EEX3262 Introduction to Object Oriented Programming

Level	3
Course Code	EEX3262
Course Title	Introduction to Object Oriented Programming
Credit value	2
Core/Optional	Optional (Computer, Electrical, Electronics and Communication)
Course Aim/s	To write Programmes in Java by applying object-oriented concepts with appropriate error handling and code refactoring techniques for a given scenario.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Explain the impact of object oriented paradigm over major Programming paradigms, object oriented principles and how object oriented Programming evolved. CLO2: Apply code refactoring techniques and naming standards to refine the source code in written Programmes. CLO3: Develop solutions for a given set of problems in Java using appropriate object oriented concepts and exception handling techniques.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Introduction to Object Oriented Programming Session 1: Overview of Object Oriented Programming Session 2: Java Programming Environment Session 3: Data Types and Variables in Java Session 4: Conditional and Recursions Session 5: Objects and Classes Session 6: Attributes, Methods and Constructors Unit 2: Object Oriented Concepts Session 7: Encapsulation Session 8: Inheritance Session 9: Polymorphism Session 10: Refactoring, Abstraction and Reusability Mechanisms Session 11: Errors, Exceptions handling and Debugging **Laboratory Work:** 1. Write simple java Programmes to get familiar with Java control structures, class and method declarations, refactoring techniques. 2. Write simple java Programmes to get familiar with access specifiers, constructor, initializing an object, instantiation of an object, message passing and mutator methods 3. Write simple java Programmes to illustrate more examples with message passing, inheritance ,overriding and overloading 4. Write a console java Programme to solve a given problem by applying object oriented concepts to achieve code reuse

EEX3266 Information Systems and Data Management

Level	3
Course Code	EEX3266
Course Title	Information Systems and Data Management
Credit value	2
Core/Optional	Optional (Computer, Electrical, Electronics and communication)
Course Aim/s	To design a relational database model by applying database design concepts, techniques and data interpretation methods for a given real world scenario.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to CLO1: Describe the evolution of different types of Information systems. CLO2: Develop Entity Relationship (ER) models using basic concepts of ER diagrams for database modelling. CLO3: Design relational database schema in ER model using the normal forms (NF). CLO4: Implement a database using suitable DBMS. CLO5: Present analysed data to interpret useful information using appropriate formats.
Content (Main topics, subtopics)	**Outline Syllabus:** Unit 1 : Evolution of Information systems Session 1: Different types of Information systems Session 2: Managing Information Systems Unit 2: Introduction to Database Session 3: Introduction to database Session 4: Degrees of Data Abstraction in Data Modelling Unit 3: Database Analysis and Design Session 5: Database system development life cycle Session 6: Entity Relationship Model Session 7: Integrity Rules and Constraints Session 8: Relational Data Model Session 9: Data Normalization and the normal forms Unit 4: Database Users, Data manipulation & interpretation and visual presentation Session 10: Database Users, Data interpretation and visual presentation Session 11: Data manipulation using Structured Query Language (SQL) **Laboratory Work:** 1. Use a DBMS to create database and to perform data manipulation. 2. Use basic SQL commands for data definition and data manipulation. 3. Use advance SQL commands to retrieve data from multiple tables. 4. Use appropriate tools to interpret and visualize data retrieved from DBMSs.

EEX3269 Introduction to Mobile Application Development

Level	3
Course Code	EEX3269
Course Title	Introduction to Mobile Application Development
Credit value	2
Core/Optional	Optional (Computer, Electrical, Electronics & Communication)
Course Aim/s	Impart the knowledge of mobile application development platforms and constituents of a mobile application
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Identify how different features are incorporated in mobile application designs in different mobile platforms. CLO2: Select appropriate mobile architecture, development platform and monetize mechanism for a mobile application. CLO3: Design mobile application wireframes for the identified software specifications.
Content (Main topics, subtopics)	**Outline Syllabus:** Unit 1: Introduction to Mobile Applications Session 1: Introduction to Mobile Applications Session 2: Components of a Mobile Application Session 3: Basics of Mobile Application Design Unit 2: Mobile Operating Systems Session 4: Introduction to Mobile Operating Systems Session 5: Basics of Android Session 6: Basics of iOS Session 7: Basics of Windows Mobile Unit 3: Mobile Hardware Session 8: Mobile Processors Session 9: Memory Session 10: Sensors Session 11: Input-Output Unit 4: Mobile Application Development Tools Session 12: Native Development Tools Session 13: Cross Platform Development tools Session 14: Publishing tools and Developer Program Session 15: Monetization and Security **Mini Research:** Carry out a survey and identify how different features are implemented in mobile application designs of multiple mobile platforms. **Design Class:** Apply fundamentals of Android app design, including how to build a simple user interface and handle user input. Come up with an idea, design the wireframes and, select the mobile architecture, development platform and monetize mechanism for a mobile application as a group **Laboratory Work:** Set up the development environment for mobile platforms.

