SCHEME OF INSTRUCTION & EXAMINATION

B.E. (INFORMATION TECHNOLOGY) – VI Semester

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P / D	Contact Hrs	CIE	SEE	Duration in Hrs	
Theory Courses										
1.	SPC 601 IT	Embedded System	3	-	-	3	40	60	3	3
2.	SPC 602 IT	Software Engineering	3	-	-	3	40	60	3	3
3.	SPC 603 IT	Artificial Intelligence & Machine Learning	3	-	-	3	40	60	3	3
4.	SPE – II	Professional Elective – II	3	-	-	3	40	60	3	3
5.	OE – I	Open Elective – I	3	-	-	3	40	60	3	3
Practical / Laboratory Courses										
6.	SPC 611 IT	Embedded System Lab	-	-	3	3	40	60	3	1.5
7.	SPC 612 IT	Artificial Intelligence & Machine Learning Lab	-	-	3	3	40	60	3	1.5
8.	SPW 6113IT	Mini Project (Software Engineering Lab)	-	-	4	4	40	60	3	2
9.	SPW614IT	Technical Seminar	-	-	2	2	50	-	3	1
10.	SPW611IT	Internship-2	The students have to undergo an internship of 4 weeks duration after VI-semester SEE				50	-	3	1
			15		**12**	**27**	**420**	**480**	**30**	**22**

Professional Elective – II		
Sl. No.	**Course Code**	**Course Title**
1.	SPE 621 IT	Information Security
2.	SPE 622 IT	Natural Language Processing.
3.	SPE 623 IT	Information retrieval systems
4.	SPE 624 IT	Ad-hoc and Sensor Networks
5.	SPE 625 IT	Parallel Algorithms

Open Elective – I		
S. No.	**Course Code**	**Course Title**
1.	SOE 611 IT	OPERATING SYSTEMS
2.	SOE 612 MBA	ENTREPRENEURSHIP
3.	SOE 613 ECE	SIGNALS AND SYSTEMS
4.	SOE 614ECE	SIGNAL ANALYSIS AND TRANSFORMATION TECHNIQUES
5.	SOE 615 CSE	OPEN SOURCE TECHNOLOGIES
6.	SOE 616 CSE	OOP USING JAVA

PC: Professional Core
SPE: Professional Elective
MC: Mandatory Course
OE: Open Elective
SEE: Semester End Evaluation (Univ. Exam)
CIE: Continuous Internal Evaluation
SI: Summer Internship
L: Lecture T: Tutorial P: Practical D: Drawing

Note:
***** The students have to undergo a Summer Internship of four weeks duration after VI semester and credits will be awarded in VII semester after evaluation.

Course Code	Course Title						Core / Elective
SPC601 IT	Embedded System						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Microprocessor	3	-	–	–	40	60	3

Course Objectives:

1. To provide knowledge on the basics, building blocks of Embedded System.
2. To teach automation using scheduling algorithms and Real time operating system.
3. To involve Discussions/ Practice/Exercise onto revising & familiarizing the concepts.

Course Outcomes:

At the end of this course, the students will be able to

1. Gain adequate understanding of the software architecture of the Embedded OS.
2. Develop simple applications for Process Management, Synchronization Techniques, Message Passing, and POSIX based application development.
3. Describe the Linux Kernel environment; build system, kernel configuration, customization and compilation.
4. Set up a Linux environment with basic understanding of kernel programming concepts like Module. Programming and Device Drivers.
5. Understand cross tooling environments and be exposed to development of device drivers for a target hardware platform.

UNIT- I

Introduction to Embedded System: Definition, Examples and components of embedded Systems, Classification of an Embedded system. Architecture of Embedded system. General purpose computers vs embedded system, Embedded System Design Process, Core of Embedded System, Memory, Sensors and Actuators, Communication Interface- Onboard communication interface, External communication interface.

UNIT-II

Hardware/Software Co-design for Embedded Systems: Microcontrollers for embedded systems, 32-bit RISC Architectures for embedded Design, ARM architectural details, The ARM programmer's model, ARM development tools, ARM microcontroller programming in C, Peripheral Interfacing with ARM, Basic Wire and Wireless Protocols like, UART, I2C, SPI, PLCC, Bluetooth, Wi-Fi, Zig-Bee and Lora for IoT applications.

UNIT- III

Embedded Operating Systems: Embedded OS overview, Study of Embedded OS principles and
requirements. Internal components of Embedded operating systems - Compare and contrast various Embedded OS platforms.

