THE NATIONAL INSTITUTE OF ENGINEERING

(AUTONOMOUS INSTITUTE UNDER V.T.U)

Department of Information Science & Engineering

Scheme of Teaching and Blown-up Syllabus

Academic Year 2025-26

Semester -7th & 8th

Department Vision

"The Department will provide quality and value-based education to produce innovative world-class computing engineers and will enhance quality research for the betterment of society."

Department Mission

- To impart high-quality training, education, and competence in the information science domain through best-in-class faculty and facilities.
- To produce globally acceptable information science graduates who can contribute professionally to industry and research activities by offering courses on emerging technologies.
- To provide platforms to work effectively and innovatively in multi-disciplinary domains.

Programme Educational Objectives (PEOs)

PEO 1: Be professionally successful in the field of Information and emerging technologies.

PEO 2: Achieve success in pursuing higher studies at globally recognized institutions.

Programme Specific Objectives (PSOs)

PSO 1: Apply the knowledge of Information Systems in the field of Engineering to provide solutions through programming skills.

PSO 2: Collaborate and communicate effectively with professionals in the field of computing, engage in continuous learning, and address societal issues.

Scheme of Teaching & Examination (2022 Scheme)

Department: Information Science & Engineering

B.E. 2022 Admitted Batch

Semester: VII

Sl.No	Type of Course	Course Code	Course Title	Teaching Department (TD)	Question Paper setting Board (PSB)	Teaching Hrs./Week			Examination				Credits
						L	T	P	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
1	IPCC	BIS701	Big Data Analytics	IS&E	IS&E	3	2	0	3	50	50	100	**4**
2	PCC	BIS702	Software Testing and Quality Assurances	IS&E	IS&E	3	0	2	3	50	50	100	**4**
3	PEC	BXX713X	Professional Elective Course - Group III	IS&E	IS&E	3	0	0	3	50	50	100	**3**
4	OEC	BXX754X	Open Elective Course - Group II	IS&E	IS&E	3	0	0	3	50	50	100	**3**
5	PROJ	BXX785	Major Project	IS&E	IS&E	0	0	12	3	100	100	200	**6**
									Total	300	300	600	**20**

Professional Elective Course - Group III

BIS713A - Deep Learning	BIS713B - Distributed File Systems	BIS713C - Embedded Systems

Open Elective Course - Group II (Offered by the IS&E Department to students of non-CS Streams)

BIS754A - Introduction to DBMS	BIS754B - Introduction to Algorithms	BIS754C - Software Engineering

The National Institute of Engineering

Scheme of Teaching & Examination (2022 Scheme)

Department: Information Science & Engineering

B.E. 2022 Admitted Batch

Semester : VIII

Sl.No	Type of Course	Course Code	Course Title	Teaching Department (TD)	Question Paper setting Board (PSB)	Teaching Hrs/Week				Examination				Credits
						L	T	P	S	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
1	PEC	BIS801X	Professional Elective - Group IV (Online Course)	IS	IS	-	-	-		-	-	50	100	**3**
2	OEC	BIS802X	Open Elective - Group III (Online Course)	IS	IS	-	-	-		-	-	50	100	**3**
3	INT	BIS803	Internship(Industry/ Research) (14-16 weeks)	IS	IS	0	0	20		3	100	100	200	**10**
Total											**100**	**200**	**400**	**16**

Professional Elective Course - Group IV (Online Courses - NPTEL)			
BIS801A	Elective –I	BIS801D	Elective –IV
BIS801B	Elective –II	BIS801E	Elective –V
BIS801C	Elective –III		

Open Elective Course - Group III (Online Courses - NPTEL)			
BIS802A	Open Elective – I	BIS802C	Open Elective –III
BIS802B	Open Elective –II	BIS802D	Open Elective –IV

THIS PAGE IS INTENTIONALLY LEFT BLANK

7th Semester

Course Code: BIS701	**Course:** Big Data Analytics
Credits: 4	**L:T:P -** 3:2:0
CIE: 100Marks	**SEE:** 100 Marks
SEE Hours: 3 hours	**Total Marks:**100

Prerequisites if any	Statistics, Python or R programming language, RDBMS and data visualization tool.
Learning objectives	1. To implement MapReduce programs for processing big data. 2. To realize storage and processing of big data using MongoDB, Pig, Hive and Spark. 3. To analyze big data using machine learning techniques.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Identify and list various Big Data concepts, tools and applications.	Apply
CO2	Develop programs using HADOOP framework.	Analyze
CO3	Use Hadoop Cluster to deploy Map Reduce jobs, PIG, HIVE and Spark programs.	Analyze
CO4	Analyze the given data set and identify deep insights from the data set.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2			2							1		1
CO2	3	2	3		2							1	1	1
CO3	3	2	3	2	3				1	2		1		
CO4	3	3	3	3	2				1	2		2	2	

