## Autonomous Programme Structure of Third Year B.Tech. Computer Engineering

### T. Y. B. Tech. Computer Engineering Semester – II

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Teaching Scheme Hours /Week</th>
<th>Examination Scheme</th>
<th>Marks</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lecture</td>
<td>Tutorial</td>
<td>Practical</td>
<td>In Semester</td>
</tr>
<tr>
<td>CE 3201</td>
<td>Theory of Computation</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>CE 3202</td>
<td>Artificial Intelligence and Machine Learning</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>CE 3203</td>
<td>Software Design and Architecture</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>PECE 3201</td>
<td>Programme Elective-II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>PECE 3202</td>
<td>Programme Elective-III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>CE 3204</td>
<td>Seminar</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>CE 3205</td>
<td>Artificial Intelligence and Machine Learning</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>PECE 3203</td>
<td>Programme Elective-III Laboratory</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>AC 3201</td>
<td>Audit Course</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>15</td>
<td>2</td>
<td>12</td>
<td>200</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td>550</td>
</tr>
</tbody>
</table>

**PECE 3201: Programme Elective-II**
1. Wireless and Mobile Communication
2. Software Testing and Quality Assurance
3. Human Computer Interaction
4. Multimedia Systems
5. Swayam Online Course

**PECE 3202: Programme Elective-III**
1. Data Mining and Data Warehousing
2. Embedded and Real-Time Systems
3. Linux Internals
4. Image Processing

**PECE 3203: Programme Elective-III Laboratory**

AC 3201 -- Audit Course: Employability Skills Development
CE 3201 Theory of Computation

Teaching Scheme
Lectures: 3 Hrs /week
Tutorial: 1 Hr /week

Examination Scheme
In Semester: 50 marks
End Semester: 50 Marks
Credits: 4

Prerequisites:  Data Structures and Algorithms II (CE 2201)
               Discrete Mathematics (CE 2103)

Course Objectives:
To facilitate the learners to -
1. Recall and understand the basics of mathematical concepts, formal languages and machines.
2. Understand and design different computational models like finite automata, regular expression, push down automata, context free grammar, and turing machine for a given language.
3. Apply inter conversion between equivalent representations of a language.
4. Learn classification of a given problem into appropriate complexity class.

Course Outcomes:
By taking this course, the learner will be able to -
1. Infer the fundamentals of mathematical concepts, formal languages and automata theory.
2. Construct different computational models like finite automata, regular expression, push down automata, context free grammar and turing machine for a given formal language.
3. Evaluate capabilities of Computational models by inter-conversion.
4. Classify a problem into appropriate complexity class.

Unit 1: Introduction  (06)
Regular Expression (RE): definition and operators, Regular Set, Algebraic Laws of Regular Expressions, Closure Properties of Regular Languages, Regular expression examples.

Unit 2: Finite Automata  (08)
Finite Automata (FA) - (Deterministic FA, Non-deterministic FA, C-NFA): Definition, Transition Function and language acceptance, Transition graph, Construction of FA.
Conversion of NFA with C moves to NFA without C moves, Conversion of NFA without C moves to DFA, Direct Conversion of NFA with C to DFA, Inter-conversion of RE and FA, Construction of RE equivalent to FA using Arden’s Theorem. Construction of FA equivalent to RE (RE to C-NFA, C-NFA to DFA). Pumping Lemma for Regular languages, Limitations of FA.

Unit 3: Context Free Grammar  (07)
Definition, representation of grammar. Context Free Grammar (CFG) - Definition, Derivation
– Leftmost, Rightmost, sentential form, parse tree, ambiguous grammar and removing ambiguity from grammar, Simplification of CFG, Normal Forms - Chomsky normal form, Greibach normal form, Closure properties of Context Free Languages (CFL), Decision properties of CFL, Chomsky hierarchy. Regular grammar- Definition, left linear, right linear grammar, Applications of grammar.

Unit 4: Push Down Automata (PDA)  (07)
Definition, Notations – Transition Table form, Types of PDA (Deterministic PDA and Non Deterministic PDA), acceptance by final state, acceptance by empty stack, Construction of PDA (DPDA, NPDA), Instantaneous Description of PDA. Equivalence of PDA and CFG - Grammar to PDA conversion, Applications of PDA.

Unit 5: Turing Machine  (07)
Turing Machine (TM) - Formal Definition, TM Instantaneous Description, Transition Function, Languages of TM, Turing Machine and halting, Deterministic Turing Machines (DTM) , Construction of DTM. Universal Turing Machine (UTM), Church -Turing hypothesis, Comparison between FA, PDA and TM. Turing Machine Halting Problem.