EEX3336 Communications and Computer Technology

Level	3
Course Code	EEX3336
Course Title	Communications and Computer Technology
Credit value	3
Core/Optional	Core (Computer, Electrical, Electronic & communication)
Course Aim/s	To provide knowledge on fundamentals of computer technology, networking and communication.
Course Learning Outcomes (CLO):	At the completion of this course, student will be able to CLO1: Describe the evolution of microprocessors and computer models, and their applications to solve real-world problems. CLO2: Describe basic components of a microprocessor and their operations with reference to different Instruction Set Architectures (ISAs), and peripherals used in computers. CLO3: Build a personal computer connected to a network using basic components and peripherals. CLO4: Write a simple Programme using a given ISA and simulate its operation in fetch-execute cycle. CLO5: Perform simple arithmetic calculations in different number systems for a given problem. CLO6: Describe the fundamentals of computer networking, Internet and Internet services, and how OSI reference model is applicable in a computer network. CLO7: Explain security threats and preventive & recovery measures in computers and computer networks. CLO8: Identify components commonly used in wired and wireless communication systems. CLO9: Perform basic calculations related to analog and digital modulation techniques and power in communication systems.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Computer Technology Session 1: Evolution of microprocessors and computer models Session 2: Arithmetic in computers: whole numbers Session 3: Arithmetic in computers: fractions Session 4: Where these arithmetic operations are done in a computer Session 5: Components connected to a microprocessor Session 6: How operations are done in a microprocessor Session 7: How to use machine instructions to solve problems Session 8: What you finally get Unit 2: Computer Communications Session 9: Introduction to networking Session 10: Connecting devices and network topologies in LAN, MAN and WAN Session 11: The Internet, World Wide Web and network Security Session 12: Introduction to wired and wireless communications Session 13: Modulation techniques in communication systems Session 14: Power calculations in communication systems **Laboratory work:** 1. a) Identify the basic components and peripherals of a personal computer and their functions.

	b) Perform assembling of a personal computer, installing operating system. 2. Study the fetch-execute cycle for a given assembly Programme. 3. Implement a simple network and get it connected to a LAN and hence to the internet. Verify the functions at transmitter and receiver ends using amplitude modulation.

EEX4347 Software Engineering Concepts

Level	4
Course Code	EEX4347
Course Title	Software Engineering
Credit value	3
Core/Optional	Core (Computer)
Course Aim/s	To provide knowledge to develop a software solution/s for a given problem using appropriate software engineering concepts
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Select appropriate software development model for the development and maintenance of diverse software products. CLO2: Analyze the requirements gathered in technical and non-technical fields to prepare a Software Requirement Specification (SRS). CLO3: Design a solution for a given problem using an appropriate design method. CLO4: Apply principles of Human Computer Interaction (HCI) in designing a software system. CLO5: Develop a software application integrated with a database for a given problem. CLO6: Use appropriate verification and validation techniques for software quality assurance. CLO7: Describe how basic concepts in project management can be applied to a software project.
Content	**Outline Syllabus:** Unit 1: Software Process models Session 1: Introduction to Software Engineering Session 2: Overview of process models Session 3: Agile method Unit 2: Software Requirement Analysis Session 4: Requirement elicitation and documentation Session 5: Requirement analysis with data flow diagram (DFD) and unified modelling language (UML) Session 6: Creating Software Requirement Specification (SRS) and a test plan based on SRS Unit 3: Introduction to Object Oriented Concepts Session 7: Decomposing a problem into classes of objects Session 8: information hiding/encapsulation, coupling and cohesion, and data abstraction Session 9: Polymorphism/Method overloading, overriding and method binding Unit 4: Software Design Session 10: Function oriented design and component based design Session 11: Object oriented design with UML

	Session 12: Principles in Human Computer Interaction (HCI) Session 13: Developing a HCI system environment Session 14: Data modelling with UML Session 15: Developing database systems Unit 5 : Software Development Session 16: Web Technologies Session 17: Web Development Unit 6: Software Testing and Reliability Session 18: Software testing methods Session 19: Software reliability Session 20: Metrics of software quality Unit 6: Project management and maintenance Session 21: Software Maintenance Session 22: Software Project management Session 23: Software Cost Estimation **Mini Project:** Develop a simple software system integrating a database for a real world problem, including requirement elicitation and analysis, design and testing.

EEX4362 Object Oriented Design & Programming

Level	4
Course Code	EEX4362
Course Title	Object Oriented Design & Programming
Credit value	3
Core/Optional	Elective (computer)
Course Aim/s	Design and develop software solution/s by applying appropriate object oriented concepts, design patterns and frameworks
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Apply the basic concepts of object oriented design and Programming for a given scenario. CLO2: Explain how abstraction mechanisms support the creation of reusable software components. CLO3: Perform object oriented design and analysis on real world problems. CLO4: Document the design solutions with UML notations and diagrams. CLO5: Select appropriate software design patterns and frameworks for different problems.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Programming Concepts(2) Session 1: Object Orient Programme development Session 2: Multi-threaded Programming Unit 2: Application of Object Oriented Concepts

	Session 3: Interfaces and abstract classes Session 4: Application of fundamental concepts Session 5: Best practices in class design and coding Unit 3: Object-oriented Analysis and Design Session 6: UML and UML notations Session 7: Requirement Analysis - Use cases Session 8: Requirement Analysis - Activity and state transition Session 9: Class and object diagrams Session 10: Sequence diagrams Session 11: Collaboration diagrams Session 12: Packaging or component diagrams Unit 4: Design Patterns and Frameworks Session 13: Basics of Design Patterns Session 14: Creational Patterns Session 15: Behavioral Patterns Session 16: Structural Patterns Session 17: Best practices and industry frameworks Unit 5: Application Development Session 18: Introduction to application development[Use of Design patterns(MVC) Session 19: Front end development(JSF, JSP, JQuery) Session20: Database Connectivity Object Relational Mapping (Hibernate) **Laboratory work:** 1. Write simple java Programmes to get familiar with Java control structures, class and method declarations, refactoring techniques. Write simple java Programmes to illustrate applying OOP concepts 2. Use of API, Interfaces and abstract classes 3. Write a Multithreaded Programming 4. Using Hibernate, Object Relational Mapping, use of Factory and Abstract factory design pattern,

EEX4366 Data Modelling and Database Systems

Level	4
Course Code	EEX4366
Course Title	Data Modelling and Database Systems
Credit value	3
Core/Optional	Elective(Computer, Electrical, Electronics and Communication)
Course Aim/s	Apply appropriate data modelling techniques to design a database system using DBMS for a real-world scenario
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Design a relational model through conceptualized EER and refine using the normal forms (NF) to reduce redundancy and improve data integrity. CLO2: Implement the relational database design in an appropriate DBMS with database storage structures and access techniques. CLO3: Use SQL to create, manipulate and query a database. CLO4: Explain the role of a database administrator.