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1	**No. of Lecture Hours**	**No. of Tutorial Hours***	**No. of Practical Hours**
1.1	Classification and Characteristics of Data	1	-	-
1.2	What is Big Data and Why is it Important?	1	-	-
1.3	Characteristics of Big Data (The 5Vs)	1	-	-
1.4	Traditional Business Intelligence vs Big Data Approaches	1	-	-
1.5	Data Warehousing vs Hadoop Environments	1	-	-
1.6	Introduction to Big Data Analytics (Definition, Classification, Importance)	1	-	-
1.7	Technologies and Tools in Big Data (NoSQL, Hadoop, Top Analytical Tools)	2	-	-
	Module – 2			
2.1	Introduction to Hadoop, Why Hadoop, Why not RDBMS, RDBMS vs Hadoop	1	-	-
2.2	History of Hadoop, Hadoop Overview, Use Cases of Hadoop	1	-	-
2.3	HDFS, Processing Data with Hadoop, YARN (Yet Another Resource Negotiator)	2	-	-
2.4	Introduction to MapReduce, Mapper, Reducer	1	-	-
2.5	Combiner, Partitioner	1	-	-
2.6	Searching, Sorting, Compression	2	-	-
	Module – 3			
3.1	What is MongoDB, Why MongoDB	2	-	-
3.2	Terms used in RDBMS and MongoDB	2	-	-
3.3	Data Types in MongoDB	2	-	-
3.4	MongoDB Query Language	2	-	-
	Module – 4			
4.1	Introduction to Hive: What is Hive, Hive Architecture, Hive data types, Hive file formats, Hive Query Language (HQL), RC File implementation, User Defined Function (UDF)	2	-	-
4.2	Introduction to Pig: What is Pig, Anatomy of Pig, Pig on Hadoop, Pig Philosophy, Use case for Pig	2	-	-
4.3	Pig Latin Overview, Data types in Pig, Running Pig, Execution Modes of Pig	1	-	-
4.4	HDFS Commands, Relational Operators, Eval Function, Complex Data Types	2	-	-
4.5	Piggy Bank, User Defined Function, Pig Vs Hive	1	-	-
	Module – 5			
5.1	Spark and Big Data Analytics: Spark, Introduction to Data Analysis with Spark	2	-	-
5.2	Text, Web Content and Link Analytics: Introduction	2	-	-
5.3	Text Mining	1	-	-
5.4	Web Mining, Web Content and Web Usage Analytics	2	-	-
5.5	Page Rank, Structure of Web and Analyzing a Web Graph	1	-	-
	Total No. of Lecture Hours	40		
	Total No. of Tutorial Hours		-	
	Total No. of Self learning Hours			-

* Refer the table below for Tutorial Component

Sl. No.	Topic	Tutorial Activity	Duration (Hours)
1	Hadoop File System Operations	Demonstrate HDFS commands like put, get, rm, mkdir, etc. through screenshots or instructor-led sessions. Discuss a sample Hadoop workflow (e.g., loading and processing log files). Provide exercises with hypothetical file paths and ask students what the output will be.	2
2	Matrix Multiplication using MapReduce	Walk through the logic and flow of a MapReduce program for matrix multiplication. Provide pseudo-code and example input/output. Include exercises for tracing intermediate key-value pairs.	2
3	Weather Data Mining with MapReduce	Explain how weather data is structured and parsed. Walk through a MapReduce solution using sample data. Discuss mapper/reducer logic and results. Include sample logs for analysis.	2
4	Movie Tags with MapReduce	Discuss the MovieLens dataset. Walk through a tag extraction and association MapReduce program. Analyze key-value pair design and partitioning. Include a matching activity for input-output mapping.	2
5	MongoDB Operations	Introduce basic MongoDB commands (count, sort, limit, skip, aggregate) with example queries and results. Provide offline exercises to write queries for a given JSON dataset.	2
6	Pig Latin Scripting	Explain data flow in Pig. Use small datasets to demonstrate sort, group, join, project, and filter. Provide Pig scripts and ask students to trace execution and expected results.	2
7	Hive Operations	Discuss Hive architecture. Walk through DDL (create, drop, alter), DML (insert, select), and query examples. Provide a schema and ask students to write HiveQL for different tasks.	4
8	Word Count in Hadoop and Spark	Compare Hadoop MapReduce and Spark for the word count problem. Use flow diagrams and pseudo-code. Include a fill-in-the-blanks activity for code fragments and data flows.	2
9	CDH and HUE	Provide screenshots and interface walkthroughs of CDH and HUE. Explain how to import datasets, run jobs, and generate visualizations. Include a quiz or worksheet on interpreting HUE dashboards.	2

Textbooks:

1. Seema Acharya and Subhashini Chellappan "**Big data and Analytics**" Wiley India Publishers, 2nd Edition, 2019.
2. Rajkamal and Preeti Saxena, "**Big Data Analytics, Introduction to Hadoop, Spark and Machine Learning**", McGraw Hill Publication, 2019.

Reference Books:

1. Adam Shook and Donald Mine, **MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems**, O'Reilly, 2012.
2. Tom White, **Hadoop: The Definitive Guide**, 4th Edition, O'Reilly Media, 2015.

3. Thomas Erl, Wajid Khattak, and Paul Buhler, **Big Data Fundamentals: Concepts, Drivers & Techniques**, Pearson India Education Service Pvt. Ltd., 1st Edition, 2016.

4. John D. Kelleher, Brian Mac Namee, Aoife D'Arcy, **Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples**, MIT Press, 2nd Edition, 2020.

Online Resources:

1. **MovieLens 20M Dataset – Kaggle**
   https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset
   A large dataset of 20 million movie ratings and metadata, ideal for building and testing recommender systems.
2. **Big Data in 5 Minutes – Simplilearn (YouTube)**
   https://youtu.be/bAyrObl7TYE?si=7xge5WaD83SvZds3
   A quick explainer video introducing big data concepts, types, and applications in under five minutes.
3. **Hadoop Tutorial for Beginners – Great Learning (YouTube)**
   https://www.youtube.com/watch?v=VmO0QgPCbZY
   A beginner-friendly crash course covering the Hadoop ecosystem and its core components.
4. **Apache Pig Tutorial – Edureka (YouTube)**
   https://www.youtube.com/watch?v=GG-VRm6XnNk
   A hands-on tutorial explaining how to write and run Apache Pig scripts for big data processing.
5. **Apache Hive Tutorial for Beginners – Simplilearn (YouTube)**
   https://www.youtube.com/watch?v=JglO2Nv_92A
   An introductory video on Apache Hive, showing how to use it for querying and managing large datasets in Hadoop.