Unit 6: Introduction to Undecidability  (07)
A Language that is not recursively enumerable, Enumerating the binary strings, diagonalization Language, An undecidable problem that is RE, Recursive language, Complements of Recursive and RE languages, universal language, Undecidability of the universal language, classes P, NP and NP-Complete Problem

Text Books:

Reference Books:

Web References:
1. NPTEL:: Computer Science and Engineering – Theory of Computation
http://nptel.ac.in/courses/106101061

Example List of Tutorials:
1. Identify Complexity (n2, log n etc.)for a given code
2. Design of Regular Expression from Language
3. Design Deterministic Finite Automata
4. NFA design and NFA to DFA conversion
5. RE to NFA with null moves and NFA with null moves to NFA without null moves
6. Formal language to CFG and CFG to language conversion
7. Simplification of CFG and Chomsky Normal Form
8. Design of Push down Automata
9. Design of Turing Machine
10. Classification of a problem into appropriate complexity classes by reduction
CE 3202 Artificial Intelligence and Machine Learning

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 Marks
End Semester: 50 Marks
Credits: 3

Prerequisite: -

Course Objectives:
To facilitate the learners to-
1. Learn overview of classic Artificial Intelligence and basics of machine learning.
2. Understand various intelligent searches and knowledge representation.
3. Understand types of learning as well as machine learning.
4. Study applications in Artificial Intelligence and Machine Learning.

Course Outcomes:
By taking this course, the learner will be able to –
1. Solve classical Artificial Intelligence (AI) problems by choosing appropriate state space analysis technique.
2. Construct intelligent search techniques to solve multiplayer gaming problem.
3. Make use of Knowledge Management methods in reasoning.
4. Apply the appropriate supervised / unsupervised Machine Learning (ML) method to solve the given problem.
5. Understand and illustrate advanced application areas of AI and ML.

Unit 1: Introduction to AI (07)

Unit 2: Heuristic Search Techniques (07)

Unit 3: Knowledge Management (07)
Unit 4: Learning

Unit 5: Machine Learning methods and models
Introduction to Supervised, Unsupervised, semi-supervised Learning, Ensemble Learning, discovery based Learning, Learning by problem solving, Reinforcement Learning, Support vector Machine, Artificial Neural Network: Perceptron, multi-layer perceptron, back propagation Neural Network, Self-organizing map.

Unit 6: Applications in Artificial Intelligence and Machine Learning

Text Books:

Reference Books:
CE 3203 Software Design and Architecture

Teaching Scheme
Lectures: 3 Hrs /Week
Tutorial: 1 Hr /Week

Examination Scheme
In Semester: 50 marks
End Semester: 50 Marks
Credits: 4

Prerequisite: Data structures and Algorithms II (CE 2201)

Course Objectives:
To facilitate the learners to -

1. Develop familiarity with the basic concepts of software architecture and various architectural styles.
2. Get exposure to the quality attribute requirements of a system.
3. Model the software requirements of the systems using Unified Modeling Language (UML) to understand the architectural, structural and behavioral aspects of the system.
4. Understand and apply various design patterns in creating an object oriented design.
5. Learn the basic concepts of refactoring.

Course Outcomes:
By taking this course, the learner will be able to:

1. Understand the concepts of software architecture and learn the different architectural styles.
2. Identify and assess the quality attributes of a system at the architectural level.
4. Analyze various design patterns to understand reusability in object oriented design.
5. Apply the refactoring methods to restructure the classes.

Unit 1: Introduction to Software Architecture
What is Software Architecture? Why Software Architecture is important, Many contexts of Software Architecture, Architectural styles: event based, layered systems and client server system.

Unit 2: Quality Attributes
Introduction to Quality Attributes, Understanding quality attributes, Quality attributes Tactics:Availability, Interoperability, Performance, Security, and Usability.

Unit 3: Design Using Unified Modeling Language (UML)
Unit 4: Creational and Structural Design Patterns (07)

Unit 5: Behavioral Design Patterns (06)
Observer, Iterator, Model View Controller (MVC), Mediator, Strategy, State, Command.

Unit 6: Refactoring (08)
What is Refactoring, Why and when to Refactor, Refactoring and Design, Refactoring and Performance, Long Method, Large Class, Alternative Classes with Different Interfaces, Extract Method, Replace Method with Method Object, Move Method, Extract Class, Replace Data value with Object, Replace Conditional with Polymorphism, Replace Constructor with Factory method, Replace error code with exception.

Text books:

Reference books:

Example list of Tutorials
1. Study architectural styles and submit a report on these styles.
2. A case study of any website or any other large system and its architecture for quality attributes requirements such as Availability, Interoperability, Performance, Security and Usability.
3. Design a Software Requirement Specification (SRS) document for a given system.
4. Draw Use case diagrams for capturing and representing requirements of a system.
5. Draw Class diagrams to identify and describe key concepts like classes, relationships and other classifiers like interfaces.
6. Draw Sequence diagrams to show message exchanges in given system.
7. Draw Package Diagram to organize and manage large and complex system.
8. Draw Deployment diagrams to model runtime architecture of given system.
9. Identify suitable design patterns for a given application.
10. Apply the refactoring techniques for given code.
PECE 3201 Wireless and Mobile Communication

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 Marks
End Semester: 50 Marks
Credits: 3

Prerequisite(s): Computer Networks (CE 3101)

Course Objectives:
To facilitate the learners to-
1. Understand and remember fundamental concepts of Wireless Communication.
2. Compare different Wireless Network Standards.
3. Understand and apply Cellular system design fundamentals.
4. Understand modern mobile network architectures from design and performance perspective.

