness, precision, recall, F-measure. Creating test collections: kappa measure, interjudge agreement.	4	10
5.	**Text Categorization and Filtering:** Introduction to text classification. Naive Bayes models. Spam filtering. Vector space classification using hyperplanes; centroids; k Nearest Neighbors. Support vector machine classifiers. Kernel functions. Boosting.	5	10
6.	**Text Clustering:** Clustering versus classification. Partitioning methods. k-means clustering. Mixture of Gaussians model. Hierarchical agglomerative clustering. Clustering terms using documents.	6	10
7	**Advanced Topics:** Summarization, Topic detection and tracking, Personalization, Question answering, Cross language information retrieval	6	15
8	**Web Information Retrieval:** Hypertext, web crawling, search engines, ranking, link analysis, Page Rank, HITS.	5	15
9	**Retrieving Structured Documents:** XML retrieval, semantic web	3	10
	TOTAL	**48**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze, Cambridge University Press, 2007.
2. Search Engines: Information Retrieval in Practice. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education, 2009.
3. Modern Information Retrieval. Baeza-Yates Ricardo and Berthier Ribeiro-Neto. 2nd edition, Addison-Wesley, 2011.
4. Information Retrieval: Implementing and Evaluating Search Engines. Stefan Buttcher, Charlie Clarke, Gordon Cormack, MIT Press, 2010.

(b) Open source software and website:

1. Awesome Information Retrieval A curated GitHub list of books, courses, datasets, software, and papers on IR.
2. Stanford IR Book Site Companion site for *Introduction to Information Retrieval* with slides and exercises.
3. Documind Blog – IR Methods Guide Step-by-step breakdown of IR models from Boolean to neural search.
4. Coveo Blog – IR Techniques & Algorithms Practical insights into IR systems used in enterprise and AI applications.
5. DocsBot AI – IR Techniques Overview Explains semantic search, relevance ranking, and evolution of IR systems

Suggested Course Practical List:

List of Laboratory/Learning Resources Required:

1. **Apache Lucene / Elasticsearch** Open-source search engines for building IR systems.
2. **FAISS (Facebook AI Similarity Search)** Efficient similarity search for dense vector representations.
3. **Haystack** Framework for building NLP-powered search systems using transformers.
4. **Kaggle Datasets** Try IR tasks like document ranking, semantic search, or query classification.
5. **TREC Collections** Benchmark datasets for evaluating IR systems.

Suggested Project List:

1. **Bibliometrics-Aided Retrieval**: Use citation networks to enhance academic search.
2. **Credibility in IR**: Develop models to rank results based on trustworthiness and source reliability.
3. **Music Information Retrieval**: Analyze and retrieve music based on audio features or lyrics.
4. **Math IR**: Semantify mathematical identifiers for better retrieval in academic papers.
5. **Ecological Syntheses IR**: Retrieve and synthesize ecological data for environmental research.
6. **Neural Information Retrieval**: Implement transformer-based models (e.g., BERT) for semantic search.
7. **Representation Learning**: Use multi-task deep neural networks for semantic classification and retrieval.
8. **Deep Sentence Embedding**: Apply LSTM or transformer models to improve document similarity.
9. **Query Auto-Completion**: Predict user queries using historical data and embeddings.

* * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

w. e. f. Academic Year:	2025-26
Semester:	7th CE
Category of the Course:	Professional Elective-4

Prerequisite:	• **Data Structures**: Trees, graphs, hash tables, inverted indexes • **Algorithms**: Searching, sorting, clustering, ranking • **Programming Skills**: Proficiency in Python, Java, or C++ (Python is widely used in modern IR)
Rationale:	**Modern Information Retrieval (IR)** lies in its critical role in helping users efficiently access relevant information from vast and complex data sources.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	understand the theoretical basis behind the standard models of IR (Boolean, Vector-space, Probabilistic and Logical models)	Understand
2	apply appropriate method of text classification or clustering.	Apply
3	use performance evaluation metric for IR	Apply
4	understand the standard methods for Web indexing and retrieval	Understand

Program Outcome

1. **Understand and explain the theoretical foundations** of standard Information Retrieval (IR) models, including Boolean, Vector Space, Probabilistic, and Logical models, and assess their applicability in different retrieval contexts.
2. **Apply appropriate techniques for text classification and clustering** to organize, categorize, and retrieve textual data efficiently in large-scale IR systems.
3. **Evaluate the performance of Information Retrieval systems** using standard IR evaluation metrics such as precision, recall, F-measure, Mean Average Precision (MAP), and normalized Discounted Cumulative Gain (nDCG).
4. **Understand and implement standard methods for Web indexing and retrieval**, including crawling, indexing, link analysis, and query processing, to design scalable and effective Web search systems.