UNIT -IV

Introduction to Device Drivers: Unix/Linux kernel fundamentals-Process Scheduling - Kernel

Synchronization, I/O devices - Architecture - Character, Block Device handling, file systems - The Ext2 file System - The Virtual File System and peripheral devices, Linux file system.

UNIT-V

Device Driver Internals: Fundamentals of device drivers-Character and Block Devices - Polling and Interrupts - The Hardware, device enumeration and configuration, Data transfer and management
mechanisms.

Text Books:

1. Raj Kamal, Embedded Systems Architecture, Programming and Design, TATA McGraw-Hill, First reprint Oct. 2003
2. Christopher Hallinan, “Embedded Linux Primer: A practical Real-World approach", Prentice Hall, 2nd Edition, 2011.
3. Daniel P. Bovet and Marco Cesati, "Understanding the Linux Kernel, O'Reilly, 3rd Edition, 2005.
4. John Madieu, "Linux Device Drivers Development: Develop customized drivers for embedded Linux", Packt Publishing, 1st Edition, 2017.
5. Jonathan Corbet, Alessandro Rubini, Greg Kroah, "Linux Device Drivers", O'Reilly, 3rd Edition, 2005.

References:

1. Steve Heath, Embedded Systems Design, Second Edition-2003, Newness,
2. David Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.
3. Wayne Wolf, Computers as Components; Principles of Embedded Computing System Design – Harcourt India, Morgan Kaufman Publishers, First Indian Reprint 2001
4. Frank Vahid and Tony Garages, Embedded Systems Design – A unified Hardware /Software Introduction, John Wiley, 2002.

Course Code	Course Title						Core / Elective
SPC602 IT	Software Engineering						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
–	3	-	–	–	40	60	3

Course Objectives:

1. To introduce the basic concepts of software development processes from defining a product to shipping and maintaining
2. To impart knowledge on various phases, methodologies and practices of software development
3. To understand the importance of testing in software development, study various testing strategies along with its relationship with software quality and metrics

Course Outcomes:

At the end of this course, the students will be able to

1. Acquired working knowledge of alternative approaches and techniques for each phase of software development
2. Judge an appropriate process model(s) assessing software project attributes and analyze necessary requirements for project development eventually composing SRS
3. Creation of visual models to describe (non-) algorithmic solutions for projects using various design principles.
4. Acquire skills necessary as an independent or as part of a team for architecting a complete software project by identifying solutions for recurring problems exerting knowledge on patterns.
5. Concede product quality through testing techniques employing appropriate metrics by understanding the practical challenges associated with the development of a significant software system.

UNIT-I

A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Models, Personal and Team Process Models, Process Technology, Product and Process.

An Agile view of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II

Software Engineering Principles: SE Principles, Communication Principles, Planning Principles, Modeling Principles, Construction Principles, Deployment.

System Engineering: Computer-based Systems, The System Engineering Hierarchy, Business Process Engineering, Product Engineering, System Modeling.

Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use-Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT-III

Building the Analysis Model: Requirements Analysis Modeling Approaches, Data Modeling Concepts, Object-Oriented Analysis, Scenario-based Modeling, Flow-oriented Modeling, Class-based Modeling, Creating a Behavioral Model.

Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design.

UNIT-IV

Creating an Architectural Design: Software Architecture, DataDesign, Architectural Styles and Patterns, Architectural Design.

Modeling Component-Level Design: Definition of Component, Designing Class-based Components, Conducting Component-level Design, Object Constraint Language, Designing Conventional Components.

Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

UNIT-V

Testing: Strategies: A Strategic Approach to Conventional Software Testing, Test Strategies for O-O Software. Tactics: Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing, O-O Testing Methods.

Debugging: Debugging Techniques, The Art of Debugging.

Product Metrics: A Framework for Product Metrics, Metrics for each phase of software development.

Software Quality: Definition, Quality Assurance: Basic Elements, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO9000 Quality Standards, SQA plan.

Textbooks:

1. Roger S. Pressman, Software Engineering: A Practitioners Approach, Seventh Edition, McGraw-Hill, 2009.

2. AliBehforoz and Frederic Johansson, Software Engineering Fundamentals, Oxford University Press, 1996.
3. PankajJalote—An Integrated Approach to Software Engineering, Third Edition, Nervosa publishing house, 2008.

Reference Books:

1. Software Engineering, a Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering: A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3. Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.