Course Code: BIS702 **Course:** Software Testing and Quality Assurance

Credits: 4 **L:T:P -** 3:0:2

CIE: 100 Marks **SEE:** 100 Marks

SEE Hours: 3 hours **Total Marks:** 100

Prerequisites if any	Basic programming knowledge, understanding of software development life cycle, and logical analytical skills.
Learning objectives	1. Understand the fundamentals of software testing, quality assurance, and related standards to ensure software reliability. 2. Design, execute, and manage effective test plans and cases using manual and automated testing techniques. 3. Apply quality assurance best practices, including documentation, defect management, and review processes in real-world projects.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Analyze and apply fundamental software testing concepts and quality assurance principles in software development.	Apply
CO2	Design and execute test cases using various manual and automated testing techniques effectively.	Apply
CO3	Develop and manage comprehensive test plans, defect reports, and documentation aligned with industry standards.	Apply
CO4	Evaluate software quality using reviews, metrics, and testing tools to support continuous improvement.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1		3				3	1					3			3
CO2		3	3	3								3		2	2
CO3			3	3				1	1			3		2	
CO4		3		1	3	3	1			1	1	3			3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1	**No. of Lecture Hours**	**No. of Tutorial Hours***	**No. of Practical Hours**
1.1	Introduction to Software Quality and Testing	1	-	-
1.2	Defect, Fault, Error, and Failure: Definitions & Relationships	1	-	-
1.3	Quality Assurance (QA) vs Quality Control (QC) vs Testing	1	-	-
1.4	Cost of Quality: Prevention, Appraisal, Failure	2	-	-
1.5	Test Process Models: V-Model, W-Model	1	-	-
1.6	Agile Testing Concepts	1	-	-
1.7	Role of Testing in SDLC and STLC	1	-	-
Book Reference: Chapters 1, 2, and selected sections of 3				
	Module – 2			
2.1	Objectives and Elements of Test Planning	1	-	-
2.2	Test Strategy and Test Estimation Techniques	1	-	-
2.3	Scheduling and Resource Allocation	2	-	-
2.4	Risk-Based Testing and Prioritization	1	-	-
2.5	Roles and Responsibilities in a Test Team	1	-	-
2.6	Defect Lifecycle and Defect Tracking Tools	1	-	-
2.7	Test Metrics: Efficiency, Effectiveness, Productivity	1	-	-
Book Reference: Chapters 5, 9, and Appendix B				
	Module – 3			
3.1	Black-Box Testing: Equivalence Class Partitioning	1	-	-
3.2	Black-Box Testing: Boundary Value Analysis	1	-	-
3.3	Decision Table and Cause-Effect Graphing	1	-	-
3.4	White-Box Testing: Statement and Branch Coverage	1	-	-
3.5	White-Box Testing: Path and Condition Coverage	1	-	-
3.6	State-Based and Use Case–Based Testing	1	-	-
3.7	Error Guessing and Exploratory Testing	2	-	-
Book Reference: Chapters 6 and 7				
	Module – 4			
4.1	Basics of Test Automation and ROI Considerations	1	-	-
4.2	Scope and Feasibility of Automation	1	-	-
4.3	Types of Automation Frameworks: Data, Keyword, Hybrid	1	-	-
4.4	Overview of Popular Tools: Selenium, JUnit/TestNG (Conceptual)	1	-	-
4.5	API Testing and Postman Basics (Conceptual)	2	-	-
4.6	CI/CD Integration in Testing: Jenkins/GitHub Actions	1		
4.7	Introduction to Non-Functional Testing: Performance (JMeter), Security (OWASP)	1		
Book Reference: Chapters 8, 11, and 12 (Conceptual coverage only)				
	Module – 5			
5.1	Overview of Software Quality Models: CMMI, ISO 9001	1	-	-
5.2	IEEE 829 and ISO/IEC/IEEE 29119 Standards	1	-	-
5.3	Review Techniques: Walkthroughs, Inspections, Audits	1	-	-
5.4	Configuration Management and Version Control Concepts	1	-	-
5.5	Test Documentation: Test Plan, Test Case, Defect Report	1	-	-
5.6	Traceability Matrix and Requirements Mapping	1	-	-
5.7	Professional Ethics and QA in Industry Practice	2	-	-
Book Reference: Chapters 4, 10, and 13				
	Total No. of Lecture Hours	40		
		Total No. of Tutorial Hours	-	
			Total No. of Self learning Hours	-

* Refer the table below for Tutorial Component

Practical Component

Sl. No.	Topic	Laboratory Activity	Duration (Hours)
1	Introduction to Software Testing	Hands-on: Analyze real-world software failures and categorize bugs using the Bugzilla bug database	2
2	Test Planning and Estimation	Create a detailed test plan and estimation using spreadsheets for a sample open-source application	2
3	Black-Box Testing Techniques	Design test cases using equivalence partitioning & boundary value analysis for a web form in XAMPP setup	2
4	White-Box Testing Techniques	Write and evaluate test cases with coverage tools like GCOV or JaCoCo for sample code snippets	2
5	Defect Reporting and Tracking	Simulate bug reporting and tracking workflow using MantisBT or Bugzilla	2
6	Test Automation Basics	Use Selenium WebDriver (in Python or Java) to automate login/test form of a sample web app	2
7	Test Metrics and Quality Measurement	Analyze sample test execution data and calculate metrics (e.g., defect density) using spreadsheets	2
8	Review Techniques	Conduct peer review on source code using GitHub pull request workflow and checklist-based reviews	2
9	Software Quality Standards	Prepare a brief quality compliance checklist for ISO 29119 using templates in Markdown	2
10	Final Case Study / Mini Project	Use Selenium and Bugzilla to test an open-source module and document the complete test lifecycle	2

Textbooks:

1. Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing: Principles and Practices*, Pearson Education, 2nd Edition, 2015.
2. Kshirasagar Naik and Priyadarshi Tripathy, *Software Testing and Quality Assurance: Theory and Practice*, Wiley India, 2011.

Reference Books:

1. Ron Patton, *Software Testing*, Pearson Education, 2nd Edition, 2007.
2. Paul C. Jorgensen, *Software Testing: A Craftsman's Approach*, CRC Press, 4th Edition, 2013.
3. Glenford J. Myers, Tom Badgett, and Corey Sandler, *The Art of Software Testing*, Wiley, 3rd Edition, 2011.

Online Resources:

1. **Software Testing Help**
   https://www.softwaretestinghelp.com
   A comprehensive portal covering manual testing, automation tools like Selenium, JUnit, TestNG, and QA interview prep.

2. **Coursera – Software Testing and Automation Specialization**
   https://www.coursera.org/specializations/software-testing-automation
   University-level video tutorials on software testing principles, tools, and real-world applications.