	CLO5: Design a database system for web applications.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1 Database Systems and Data Modelling Techniques Session 01: Database Management Systems & Data Modelling Session 02: Enhanced Entity - Relationship model Session 03: EER to Relational mapping Session 04: Database Design tool - UML Session 05: Higher forms of Normalisation Unit 2 Database file Organization Session 06: Data Storage- Disk and Files Session 07: File Organization for DBMS Session 08: Indexing structures Unit 3 Database Operations Session 09: Fundamentals of Database Operations Session 10: The Relational Database standards Session 11: Basic SQL Session 12: SQL complex queries Session 13: Triggers and transactions in SQL Session 14: Fundamentals of PL/SQL Session 15: NoSQL databases Unit 4 Database Administration Session 16: Database Administration Session 17: Key issues in Database Administration Unit 5 Databases for Web Applications Session 18: XML for Web applications Session 19: XML Languages Session 20: Web application development with XML **Laboratory Work:** 1. Create tables and write queries to extract data 2. Create views and triggers 3.Use appropriate SQL - Stored Procedures 4. Design a Web application using XML Get familiar with : -Specifying path expression in XML -* Query in XML **Mini Project:** Design a normalized relational database for a given case and implement using an appropriate DBMS.

EEX4435 Data Structures and Algorithms

Level	4
Course Code	EEX4435
Course Title	Data Structures and Algorithms
Credit value	4
Core/Optional	Core (Computer), Elective(Electronic and Communication)
Course Aim/s	To teach students to formulate and implement an algorithm using appropriate data structures to provide solutions for given problems.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Use appropriate data structures in real world software applications. CLO2: Design algorithms for solving complex problems. CLO3: Evaluate the complexity and the efficiency of different algorithms. CLO4: Implement an algorithm using a suitable Programming language.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Algorithms Session 1: Introduction to data structures, Algorithms and Programming Session 2: Fundamentals of Algorithmic Problem Solving Session 3: Analysis of Algorithm Efficiency Session 4: Recursion Unit 2: Fundamental Data Structures Session 5: Lists Session 6: Stacks Session 7: Queue Session 8: Binary Trees Session 9: Binary Search Trees Session 10: Heaps and Priority Queues. Properties Implementation Unit 3: Sorting and Searching Session 11: Internal Sorting Session 12: External Sorting Session 13: Searching Session 14: Hashing Unit 3: Advance Data Structure Session 15: Graphs Session 16: Shortest path problems Unit 4 Theory of Algorithms Session 17: Algorithm Designing Techniques Session 18: Recurrence Relation Session 19: Algorithmic complexity Session 20: Emerging trend in the field of data structures and algorithms Session 21: Parallel algorithms **Laboratory Work:** 1. Implement basic data structures using a Programming language 2. Demonstrate the application of sorting and searching methods for a given dataset 3. Implement trees and graph data structures. **Mini Project:**

	Apply and implement appropriate data structures and algorithms to device a solution for a given real world problem scenario.

EEX5270 Information Security

Level	5
Course Code	EEX5270
Course Title	Information Security
Credit value	2
Core/Optional	Core (Computer)
Course Aim/s	To impart the knowledge of the principles and practices of how information security functions in an organization.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Describe the information security overview relating to the history, standards, governance and regulations. CLO2: Describe the vulnerabilities relating to technical and human factors in information security. CLO3: Demonstrate the relationship between information security and risk management. CLO4: Describe algorithms and measures for enhancing security. CLO5: Distinguish strategic and tactical design issues in information security.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Information Security overview Session 01: History and overview Session 02: Security principles and mechanisms Session 03: Data protection standards Session 04: Computer Forensics

	Unit 2: Vulnerabilities: technical and human factors Session 05: Role of human behavior in security system design Session 06: Fault tree analysis and code reviews Session 07: Security Attacks Session 08: Ransomware and Malware Unit 3: Resource protection models, Message authentication codes, Secret and public key cryptography Session 9: Resource protection models Session 10: Cryptography Session 11: Symmetric algorithm design and implementation issues Session 12: Encryption using secret key Session 13: Message authentication codes Session 14: Public Key Infrastructure(PKI) Unit 4: Network and web security Session 15: Transport layer security (TLS) Session 16: Secure Network Access Session 17: Security in Software Unit 5: Secure Computing Session 18: Authentication Session 19: Trusted computing Session 20: Side-channel attacks **Case Study:** Analyse flow of information in an organization and identify the security measures implemented and potential risks

EEX5280 Creative Design

Level	5
Course Code	EEX5280
Course Title	Creative Design
Credit Value	2
Core/ Optional	Optional (Computer, Electrical, Electronics & Communication)
Course Aim/s	The aim of this course is to develop ability to understand, contextualize and analyze engineering designs catered to solve an identified problem and communicate to stakeholders
Course Learning Outcomes (CLOs)	At the completion of this course student will be able to: CLO1: Explain the principles of design process with respect to the design environment and develop a holistic view. CLO2: Carry out a need analysis and a requirement analysis. CLO3: Design products to cater the need applying innovative ideas. CLO4: Evaluate the proposed design against social, economic and environment impacts.