Course Code	Course Title						Core / Elective
SPC602 IT	Artificial Intelligence & Machine Learning						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Programming language, Data Structures	3	-	–	–	40	60	3

Course Objectives:

1. To learn the concepts of Artificial Intelligence , machine learning and types of learning along with evaluation metrics.
2. To study various supervised learning algorithms and learn ensemble techniques and various unsupervised learning algorithms.
3. To explore Neural Networks and to learn reinforcement learning and study applications of machine learning.

Course Outcomes:

At the end of this course, the students will be able to

1. Describe the concepts and applications of artificial intelligence and Machine Learning.
2. Understand and Compute the performance metrics for regression and classification problems.
3. Extract features that can be used for a particular machine learning approach in various applications.
4. Apply ensemble techniques for improvement of classifiers.
5. Understand reinforcement learning and Apply classification, clustering and reinforcement learning to various applications.

UNIT-I

Overview of Artificial Intelligence: Introduction. The Turing Test, Strong AI Versus Weak, AI, Identifying Problems Suitable for AI, Applications and Methods, Early History of AI, Recent History of AI to the Present, AI in the New Millennium

Introduction: Learning, Concept of learning system, Machine Learning, Types of Machine Learning, data and data set, attribute types, Training, Testing, and Validation Sets, The Confusion Matrix, Accuracy Metrics, The Receiver Operator Characteristic (ROC) Curve, Unbalanced Datasets, Some basic statistics: Averages, Variance and Covariance, The Gaussian, the bias-variance trade-off.

UNIT-II

Regression: Linear Regression, Logistic Regression.
Supervised Learning: Classification Learning with Trees: Using Decision Trees, Constructing Decision Trees, CART, Naïve Bayes, K-Nearest Neighbors, Support Vector Machines(SVM)

UNIT-III

Unsupervised Learning(clustering): Introduction, Similarity and Distance Measures, Outliers, Partitional Algorithms, Hierarchical Methods, Clustering Large Databases, Clustering with Categorical Attributes, Comparison

Dimensionality Reduction: Curse of Dimensionality, Linear Discriminant Analysis, Principal Component Analysis

UNIT-IV

Neural Networks: Introduction Artificial neural Network, perceptron, Multilayer Perceptron, Back propagation Algorithm, Activation Functions, Gradient Descent for Machine Learning.

Ensemble Algorithms: Bagging, Boosting. Random forest,

UNIT-V

Reinforcement learning: Overview, State and Action Spaces, The Reward Function, Discounting, Action Selection, Policy, Markov decision processes, Q-learning, Uses of Reinforcement learning. Applications of Machine Learning.

Textbooks

1. Stuart Russell, Peter Norvig: Artificial Intelligence, A Modern Approach, Pearson Education, Global Edition. 2021
2. Stephen Marsland, Machine Learning: An Algorithmic Perspective, Second Edition, Chapman & Hall/CRC, 2014.
3. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

Reference Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, Springer Series in Statistics.(2009).
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 1st edition, 2016.
3. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.

Course Code	Course Title						Core / Elective
SPC611 IT	Embedded System Lab						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
C Programming Language	-	-	–	3	40	60	1.5

Course Objectives:

1. To understand basic concepts and structure of embedded systems.
2. To design and develop real time applications of embedded systems

Course Outcomes:

At the end of this course, the students will be able to

1. Apply the basic concepts to develop an Interface for 8051 and ARM processors.
2. How to interface input and output units.
3. Develop control applications
4. Demonstrate the RTOS Concepts by designing real time applications.
5. Demonstrate multi-tasking, scheduling, priority inversion and Interrupt service routines in RTOS.

List of Experiments:

1. **Interface Input-Output and other units such as:** Relays, LEDs, LCDs, Switches, Keypads, Stepper Motors, Sensors, ADCs, Timers
2. **Demonstrate Communications**: RS232, IIC and CAN protocols
3. **Develop Control Applications such as**: Temperature Controller, Elevator Controller, Traffic Controller
4. **Understanding Real Time Concepts using any RTOS through Demonstration of:**
   i. Task management and Software timers.
   ii. Real-time message queues, semaphores, and mutexs.
   iii. Process management and Thread management.
   iv. Scheduling policies and preemptions.
   v. Embedded Linux Development environment set-up.
   vi. Linux Kernel configuration.
   vii. Building Embedded Linux Device Trees.

viii. Linux Kernel Modules and Device model.

ix. Sysfs, Char device / drivers.

x. Platform device/driver.