Course Code: BIS713A	**Course:** Deep Learning
Credits: 3	**L:T:P -** 3:0:0
CIE: 100Marks	**SEE:** 100 Marks
SEE Hours: 3 hours	**Total Marks:** 100

Prerequisites if any	- Basic understanding of Machine Learning - Proficiency in Python programming - Linear Algebra, Calculus, and Probability fundamentals
Learning objectives	1. To introduce fundamental concepts of deep learning and neural networks. 2. To develop an understanding of various deep learning architectures and their applications. 3. To gain practical exposure to building and training deep learning models using popular frameworks. 4. To evaluate and interpret model performance using standard metrics and visualization techniques.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand the fundamentals of deep learning and neural networks	Understand
CO2	Design and implement feedforward and convolutional neural networks	Apply
CO3	Analyze and optimize deep learning models using techniques like dropout, batch normalization, and tuning	Analyze
CO4	Apply deep learning techniques to real-world problems such as image recognition, NLP, and generative models	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2											2	
CO2	3	3	3	2									3	
CO3	2	3	3	2	2								3	2
CO4	3	3	3	3	3								3	3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	Overview of Deep Learning vs. Machine Learning	1	-	-
1.2	Biological Neurons and Artificial Neurons	1	-	-
1.3	Perceptron, Activation Functions	1	-	-
1.4	Gradient Descent and Backpropagation	2	-	-
1.5	Neural Network Architectures	1	-	-
1.6	Overfitting and Regularization	1	-	-
1.7	Introduction to TensorFlow and PyTorch	1	-	-
	Module – 2			
2.1	Multilayer Perceptrons (MLPs)	1	-	-
2.2	Forward and Backward Pass in MLPs	1	-	-
2.3	Hyperparameter Tuning and Optimization	2	-	-
2.4	Batch Normalization and Dropout	1	-	-
2.5	Weight Initialization Techniques	1	-	-
2.6	Learning Rate Schedulers and Early Stopping	1	-	-
2.7	Model Evaluation Metrics	1	-	-
	Module – 3			
3.1	Introduction to CNNs	1	-	-
3.2	Convolution and Pooling Layers	1	-	-
3.3	CNN Architectures: LeNet, AlexNet, VGG, ResNet	1	-	-
3.4	Transfer Learning and Fine-Tuning	1	-	-
3.5	Object Detection Basics	1	-	-
3.6	Practical CNN Implementation	1	-	-
3.7	Case Study: Image Classification Task	2	-	-
	Module – 4			
4.1	RNNs, Vanishing Gradient Problem	1	-	-
4.2	LSTM and GRU Architectures	1	-	-
4.3	Applications of RNNs in NLP	1	-	-
4.4	Attention Mechanism and Transformers	1	-	-
4.5	BERT and GPT Overview	2	-	-
4.6	Sequence-to-Sequence Models	1		
4.7	Language Modeling and Text Generation	1		
	Module – 5			
5.1	Introduction to Autoencoders	1	-	-
5.2	Variational Autoencoders (VAEs)	1	-	-
5.3	Generative Adversarial Networks (GANs)	1	-	-
5.4	Applications of GANs	1	-	-
5.5	Explainability in Deep Learning Models	1	-	-
5.6	Ethics and Fairness in Deep Learning	1	-	-
5.7	Industry Case Studies	2	-	-
	Total No. of Lecture Hours	40		
	Total No. of Tutorial Hours		-	
	Total No. of Self learning Hours			-

Textbooks:

1. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
2. Chollet, François. *Deep Learning with Python*. Manning Publications, 2018.

Reference Books:

1. Nielsen, Michael. *Neural Networks and Deep Learning*. Determination Press, 2015.
2. Aggarwal, Charu C. *Neural Networks and Deep Learning*. Springer, 2018.

Online Resources:

1. **DeepLearning.AI – Deep Learning Specialization**
   https://www.coursera.org/specializations/deep-learning
   A foundational Coursera specialization by Andrew Ng covering neural networks, CNNs, sequence models, and more using real-world applications.

2. **Stanford CS231n – Convolutional Neural Networks for Visual Recognition**
   http://cs231n.stanford.edu/
   A renowned Stanford course providing in-depth lectures, assignments, and notes on computer vision and deep learning architectures.

3. **TensorFlow and PyTorch Documentation**
   https://www.tensorflow.org/ and https://pytorch.org/
   Official documentation for two of the most widely used deep learning frameworks, including API references, guides, and tutorials.

4. **Fast.ai – Practical Deep Learning for Coders**
   https://course.fast.ai/
   A hands-on, code-first deep learning course aimed at enabling rapid prototyping and deployment of deep learning models.

Course Code: BIS713B	**Course:** Distributed File Systems
Credits: 3	**L:T:P -** 3:0:0
CIE: 100 Marks	**SEE:** 100 Marks
SEE Hours: 3 hours	**Total. Marks:** 100

Prerequisites if any	Basic knowledge of operating systems and computer networks
Learning objectives	4. To understand the characteristics and models of distributed systems. 5. To explore communication mechanisms among distributed components. 6. To study operating system support and file systems in distributed environments. 7. To learn about global states, synchronization, and coordination in distributed systems. 8. To understand distributed transactions, concurrency, and recovery mechanisms.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain the characteristics of a distributed system along with its design challenges.	Understand
CO2	Illustrate the mechanism of IPC between distributed objects.	Apply
CO3	Describe the distributed file service architecture and the important characteristics of Sun NFS.	Analyze
CO4	Discuss concurrency control algorithms applied in distributed transactions.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	2	1	-	-	-	-	-	-	-	3	-
CO2	3	3	2	2	2	-	-	-	-	-	-	-	3	1
CO3	3	2	3	2	2	-	-	-	-	-	-	-	3	1
CO4	3	3	3	3	2	-	-	-	-	-	-	-	3	2