Course Outcomes:
By taking this course, the learner will be able to–
1. Understand basics of wireless communication and wireless standards.
2. Understand mobility management.
3. Recognize and analyze the important issues and concerns of Cellular system design.
4. Analyze evolution of mobile communication with recent trends and emerging technologies.

Unit 1: Introduction to Wireless Communication (07)
Introduction to wireless communication: Evolution, Types of wireless communication, Signals, antennas, signal propagation, mobile radio systems -examples, trends in cellular radio and personal communications, multiple access technologies: Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA).

Unit 2: Wireless LAN Standards (07)
Overview of 802.11 a, b, g, n standards, Concept of Spread Spectrum- Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), Comparison amongst 802.11 standards, Introduction and overview of MAC for 802.11 networks Carrier sense multiple access (CSMA/CA), Overview of IEEE 802.16 WiMax.

Unit 3: Global System for Mobile Communication (GSM) System (07)
Introduction, GSM background, GSM operational and technical requirements, Cell layout, GSM system architecture, elements of GSM architecture, Signal processing in GSM, Mobility management-Signaling protocols, Basic steps in the formation of a call, Handoff management.

Unit 4: General Packet Radio Service (GPRS) System (07)
Introduction and Need, GPRS system architecture, GPRS interfaces, GPRS transmission
plane, GPRS Mobility Management, MS State Transition, GPRS, GPRS routing and application.

Unit 5: Long Term Evolution Technologies (07)
Long Term Evolution (LTE) Technologies-Evolution to 4G, Orthogonal Frequency Division Multiplexing (OFDM), Downlink capacity comparison, Multiple Input Multiple Output (MIMO) spatial multiplexing, code words and layer mapping, Channel Coding schemes in LTE, Frequency Division Duplex (FDD) and Time Division Duplex (TDD).

Unit 6: Cellular System Design Fundamentals (07)
Introduction to Cellular system design concept, Importance of Frequency Reuse, Concept of Channel assignment and Handoff strategies, Interference and System capacity- Co-channel and Interference and System capacity, Channel planning for Wireless Systems Introduction to Trunking and Grade of service, Importance of Erlang B and C formula and Problem solving.

Text Books:

Reference Books:

Web References:
2. NPTEL: Introduction to Wireless and Cellular Communications- onlinecourses.nptel.ac.in/noc17_cs37/preview
PECE 3201 Software Testing and Quality Assurance

**Teaching Scheme**
Lectures: 3 Hrs /week

**Examination Scheme**
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

**Prerequisite:** -

**Course Objectives:**
To facilitate the learner to -
1. Develop familiarity with the fundamental concepts and the process of software testing.
2. Gain comprehensive knowledge about various software testing techniques and methods.
3. Study various software testing strategies.
4. Get exposure to the quality assurance process and its role in software development.
5. Learn the essential features of various automated testing tools used for testing different types of applications.

**Course Outcomes:**
By taking this course, the learner will be able to-
1. Understand the various concepts and process of software testing, testing metrics and quality assurance.
2. Apply various software testing techniques and strategies suitable to different problem areas.
3. Design the essential test cases at various phases of software testing life cycle.
4. Compare modern testing tools for testing various types of applications.

**Unit 1: Introduction** (06)

**Unit 2: Black Box Testing** (08)
Introduction, Need of black box testing, Requirements Analysis, Testing Methods - Requirements based testing, Positive and negative testing, Boundary value analysis, Equivalence Partitioning class, Domain testing, Design of test cases, Case studies of Black-Box testing.

**Unit 3: Testing Strategies and System Testing** (07)
Unit, Integration, System, Acceptance testing, Usability testing, Regression testing, Scenario testing, Adhoc testing, Functional, Performance testing, Stress testing, Security testing, Alpha-Beta testing.

**Unit 4: Testing Metrics and Quality Assurance** (07)
Testing Metrics and measurements, Types of metrics – Project, Progress, Productivity, Software quality, Quality control and assurance, Quality factors, Software Quality Assurance(SQA) Model - Six Sigma, Ishikawa’s Seven Basic Tools.
Unit 5: White Box Testing
Introduction, Need of white box testing, Testing types, Static testing by humans, Structural Testing – Control flow testing, Loop Testing, Design of test cases, Challenges in White box testing, Case-studies of White-Box testing.

Unit 6: Recent Trends and Automated Testing
Agile Testing, Model based testing, Need for Automation, Keyword driven automation, Data driven automation, Manual testing versus Automated testing, Automated Testing Tools, Selection of tool, Study of Testing tools and frameworks (such as Selenium, Junit, Bugzilla).

Text books:

Reference Books:

Web References
2. https://www.tutorialspoint.com/junit
3. https://www.bugzilla.org
PECE 3201 Human Computer Interaction

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

Prerequisite: -

Course Objectives:
To facilitate the learner to-
1. Determine the relationship between the user experience and usability.
2. Identify the main modes of human computer interaction.
3. Identify the common pitfalls in data analysis, interpretation and presentation.
4. Understand the use of scenarios and prototypes in design.
5. Identify different evaluation methods for different purposes at different stages of the design process.
6. Understand the advanced techniques of explicit Human Computer Interaction.