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

CO PO Mapping							
CO No.	**Course Outcome Description**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6-PO12 (Optional)**
CO1	Understand the theoretical basis behind the standard models of IR (Boolean, Vector-space, Probabilistic, and Logical models)	✓ High	✓ Medium				PO12 (Lifelong Learning) – Low
CO2	Apply appropriate method of text classification or clustering	✓ High	✓ High	✓ Medium	✓ Medium		PO9 (Individual and Team Work) – Low
CO3	Use performance evaluation metrics for IR	✓ High	✓ Medium		✓		

Teaching and Examination Scheme:

Hours per Week			**Course Credits**	**Examination Marks**				
				Theory Marks		**Practical Marks**		
L	**T**	**P**		**E**	**M**	**I**	**V**	**100**
3	**0**	**0**	**3**	**60**	**40**	**00**	**00**	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103705

Course / Subject Name: Modern Information Retrieval

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Information Retrieval:** The nature of unstructured and semi-structured text. Inverted index and Boolean queries.	5	10
2.	**Text Indexing, Storage and Compression:** Text encoding: tokenization, stemming, stop words, phrases, index optimization. Index compression: lexicon compression and postings lists compression. Gap encoding, gamma codes, Zipf's Law. Index construction. Postings size estimation, merge sort, dynamic indexing, positional indexes, n-gram indexes, real-world issues.	7	10
3.	**Retrieval Models:** Boolean, vector space, TFIDF, Okapi, probabilistic, language modeling, latent semantic indexing. Vector space scoring. The cosine measure. Efficiency considerations. Document length normalization. Relevance feedback and query expansion. Rocchio.	7	10
4.	**Performance Evaluation:** Evaluating search engines. User happiness, precision, recall, F-measure. Creating test collections: kappa measure, interjudge agreement.	4	10
5.	**Text Categorization and Filtering:** Introduction to text classification. Naive Bayes models. Spam filtering. Vector space classification using hyperplanes; centroids; k Nearest Neighbors. Support vector machine classifiers. Kernel functions. Boosting.	5	10
6.	**Text Clustering:** Clustering versus classification. Partitioning methods. k-means clustering. Mixture of Gaussians model. Hierarchical agglomerative clustering. Clustering terms using documents.	6	10
7	**Advanced Topics:** Summarization, Topic detection and tracking, Personalization, Question answering, Cross language information retrieval	6	15
8	**Web Information Retrieval:** Hypertext, web crawling, search engines, ranking, link analysis, Page Rank, HITS.	5	15
9	**Retrieving Structured Documents:** XML retrieval, semantic web	3	10
	TOTAL	**48**	**100**

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	40	40	-	-	-

Suggested Specification Table with Marks (Theory):
Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze, Cambridge University Press, 2007.
2. Search Engines: Information Retrieval in Practice. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education, 2009.
3. Modern Information Retrieval. Baeza-Yates Ricardo and Berthier Ribeiro-Neto. 2nd edition, Addison-Wesley, 2011.
4. Information Retrieval: Implementing and Evaluating Search Engines. Stefan Buttcher, Charlie Clarke, Gordon Cormack, MIT Press, 2010.

(b) Open source software and website:

1. Awesome Information Retrieval A curated GitHub list of books, courses, datasets, software, and papers on IR.
2. Stanford IR Book Site Companion site for *Introduction to Information Retrieval* with slides and exercises.
3. Documind Blog – IR Methods Guide Step-by-step breakdown of IR models from Boolean to neural search.
4. Coveo Blog – IR Techniques & Algorithms Practical insights into IR systems used in enterprise and AI applications.
5. DocsBot AI – IR Techniques Overview Explains semantic search, relevance ranking, and evolution of IR systems

Suggested Course Practical List:

List of Laboratory/Learning Resources Required:

1. **Apache Lucene / Elasticsearch** Open-source search engines for building IR systems.
2. **FAISS (Facebook AI Similarity Search)** Efficient similarity search for dense vector representations.
3. **Haystack** Framework for building NLP-powered search systems using transformers.
4. **Kaggle Datasets** Try IR tasks like document ranking, semantic search, or query classification.
5. **TREC Collections** Benchmark datasets for evaluating IR systems.

Suggested Project List:

1. **Bibliometrics-Aided Retrieval**: Use citation networks to enhance academic search.
2. **Credibility in IR**: Develop models to rank results based on trustworthiness and source reliability.
3. **Music Information Retrieval**: Analyze and retrieve music based on audio features or lyrics.
4. **Math IR**: Semantify mathematical identifiers for better retrieval in academic papers.
5. **Ecological Syntheses IR**: Retrieve and synthesize ecological data for environmental research.
6. **Neural Information Retrieval**: Implement transformer-based models (e.g., BERT) for semantic search.
7. **Representation Learning**: Use multi-task deep neural networks for semantic classification and retrieval.
8. **Deep Sentence Embedding**: Apply LSTM or transformer models to improve document similarity.
9. **Query Auto-Completion**: Predict user queries using historical data and embeddings.