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1: Characterization of Distributed Systems	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	Introduction	1	-	-
1.2	Examples of Distributed Systems	1	-	-
1.3	Resource Sharing and the Web	1	-	-
1.4	Challenges	2	-	-
1.5	Architectural Models	1	-	-
1.6	Fundamental Models	1	-	-
1.7	Summary and Review	1	-	-
	Module – 2: Inter Process Communication			
2.1	Introduction	1	-	-
2.2	API for Internet Protocols	1	-	-
2.3	External Data Representation and Marshalling	2	-	-
2.4	Client – Server Communication	1	-	-
2.5	Group Communication	1	-	-
2.6	Distributed Objects and RMI: Introduction	1	-	-
2.7	Communication between Distributed Objects, RPC, Events and Notifications	1	-	-
	Module – 3: Operating System Support			
3.1	Introduction	1	-	-
3.2	The OS Layer	1	-	-
3.3	Protection	1	-	-
3.4	Processes and Threads	1	-	-
3.5	Communication and Invocation	1	-	-
3.6	Operating System Architecture	1	-	-
3.7	Distributed File Systems: Introduction, File Service Architecture, Sun NFS	2	-	-
	Module – 4: Time and Global States			
4.1	Introduction	1	-	-
4.2	Clocks, Events and Process Status	1	-	-
4.3	Synchronizing Physical Clocks	1	-	-
4.4	Logical Time and Logical Clocks	1	-	-
4.5	Global States	2	-	-
4.6	Coordination and Agreement: Introduction	1		
4.7	Distributed Mutual Exclusion, Elections	1		
	Module – 5: Distributed Transactions			
5.1	Introduction	1	-	-
5.2	Flat and Nested Distributed Transactions	1	-	-
5.3	Atomic Commit Protocols	1	-	-
5.4	Concurrency Control in Distributed Transactions	1	-	-
5.5	Distributed Deadlocks	1	-	-
5.6	Summary Concepts	1	-	-
5.7	Case Studies and Review	2	-	-
	Total No. of Lecture Hours	40		
		Total No. of Tutorial Hours	-	
			Total No. of Self learning Hours	-

Textbooks:

1. George Coulouris, Jean Dollimore, and Tim Kindberg: ***Distributed Systems – Concepts and Design*, 5th Edition**, Pearson Publications, 2009.

Reference Books:

1. Andrew S. Tanenbaum: ***Distributed Operating Systems***, 3rd Edition, Pearson Publications, 2007.
2. Ajay D. Kshemkalyani and Mukesh Singhal: ***Distributed Computing: Principles, Algorithms and Systems***, Cambridge University Press, 2008.
3. Sunita Mahajan and Seema Shah: ***Distributed Computing***, Oxford University Press, 2015.

Online Resources:

1. **MIT OpenCourseWare – Distributed Systems (6.824)**
   https://ocw.mit.edu
   Offers lecture videos, assignments, and readings by Prof. Frans Kaashoek.
2. **Coursera – Cloud Computing Specialization (University of Illinois)**
   https://www.coursera.org/specializations/cloud-computing
   Contains modules on distributed file systems, coordination, and transactions.

3. **Apache Hadoop – HDFS Architecture Guide**
   https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

4. **Microsoft Azure Architecture Center – Distributed Systems Design**
   https://learn.microsoft.com/en-us/azure/architecture/guide/design-principles/distributed-system

5. **YouTube Channel – Gaurav Sen**
   https://www.youtube.com/c/GauravSen

Course Code:	BIS713C	**Course:**	Embedded Systems
Credits:	3	**L:T:P -**	3:0:0
CIE:	100 Marks	**SEE:**	100 Marks
SEE Hours:	3 hours	**Total. Marks:**	100

Prerequisites if any	None – Designed for students with no prior exposure to Microcontrollers, IoT, or Microprocessors.
Learning objectives	1. To understand the architecture and functioning of embedded systems. 2. To learn how microcontrollers are programmed and interfaced with peripherals. 3. To introduce real-time concepts and design methodologies. 4. To enable the application of embedded concepts in practical scenarios.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand embedded system concepts and architecture.	Understand
CO2	Apply embedded C programming to interface peripherals.	Apply
CO3	Analyze embedded system behavior using real-time OS concepts.	Analyze
CO4	Evaluate system design trade-offs for embedded applications.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	1	-	-	-	-	-	-	-	-	-		3	1
CO2	3	3	2	2	2	-	-	-	2	1	-	1		3	2
CO3	2	2	3	2	2	-	-	-	1	2	-	2		2	3
CO4	2	2	3	3	1	1	-	-	2	2	1	2		2	3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1: Introduction to Embedded Systems	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	Definition, characteristics and application domains	1	-	-
1.2	Embedded system vs general-purpose computing	1	-	-
1.3	Core components and architecture	1	-	-
1.4	Classification: small, medium, large scale	2	-	-
1.5	Embedded development lifecycle	1	-	-
1.6	Design metrics and constraints	1	-	-
1.7	Case study: Home automation system	1	-	-
	Module – 2: Microcontrollers and Interfacing Basics			
2.1	Microcontroller vs Microprocessor	1	-	-
2.2	Architecture of 8051 or ARM Cortex-M	1	-	-
2.3	Memory types and mapping	2	-	-
2.4	I/O ports, GPIO	1	-	-
2.5	LED, switch, LCD interfacing	1	-	-
2.6	ADC/DAC basics	1	-	-
2.7	Timers, counters, and PWM	1	-	-
	Module – 3: Operating System Support in Embedded Systems			
3.1	Need for RTOS	1	-	-
3.2	Concepts of multitasking and task scheduling	1	-	-
3.3	Interrupt handling and context switching	1	-	-
3.4	Semaphores and Mutex	1	-	-
3.5	Priority inversion	1	-	-
3.6	RTOS architecture (FreeRTOS or others)	1	-	-
3.7	Simple program execution in RTOS	2	-	-
	Module – 4: Time and Global States			
4.1	Time management in real-time systems	1	-	-
4.2	Event handling and event queues	1	-	-
4.3	Time-triggered vs event-triggered systems	1	-	-
4.4	Watchdog timers	1	-	-
4.5	Clock synchronization basics	2	-	-
4.6	State machines	1		
4.7	Power saving in embedded systems	1		
	Module – 5: Distributed Transactions and Embedded Communication			
5.1	Basics of serial communication: UART, SPI, I2C	1	-	-
5.2	Interfacing with sensors (temperature, ultrasonic, IR)	1	-	-
5.3	CAN, USB, Bluetooth overview	1	-	-
5.4	Embedded networking concepts	1	-	-
5.5	Remote firmware updates	1	-	-
5.6	Debugging techniques and tools	1	-	-
5.7	Mini project outline and integration	2	-	-
	Total No. of Lecture Hours	40		
		Total No. of Tutorial Hours	-	
			Total No. of Self learning Hours	-