Course Outcomes:
By taking this course, the learner will be able to:
1. Understand importance of human centered software development.
2. Identify the interaction possibilities beyond mouse and pointer interfaces.
3. Illustrate data gathering needs for design requirements.
4. Analyze interaction designs.
5. Evaluate the different stages of design process.
6. Understand the advanced techniques in Human Computer Interaction.

Unit 1: Understanding Users and Introduction to Interactive Design (08)
Introduction to cognition, Cognitive framework, Good and poor design and components of Interaction design, The User Experience, Understanding the problem space and conceptualizing Design, Conceptual models, Interface Metaphors, Interaction types, Paradigms, Theories, Models and Frameworks.

Unit 2: Interaction Styles (08)
Direct Manipulation and Virtual Environments, Menu selection, Form Fill-in and Dialog Boxes, Command and Natural Languages, Interaction Devices, Emotional Interaction, Collaboration and Social Media Participation.

Unit 3: Establishing Requirements (07)
Understanding importance of identifying the requirements, Different kinds of requirements, Data gathering for requirements, Data analysis, Data interpretation and presentation, Task description and analysis.

Unit 4: Design, Prototyping, and Construction (06)
Prototyping and construction, Conceptual design, Physical design, Using scenarios in design, Using prototypes in design and support for design.

Unit 5: Evaluation Approaches (06)
Importance of evaluation, Evaluation approaches and methods, Evaluation case studies,
Determine, Explore, Choose, Identify, Decide, Evaluate (DECIDE): A Framework to guide evaluation.

**Unit 6: New Interaction Technologies** (07)
Explicit and Implicit Human Computer Interaction, User Interfaces and Interaction for Four Widely Used Devices, Hidden User Interface via Basic smart Devices, Hidden User Interface via Wearable and Implanted Devices.

**Text books:**


**Reference Books:**

PECE 3201 Multimedia Systems

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

Prerequisite: -

Course Objectives:
To facilitate the learners to-
1. Understand Multimedia basics.
2. Understand various file formats.
3. Learn Multimedia editing tools.
4. Analyze various compression techniques.
5. Learn advances in Multimedia.

Course Outcomes:
By taking this course, the learner will be able to-
1. Infer various media characteristics.
2. Apply digital image processing techniques in related applications.
3. Analyze various multimedia signals.
4. Relate and choose multimedia tools and technologies.
5. Understand advances in Multimedia.

Unit 1: Introduction to Multimedia (06)
What is Multimedia? (Text, Graphics, Audio, Video, Animation), Multimedia presentation and production, Multimedia Authoring Tools (Various tools for creation and editing of Multimedia Projects), Hardware and Software requirement for Multimedia, Multimedia Applications.

Unit 2: Text and Audio (08)
Text - Introduction, About Fonts and Faces, Using Text in Multimedia, Font Editing and Design Tools, Text Compression (HUFFMAN, LZ, LZW), File Formats (TXT, DOC, RTF, PDF, PS), Hypertext and Hypermedia.

Unit 3: Images (07)
Digital Image, Basic steps for image processing, Image file formats (BMP, TIFF), Image Compression (RLE, JPEG), Image Manipulation, Image processing softwares.

Unit 4: Video (07)
Types of Video Signals, Analog Video, Digital Video, Video File Formats and CODEC (AVI, MPEG), Video Editing Softwares.

Unit 5: Animation and Virtual Reality (07)
Animation- Introduction, Uses, Types, Principles, Animation on Web, 3D animation, Rendering, Animation Softwares.
Virtual reality - Introduction, Forms, Applications, Software Requirements, Devices, VRML.

Unit 6: Advances in Multimedia (07)
Introduction, Challenges of Multimedia Information processing, Watermarking, Organization, Storage and retrieval Issues, Neural Networks for multimedia processing, Multimedia Processors.
Text Books:


Reference Books:

PECE 3202       Data Mining and Data Warehousing

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 Marks
End Semester: 50 marks
Credits: 3

Prerequisite: Database Management Systems (CE 3102)

Course Objectives:
To facilitate the learners to -
1. Understand the concepts and techniques of data mining and data warehousing.
2. Apply various data pre-processing and visualization techniques.
3. Design and model a data warehouse and its components.
4. Compare and analyze various Data Mining algorithms based on performance parameters.
5. Understand advances in the field of Data Mining.

Course Outcomes:
By taking this course, the learner will be able to -
1. Demonstrate the need, importance and procedure of building a Data Warehouse (DW) to solve any Business Intelligence (BI) problem
2. Choose and apply appropriate pre-processing techniques to make data ready for further analysis
3. Design a Data warehouse model for the given application
4. Compare and analyze the strengths and weaknesses of various data mining algorithms
5. Understand the advances in the field of Data Mining.