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name:

EngineeringLevel: UG

Course / Subject Code: BE103706

Course / Subject Name: Distributed System

w. e. f. Academic Year:	2025-26
Semester:	7th CE
Category of the Course:	Professional Elective-5

Prerequisite:	• TCP/IP, HTTP, DNS, sockets, routing • Protocols and communication models (e.g., client-server, peer-to-peer) • Proficiency in languages like Java, Python, Go, or C++ • Familiarity with multithreading and asynchronous programming
Rationale:	**Distributed Systems** lies in solving problems that single machines can't handle efficiently, reliably, or at scale.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Understand the fundamental concepts and goals of distributed systems	Understand
2	Analyze system architectures, process models, and communication mechanisms in distributed systems	Apply
3	Apply naming and synchronization techniques in distributed systems	Apply
4	Understand consistency models, replication strategies, and fault tolerance mechanisms	Understand
5	Evaluate security threats and implement basic security mechanisms in distributed systems	Understand

Program Outcome:

Graduates will be able to understand and analyze the core principles and goals of distributed systems, including various architectural and process models. They will gain the ability to evaluate and apply communication protocols, naming conventions, and synchronization techniques essential to distributed environments. Students will demonstrate competence in understanding consistency models, designing replication strategies, and implementing fault-tolerant systems. Additionally, they will be equipped to identify

security threats and apply foundational security mechanisms to ensure system integrity and confidentiality in distributed computing contexts.

Course Outcomes (COs)	Description	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	Understand the fundamental concepts and goals of distributed systems	✓											
CO2	Analyze system architectures, process models, and communication mechanisms in distributed systems	✓	✓	✓									
CO3	Apply naming and synchronization techniques in distributed systems	✓	✓	✓									
CO4	Understand consistency models, replication strategies, and fault tolerance mechanisms	✓	✓	✓	✓								

Teaching and Examination Scheme:

Hours per Week			Course Credits	Examination Marks				
				Theory Marks		Practical Marks		
L	T	P		E	M	I	V	150
3	0	2	4	60	40	20	30	

SWAMINARAYAN UNIVERSITY

Program Name:

EngineeringLevel: UG

Course / Subject Code: BE103706

Course / Subject Name: Distributed System

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Fundamentals of Distributed System:** Definition of a Distributed System, Goals of a Distributed System, Types of Distributed Systems, Basics of Operating System and Networking.	2	10
2.	**Basics of Architectures, Processes, and Communication:** Architectures - Types of System Architectures, Self Management in Distributed Systems; Processes - Basics of Threads, Virtualization, Roles of Client and Server, Code Migration; Communication - Types of Communications, Remote Procedure Calls, Message-Oriented Communication, Stream-Oriented Communication, Multicasting	8	20
3.	**Naming** - Names, Identifiers, And Addresses, Flat Naming, Structured Naming, Attribute-Based Naming	2	10
4.	**Synchronization** - Clock Synchronization, Logical Clocks, Mutual Exclusion, Global Positioning Of Nodes, Election Algorithms	4	10
5.	**Consistency, Replication and Fault Tolerance**-Introduction To Replication, Data-Centric Consistency Models, Client-Centric Consistency Models, Replica Management, Consistency Protocols, Basics of Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit, Recovery	10	20
6.	**Security:** Introduction to Security- Security Threats, Policies, and Mechanisms, Design Issues, Basics of Cryptography, Secure Channels- Authentication, Message Integrity and Confidentiality, Secure Group Communication; Access Control- General Issues in Access Control, Firewalls, Secure Mobile Code, Denial of Service; Security Management-Key Management, Secure Group Management, Authorization Management	7	20
7	**Categories of Distributed System:** Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, Security: Distributed Object-based System;Distributed File System;Distributed Web-based System; Distributed Coordination based System	9	10
	TOTAL	**42**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per

References/Suggested Learning Resources:

(a) Books:

1. Tanenbaum, A.S. and Van Steen, M., 2007. Distributed systems: principles and paradigms. Prentice-Hall.
2. Sinha, P.K., 1998. Distributed operating systems: concepts and design. PHI Learning Pvt. Ltd..
3. Liu, M.L., 2003. Distributed computing: principles and applications. Pearson Education Inc..
4. Lynch, N.A., 1996. Distributed algorithms. Elsevier.
5. Coulouris, G.F., Dollimore, J. and Kindberg, T., 2005. Distributed systems: concepts and design.pearson education.

(b) Open source software and website:

Open Source Software

1. **Apache Hadoop** – Distributed storage and processing using MapReduce
2. **Apache Spark** – Fast distributed data processing engine
3. **Kubernetes** – Container orchestration for distributed applications
4. **Apache Kafka** – Distributed event streaming platform
5. **Redis** – In-memory key-value store with distributed capabilities
6. **CockroachDB** – Distributed SQL database with strong consistency

Resource Websites

1. GeeksforGeeks Distributed Systems Tutorial
2. TutorialsPoint Distributed Systems
3. NPTEL Online Course on Distributed Systems
4. Educative Blog on Distributed Systems
5. FreeCodeCamp Introduction to Distributed Systems
6. SourceForge Distributed Computing Directory
7. LibHunt Distributed Computing Projects

Suggested Course Practical List:

1. Write a Program to implement Concurrent Echo Client Server Application.
2. Write at least 2 Programs for Remote Procedure call.
3. Write at least 2 Programs for Remote Method Invocation.
4. Write the Programs for Thread Programming in JAVA.
5. Implement Network File System (NFS).
6. Creation of a BPEL (Business Process Execution Language) Module and a Composite Application.
7. Implement CORBA file.
8. Study of Web Service Programming
9. Study of open source key management tool.