Textbooks:

2. Kamal, Raj. *Embedded Systems: Architecture, Programming and Design*. 3rd ed., McGraw Hill Education, 2017.
3. Mazidi, Muhammad Ali, Janice Gillispie Mazidi, and Rolin D. McKinlay. *The 8051 Microcontroller and Embedded Systems: Using Assembly and C*. 2nd ed., Pearson Education, 2006.

Reference Books:

4. Peckol, James K. *Embedded Systems: A Contemporary Design Tool*. 2nd ed., Wiley, 2019.
5. Marwedel, Peter. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. 2nd ed., Springer, 2011.
6. ARM Limited. *ARM Cortex-M Series Technical Reference Manual*. ARM Holdings, www.arm.com.

Online Resources:

1. "Embedded Systems - Introduction." *GeeksforGeeks*, 15 Sept. 2021, https://www.geeksforgeeks.org/embedded-systems-introduction/

2. "Embedded Systems Basics." *NPTEL*, National Programme on Technology Enhanced Learning, https://nptel.ac.in/courses/108/105/108105102/

3. "Introduction to Embedded Systems." *TutorialsPoint*, https://www.tutorialspoint.com/embedded_systems/index.htm

Course Code: BIS754A **Course:** Introduction to DBMS

Credits: 3 **L:T:P -** 3:0:0

CIE: 100 Marks **SEE:** 100 Marks

SEE Hours: 3 hours **Total. Marks:** 100

Prerequisites if any	• Basic understanding of data and its representation (no programming required) • Logical reasoning and problem-solving ability
Learning objectives	1. Understand the role and components of database management systems. 2. Model real-world data and design basic relational databases. 3. Use SQL to create, query, and manipulate data effectively.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand the basic concepts, architecture, and components of a Database Management System.	Understand
CO2	Apply Entity-Relationship modeling to design a database for a real-world engineering application.	Apply
CO3	Apply SQL commands to create, retrieve, and manipulate data in a relational database.	Analyze
CO4	Analyze the normalization process to improve the structure and efficiency of a database.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	-	-	-	-	-	-	-	-	-	-	-		2	-
CO2	3	2	2	-	2	-	-	-	-	-	-	-		3	1
CO3	3	2	2	-	2	-	-	-	-	-	-	-		3	1
CO4	3	2	2	2	-	-	-	-	-	-	-	-		2	-

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1:Introduction to Databases and DBMS	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	Data, Information, and Database Concepts	1	-	-
1.2	Characteristics of Database Systems	1	-	-
1.3	File-based Systems vs DBMS	1	-	-
1.4	DBMS Architecture: Components and Functions	2	-	-
1.5	Data Models: Hierarchical, Network, Relational	1	-	-
1.6	Introduction to Relational Databases and Terminology	1	-	-
1.7	Applications of Databases in Engineering Domains	1	-	-
	Module – 2:Data Modeling and ER Diagrams			
2.1	Introduction to Data Modelling	1	-	-
2.2	Entity, Attributes, and Entity Sets	1	-	-
2.3	Relationships and Relationship Sets	2	-	-
2.4	Keys and Constraints	1	-	-
2.5	ER Diagrams: Symbols and Notations	1	-	-
2.6	Enhanced ER Models (Specialization, Generalization) – Intro only	1	-	-
2.7	Mapping ER Models to Relational Schemas	1	-	-
	Module – 3:Relational Model and Structured Query Language (SQL)			
3.1	Relational Model Concepts (Relations, Tuples, Attributes)	2	-	-
3.2	Relational Algebra – Basic Operations	1	-	-
3.3	Introduction to SQL: Data Types, DDL (CREATE, ALTER, DROP)	1	-	-
3.4	DML Operations (INSERT, UPDATE, DELETE)	1	-	-
3.5	Basic SQL Queries (SELECT, WHERE, ORDER BY)	2	-	-
3.6	Joins and Aggregate Functions	1	-	-
	Module – 4:Database Design and Normalization			
4.1	Characteristics of a Good Database Design	1	-	-
4.2	Functional Dependencies	1	-	-
4.3	Introduction to Normalization: 1NF, 2NF, 3NF	2	-	-
4.4	Anomalies in Unnormalized Databases	2	-	-
4.5	Design Example: Normalizing a Sample Database	2	-	-
	Module – 5:Transactions, Storage, and Database Applications			
5.1	Introduction to Transactions and ACID Properties	2	-	-
5.2	Concurrency Control (basic idea only)	1	-	-
5.3	Database Storage Concepts and File Organization (overview only)	1	-	-
5.4	Indexing and Query Optimization (basic concepts)	1	-	-
5.5	Backup, Recovery, and Security Basics	1	-	-
5.6	Real-World Database Applications in Core Engineering Disciplines.	2	-	-
	Total No. of Lecture Hours	40		
		Total No. of Tutorial Hours	-	
			Total No. of Self learning Hours	-

Textbooks:

1. "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan – 7th Edition, McGraw-Hill Education.

Reference Books:

1. "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe – 7th Edition, Pearson Education.