Unit 1: Introduction to Data Warehousing and Data Mining (06)
Introduction to data warehousing and data mining. Evolution of decision support systems, operational data Vs. historical data (Data Warehouse data), importance of data preparation for data mining, types of data mining techniques, various data mining functionalities, data mining task primitives, integration of operational system and Data Warehousing system.

Unit 2: Data Preprocessing (08)
Introduction / overview of data pre-processing; Descriptive data summarization – Measuring central tendency, dispersion, range, quartiles, variance and standard deviation of data, Graphical displays of descriptive data summaries; Data cleaning, Data Integration, Data Transformation, Data Reduction.

Unit 3: Data Warehouse and Online Analytical Processing (OLAP) Technology (07)
3-tier Data Warehouse architecture, data warehouse design process; Modelling subject(s), dimensions and measures, Multidimensional data modelling, Introduction to OLAP, OLAP operations, Data cube generation, Concept hierarchy generation, Case study on designing a Data warehouse for a given application.
Unit 4: Data mining Functionalities - I
Data mining process, Types of Data Mining Systems; Cluster Analysis - Types of Data In Cluster Analysis, Categorization of Major Clustering Methods, k-means clustering, Density based Clustering.

Unit 5: Data mining Functionalities - II
Classification and Regression, Decision Tree Induction, Bayesian Classification, Nearest Neighbor approach; Mining frequent patterns and Association Rules – Apriori Algorithm, Outlier analysis.

Unit 6: Advances in Data Mining
Information Retrieval and Text Mining, Multimedia Data Mining, Graph Mining, Mining World Wide Web, Stream, Time series and Sequence data mining, Applications and trends in Data Mining.

Text Books:

Reference Books:

Web References:
1. www.autonlab.org/tutorials : Statistical Data mining Tutorials
2. www-db.stanford.edu/ullman/mining/mining.html : Data mining lecture notes
3. ocw.mit.edu/ocwweb/sl-School-of-management/15-062Data-MiningSpring2003/course home/index.htm : MIT Data mining open course ware
4. www.kdnuggets.com : Data mining resources
PECE 3202 Embedded and Real-Time Systems

**Teaching Scheme**
Lectures: 3 Hrs /week

**Examination Scheme**
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

**Prerequisite:** Microprocessor Architecture (CE 2204)

**Course Objectives:**
**To facilitate the learners to -**

1. Understand processors, its components use for embedded product.
2. Implement use of system hardware in various embedded designs.
3. Differentiate between use of embedded communications protocols and its interfacing to memory and processor.
4. Execute smaller codes written for embedded system programming using different languages.
5. Understand real time operating systems and compare different scheduling algorithms.

**Course Outcome:**
**By taking this course, the learner will be able to –**

1. Summarize embedded systems with different components and design process.
2. Design an embedded system for a given application using system hardware components.
3. Analyze processor, memory, input/output and communication protocols requirement for a given embedded system.
4. Develop skills for embedded system programming.
5. Understand Real Time Operating System (RTOS) and exemplary operating system use for various embedded applications.

**Unit 1: Introduction to Embedded Systems** (06)
Components of Embedded System & its Classification, Characteristic of embedded system.
Structural Units of Processor, Comparison of Microprocessors & Microcontrollers (8051 block diagram).
Introduction to embedded processor, Digital Signal Processor, Application Specific System Processor, Multiprocessor systems using General Purpose Processor. CISC and RISC Processor architectures.

**Unit 2: System Hardware** (08)
ARM7 Processor - Architecture, Register set, Modes of operation, Interrupt Structure and

Unit 3: Memory and I/O interfacing and Communication Buses (07)

Unit 4: Programming concepts, Embedded System Programming (07)
Programming in Assembly language using ARM processor, Use of High level Language – C and Python for Embedded System Applications, Selection of data structures, Micro, function, statement, loops etc. Embedded system programming using Development boards, Development tools – Simulator / emulator / debugger.

Unit 5: Real time Operating System (07)

Unit 6: Exemplary Operating Systems and Representative Embedded System (07)
Examples of Real Time OS, embedded System OS and Handheld OS. Representative Embedded Systems – Digital Thermometer, smart card Design Examples and case study of Automatic Vending machine / Automatic Cruise control System their Block diagram, class diagrams.

Text Books:

Reference Books:
2. ARM 7 Manual.
PECE 3202 Linux Internals

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

Prerequisites: Operating Systems (CE 2203)

Course Objectives:
To facilitate the learners to -
1. Understand basic concepts of Unix Operating System
2. Understand Linux kernel and environment.
3. Understand process and memory management in Linux.
4. Learn basics of Inter process communication w.r.t. Unix/Linux and perform socket programming.
5. Learn AWK tool and programming in AWK.

Course Outcomes:
By taking this course, the learner will be able to –
1. Understand kernel architecture, files and buffer management in Unix/Linux and summarize variants in Linux.
2. Apply process, thread and memory management techniques in Unix/Linux.
3. Apply various Inter process communication primitives.
4. Select appropriate Unix/Linux tools and build scripts for various operations under Unix/Linux.