List of Laboratory/Learning Resources Required:

1. MIT 6.824: Distributed Systems – Graduate-level course with hands-on labs in Go
2. NPTEL Distributed Systems Course – University-level course from IITs
3. GeeksforGeeks Distributed Systems Tutorial – Covers basics to advanced topics
4. TutorialsPoint Distributed Systems – Structured lessons with diagrams
5. Coursera Distributed Systems Courses – Courses from top universities and companies
6. Class Central Distributed Systems Catalog – Aggregated list of free and paid courses

Suggested Project List:

1. **Distributed Blockchain Ledger** – Implement a basic blockchain with consensus
2. **Distributed Microservices Architecture** – Build a system with service discovery and orchestration
3. **Distributed SQL Database** – Create a scalable, consistent database like CockroachDB
4. **Event Streaming Platform** – Build a Kafka-like system for real-time data pipelines
5. **Decentralized Firebase Alternative** – Explore DB3 Network or similar open-source projects

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103707

Course / Subject Name: Information Security

w. e. f. Academic Year:	2025-26
Semester:	7th CE
Category of the Course:	Professional Elective-5

Prerequisite:	• Encryption and decryption • Hashing algorithms (SHA, MD5) • Public Key Infrastructure (PKI)
Rationale:	**Information Security** lies in its critical role in safeguarding the integrity, confidentiality, and availability of data in an increasingly digital world.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Explore the basic principles of the symmetric cryptography and techniques with their strengths and weaknesses from perspective of cryptanalysis	Apply
2	Implement and analyze various symmetric key cryptography algorithms and their application in different context.	Analyze
3	Compare public key cryptography with private key cryptography and Implement various asymmetric key cryptography algorithms.	Understand
4	Explore the concept of hashing and implement various hashing algorithms for message integrity.	Apply
5	Explore and use the techniques and standards of digital signature, key management and authentication.	Apply

Program Outcome: Cryptography and Network Security

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

1. **Demonstrate foundational knowledge of symmetric key cryptography**, including encryption techniques such as block and stream ciphers, and critically analyze their strengths and weaknesses from a cryptanalytic perspective.
2. **Implement and evaluate symmetric encryption algorithms** (e.g., DES, AES, RC4) in various application scenarios, demonstrating understanding of their security implications and performance characteristics.
3. **Compare and contrast symmetric (private key) and asymmetric (public key) cryptography**, identifying use-case scenarios, advantages, and limitations of each.

4. **Design and implement asymmetric cryptographic algorithms** such as RSA, ElGamal, and ECC, applying them in real-world contexts like secure key exchange and digital signatures.
5. **Understand the principles of hashing** and implement secure hash functions (e.g., MD5, SHA family) to ensure data integrity in communication and storage systems.
6. **Apply digital signature schemes, key management practices, and user authentication techniques** to build secure communication systems, following current cryptographic standards and protocols.
7. **Analyze and evaluate cryptographic systems for potential vulnerabilities**, and recommend improvements or alternative approaches based on security needs and performance trade-offs.

CO PO MAPPING							
CO No.	**Course Outcome Description**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6-PO12 (as applicable)**
CO1	Explore the basic principles of symmetric cryptography and techniques with their strengths and weaknesses from the perspective of cryptanalysis	3	2	1	1	1	-
CO2	Implement and analyze various symmetric key cryptography algorithms and their application in different contexts	3	3	2	2	2	PO12 (1)
CO3	Compare public key cryptography with private key cryptography and implement various asymmetric key cryptography algorithms	3	3	3	2	2	PO12 (1)
CO4	Explore the concept of hashing and implement various hashing algorith						

Teaching and Examination Scheme:

Hours per Week			**Course Credits**	**Examination Marks**				
				Theory Marks		**Practical Marks**		
L	**T**	**P**		**E**	**M**	**I**	**V**	**150**
3	**0**	**2**	**4**	**60**	**40**	**20**	**30**	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103707

Course / Subject Name: Information Security

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Symmetric Cipher Model, Cryptography, Cryptanalysis and Attacks; Substitution and Transposition techniques	3	10
2.	Stream ciphers and block ciphers, Block Cipher structure, Data Encryption standard (DES) with example, strength of DES, Design principles of block cipher, AES with structure, its transformation functions, key expansion, example and implementation	8	20
3.	Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode	4	10
4.	Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security, Diffie-Hillman Key Exchange algorithm, Man-in-Middle attack	8	20
5.	Cryptographic Hash Functions, their applications, Simple hash functions, its requirements and security, Hash functions based on Cipher Block Chaining, Secure Hash Algorithm (SHA)	5	10
6.	Message Authentication Codes, its requirements and security, MACs based on Hash Functions, Macs based on Block Ciphers	5	10
7.	Digital Signature, its properties, requirements and security, various digital signature schemes (Elgamal and Schnorr), NIST digital Signature algorithm	4	10
8.	Key management and distribution, symmetric key distribution using symmetric and asymmetric encryptions, distribution of public keys, X.509 certificates, Public key	4	10
	TOTAL	**41**	**100**

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103707

Course / Subject Name: Information Security

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per

References/Suggested Learning Resources:

(a) Books:

1. Cryptography And Network Security, Principles And Practice Sixth Edition, William Stallings, Pearson
2. Information Security Principles and Practice By Mark Stamp, Willy India Edition
3. Cryptography & Network Security, Forouzan, Mukhopadhyay, McGrawHill
4. Cryptography and Network Security Atul Kahate, TMH
5. Cryptography and Security, C K Shyamala, N Harini, T R Padmanabhan, Wiley-India
6. Information Systems Security, Godbole, Wiley-India
7. Information Security Principles and Practice, Deven Shah, Wiley-India
8. Security in Computing by Pfleeger and Pfleeger, PHI
9. Build Your Own Security Lab : A Field Guide for network testing, Michael Gregg, Wiley India

(b) Open source software and website:

Open Source Software for Information Security

1. **Wireshark** – Network protocol analyzer for packet inspection
2. **Nmap** – Network scanner for host discovery and port scanning
3. **Metasploit Framework** – Penetration testing and exploit development
4. **John the Ripper** – Password cracking tool
5. **Snort** – Intrusion detection and prevention system
6. **Aircrack-ng** – Wi-Fi network security testing suite
7. **OWASP ZAP** – Web application vulnerability scanner

Open Source Websites for Information Security

1. OWASP Foundation – Free tools for secure software development
2. LetsDefend GitHub Repository – Free cybersecurity tools, blogs, courses, and news
3. SourceForge Security Directory – Browse hundreds of open-source security software projects
4. Data Stack Hub – List of top open-source cybersecurity tools
5. Solutions Review – Tools for identity management, endpoint security, and SIEM
6. All About Testing – Top OSINT tools for ethical hacking and reconnaissance

Suggested Course Practical List:

List of Laboratory/Learning Resources Required:

1. Cybrary Offers free practice tests, labs, and assessments for all skill levels
2. **TryHackMe** – Gamified cybersecurity learning with real-world scenarios
3. **Hack The Box** – Advanced penetration testing labs
4. **OverTheWire** – Wargames for learning Linux and security concepts

Suggested Project List:

1. Web Application Vulnerability Scanner
2. Create a tool that scans for XSS, CSRF, and other common web vulnerabilities.
3. Digital Watermarking for Image Authentication
4. Embed hidden data in images to verify authenticity.
5. Android Text Encryption App
6. Build a mobile app that encrypts messages using various algorithms.
7. Secure Persona Prediction & Data Leakage Prevention
8. Use Python to analyze user behavior and prevent sensitive data leaks.
9. Detecting Phishing Websites Using Machine Learning
10. Train a model to identify phishing URLs based on features like domain age and SSL status

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103708

Course / Subject Name: Parallel and Distributed Computing

w. e. f. Academic Year:	2025-26
Semester:	7th CE
Category of the Course:	Professional Elective-5

Prerequisite:	• Programming Skills • Parallel Programming Models • Networking Basics • Linux/Unix Environment
Rationale:	**Parallel and Distributed Computing (PDC)** lies in its ability to solve complex, large-scale problems efficiently by leveraging multiple computing resources simultaneously.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.	Apply
2	Analyze the various design principles of parallel algorithms.	Analyze
3	Learn the intricacies of parallel and distributed programming	Understand
4	Develop and execute basic parallel and distributed applications using basic programming models and tools.	Apply
5	Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.	Apply

Program Outcome (PO):

Standard Program Outcomes (POs):

1. **Engineering knowledge**
2. **Problem analysis**
3. **Design/development of solutions**
4. **Conduct investigations of complex problems**
5. **Modern tool usage**
6. **The engineer and society**
7. **Environment and sustainability**

8. **Ethics**
9. **Individual and team work**
10. **Communication**
11. **Project management and finance**
12. **Life-long learning**

CO PO MAPPING									
CO\PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO9**	**PO12**	**PSO1**	**PSO2**
CO1	✓	✓			✓			✓	
CO2		✓	✓	✓				✓	
CO3	✓				✓		✓	✓	✓
CO4			✓		✓	✓			

Teaching and Examination Scheme:

Hours per Week			Course Credits	Examination Marks				
				Theory Marks		Practical Marks		
L	T	P		E	M	I	V	150
3	0	2	4	60	40	20	30	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103708

Course / Subject Name: Parallel and Distributed Computing

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Parallel Computing:** The Idea of Parallelism, Power and potential of parallelism, Examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instructions streams.	5	10
2.	**Parallel Architecture:** Pipeline architecture, Array processor, Multi processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).	10	20
3.	**Parallel Algorithm Design Principles and Programming:** Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.	10	20
4.	**Introduction to Distributed Systems:** Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.	6	20
5.	**Distributed Computing and Communication design principles:** AModel of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.	8	20
6.	**Parallel and Distributed Programming Frameworks** Overview of CUDA, OpenMP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.	6	10
	TOTAL	**45**	**100**

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103708

Course / Subject Name: Parallel and Distributed Computing

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per

References/Suggested Learning Resources:

(a) Books:

1. Introduction to Parallel Computing (2nd Edition), Ananth Grama, Anshul Gupta, and George Karypis, Vipin Kumar, Addition Wesley
2. Parallel and Distributed Systems 2nd Edition, Arun Kulkarni, Nupur Prasasd Giri, Nikhilesh Joshi, Bhushan Jadhav, Wiley
3. Introduction To Parallel Programming, Steven Brawer, Academic Press
4. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley
5. Distributed Operating Systems, A.S. Tanenbaum, Prentice Hall
6. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
7. Parallel Programming in C with MPI and OpenMP Michael J.Quinn, McGrawHill Higher Education