Online Resources:

1. **NPTEL – Database Management Systems** by Prof. P. P. Chakrabarti, IIT Kharagpur
   https://nptel.ac.in/courses/106105175

2. **w3schools SQL Tutorial**
   https://www.w3schools.com/sql\

Course Code: BIS754B **Course:** Introduction to Algorithms

Credits: 3 **L:T:P -** 3:0:0

CIE: 100 Marks **SEE:** 100 Marks

SEE Hours: 3 hours **Total. Marks:** 100

Prerequisites if any	Programming language C, C++ or Python
Learning objectives	1. Understand and apply the principles of algorithm analysis, including asymptotic notations (Big-O, Big-Ω, and Big-Θ) to evaluate algorithm efficiency. 2. Employ algorithm design paradigms such as brute force, divide and conquer, greedy strategies, and dynamic programming to solve real-world computational problems. 3. Assess the performance, correctness, and suitability of algorithms and data structures for specific problem contexts, including analysis of both recursive and iterative forms.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain the principles of algorithm analysis and the use of asymptotic notations to describe algorithm efficiency.	Understand
CO2	Apply various algorithm design techniques such as brute force, divide and conquer, greedy, dynamic programming, and others to solve computational problems.	Apply
CO3	Analyze the performance and correctness of algorithms, including both recursive and non-recursive forms.	Analyze
CO4	Analyze the suitability of different data structures and algorithmic strategies for solving specific computational problems.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2	3	-	-	-	-	2	-	-	-		3	3
CO2	3	3	3	2	-	2	-	-	2	-	-	2		3	3
CO3	3	3	3	2	-	-	-	-	2	-	-	-		3	3
CO4	3	3	3	3	-	2	-	-	2	-	-	2		3	3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1: Introduction to Algorithm	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
1.1	What is an Algorithm? It's Properties. Fundamentals of Algorithmic Problem solving	1	-	-
1.2	Analysis Framework-Time efficiency and space efficiency, Worst-case, Best-case and Average case efficiency	1	-	-
1.3	Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (θ) with examples	1	-	-
1.4	Basic efficiency classes, Mathematical analysis of Non-Recursive and Recursive Algorithms with Examples	2	-	-
1.5	Brute Force Method: Selection Sort and Bubble Sort	1	-	-
1.6	Sequential Search and Brute-Force String Matching	1	-	-
1.7	Depth-First Search and Breadth-First Search	1	-	-
	Module – 2: Divide and Conquer, Decrease and Conquer Approaches			
2.1	Divide and Conquer: General method, Recurrence equation for divide and conquer, solving it using Master's theorem.	2	-	-
2.2	Divide and Conquer algorithms and complexity Analysis of Binary search, Merge sort, Quick sort.	3	-	-
2.3	Decrease and Conquer Approach: Introduction, Insertion sort&efficiency analysis.	1	-	-
2.4	Graph searching algorithms, Topological Sorting& efficiency analysis.	2	-	-
	Module – 3: Greedy and Transform & Conquer Approaches			
4.1	Greedy Method: General method, Coin Change Problem, Knapsack Problem	1	-	-
4.2	Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm with performance analysis.	2	-	-
4.3	Single source shortest paths: Dijkstra's Algorithm.	1	-	-
4.4	Optimal Tree problem: Huffman Trees and Codes.	1	-	-
4.5	Transform and Conquer Approach: Introduction, Heaps and Heap Sort.	1	-	-
4.6	Balanced Search Trees- AVL trees, 2-3 trees	2	-	-
	Module – 4: Dynamic Programming and Space-Time Tradeoff Approaches			
4.1	Dynamic Programming: General method, Three basic example.	2	-	-
4.2	Transitive Closure: Warshall's Algorithm.	1	-	-
4.3	All Pairs Shortest Paths: Floyd's Algorithm	1	-	-
4.4	The Knapsack Problem and Memory Functions	1	-	-
4.5	Space-Time Tradeoffs: Introduction, Sorting by Counting,	2	-	-
4.6	Input Enhancement in String Matching-Harspool's algorithm.	1	-	-

Continued…

	Module – 5: Backtracking and Branch & Bound Approaches			
5.1	Backtracking: General method, n-Queens problem	1	-	-
5.2	Sum of subsets problem, Hamiltonian Circuit Problems.	2	-	-
5.3	Branch and Bound: Assignment Problem,	1	-	-
5.4	Travelling Salesperson problem, 0/1 Knapsack problem	2	-	-
5.5	NP-Complete and NP-Hard problems: Basic concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes.	2	-	-
Total No. of Lecture Hours		40		
Total No. of Tutorial Hours			-	
Total No. of Self learning Hours				-

Textbooks:

1. Levitin, Anany. ***Introduction to the Design and Analysis of Algorithms***. 2nd ed., Pearson, 2009.
2. Horowitz, Ellis, Sartaj Sahni, and Sanguthevar Rajasekaran. ***Computer Algorithms in C++.*** 2nd ed., Universities Press, 2014.

Reference Books:

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. ***Introduction to Algorithms***. 3rd ed., PHI Learning, 2009.
2. Sridhar, S. ***Design and Analysis of Algorithms***. Oxford University Press, Higher Education, 2014.

Online Resources:

1. **VTU e-Learning – Design and Analysis of Algorithms (06CS43)**
   http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS43.html
   Video lectures by VTU faculty aligned with the 06CS43 syllabus on algorithm design and analysis.

2. **NPTEL – Design and Analysis of Algorithms by Prof. Naveen Garg, IIT Delhi**
   https://nptel.ac.in/courses/106/101/106101060/
   Comprehensive NPTEL course covering algorithmic techniques and complexity analysis.

3. **VTU FEP – Algorithms and Data Structures**
   http://elearning.vtu.ac.in/econtent/courses/video/FEP/ADA.html
   Foundation e-Program (FEP) videos on Algorithms and Data Structures by VTU.

4. **Virtual Labs – Algorithm Virtual Lab, IIIT Hyderabad**
   http://cse01-iiith.vlabs.ac.in/
   Interactive simulations and experiments on algorithm performance and behaviour.

5. **Stanford Online – Introduction to Algorithms**
   http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
   Stanford University's open online course providing foundational knowledge in algorithms.