Course Code	Course Title						Core / Elective
SPC612 IT	AI & Machine Learning Lab						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Python Programming	-	-	–	3	40	60	1.5

Course Objectives:

The main objectives of this course are:

1. Demonstration of different classifiers on different data.
2. Demonstrate unsupervised learning algorithms and dimensionality reduction techniques.
3. Make use of real-world data to implement machine learning models

Course Outcomes:

At the end of this course, the students will be able to

1. Understanding the exploratory data analysis and data visualization
2. Understand the implementation procedures for the machine learning algorithms.
3. Design Python programs for various Learning algorithms.
4. Apply appropriate data sets to the Machine Learning algorithms.
5. Identify and apply Machine Learning algorithms to solve real world problems.

List of Experiments:

1. Write a python program to import and export data using Pandas library functions
2. Demonstrate various data pre-processing (EDA) techniques for a given dataset
3. Write a Python program to demonstrate various Data Visualization Techniques
4. Develop Decision Tree Classification model for a given dataset and use it to classify a new sample
5. Implement Simple and Multiple Linear Regression Models
6. Develop Logistic Regression Model for a given dataset
7. Implement Naïve Bayes theorem to classify the text data
8. Implement k-nearest neighbours classification using python
9. Write a python program to implement K-Means clustering Algorithm
10. Implement Dimensionality reduction using Principle Component Analysis (PCA) method

11. Build Artificial Neural Network model with back propagation on a given dataset

12. Implement Random forest ensemble method on a given dataset.

13. Implement Boosting ensemble method on a given dataset

Course Code	Course Title						Core / Elective
SPW6113 IT	Mini Project (Software Engineering Lab)						Core
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Programming for Problem Solving	-	-	–	3	40	60	2

Course Objectives:

1. To have hands on experience in developing a software project by using various software engineering.
2. Principles and methods in each of the phases of software development.
3. Translate end-user requirements into system and software requirements

Course Outcomes:

At the end of this course, the students will be able to

1. Ability to generate a high-level design of the system from the software requirements
2. Gain experience and/or awareness of testing problems and will be able to develop a simple testing report
3. Understand the software engineering methodologies involved in the phases for Project development.
4. Gain knowledge about open-source tools used for implementing software engineering methods.
5. To develop product-prototypes implementing software engineering methods.

I. FORWARD ENGINEERING

Students have to form a team with a batch size of two or three and take up a case study based project to analyze, plan, design UML models and create a prototypical model (identifying deliverables) by coding the developed design s and finally documenting considering any one example of the following domains:

I. Academics (Course registration System , Student marks analyzing system)

II. Health Care (Expert system to prescribe medicines forgiven symptoms, Remote Diagnostics, Patient/Hospital Management System)

III. Finance (Banking: ATM/ Net Banking, UPI: Pay TM / Phone Pay)

IV. E-Commerce(any Online shopping portal) Logistics(Postal/Courier: India Post /DTDC /UPS)

V. Hospitality (Tourism Management: Telangana Tourism/Incredible India, Event Management)

VI. Social Networking(LinkedIn, Face Book, Shaadi.com, Bharat Matrimony, Tinder)

VII. Customer Support(Banking Ombudsman ,Indian Consumer Complaints Forum)

VIII. Booking/Ticketing(Online Food Ordering ,Travel: {Cars: Uber/OLA/Zoom, Railways: IRCTC, Buses: Online TSRTC/Red Bus /Abhi Bus, Flights: Make MyTrip)

II. **REVERSE ENGINEERING**: Students have to refer any project repository: GitLab/ GitHub, execute the code in order to observe its functionalities/features/requirements and by the help of any tool derive the designs from the code for understanding the relationships among various subsystems/ classes/ components and If the tool partially generates models then identify by associating elements to judge/make the appropriate relationships.

III. **TESTING: Prepare Test Plan and develop Test Case**

Hierarchy monitor run cover/report errors using manual/automated testing tools Software Required: Star UML/ Umbrello Net Beans /Eclipse IDE , XAMP/MEAN stack, JUnit, JMeter, Selenium ,Bugzilla.

IV. **GUIDELINES FOR MINI PROJECT**

The students are required to carry out mini projects in any of the areas such as Data Structures, Micro processors and Interfacing, Database Management Systems, Operating Systems, Design and Analysis of Algorithms, Software Engineering, Data Communications, Web Programming & Services, Computer Networks, Compiler Construction, Object Oriented System Development.

Problems Statements are suggested to be taken can also be taken from Smart India Hackathon (SIH) Portal invited from the Ministries /PSUs /MNCs/ NGOs to be worked out through.

The project could be classified as hard ware, software, modeling, simulation etc. The project should involve one or many elements of techniques such as analysis, design, and synthesis.