Content (Main topics, subtopics)	**Outline Syllabus:** Session 1: Introduction and Principles of design Problem identification, Ideate and Innovation, what is design, Design Environment, Holistic design, Intellectual Property, 10 step design process- Dieter Rams, Different design methods, Physical laws, Principles of reliability (design, cost and probability) Session 2: Design in practice How people treat different things, Research: stakeholder analysis / market research, Environmental impacts, Socio economic impact / constraints, Design process, Descriptive, prescriptive, Cost models, Design for manufacture and assembly Session 3: Psychology in design Theoretical aspects, Social psychology- establishing relationships, Motivation, Different thinking pattern- divergent, convergent, Evaluating ideas, Different mental, approaches- mind mapping, Left brain-right brain, Reflection- as a process of learning, Pattern Breaking-(Thinking differently ,Changing your point of view), Idea-collection processes-(Brainstorming/Brain-writing, Metaphoric thinking, Outrageous thinking), Mapping thoughts, Systematic logical thinking, Using math concepts Session 4: Designing tools and Product Development K-scripts for dialogue systems, State machines, Usability testing methods, Development concept, Product development process, Scope of the product development-(Technical questioning, Harvard business case methodology), Functional decomposition, Reverse engineering, Bench making and establishing engineering specification Session 5: Evaluation Evaluation Mechanisms, Social impact, cost impact, Usability testing, Performance evaluation.

EEX5346 Embedded Systems

Level	5
Course Code	EEX5346
Course Title	Embedded Systems
Credit value	3
Core/Optional	Core (Computer),Elective(Electronic and Communication)
Course Aim/s	To provide necessary knowledge and skills to design and develop embedded systems for different applications
Course Learning Outcomes (CLO):	At the completion of this course student will be able to CLO1: Evaluate important trade-offs among power, performance, and cost in embedded system design. CLO2: Apply advanced software design concepts to develop and debug applications for a target embedded system using modern tools and approaches. CLO3: Describe mobile and wireless embedded systems using both short-range and long-range in various interconnection architectures. CLO4: Apply the advanced interfacing techniques and persistent storage in embedded system. CLO5: Classify the modern computing platform for embedded systems. CLO6: Develop an embedded system for a real-world application fulfilling the given specifications.
Content	**Outline Syllabus:** Unit 1: Programming complex embedded systems Session 01: Programming tools and techniques Session 02: Structured approaches in writing complex embedded applications

	Session 03: Programming Techniques used in event-driven state machine frameworks Session 04: Programming embedded system applications Unit 2: Low power operation Session 05: Power saving approaches in embedded system design Session 06: Programming Techniques for low power operations Session 07: Energy storage techniques in embedded system Unit3: Mobile and network embedded systems Session 08 : Internet of Things Session 09 : Wireless Connectivity Session 10: Security in wireless communication. Unit 4: Advanced Interfacing techniques Session 11: Advanced input/output buses Session 12: Advanced serial bus protocols Session 13: Persistent storage for embedded systems Unit 5: Embedded systems Design Session 14: Multimedia peripherals in advanced embedded System. Session 15: Computing platforms in embedded applications Session 16: Embedded systems design methodologies **Laboratory work:** 1. Demonstrate the functionalities of the development system board using self-test method. 2. Implement the embedded system using development system board as a prototype model 3. Analyze energy usage of the prototype model for different configurations. **Design Project:** Design an embedded system for a real-world application fulfilling the given specifications.

EEX5351 Digital Electronic Systems

Level	5
Course Code	EEX5351
Course Title	Digital Electronic Systems
Credit value	3
Core/Optional	Core
Course Aim/s	To provide knowledge to design and skills for analysis to implement digital systems on appropriate hardware platforms.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to CLO1: Apply the concepts of basic timing issues, including clocking, timing constraints, and propagation delays during the design process. CLO2: Develop complex digital systems in a hierarchical fashion using top-down and bottom-up design approaches. CLO3: Verify the digital systems design via digital circuit simulation using a hardware descriptive language. CLO4: Implement digital system designs using Programmable platforms such as FPGAs and PLDs. CLO5: Explain the types and characteristics of common faults that occur in digital electronic systems. CLO6: Discuss different computer-aided testing tools for circuit simulation, fault diagnosis and ATPG.

Content (Main topics, sub topics)	Outline Syllabus:

Unit 01: Design Approaches

- Session 01: Design Approaches and Constraints of Digital Systems
- Session 02: Digital System Modeling Techniques
- Session 03: Designing with Sequential MSIs

Unit 02: Digital Design Hardware

- Session 04: Architecture of PLDs, PLAs, CPLDs, and FPGAs
- Session 05: Implementation of digital systems using PLDs, PLAs, CPLDs, FPGAs

Unit 03: Clock Generation and Timing Analysis

- Session 06: Clock Generation
- Session 07: Propagation, delay, Timing and Duty Cycle parameters

Unit 04: Finite State Machines and State based Representations

- Session 08: FSM and State Diagrams
- Session 09: Design and implementation using FSMs

Unit 05: CMOS Logic Gates

- Session 10: CMOS Design of Logic Gates
- Session 11: Analysis of the operation in each logic gate design

Unit 06: Logic Faults

- Session 12: Logic faults and Fault analysis of the CMOS logic gates
- Session 13: Fault Diagnosis: Sequential and Combinational Circuits

Unit 07: VHDL Programming

- Session 14: Introduction VHDL - Basic terminology and language elements
- Session 15: Behavioral Modeling
- Session 16: Dataflow Modeling
- Session 17: Structural Modeling
- Session 18: Test bench Modeling

Unit08: VLSI Testability

- Session19: Introduction VLSI, VLSI Testability and ATPG
- Session20: Design Styles: SOC and ASIC

	Laboratory work: 1. Design a digital electronic system for a given application scenario. 2. Implement the design on FPGA and verify the implementation **Mini Project** Provide a critical analysis based on a comprehensive literature survey on a given topic.