Course Code: BIS754C **Course:** Software Engineering

Credits: 3 **L:T:P -** 3:0:0

CIE: 100 Marks **SEE:** 100 Marks

SEE Hours: 3 hours **Total. Marks:** 100

Prerequisites if any	✓ Basic knowledge of programming and problem-solving.
Learning objectives	1. Provide a foundation in software engineering principles and practices. 2. Introduce software development life cycle models, requirement analysis, design, implementation, testing, and maintenance. 3. Emphasize the importance of quality software processes and documentation.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand fundamental concepts and process models of software engineering.	Understand
CO2	Apply software engineering principles in real-world scenarios.	Apply
CO3	Analyze and model requirements and design solutions.	Analyze
CO4	Evaluate software quality assurance strategies and tools.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	1	-	-	-	2	-	1	2	1
CO2	2	2	2	-	1	-	-	-	1	1	-	1	3	1
CO3	3	2	3	2	-	-	-	-	-	1	-	1	3	-
CO4	2	1	2	3	1	1	-	-	-	2	-	2	2	3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1: Introduction to Software Engineering	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	Definition and Evolution of Software Engineering	1	-	-
1.2	Software Characteristics and Applications	1	-	-
1.3	Software Process Models: Waterfall, Prototyping, Spiral, Agile	1	-	-
1.4	Overview of Software Project Management	2	-	-
1.5	Software Metrics and Estimation	1	-	-
1.6	Software Engineering Challenges	1	-	-
1.7	Overview of Software Quality	1	-	-
Module – 2: Requirements Engineering				
2.1	Requirements Engineering Process	1	-	-
2.2	Types of Requirements: Functional and Non-functional	1	-	-
2.3	Requirement Elicitation Techniques	2	-	-
2.4	Requirement Analysis and Modeling	1	-	-
2.5	Software Requirement Specification (SRS)	1	-	-
2.6	Use Case Modeling	1	-	-
2.7	Requirements Validation and Management	1	-	-
Module – 3: Software Design				
3.1	Design Concepts and Principles	2	-	-
3.2	Architectural Design and Styles	1	-	-
3.3	Component-level Design	1	-	-
3.4	Data Design and Class Design	1	-	-
3.5	User Interface Design	2	-	-
3.6	Design Documentation	1	-	-
Module – 4: Software Implementation and Testing				
4.1	Coding Standards and Guidelines	1	-	-
4.2	Programming Best Practices	1	-	-
4.3	Introduction to Testing: Types and Levels	2	-	-
4.4	White-box and Black-box Testing Techniques	2	-	-
4.5	Test Case Design and Test Automation	2	-	-
Module – 5: Software Maintenance and Quality Management				
5.1	Types of Software Maintenance	2	-	-
5.2	Software Configuration Management	1	-	-
5.3	Risk Management	1	-	-
5.4	Software Quality Assurance (SQA)	1	-	-
5.5	Software Process Improvement Models (CMMI, ISO)	1	-	-
5.6	Tools for Software Engineering (Version control, CASE tools)	2	-	-
	Total No. of Lecture Hours	40		
		Total No. of Tutorial Hours	-	
			Total No. of Self learning Hours	-

Textbooks:

2. Pressman, Roger S., and Bruce R. Maxim. *Software Engineering: A Practitioner's Approach*. 8th ed., McGraw-Hill, 2015.

Reference Books:

2. Sommerville, Ian. *Software Engineering*. 10th ed., Pearson, 2015.
3. Jalote, Pankaj. *An Integrated Approach to Software Engineering*. Springer, 2013.

Online Resources:

1. National Programme on Technology Enhanced Learning (NPTEL). *Software Engineering*. Government of India, https://nptel.ac.in/courses/106105087

2. IEEE Xplore Digital Library. *IEEE Software Magazine*. IEEE, https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=52. Accessed 12 June 2025.

3. GitHub. *GitHub: Where the World Builds Software*. GitHub, Inc., https://github.com.

4. Stack Overflow. *Stack Overflow: Where Developers Learn, Share, & Build Careers*. Stack Exchange Inc., https://stackoverflow.com

Course Code: BXX785

Credits: 6

CIE: 100 Marks

SEE Hours: 3

Course: Major Project

L:T:P:S - 0:0:12

SEE: 100 Marks

Max. Marks: 100

Prerequisites if any	Mini Project should be completed.
Learning objectives	1. Develop and implement cost-effective design methods. 2. Compute the results obtained from the implementation and validate it. 3. Demonstrate and present the project in a team and prepare the report of the project work.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Identify different areas of interest, feasible for the project team.	L4
CO2	Formulate the problem and perform analysis.	L4
CO3	Develop the cost effective design methods to solve the identified problem	L5
CO4	Compute the results obtained from the implementation	L5
CO5	Validate using various test cases and demonstrate the project work in team and prepare the report of the project work.	L5

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	2	3	3	2	3	1	3	2	2	
CO2	3	3	3						2		2		2	
CO3	3	3	2	2					2		2		2	
CO4									3		2			2
CO5									3		3			2

3 – Strong 2 – Medium 1 – Low

8th Semester

Course Code: BIS803	**Course:** Internship(Industry/ Research)
Credits: 10	**L:T:P:S -** 0:0:20
CIE: 100 Marks	**SEE:** 100 Marks
SEE Hours: 3	**Max. Marks:** 100

Prerequisites if any	NIL
Learning objectives	1. To provide opportunities for experiential learning in varied areas of the discipline beyond ‘teaching-training’ and enhance professional growth of the students. 2. To help students prepare for career in Information science and Engineering and develop a road map for the same.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom’s level**
CO1	Interpret the problem to be analyzed in the internship.	L5
CO2	Apply the Information Technology skillset and management principles to address real-world problems.	L3
CO3	Demonstrate the solution to the problem.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	3	1	3		2	2	1	3	2	3	3			
CO2	3	3	3		3		2	1	3		3	3		3	
CO3	3	2	1		2			1	3	3	3	3			3

3 – Strong 2 – Medium 1 – Low

The National Institute of Engineering, Mysuru

Department of Information Science and Engineering

“Education is the power that shapes personality”

"ಶಿಕ್ಷಣವೇವ್ಯಕ್ತಿತ್ವನಿರ್ಮಾಣದಶಕ್ತಿ."
— ಸರ್ವಪಳ್ಳಿರಾಧಾಕೃಷ್ಣನ್