(b) Open source software and website:

Open Source Software for Parallel and Distributed Computing

1. **Apache Hadoop** – Distributed storage and processing using MapReduce
2. **Apache Spark** – Fast, in-memory distributed data processing
3. **Dask** – Parallel computing in Python for data science workflows
4. **MPI (OpenMPI / MPICH)** – Message Passing Interface for parallel programming
5. **Ray** – Distributed computing framework for Python applications
6. **Charm++** – Parallel programming framework based on C++

Websites for Parallel and Distributed Computing Resources

1. GitHub – Awesome Parallel Computing A curated list of libraries, frameworks, books, and tutorials for parallel computing.
2. Dask Official Website Learn how to scale Python tools for parallel and distributed workloads.
3. AnalyticsPedia Guide Detailed guide on using Hadoop, Spark, Kubernetes, and Kafka for distributed computing.
4. SourceForge Distributed Computing Projects Browse hundreds of open-source distributed computing software projects.

5. LibHunt Distributed Computing Projects Discover trending open-source projects in distributed computing.

Suggested Course Practical List:

List of Laboratory/Learning Resources Required:

1. **OpenMP / MPI** – For parallel and distributed programming
2. **Apache Hadoop / Spark** – For big data distributed processing
3. **CUDA / OpenCL** – For GPU-based parallel computing
4. **Docker / Kubernetes** – For containerized distributed systems
5. **Git & GitHub** – For version control and collaboration

Suggested Project List:

1. **Load Balancing in Distributed Systems**
2. Design algorithms to evenly distribute tasks across nodes to avoid bottlenecks.
3. **Parallelizing Octave (Matlab Lookalike)**
4. Modify Octave to support parallel execution of mathematical functions.
5. **Checkpointing and Recovery in Distributed Applications**
6. Implement fault tolerance using state-saving and rollback mechanisms.
7. **Master-Worker Paradigm for Hybrid Memory Models**
8. Combine shared and distributed memory techniques for scalable computation.
9. **Superlinear Speedup Analysis in Parallel Systems**
10. Investigate scenarios where parallel execution outperforms linear scaling

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103709

Course / Subject Name: Intelligent Data Analysis

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	**-Professional Elective -6**

Prerequisite:	Programming skills
Rationale:	Today's world is a data-driven world. Increasingly, the efficient operation of organizations across sectors relies on the effective use of vast amounts of data. Big data analytics helps us to examine these data to uncover hidden patterns, correlations, and other insights. It is a fast-growing field and skills in the area are some of the most in-demand today.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	identify big data application areas models and tools.	Understand
2	use big data framework	Apply
3	model and analyze data by applying selected techniques	Apply
4	demonstrate an integrated approach to big data	Apply
5	identify big data application areas models and tools.	Understand

Program Outcome		
Course Learning Outcome (CLO)	**Mapped Program Outcomes (POs)**	**Justification / Description**
1. Identify big data application areas, models, and tools.	PO1, PO2	✓ PO1: Apply knowledge of computing fundamentals.✓ PO2: Identify computing requirements for big data problems and applications across industries (e.g., healthcare, finance, retail).

Program Outcome		
Course Learning Outcome (CLO)	**Mapped Program Outcomes (POs)**	**Justification / Description**
2. Use big data frameworks (e.g., Hadoop, Spark) for managing and processing large datasets.	PO3, PO5	✓ PO3: Design solutions using modern tools (Hadoop, Spark, Hive).✓ PO5: Use appropriate techniques and tools for complex data processing tasks.
3. Model and analyze data using selected data analytics techniques (e.g., machine learning, data mining).	PO4, PO5	✓ PO4: Conduct investigations and use data analysis for meaningful insights.✓ PO5: Apply data modeling tools (e.g., R, Python, Weka, MLlib) for analysis.
4. Demonstrate an integrated approach to big data analytics in real-world scenarios.	PO6, PO11	✓ PO6: Understand societal and professional responsibilities in applying big data.✓ PO11: Understand emerging trends and lifelong learning for evolving big data ecosystems.

CO PO Mapping												
Course Outcome (CO)	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1: Identify big data application areas, models, and tools	2	2	1	-	3	-	-	-	-	-	-	2
CO2: Use big data frameworks (like Hadoop, Spark) for managing and processing data	3	2	3	2	3	-	-	-	-	-	-	3
CO3: Model and analyze data using selected big data techniques (e.g., ML, clustering)	3	3	3	2	3	-	-	-	-	-	-	3
CO4: Demonstrate an integrated approach to handling, processing, and analyzing big data	3	3	3	3	3	-	-	-	-	-	-	3

Teaching and Examination Scheme:

Hours per Week			**Course Credits**	**Examination Marks**				
				Theory Marks		**Practical Marks**		
L	**T**	**P**		**E**	**M**	**I**	**V**	**150**
3	**0**	**2**	**4**	**60**	**40**	**20**	**30**	

Course Content:

UnitNo.	Content	No. of Hours	% of Weightag
1.	**Introduction to Big Data:** Introduction to Big Data, Big Data characteristics, Challenges of Conventional System, Types of Big Data, Intelligent data analysis, Traditional vs. Big Data business approach, Case Study of Big Data Solutions.	4	
2.	**Hadoop:** History of Hadoop, Hadoop Distributed File System: Physical organization of Compte Nodes, Components of Hadoop Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Design of HDFS,Java interfaces to HDFS Basics, Developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Map Reduce Types and Formats, Map Reduce Features, Hadoop environment. Setting up a Hadoop Cluster, Cluster specification, Cluster Setup and Installation, Hadoop Configuration, ecurity in Hadoop, Administering Hadoop, Monitoring-Maintenance, Hadoop benchmarks, Hadoop in the cloud	12	
3.	**NoSQL:** What is NoSQL? NoSQL business drivers; NoSQL case studies; NoSQL data architecture patterns: Key-value stores, Graph stores, Column family (Bigtable) stores, Document stores, Variations of NoSQL architectural patterns; Using NoSQL to manage big data: What is a big data NoSQL solution? Understanding the types of big data problems; Analyzing big data with a shared-nothing architecture; Choosing distribution models: master-slave versus peer-to-peer; Four ways that NoSQL systems handle big data problems	7	
4.	**Mining Data Stream:** Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating moments, Counting oneness in a Window, Decaying Window, Real time Analytics Platform (RTAP) applications, Case Studies, Real Time Sentiment Analysis, Stock Market Predictions. Using Graph Analytics for Big Data: Graph Analytics	10	
5.	**Frameworks:** Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services, HiveQL, Querying Data in Hive, fundamentals of HBase and Zoo Keeper, IBM Info Sphere Big Insights and Streams.	12	
6.	**Spark:** Introduction to Data Analysis with Spark, In-Memory Computing with Spark, Spark Basics, Interactive Spark with PySpark, Writing Spark Applications		
	Total	**48**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

(b) Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007

(c) Bill Franks , "Taming The Big Data Tidal Wave: Finding Opportunities In Huge Data Streams With Advanced Analytics", Wiley

(d) Anand Rajaraman and Jeff Ullman "Mining of Massive Datasets", Cambridge University Press,

(e) Michael Minelli, Michele Chambers, Ambiga Dhiraj, "Big Data Big Analytics: Emerging Business Intelligence And Analytic Trends For Today's Businesses", Wiley India

(f) Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", Wiley.

(g) Chris Eaton, Dirk derooset al., "Understanding Big data", McGraw Hill, 2012.

(h) BIG Data and Analytics , Seema Acharya, Subhashini Chhellappan, Willey

(i) MongoDB in Action, Kyle Banker, Piter Bakkum , Shaun Verch, Dream tech Press

(j) Tom White, "HADOOP: The Definitive Guide", O Reilly 2012.

(k) Vignesh Prajapati, "Big Data Analytics with R and Hadoop", Packet Publishing 2013.

(l) Learning Spark: Lightning-Fast Big Data Analysis Paperback by Holden Karau.

Open source software and website:

Tool	Purpose	Type
Apache Hadoop	Distributed storage and processing	Framework (Batch)
Apache Spark	In-memory computing, real-time processing	Framework (Real-time)
Apache Kafka	Real-time messaging system	Streaming
MongoDB	NoSQL database for unstructured data	Storage/Database
Elasticsearch	Search engine for analytics and text data	Query/Indexing
Apache Flink	Stream and batch processing	Framework
Jupyter Notebook	Interactive data exploration & visualization	Notebook/IDE
TensorFlow / PyTorch	ML and Deep Learning	Analytics/Modeling

Suggested Course Practical List:

Case Study:

Stage 1:

Selection of case study topics and formation of small working groups of 2/3 students per group. Students engage with the cases, read through background material provided in the session and work through an initial set of questions to deepen the understanding of the case. Sample applications and data will be provided to help students familiarize themselves with the cases and available (big) data.

Stage 2:

The groups are given a specific task relevant to the case in question and are expected to develop a corresponding big data concept using the knowledge gained in the course and the parameters set by the case study scenario. A set of questions that help guide through the scenarios will be provided.

Stage 3:

Each group prepares a short 2 – 5 page report on their results and a 10 min oral presentation of their big data concept.

Apart from case student students will perform at the following programming exercises:

1. Implement following using Map- Reduce
a. Matrix multiplication
b. Sorting
c. Indexing
2. Distributed Cache & Map Side Join, Reduce side Join Building and Running a Spark Application Word count in Hadoop and Spark Manipulating RDD
3. Implementation of Matrix algorithms in Spark Sql programming
4. Implementing K-Means Clustering algorithm using Map-Reduce
5. Implementing any one Frequent Item set algorithm using Map-Reduce
6. Create A Data Pipeline Based On Messaging Using PySpark And Hive - Covid-19 Analysis

List of Laboratory/Learning Resources Required:

Suggested Project List:

* * * * * * *

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103711

Course / Subject Name: – Introduction to Machine Learning

w. e. f. Academic Year:	2025-26
Semester:	7th
Category of the Course:	- Professional Elective -6