EEX5467 Software Testing and Quality Assurance

Level	5
Course Code	EEX5467
Course Title	Software testing and quality assurance
Credit value	4
Core/Optional	Elective (Computer)
Course Aim/s	To develop competencies of students to apply principles and concepts of testing and quality assurance to perform as a professional in software development process.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Demonstrate the ability to plan tests including the selection of techniques and methods to be used to validate the product against its approved requirements and design. CLO2: Demonstrate the ability to execute tests, design test cases; use test tools; and monitor testing to ensure correctness and completeness in a software quality assurance team. CLO3: Apply different testing techniques in an effective and efficient manner to test different software problems. CLO4: Analyse a software process to evaluate how effective it is at promoting quality. CLO5: Identify industry wide quality standards and processes to apply in different type of businesses.
Content (Main topics, subtopics)	**Outline Syllabus:** Unit 1: Introduction to Software Quality Assurance Session 1: Definitions of Quality and Quality attributes of software and Quality Vocabulary Session 2: Societies concerns for Quality, the costs/impact of bad quality and Cost of Quality Model Session 3: The dimensions of quality engineering and roles of people Unit 2: Software Testing Process Session 4: Test Planning, Test Analysis & Design Session 5: Test Implementation & Execution, Monitoring and Control Session 6: Software defects management Session 7: Evaluation of Test Exit Criteria and Reporting Session 8: Test Closure Activities Unit 3: Requirement Validation and Reviews Session 9: Formal Requirement Analysis Session 10: Desk Checking, Walk through, Inspections, Management Reviews & Audits Session 11: Prototyping to validate requirements Unit 4: Project Risk Management by Software Testing and Quality Assurance Session 12: Software Risk Management (Identify, Plan, Analyze, Control, Review)

	Session 13: Software Verification and Validation Session 14: Quality Metrics Unit 5: Testing types and techniques Session 15: Unit testing Functional testing, White-box testing techniques, Black-box testing techniques Session 16: Regression Testing, System testing and Integration testing Session 17: Operational Profile based testing, User Acceptance testing, Session 18: Deployment Process & Deployment testing Session 19: Non-functional testing Unit 6: Test Artefacts, Tools and Automation Session 20: Test Policy, Test Strategy, Master Test Plan, Level Test Plan, Internal Acceptance criteria Session 21: Developing test cases using software requirements/user stories (Mind Mapping, Scripting) Session 22: Introduction to Test Automation Session 23: Testing tools Tool selection (Open source tools, custom tools, ROI), Tool lifecycle and metrics, Advantages and disadvantages of automation Session 24: Practical session on functional test automation Session 25: Practical session on non-functional test automation Unit 7: Quality Standards and Processes Session 26: Software life cycle processes (Traditional, Agile, RUP etc.), Test Driven Development Session 27: Organizational implementation of standards Session 28: IEEE Software Quality related standards Session 29: ISO Quality Management Systems (9001:2015 and 9001:27001), ISO 15504, SO/IEEE Standard 12207 Session 30: Capability Maturity Model Integration (CMMI and TMMI) Session 31: Lean Six Sigma Session 32: Root Cause analysis (Pareto, Fishbone and Etc.) Session 33: Statistical Process Control and Continuous Quality Improvement
	Case Study: Provide a critical analysis based on a comprehensive literature survey on a selected topic.

EEX5536 Computer Architecture

Level	5
Course Code	EEX5536
Course Title	Computer Architecture
Credit value	5
Core/Optional	Core(Computer)
Course Aim/s	To provide the necessary knowledge to analyse, organise and design various computer systems.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Analyze performance of computer systems using different performance metrics and benchmarks considering their strengths and weaknesses. CLO2: Explain the functions of components, data-path and the control unit of a processor when executing its instructions. CLO3: Evaluate hierarchical memory design and memory organizations for improving overall performance of a computer system.