The department will appoint a project Coordinator who will coordinate the following:

I. Grouping of students (maximum of 3 students in a group)

II. Allotment of projects and project guides.

III. All projects allotment is to be completed by the 4th week of the semester so that the students get sufficient time for completion of the project.

IV. Disseminate guidelines given by monitoring committee comprising of senior faculty members to the students and their guides.

V. Three periods of contact load will also be assigned to each project guide for project guidance and monitoring at regular intervals.

VI. Sessional marks are to be awarded by the monitoring committee.

VII. Common norms will be established for the final presentation and documentation of the project report by the respective departments.

VIII. Students are required to submit a presentation and report on the mini project at the end of the semester.

Text Books:

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 7thEdition, McGraw Hill, 2009.
2. Ali Beh forooz and Frederick J.Hudson, Software Engineering Fundamentals, Oxford University Press, 1996.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2008.

References Book:

1. Summerville, Ian Software Engineering, Addison-Wesley , Boston, MA. (2011).
2. Stephens, Rod (2015) Beginning Software Engineering, Wrox**.**

Course Code	Course Title						Core / Elective
SPE 621 IT	Information Security						Elective
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Computer Network	3	-	–	–	40	60	3

Course Objectives:

1. Familiarize students with basics of Network Security and cryptography.
2. To understand the network security, services, attacks, mechanisms, types of attacks on TCP/IP protocol suite and comprehend and apply network layer security protocols, Transport layer security protocols, Web security protocols.
3. To comprehend and apply authentication services, authentication algorithms and Provide a solid understanding of main issues related to network security

Course Outcomes:

At the end of this course, the students will be able to

1. Understand the most common type of information and network threat sources.
2. Understand basic Symmetric and Asymmetric cryptographic algorithms.
3. Apply the various Authentication schemes to simulate different applications.
4. Understand various email security services and IP security protocol
5. Understand basic web security protocol and firewalls

UNIT-I

Attacks on Computers and Computer Security: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security

Cryptography: Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

UNIT-II

Symmetric key Ciphers: Block Cipher principles & Algorithms (DES, AES,Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RC4,Location and placement of encryption function, Key distribution

Asymmetric key Ciphers: Principles of public key cryptosystems, Algorithms(RSA, Diffie-Hellman,ECC), Key Distribution

UNIT-III

Message Authentication Algorithms and Hash Functions: Authentication requirements, Functions, Message authentication codes, Hash Functions, Secure hash algorithm, Whirlpool, HMAC, CMAC, Digital signatures, knapsack algorithm.

UNIT-IV

E-Mail Security: Pretty Good Privacy, S/MIME IP Security: IP Security overview, IP Security architecture, Authentication Header, encapsulating security payload, combining security associations, key management

UNIT-V

Web Security: Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction Intruders, Virus and Firewalls: Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls Case Studies on Cryptography and security: Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability, Virtual Elections.

Text books:

1. Cryptography and Network Security: William Stallings, Pearson Education, 5 th Edition
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 2 nd Edition.
3. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

Reference Books:

1. Security in Computing, Fourth Edition, by Charles P. Pfleeger, Pearson Education
2. Cryptography and Network Security Principles and Practice, Fourth or Fifth Edition, William Stallings, Pearson
3. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall.
4. Network Security Essentials: Applications and Standards, by William Stallings.

Course Code	Course Title						Core / Elective
SPE623IT	Natural language processing						Elective
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Theory Of automata	-	-	–	–	40	60	3

Course Objectives:

1. To learn the fundamentals of natural language processing
2. To understand the role of syntax, semantic, pragmatics in NLP
3. To apply the NLP techniques to Text processing application.

Course Outcomes:

At the end of this course, the students will be able to

1. Tag a given text with basic Language features
2. Design an innovative application using NLP components
3. Implement a rule-based system to tackle morphology/syntax of a language.
4. Design a tag set to be used for statistical processing for real-time applications.
5. Compare and contrast the use of different statistical approaches for different types of NLP applications.

UNIT- I

Definition, History of NLP, corpus, type of corpus, Ambiguity in Language, NLP phases , lexical resources, WordNet, Regular Expressions, Text Processing with Unicode, Regular Expressions for Detecting Word Patterns, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation, word count, type and token ratio, zips law

UNIT -II

Text Representation: one-hot encoding model, Bag-of-word model, count vectors and TF-IDF vectors, N-gram language model, Word2VEC Embedding, Glove Embedding, Fast Text Embedding Morphology, types of morphology, lionization, stemming, Transducers for lexicon and rules,

Word Classes, tag and tag set, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in Pops tagging – Hidden Markov and Maximum Entropy models.