Unit 1: Foundation of Unix Operating System (07)
Introduction, Kernel architecture, types of kernel, Operating system: Booting process, Grub I, Grub II, Representation of files, systems call File system, Concept of Buffer management in Unix/Linux.

Unit 2: Process and Threads in Linux (08)
Process states and transitions, layout of system memory, Context of a process, saving the context of a process, Concept of threads, Linux processes and thread management, introduction to threads (advantages and implementation), Process management and Linux scheduler.

UNIT 3: Memory management policies (07)
Swapping, Demand Paging, A hybrid system with swapping and demand paging, Linux memory management.

UNIT 4: Inter-process Communication (IPC) in Linux (07)
Process tracing, system V IPC, Network communication, sockets, Multiprocessor systems: problem with multiprocessor systems, solution with master slave processes, Shared memory.
pipes. Linux Inter process communication: User level IPC mechanism, Kernel synchronization, socket programming

Unit 5: Basic Tools (07)
Search and Sort tools: grep, egrep, fgrep, MAKE tool: When to use MAKE, Macros, abstractions and shortcuts, make, nmake, cmake. Awk tool: AWK syntax, AWK grammar, AWK scripting

Unit 6: Variants in Linux (06)
Hand-held systems: requirements, Linux as hand-held operating system, Linux for distributed systems, technology overview, Case study: Google Android

Text books:


Reference books:

PECE 3202  Image Processing

Teaching Scheme
Lectures: 3 Hrs /week

Examination Scheme
In Semester: 25 marks
End Semester: 50 marks
Credits: 3

Prerequisite: -

Course Objectives:
To facilitate the learner to-
1. Understand basic concepts of digital image processing.
2. Learn and apply image enhancement and Image Segmentation techniques.
3. Understand object Recognition, Image Restoration and reconstructions.
4. Learn and apply image compression techniques and Understand image processing applications.

Course Outcome:
By taking this course, the learner will be able to -
1. Relate basic knowledge of two dimensional array with digital image processing.
2. Apply image enhancement techniques and Image Segmentation on images.
3. Apply Image Restoration, reconstructions techniques and perform object recognition.
4. Analyze image compression techniques and review image processing applications.

Unit 1: Introduction to Image Processing (07)
Introduction to digital image processing: Origin, usage and application of image processing, Fundamental steps and component of image processing system, introduction to Human Visual System, Image sensing and acquisition, Basic concepts in sampling and quantization, representation of digital images. Elements of matrix theory.

Unit 2: Image Enhancement Techniques (08)
Basic image preprocessing (contrast enhancement, simple noise reduction, color balancing), some basic gray level transformations, Histogram Processing, Arithmetic Operations, Spatial filtering, Smoothing and Sharpening Spatial filters, Image Enhancement in the Frequency Domain, Gaussian filters, Homomorphic filtering.

Unit 3: Image Compression (07)
Introduction to Image Compression and its need, Coding Redundancy, Classification of Compression Techniques (Lossy and Lossless - JPEG, RLE, Huffman, Shannon fano), Scalar & Vector Quantization.

Unit 4: Image Restoration and Reconstruction (06)
Model of Image degradation, Noise Models, Classification of image restoration techniques, Blind-deconvolution techniques, Lucy Richardson Filtering, Wiener Filtering.

Unit 5: Image Segmentation, Analysis and Object Recognition (08)
Introduction to feature extraction: Edges, Lines and corners detection, Texture and shape measures. Segmentation and thresholding, region extraction, edge (Canny) and region based approach, use of motion in segmentation.
Introduction to Object Recognition, Object Representation (Signatures, Boundary Skeleton), Simple Boundary Descriptors, Regional descriptors (Texture).

Unit 6: Advances in Image Processing Applications (06)
Medical Image Processing, Face detection, Iris Recognition, Synthetic-aperture radar (SAR) Image Processing

Text Books:

Reference Books:

Web Reference:
1. https://serc.carleton.edu/usingdata/datasheets/SARdata.html
CE 3204 Seminar

Teaching Scheme
Practical: 4 Hrs/week

Examination Scheme
In Semester: 25 marks
Oral: 25 marks
Credits: 2

Course Objectives:
To facilitate the learners to-
1. Identify the topic based on computer science or engineering trends/ current social problems/ new technologies.
2. Explore the basic principles of communication (verbal and non verbal) and active, empathetic listening, speaking and writing techniques.
3. Produce relevant technical documents by following best practices of technical writing.
4. Understand the basic principles of presentation and technical writing techniques for seminar.

Course Outcomes:
By taking this course, the learner will be able to-
1. Select appropriate/research topic and write a technical report and present it to audience.
2. Be familiar and use the basic technical writing concepts and terms such as audience analysis, jargon, format, visuals and presentation.
3. Enhance skills to read, understand and interpret material on technology.
4. Strengthen technical communication and presentation skills.

General Guidelines for Seminar:
- Seminar is an individual student activity.
- The area/domain must be selected under the guidance of institute guide.
- Each student will select a topic in the current/new trends of Computer Engineering and Technology beyond the scope of syllabus avoiding the repetition in consecutive years.
- Student should do - literature survey based on IEEE/ACM/Springer/Digital Library papers or technical Magazines/books, specify knowledge area, brief technical knowledge about the topic.
- Each student will make a seminar presentation based on the domain topic using audio/video aids for a duration of 20-25 minutes.
- Students will have to submit the technical seminar report in the department.