Prerequisite:	Programming and Data Structure, Algorithms, Probability and Statistics.
Rationale:	The objective of the course is to introduce the students with concepts of machine learning, machine learning algorithms and building the applications using machine leaning for various domains.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Explore the fundamental issues and challenges in Machine Learning	Understand
2	including data and model selection and complexity	Apply
3	Appreciate the underlying mathematical relationships within and across	Apply
4	Machine Learning algorithms	Apply
5	Evaluate the various Supervised Learning algorithms using appropriate	Understand

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

1. **Explore** the fundamental issues and challenges in Machine Learning, including data acquisition and preprocessing, model selection, overfitting, underfitting, and managing model complexity.
2. **Demonstrate an understanding** of the underlying mathematical foundations and relationships that govern various Machine Learning algorithms, including concepts from linear algebra, probability, optimization, and statistics.

3. **Evaluate** and compare the performance of various Supervised Learning algorithms (e.g., Linear Regression, Decision Trees, SVM, k-NN, Neural Networks) using appropriate performance metrics and validation techniques (e.g., cross-validation, confusion matrix, ROC curves).

CO PO Mapping					
CO \ PO	**PO1**	**PO2**	**PO3**	**PO5**	**PO12**
CO1	✓	✓		✓	✓
CO2	✓	✓			✓
CO3	✓	✓	✓	✓	

Teaching and Examination Scheme:

Hours per Week			Course Credits	Examination Marks				
				Theory Marks		Practical Marks		
L	T	P		E	M	I	V	150
3	0	2	4	60	40	20	30	

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103711

Course / Subject Name: – Introduction to Machine Learning

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Machine Learning:** Overview of Human Learning and Machine Learning, Types of Machine Learning, Applications of Machine Learning, Tools and Technology for Machine Learning .	2	4
2.	**Preparing to Model:** Machine Learning activities, Types of data in Machine Learning, Structures of data, Data quality and remediation, Data Pre-Processing: Dimensionality reduction, Feature subset selection.	4	8
3.	**Modeling and Evaluation:** Selecting a Model: Predictive/Descriptive, Training a Model for supervised learning, model representation and interpretability, Evaluating performance of a model, Improving performance of a model.	5	11
4.	**Basics of Feature Engineering:** Feature and Feature Engineering, Feature transformation: Construction and extraction, Feature subset selection : Issues in high-dimensional data, key drivers, measure and overall process	3	6
5.	**Overview of Probability:** Statistical tools in Machine Learning, Concepts of probability, Random variables, Discrete distributions, Continuous distributions, Multiple random variables, Central limit theorem, Sampling distributions, Hypothesis testing, Monte Carlo Approximation	4	8
6.	**Bayesian Concept Learning:** Impotence of Bayesian methods, Bayesian theorem, Bayes' theorem and concept learning, Bayesian Belief Network	5	11
7	**Supervised Learning:** Classification and Regression: Supervised Learning, Classification Model, Learning steps, Classification algorithms, Regression, Regression algorithms	10	22
8	**Unsupervised Learning:** Supervised vs. Unsupervised Learning, Applications, Clustering, Association rules	6	13
9	**Neural Network:** Introduction to neural network, Biological and Artificial Neurons, Types of Activation functions, Implementation of ANN, Architecture, Leaning process, Back propagation, Deep Learning	6	13
	Total	45	100

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	**40**	**40**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Books:

(a) Machine Learning, Saikat Dull, S. Chjandramouli, Das, Pearson
(b) Machine Learning with Python for Everyone, Mark Fenner, Pearson
(c) Machine Learning, Anuradha Srinivasaraghavan, Vincy Joseph, Wiley
(d) Machine Learning with Python, U Dinesh Kumar Manaranjan Pradhan, Wiley
(e) Python Machine Learning, Sebastian Raschka, Vahid Mirjalili, Packet Publishing

Open source software and website:

Library	Language	Description
TensorFlow	Python, C++	Developed by Google, widely used for both research and production. Great for deep learning.
PyTorch	Python	Developed by Facebook. Dynamic computation graph, intuitive, dominant in academia and research.
scikit-learn	Python	Built on NumPy/SciPy. Best for traditional ML algorithms (SVM, Random Forest, etc.).
JAX	Python	From Google. High-performance ML with composable function transformations and GPU/TPU support.
Keras	Python	High-level API for building and training deep learning models, now tightly integrated with TensorFlow.
ML.NET	C#	Microsoft's ML framework for .NET developers.
Shogun	C++ (with interfaces for Python, R, Java)	Focuses on kernel methods and SVMs.

Suggested Course Practical List:

1. Supervised Learning Basics
2. Data Preprocessing
3. Tree-based Models
4. Model Evaluation and Tuning
5. Unsupervised Learning
6. Natural Language Processing (NLP)
7. Computer Vision
8. Time Series Forecasting
9. Deep Learning
10. Model Deployment
11. End-to-End ML Pipeline

SWAMINARAYAN UNIVERSITY
Program Name: Engineering
Level: UG
Course / Subject Code: BE103405
Course / Subject Name: Computer Organization & Architecture

List of Laboratory/Learning Resources Required:

Suggested Project List:

* * * * * * *