UNIT -III

Context-Free Grammar, Parsing with Context-Free Grammar, Dependencies and Dependency Grammar, Grammar Development, Shallow parsing – Probabilistic CFG

UNIT -IV

Semantics: Representing meaning – Semantic analysis, Lexical semantics, Word Senses, Relations between Senses, Thematic Roles, Word sense disambiguation (WSD) , WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods, Word Similarity using Thesaurus and Distributional methods.

UNIT -V

Pragmatics: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution, Coreference Resolution, Dialog and Conversational agents, Natural language generation, Statistical alignment and Machine translation: Text alignment, word alignment, statistical machine translation,

Textbooks:

1. Daniel Jurafsky, James H. Martin—Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, —Natural Language Processing with Python‖, First Edition, O_Reilly Media, 2009.

References:

1. Breck Baldwin, —Language processing with Java and Ling Pipe Cookbook, Atlantic Publisher, 2015.
2. Richard M Reese, —Natural Language Processing with Java, Oreille Media, 2015.
3. Nitin Indurkhya and Fred J. Damerau, —Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.
4. Tanveer Siddiqui, U.S. Tiwary, —Natural Language Processing and Information Retrieval‖, Oxford University Press, 2008.
5. Daniel and Martin J. H., “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, Prentice Hall, 2009.

Course Code	Course Title						Core / Elective
SPE 624IT	Information Retrieval Systems						Elective
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Database Management System	-	-	–	–	40	60	3

Course Objectives:

1. To understand indexing and querying in information retrieval systems
2. To learn the different models for information retrieval
3. To expose the students to text classification and clustering and to learn about web searching

Course Outcomes:

At the end of this course, the students will be able to

1. Apply IR principles to locate relevant information in large collections of data
2. Design different document Indexing and clustering algorithms
3. Understand various classes of Automatic Indexing and term clustering.
4. Design an Information Retrieval System for web search tasks.
5. Understand different text search and multimedia retrieval systems.

UNIT-I

Introduction: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses, Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

UNIT-II

Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction

Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models

UNIT-III

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT-IV

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies

UNIT-V

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

Textbook:

1. Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. kowalski, Mark T. Maybury, Springer

References:

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2. Information Storage & Retrieval by Robert Korfhage – John Wiley & Sons.
3. Modern Information Retrieval by Yates and Neto Pearson Education.

Course Code	Course Title						Core / Elective
SPE625 IT	Ad Hoc & Wireless Sensor Networks						Elective
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
Computer Networks	-3	-	–	–	40	60	3

Course Objectives:

1. Learn Ad hoc network and Sensor Network fundamentals
2. Understand the different routing protocols, sensor network architecture and design issues
3. Understand the transport layer and security issues possible in Ad hoc and Sensor networks

Course Outcomes:

At the end of this course, the students will be able to

1. Know the basics of Ad hoc networks and Wireless Sensor Networks
2. Apply this knowledge to identify the suitable routing algorithm based on the network and user requirement
3. Apply the knowledge to identify appropriate physical and MAC layer protocols
4. Understand the transport layer and security issues possible in Ad hoc and sensor networks
5. Familiar with the OS used in Wireless Sensor Networks and build basic modules.

UNIT-1

Elements of Ad hoc Wireless Networks, Issues in Ad hoc wireless networks, Example commercial applications of Ad hoc networking, Ad hoc wireless Internet, Issues in Designing a Routing Protocol for Ad Hoc Wireless Networks, Classifications of Routing Protocols, Table Driven Routing Protocols – Destination Sequenced Distance Vector (DSDV), On–Demand Routing protocols –Ad hoc On–Demand Distance Vector Routing (AODV).

UNIT-II

Challenges for Wireless Sensor Networks, Enabling Technologies for Wireless Sensor Networks, WSN application examples, Single-Node Architecture – Hardware Components, Energy Consumption of Sensor Nodes, Network Architecture – Sensor Network Scenarios, Transceiver Design Considerations, Optimization Goals and Figures of Merit.

UNIT-III

MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols and Wakeup Concepts – S-MAC, The Mediation Device Protocol, Contention based protocols – PAMAS, Schedule based protocols – LEACH, IEEE 802.15.4 MAC protocol, Routing Protocols Energy Efficient Routing, Challenges and Issues in Transport layer protocol.

UNIT-IV

Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Layer wise attacks in wireless sensor networks, possible solutions for jamming, tampering, black hole attack, flooding attack. Key Distribution and Management, Secure Routing – SPINS, reliability requirements in sensor networks.