Guidelines for assessment:
- Internal guide will evaluate students on understanding, punctuality, timely completion, active participation and other criteria as thought relevant.
- A panel of examiner(s) will assess the seminar work based on parameters like understanding, presentation, question and answers, active participation and the other criteria as thought relevant by the panel of examiner(s).

References:
1. Research papers from reputed journals/transactions- references necessary for the Project.
2. Reference books/Magazines for conceptual technical support.
CE 3205 Artificial Intelligence and Machine Learning Laboratory

Teaching Scheme
Practical: 4 Hrs /week

Examination Scheme
Practical: 50 marks
Credits: 2

Course Objectives:
To facilitate the learners to-
1. Experiment Artificial Intelligence and machine learning concepts from syllabus.
2. Experiment AI searches like $A^*$, Min-max algorithm.
3. Understand monotonic and non-monotonic knowledge representation.
4. Experiment classification and clustering algorithms.

Course Outcomes:
By taking this course, the learner will be able to-
1. Implement and analyse various intelligent searching techniques.
2. Apply Knowledge Management techniques to implement truth maintenance system / Expert system.
3. Choose the appropriate supervised Machine Learning (ML) method and solve the given problem.
4. Choose the appropriate Unsupervised ML method and solve the given problem.

Example list of Assignments:

Assignments Group A (Mandatory)
1. Study: Learning simple statements in Prolog
2. Implement DFS/BFS for simple water jug problem
3. Implement $A^*$ algorithm for 8 puzzle problem
4. Implement Unification algorithm
5. Represent knowledge using Prolog by implementing small

Assignments Group B (Any 3)
1. Write a program to implement Min-max algorithm
2. Write a program to implement Perceptron in artificial neural network
3. Write a program to implement SOM
4. Write a program to implement SVM

Assignment Group C
Develop any one machine learning tool for application: character/sign classification
PECE 3203 Data Mining and Data Warehousing Laboratory

Teaching Scheme
Practical: 2 Hrs / week

Examination Scheme
Oral: 25 marks
Credit: 1

Course Objectives:
To facilitate the learners to -
1. Model and build a data mart / data warehouse.
2. Study and analyze various open source data sets to pre-process them using open source data mining tools.
3. Implement data mining algorithms to discover interesting patterns.
4. Analyze results of data mining algorithms

Course Outcome:
By taking this course, the learner will be able to –
1. Study and process raw data to model and build a data warehouse, using appropriate schema
2. Experiment with large open source datasets by applying pre-processing tools and techniques
3. Build and analyze various data mining algorithms on real time data
4. Implement advanced Data Mining functionalities such as Text Mining and Mining unstructured data.

Example List of Assignments

Assignments Group A (Mandatory)
1. Explore WEKA Data Mining / Machine Learning Toolkit and perform the following operations: Understand the features of WEKA toolkit, Study the arff file format, explore the available data sets in WEKA.
2. Load any one dataset in Weka and observe the following : List the attribute names and their types, Number of records in each dataset, class attribute (if any), Plot Histogram, Determine the number of records for each class, Visualize the data in various dimensions; Apply various pre-processing tasks; Apply classification OR clustering algorithms on the chosen dataset and observe the results
3. Implement K-means clustering algorithm using a programming language that you are familiar with such as Java / Python. Compare the performance of your algorithm on the dataset, used in Weka, on different parameters such as accuracy, scalability, efficiency etc. by changing input parameter value such as K.

Assignments Group B (Any 2)
1. Implement DBSCAN clustering algorithm. Compare the performance of your algorithm on the dataset, used in Weka, on different parameters such as accuracy, scalability, efficiency etc.
2. Implement a decision tree classification algorithm. Compare the performance of your algorithm on the dataset, used in Weka, on different parameters such as accuracy, scalability, efficiency etc.

3. Implement Apriori, a Frequent Pattern Analysis algorithm. Assume suitable data. Compare the performance of your algorithm on the dataset, used in Weka, on different parameters such as accuracy, scalability, efficiency etc.


**Assignments Group C (Any 1)**

1. Build a Data Warehouse / Data Mart (using open source tools like Pentaho or other data warehouse tools like Microsoft-SSIS etc.)
   Identify source tables and populate sample data
   Analyze which multidimensional model (Star, snowflake and Fact constellation) will be best suited for the given application and design the schema (Example Applications can be Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.)

2. Study any of the existing data warehouse / data repository / ... and prepare your report based on data / model / tools and techniques / software used etc.

3. Download, install and study the features of any open source data mining tool and compare its features with Weka.
PECE 3203 Embedded and Real-Time Systems Laboratory

Teaching Scheme
Practical: 2 Hrs / week

Examination Scheme
Oral: 25 marks
Credit: 1

Course Objectives:
To facilitate the learners to-
1. Understand various embedded development boards.
2. Implement different components of embedded systems on development boards.
3. Implement using assembly level language or high level language.
4. Develop mini applications based on embedded systems knowledge with proper design process.