	CLO4: Explain Input/output systems including I/O devices and interfacing techniques used in a computer. CLO5: Demonstrate the ability to do low level Programming for a target processor and to write I/O drivers using assembly language. CLO6: Perform a survey on contemporary computer architectures and their impact to the field of computing and the society CLO7: Explain computer arithmetic, arithmetic unit of a processor and its effect on overall performance.
Content (Main topics, sub topics)	**Outline Syllabus:** History and overview (1 session): ▪ Articulate differences between computer organization and computer architecture. ▪ Explain the reasons and strategies for different computer architectures and indicate some strengths and weaknesses inherent in each. ▪ Describe how computer engineering uses or benefits from computer architecture and organization. *Relevant tools, standards and/or engineering constraints* ▪ Discuss the type of information contained in the IEEE 754 standard for floating-point arithmetic. ▪ Discuss the type of information contained in one or more components interconnect standards. ▪ Discuss how architecture design choices and tradeoffs influence important consequences such as performance, power, etc. Instruction set architecture(3 sessions): ▪ Explain the organization of a von Neumann machine and its major functional units. ▪ Articulate the strengths and weaknesses of the von Neumann architecture, compared to a Harvard or other architecture ▪ Explain the relationship between the encoding of machine-level operations at the binary level and their representation in a symbolic assembly language. Explain different instruction format options, such as the number of addresses per instruction and variable-length versus fixed-length formats. ▪ Describe reduced (RISC) vs complex (CISC) instruction set computer architectures. Measuring performance (3 sessions): ▪ Understand the factors that contribute to computer performance. ▪ Articulate the rationale for and limitations of commonly used computer performance metrics, such as clock rate, MIPS, cycles per instruction, throughput, and bandwidth. ▪ Describe the rationale for and limitations of benchmark Programmes. ▪ Name and describe two commonly used benchmarks for measuring computer performance, and compare two different computer systems using published benchmark results. ▪ Select the most appropriate performance metrics and/or benchmarks for evaluating a given computer system, for a target application. ▪ Explain the role of Amdahl's law in computer performance and the ways control and data path design can affect performance. Computer arithmetic(3 sessions): ▪ Determine the characteristics of commonly used number systems such as range, precision, accuracy, and conditions that lead to arithmetic overflow and tradeoffs between characteristics of different number systems. ▪ Describe the limitations of computer arithmetic and the effects of errors on calculations. ▪ Describe basic arithmetic algorithms for addition, subtraction, multiplication and division of integer binary numbers.

	Convert floating-point numbers to and from binary format, using a standard for floating-point arithmetic. ▪ Describe algorithms for addition, subtraction, multiplication, and division of floating-point numbers. ▪ Describe how multi-precision arithmetic is performed in a computer system. Processor organization (8 sessions): ▪ Discuss the relationship between instruction set architecture and processor organization. ▪ Compare and discuss tradeoffs between alternative implementations of data paths for a Von Neumann machine. ▪ Explain basic instruction-level parallelism (ILP) using pipelining, the effect of pipelining on performance, and the major hazards that may occur, including performance penalties resulting from hazards. ▪ Explain the steps needed to overcome the effect of pipeline hazards caused by branches. ▪ Describe common exception and interrupt handling mechanisms used in computer systems. ▪ Describe the characteristics of superscalar architectures, including multi-issue operation, in-order and out-of-order execution. ▪ Describe how each of the functional parts of a computer system affects its overall performance. Memory system organization and architecture (8 sessions): ▪ Design a main memory with specified parameters using given memory devices. ▪ Discuss how memory performance metrics, such as latency, cycle time, bandwidth, and interleaving, are used to measure the effects of memory on overall system performance. ▪ Explain the use of memory hierarchy to reduce the effective memory latency in a system. ▪ Describe common cache memory organizations, explain the use of cache memory to improve performance, and discuss cost-performance trade-offs of different cache organizations. ▪ Describe mechanisms used to provide cache coherence, invalidation/snooping, and shared/exclusive access control. ▪ Describe characteristics of current secondary storage technologies, such as magnetic, optical, and solid-state drives. Input/output interfacing and communication (6 sessions): ▪ Describe how a processor interacts with input/output (I/O) devices, including peripheral addressing (isolated vs memory-mapped) handshaking, and buffering ▪ Explain the use of interrupts to implement I/O control and data transfers, including vectored and prioritized interrupts, and discuss factors that contribute to interrupt overhead and latency. ▪ Write small interrupt service routines and I/O drivers using assembly language. ▪ Discuss the use of direct memory access (DMA) to interact with IO devices. ▪ Determine trade-offs between Programme-controlled IO, interrupt-driven IO, and DMA for a given application. ▪ Describe the characteristics of a parallel bus, including data transfer protocols. ▪ Describe characteristics of asynchronous and synchronous serial communication protocols. ▪ Discuss trade-offs between parallel and serial data transmission between devices. Peripheral subsystems (2 sessions): ▪ Describe the characteristics of one or more computer system expansion buses and select an appropriate bus for connecting given components/subsystems to a computer system.

	Describe data access from a secondary storage device such as a magnetic or solid-state disk drive. ▪ Describe storage subsystems: storage technologies, storage controllers. ▪ Describe display subsystems: audio/video subsystems, display controllers. ▪ Describe input device subsystems (e.g., keyboard, mouse). ▪ Describe communication subsystems: network controllers, serial and parallel communication functions **Laboratory Work:** Interface I/O devices with computers developing necessary software. Requirements: analyze pipelines, parallel port, serial port, ISA, PCI card

EEX6181 Research Methodology and Project Identification (Computer Engineering)

Level	6
Course Code	EEX6181
Course Title	Research Methodology and Project Identification (Computer Engineering)
Credit value	1
Core/Optional	Core
Course Aim/s	To provide the student with knowledge of research-based approach in providing solutions to engineering problems
Course Learning Outcomes (CLO):	After completion of this course student will be able to: CLO1: Demonstrate the knowledge of types of research designs, quantitative and qualitative forms of research, and use the most appropriate design for a given issue. CLO2: Create problem statement, purpose statement, research problem and objective of a research. CLO3: Conduct a literature survey effectively in view of information collection with regard to a given issue, critically evaluate outcomes of literature survey, and document them with recognized referencing methods. CLO4: Prepare a research proposal based on preliminary investigations. CLO5: Describe ethical issues concerning research and identify and avoid such issue in engineering research. CLO6: Plan and execute substantially research based projects with innovativeness and initiative and with a personal autonomy and accountability. CLO7: Prepare research abstracts and full length articles according to standard formats.
Content (Main topics, sub topics)	**Outline Syllabus:** Session 01: Meaning and objectives of research Session 02: Types of research Session 03: Literature survey and referencing methods Session 04: Defining a research problem Session 04: Preparation of a research proposal Session 05: Ethics in research Session 06: Dissemination of research outcomes and patenting **Activities:** Workshop on research problem formulation