UNIT-V

Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms – TinyOS, nesC, CONTIKIOS, Node-level Simulators – NS2 and its extension to sensor networks, COOJA, TOSSIM, Programming beyond individual nodes – State centric programming.

Textbooks:

1. C.Siva Ram Murthy and B.S.Manoj, ―Ad Hoc Wireless Networks – Architectures and 2 Protocols‖, Pearson Education, 2006.
2. Holger Karl, Andreas Willing, ―Protocols and Architectures for Wireless Sensor Networks‖, John Wiley & Sons, Inc., 2005.

References:

1. Subir Kumar Sarkar, T G Basavaraju, C Puttamadappa, ―Ad Hoc Mobile Wireless Networks, Auerbach Publications, 2008.
2. Carlos De Morais Cordeiro, Dharma Prakash Agrawal, ―Ad Hoc and Sensor Networks: Theory and Applications (2nd Edition), World Scientific Publishing, 2011.
3. Waltenegus Dargie, Christian Poellabauer, ―Fundamentals of Wireless Sensor Networks Theory and Practice, John Wiley and Sons, 2010
4. Xiang-Yang Li, "Wireless Ad Hoc and Sensor Networks: Theory and Applications, 1227th edition, Cambridge university Press, 2008.

Course Code	Course Title						Core / Elective
SPC601 IT	Parallel Algorithms						Elective
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
–	3	-	–	–	40	60	3

Course Objectives:

1. To learn parallel Algorithm development techniques for shared memory and message passing models.
2. To study the main classes of parallel algorithms
3. To study the complexity and correctness models for parallel algorithms.

Course Outcomes:

At the end of this course, the students will be able to

1. Analyze Parallel Algorithms.
2. Implements PRAM and Basic Algorithms
3. Instrument with shared Memory Algorithms
4. Design Sorting and Selection Networks
5. Understand Distributed Algorithms.

UNIT-1

Introduction to Parallelism: Parallel Processing Ups and Downs, Types of Parallelism: A Taxonomy

Parallel Algorithms: Simple Computations and Architectures, Algorithms for linear Array, Algorithms for a Binary Tree, Algorithm for 2D Mesh, Algorithm for a Linear Array, Algorithms for a Binary, Algorithm for 2D Mesh, Algorithms with Shared Variables**.**

UNIT-II

PRAM and Basic Algorithms: PRAM Sub-models and Assumptions, Data Broadcasting, Semi-group or Fan, In Computations, Parallel Prefix Computations, Ranking the Elements of a Linked List, Matrix Multiplications.

UNIT-III

Shared Memory Algorithms: Sequential Rank Based Selection, A Parallel Selection Algorithm, A Selection Based Sorting Algorithm, Alternative Sorting Algorithm, Bionic algorithm

UNIT-IV

Mesh Base Architecture: Recursive Sorting Algorithms, Greedy Routing Algorithms, Graph Algorithms, Image Processing Algorithms.

UNIT-V

Distributed Algorithms: Models and Complexity measures, Safety, liveness, termination, logic time and event ordering, Global state and snapshot algorithms, Mutual exclusion and Clock Synchronization and Distributed Graph algorithms.

Text Books:

1. Ananth Grama, Anushul Gupta,George Karypis and Vipin Kumar, Introduction to Parallel Computing, Second edition, Addison-Wesley/Pearson,1994/2003.
2. Introduction to Parallel Processing Algorithms and Architecture, Behrooz Parhami,Platinum series of Computer science

Reference Books:

1. Vijaya K Garg," Elements of Distributed computing", Wiley Joseph F Jaja, An Introduction to Parallel Algorithms, Addison Wesley,1992.
2. MiChael J Quinn, Parallel Programming in C with MPI and OpenMP, first edition, McGraw Hill 2004/2003
3. Micheal J Quinn, Parallel Computing: Theory and Practice, Second edition, McGraw, 1994/2002.
4. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
5. Andrew S. Tanenbaum, Distributed Operating System, ACM Press.

Course Code	Course Title						Core / Elective
SPW614IT	Technical Seminar						Seminar
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
_	-	-	–	2	50	-	1

Course Objectives:

Seminar is an important component of learning in an Engineering College, where the student gets acquainted with preparing a report & presentation on a topic.

Course Outcomes:

At the end of this course, the students will be able to

1. Formulate a specific problem and give solution
2. Develop model/models either theoretical/practical/numerical form
3. Solve, interpret/correlate the results and discussions
4. Conclude the results obtained
5. Write the documentation in standard format

Guidelines for preparing Technical Seminar

1. **Selection of topic/area**

   Select a paper according to the specialisation of students. Papers from any other approved journals can also be selected.