Course Outcomes:
By taking this course, the learner will be able to –
1. Apply the knowledge of embedded programming for solving the given problem.
2. Develop various Input/Output interfacing modules for a given embedded board.
3. Apply the knowledge of embedded design process to implement smaller applications.

Example List of Laboratory Assignments

Assignments Group A (Mandatory)
1. Study of Operating System based Evaluation/development Board (16 or 32 bit Microcontroller based) - Hardware and IDE Software.
2. Write a Program to read input from the switches and display on LED using Microcontroller development board.
3. Write a Program to read key press from keypad and display the key ID on LED or LCD.
4. Write a program to control the replay operation as per switch position and to indicate its status on LED or LCD.
5. Write a program to communicate with PC serially.

Assignments Group B (Any 3)
1. Write a program for Data Acquisition system to Acquire data from ADC Channel, Convert it into Digital Format and transmit to PC.
2. Write a program to perform serial communication, which generates packets of 32 bits, where first bit of packet indicates whether the packet is control packet or data packet.
3. Write a program to implement process control application/s using the peripherals such as LED or LCD, Keyboard, ADC, Relays, Switches etc.
4. Write a shell script that displays the no of readable, writeable and executable files in specified directory.
5. Write a program that demonstrates the communication between two processes.
6. Write a program to simulate traffic signal using Beagle bone Black board.
7. Write a program to simulate lift elevator using Beagle bone Black board.
8. Write a program to simulate robotics (stepper motor) using Beagle bone Black board.

Assignments Group C (Any 1)

1. Study of Compiling the Embedded Linux Kernel
   1. selecting the kernel source
   2. configuring the kernel
   3. compiling or building the kernel modules
   4. installing the kernel modules
2. Building the File Systems
   1. Basic structure of the root file system
   2. kernel modules
   3. kernel images
   4. device files
   5. BusyBox
   6. Selecting a file system
   7. RAM disk

or

Building the ToolChain
   1. binutils
   2. gcc
   3. Design any course project/application using the various studied components.
Course Objectives:
To facilitate the learners to-
1. Use Unix commands and write shell script commands and shell script for Unix.
2. Use various system calls for disk and memory management in Unix.
3. Use various Inter Process Communication (IPC) mechanisms, such as sockets, pipes for communication between two processes.
4. Understand use of basic tools in Linux.

Course Outcomes:
By taking this course, the learner will be able to –
1. Develop script for a given problem.
2. Implement basic system calls in Unix/Linux.
3. Implement sample programs using IPC primitives in Unix/Linux.
4. Develop programs for disk and file management system in Unix/Linux environment.

Example list of Assignments:

Assignments Group A (Mandatory)
1. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
2. Write a shell script to find factorial of a given integer.
3. Implement in Java (modular programming) the following UNIX commands using System calls
   - cat
   - ls
   - mv
4. Write an IPC program using pipe. Process A accepts a character string and Process B inverses the string. Pipe is used to establish communication between A and B processes using Python or Java
5. grep, Make, nmake commands

Assignments Group B (Any 4)
1. Write an Awk script to count the number of lines in a file that do not contain vowels.
2. Write client and server programs (using Java) for interaction between server and client processes using Unix Domain sockets.
3. Write a python program for creating virtual file system on Linux environment.
4. Write a program in Java/Python to create a RAMDRIVE and associate an acyclic directory structure to it.
5. Write a Java program to create a Zombie process.
6. Write a Java/Python program that illustrates two processes communicating using shared memory.

Assignment Group C

1. Make tool (dependency file structure).
PECE 3203 Image Processing Laboratory

Teaching Scheme
Practical: 2 Hrs/week

Examination Scheme
Oral: 25 Marks
Credits: 1

Course Objectives:
To facilitate the learners to:
1. Learn basic image processing operations like image Read, Write, add, subtract.
2. Understand and apply algorithms used for image enhancement, edge detection.
3. Design image processing application using various techniques.
4. Learn and use different Image Processing Tools.

Course Outcome:
By taking this course, the learner will be able to:
1. Perform basic operations on given image.
2. Apply algorithms for image enhancement and edge detection effectively.
3. Develop small image processing application using various techniques.
4. Utilize Image Processing Tools to perform various image processing techniques.

Example list of Assignments:

Assignments Group A (Mandatory)
1. Write a program to create a simple image file in .tiff format, and display it.
2. Write a program to perform Intensity Transformation techniques on given image.
3. Write a program for image enhancement techniques.

Assignments Group B (Any 3)
1. Write a program using derivative filtering techniques for edge detection.
2. Write a program to illustrate Morphological transformation using Dilation.
3. Write a program to illustrate Morphological transformation using Erosion.
4. Write a program to illustrate Image Restoration techniques.

Assignments Group C (Any 1)
Develop any one of the Image processing application using MATLAB/OpenCV (in Limited Scope).
1. Medical Image Processing
2. Face detection
3. Iris Recognition
4. Finger Print detection