EEX6236 Advanced Computer Architecture

Level	6
Course Code	EEX6236
Course Title	Advanced Computer Architecture
Credit value	2
Core/Optional	Core (Computer)
Course Aim/s	To disseminate knowledge of techniques to improve the performance of processor and computer system.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Observe the evolution of instruction sets to improve performance and the enhancement of instruction sets for different application domains. CLO2: Compare various techniques and approaches used for parallel processing and Programming. CLO3: Distinguish Multi/many-core architectures and Distributed system architectures including their Programming techniques, models, frameworks, and languages. CLO4: Explain memory organization in parallel computing systems and mechanisms used to improve performance and the reliability of the memory system. CLO5: Create parallel Programmes to a target platform for given problems.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Instruction set Architecture, Computer Arithmetic and Processor organization (3 sessions) Describe features and applications of short vector instruction sets: Streaming extensions, AltiVec, relationship between computer architecture and multimedia applications. Describe algorithms for higher-complexity functions, such as square roots and transcendental functions. Unit 2: Memory system organization and architecture(2 sessions) Understand how errors in memory systems arise, and describe several mechanisms used to resolve them, such as error detecting and error correcting systems, and RAID structures. Unit 3: Multi/Many-core architectures *Minimum core coverage time: (5* sessions) Discuss the performance limitations of single-core processors due to clock-frequency and power walls. Describe the basic organization of a multi/many-core, shared memory processor. Discuss the benefits of homogeneous vs heterogeneous multi/many-core architectures, and trade-offs between different architectures. Discuss on-chip interconnect networks and memory controller issues. Describe how Programmes are partitioned for execution on multi/many-core processors. Articulate current Programming techniques, models, frameworks, and languages for multi/many-core processors. Unit 4: Parallel algorithms and multi-threading (3 sessions) o Analyze the parallelism inherent in a simple sequential algorithm. o Explain why communication and coordination are critical to ensure correctness.

	○ Calculate the speedup attainable in theory and explain factors limiting attainable speedup. ○ Explain limitations to scalability. Unit 5: Distributed system architectures(2 sessions) ○ Explain the differences and trade-offs between different distributed system paradigms and their usefulness and applicability. ○ Discuss granularity and levels of parallelism in distributed systems, including threads, thread-level parallelism and multithreading. ○ Describe the topology, degrees of coupling, and other characteristics of several current multiprocessor/multicomputer architectures. ○ Understand how the client-server model works in a decentralized fashion. ○ Understand how agents work and how they solve simple tasks. ○ Articulate current Programming techniques, models, frameworks, and languages for distributed, parallel processing. ○ Describe the concept of logical clocks versus physical clocks and show how they affect implementation of distributed systems. ○ Be familiar with simple election and mutual exclusion algorithms and their applicability. ○ Describe approaches to design for parallelism, synchronization, thread safety, concurrent data structures. Discuss distributed transactions: models, classification, and concurrency control **Mini Research:** Review research papers to analyze contemporary computer architectures and their impacts on computer performance

EEX6335 Compiler Design

Level	6
Course Code	EEX6335
Course Title	Compiler Design
Credit value	3
Core/Optional	Core (Computer)
Course Aim/s	Aim of this course is to provide required knowledge and skills to develop compiler for real applications.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to : CLO1: Develop grammar for a compiler by analysing a target application CLO2: Apply the principles of theory of computation to develop the compiler based on the grammar developed. CLO3: Create scanner, parser and code generator of the compiler using LEX and YACC tools. CLO4: Construct the compiler of the target application. CLO5: Validate the constructed compiler with a selected set of samples for the target application.
Content (Main topics, subtopics)	**Outline Syllabus:** Unit 1: Theory of Computation

	Session 01: Grammars: Properties of Context Free Grammars Session 02: Finite State Automata and Regular Expressions Session 03: Pushdown Automata Session 04: Turing Machines Unit 2: Engineering a Compiler Session 05: An Overview of a compiler Session 06: Lexical analysis Session 07: Syntax analysis Session 08: Semantic analysis Session 09: Intermediate code generation Session 10: Run-time environments Session 11: Local optimizations Session 12: Machine code generation Session 13: Global register allocation Session 14: Machine-independent optimization Session 15: Overview of LEX Session 16: Overview of YACC **Design Project:** Apply theory of computation to design a compiler for a target application Implement Lexical analyser, syntax analyser and code generator for target application Evaluate grammar based on real example to verify the regular expression of the grammar.

EEX7171 Emerging Technologies

Level	7
Course Code	EEX7171
Course Title	Emerging Technologies
Credit Value	1
Core/ Optional	Elective (Computer, Electrical)
Course Aim/s	To provide an understanding of emerging technologies, their impact on the society and how to use them to find solutions for today's problems.
Course Learning Outcomes (CLOs)	At the completion of this course student will be able to: CLO1: Exhibit the inquisitiveness to appreciate contemporary and emerging trends. CLO2: Critically evaluate contemporary and emerging technologies to assess the impact in a relevant field. CLO3: Demonstrate the ability to research and present the findings.
Content (Main topics, subtopics)	Various contemporary and future trends in technology, delivered by thought leaders such as Industry representatives, leading researchers and other academics.