2. **Approval to the selected topic**

   After selecting the paper, get approval from the concerned faculty in charge.

3. **Study of topic**

   Students are requested to acquire a thorough knowledge on the subject by referring back papers and reference books (These may be included as references at the end of the paper) on the corresponding area.

4. **Preparation of slides for presentation**

   Slides may be presented in MS power point. Time allowed for presentation is 20 minutes for presentation and 5 minutes for discussions. So, number of slides may be around 20 - 25 to adhere the time limit.

Seminar Report Guidelines

1. The references shall be written using IEEE style of reference writing.
2. The report shall be computer typed (English- British, Font -Times Roman, Size-12 point, Double spacing between lines) and printed on A4 size paper.
3. The report shall be hard bound with cover page in white colour. The name of the candidate, degree (specifying the specialization), month & year of submission, name of

the University including college name shall be printed in black on the cover [Refer sample sheet (outer cover)]

4. The report shall be typed on one side only with double space with a margin 3.5 cm on the left, 2.5 cm on the top, and 1.25 cm on the right and at bottom.

5. In the report, the title page [Refer sample sheet (inner cover)] should be given first then the Certificate by the candidate, followed by an abstract of the report (not exceeding 1500 words). This should be followed by the acknowledgment, list of figures/list of tables, notations/nomenclature, and then contents with page nos.

6. The diagrams should be printed on a light/white background, Tabular matter should be clearly arranged and the font of the Tabular matter should be Font -Times Roman, Size-10 point, Single spacing between lines. Decimal point may be indicated by full stop (.). The caption for Figure must be given at the BOTTOM (center aligned) of the Fig. and Caption for the Table must be given at the TOP (center aligned) of the Table. The font for the captions should be Times Roman, Italics, Size-10 point.

7. The font for the chapter titles should be Times Roman, Bold, Capital, Size-16 point and center aligned. The font for the Headings should be Times Roman, Bold, and Size-14 point. The font for the sub-headings should be Times Roman, Bold, and Size-12 point.

8. Conclusions must not exceed more than one page.

9. The graphs should be combined for the same parameters for proper comparison. Single graph should be avoided as far as possible.

10. The report must consist of following chapters:

    CHAPTER 1 - INTRODUCTION

    CHAPTER 2 - CURRENT STATUS

    CHAPTER 3 –IMPORTANCE AND APPLICATIONS

    CHAPTER 4 – LATEST RESEARCH

    CHAPTER 5 - FUTURE TRENDS

    CHAPTER 6 - CONCLUSIONS

    REFERENCES

    APPENDIX (IF ANY)

    ANNEXURES-I, II, III (IF ANY)

Paste a CD containing the soft copy of Report (in Docx and PDF), Reference papers and other material related to the work, on the inner side of back hard cover.

Course Code	Course Title						Core / Elective
SPW611IT	INTERNSHIP-II						Seminar
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
–	-	-	–	-	50	-	1

Course Objectives:

Student is able to test theoretical learning in practical situations by accomplishing the tasks assigned during the internship period.

Course Outcomes:

1. Student is able to construct the company profile by compiling the brief history, management structure, products / services offered, key achievements and market performance for his / her organization of internship.
2. For his / her organization of internship, the student is able to assess its Strengths, Weaknesses, Opportunities and Threats (SWOT)
3. Student is able to determine the challenges and future potential for his / her internship organization in particular and the sector in general.
4. Student is able to test the theoretical learning in practical situations by accomplishing the tasks assigned during the internship period.
5. Student is able to apply various soft skills such as time management, positive attitude and communication skills during performance of the tasks assigned in internship organization

Summer Internship is introduced as part of the curricula of encouraging students to work on problems of interest to industries. A batch of three students will be attached to a person from the Computer Industry/Software Companies/R&D Organization for a period of 4-6 weeks. This will be during the summer vacation following the completion of the III year Course. One faculty coordinator will also be attached to the group of 3 students to monitor the progress and to interact with the industry co-ordinate (person from industry).

After the completion of the project, student will submit a brief technical report on the project executed and present the work through a seminar talk to be organized by the Department. Award of sessionals are to be based on the performance of the students, to be judged by a committee constituted by the department. One faculty member will co-ordinate the overall activity of Industry Attachment Program.

Each group will be required to:

1. Submit a one page synopsis before the seminar for display on notice board.

2. Give a 30 minutes presentation followed by 10 minutes discussion.

3. Submit a technical write-up on the talk.