EEX7241 Neural Networks and Fuzzy Logic

Level	7
Course Code	EEX7241
Course Title	Neural Networks and Fuzzy Logic
Credit Value	2
Core/ Optional	Optional (Computer, Electrical)

Course Aim/s	To be able to apply neural network concepts and fuzzy logic for learning and prediction in intelligent systems.
Course Learning Outcomes (CLOs)	At the completion of this course student will be able to: CLO1: To create neural networks with suitable learning algorithms and architectures to solve a given problem. CLO2: To generate Fuzzy rules in order to create fuzzy interference systems CLO3: To apply neuro-fuzzy systems in Signal processing, communication systems and optimization system.
Content (Main topics, sub topics)	**Outline Syllabus:** Unit 1: Session 1: Review of Fundamentals of Artificial Neural Networks (ANNs): Neuron Physiology, Network architectures and learning processes, Single and multilayer perceptions, Back propagation algorithm Session 2: Learning Algorithms: Learning paradigms (Supervised, unsupervised, recurrent) with examples Session 3: Different types of Neural network models: Multilayer systems, radial basis function, simulation packages Session 4: Competitive networks: Self-organizing maps and applications Session 5: Deep learning Networks: Introduction, Convolution networks, Issue with Deep learning Unit 2: Session 6: A review of fuzzy set theory: crisp values, linguistic variables, fuzzy sets, fuzzy rules and fuzzy reasoning Session 7: Fuzzy relations and fuzzy rule-based systems: fuzzifier, fuzzy rule base, fuzzy inference systems, defuzzifier Session 8: Fuzzy systems: Design and implementation of fuzzy systems and applications Session 9: Principal components analysis: Application of PCA in Neural network and Fuzzy systems, real world applications, short comings in PCA Session 10: Neuro-Fuzzy Modeling: Fuzzy neural networks, Neuro-Fuzzy Control, designing neuro-fuzzy systems,

EEX7244 Data Mining

Level	7
Course Code	EEX7244
Course Title	Data Mining
Credit Value	2
Core/ Optional	Optional (Computer)
Course Aim/s	To equip the students with different Data mining algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data.
Course Learning Outcomes (CLO)	At the completion of this course students will be able to: CLO1: Describe and apply basic concepts and techniques of data mining. CLO2: Apply different methods to extract processed information from large amounts of data, both in theory and in practical applications. CLO3: Apply the learnt methods with the use of recent data mining software tools for a given scenario/dataset. CLO4:- Extract and present new information and insights with respect to the analysed scenario/data set.

Content (Main topics, subtopics)	**Outline Syllabus:** Session 01: Introduction to Data Mining - What is data mining?, Related technologies - Machine Learning, DBMS, OLAP, Statistics, Session 02: Basics of Data mining - Data Mining Goals, Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods Session 03: Data pre-processing - Data cleaning, Data transformation, Data reduction, Discretization and generating concept hierarchies Session 04: Introduction to Data mining System – Installing of the Data Mining System, Experiments with the system - filters, data cleaning Session 05: Data mining knowledge representation - Task relevant data, Background knowledge, Interestingness measures, Representing input data and output knowledge, Visualization techniques Session 06: Attribute-oriented analysis - Attribute generalization, Attribute relevance, Class comparison, Statistical measures, use of filters Session 07: Data mining algorithms- Association rules - Motivation and terminology, generating item sets and rules efficiently, Correlation analysis, mining association rules Session 08: Data mining algorithms: Classification - Basic learning/mining tasks, inferring rudimentary rules: 1R algorithm, Decision trees, Covering rules, Implementation with a DM software Session 09: Data mining algorithms: Prediction - Prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbour), Linear models, Implementation with a DM software Session 10: Clustering- Introduction to clustering, Partitioning methods: k-means, expectation maximization (EM), Hierarchical methods: distance-based agglomerative and divisible clustering, Implementation with a DM software Session 11: Clustering different source data - Clustering streaming data, Clustering graph data and network data, Constraint-based clustering and semi-supervised clustering, Application examples of cluster analysis Session 12: Mining real data - Pre-processing data from a real domain, applying various data mining techniques to create a comprehensive and accurate model of the data.

EEX7337 System Design in Groups

Level	7
Course Code	EEX7337
Course Title	System Design in Groups
Credit value	3
Core/Optional	Core (Computer)
Course Aim/s	Design and develop a solution for a given problem as a team.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Apply systems engineering principles throughout a computer system's life cycle, including important trade-offs to design a target application. CLO2: Reflect capabilities of project management to support development of computer systems including interdisciplinary issues. CLO3: Develop user experience design for a target computer system using appropriate tools. CLO4: Assess the design for manufacturability, sustainability, and maintainability considering levels of risk, dependability, safety, fault tolerance and impacts on society, economy and environment throughout the system's life cycle.

	CLO5: Demonstrate ability to model, simulate, prototype the design using appropriate tools and methods, and communicate the design to the client and the public.

Content (Main topics, sub topics)	1. Hardware and Software Processes (3) o Explain the need for a disciplined approach to system development and the elements of a disciplined approach in specific contexts. o Describe the nature of a life cycle, the role of life cycle models, quality in relation to the life cycle, the influence of system nature, and the size on choice of life cycle model. o Describe some common software and hardware development models and show how to use these models during the development of a computer-based system. o Explain how to gather data to inform, assess, and improve system design processes. o Describe the benefits of agile methods for hardware and software design. o Discuss the importance of modular design processes, and the design for modularity and reuse in the development of a computer-based system. o Select, with justification, system development models most appropriate for the development and maintenance of diverse computer-based systems. 2. Requirements analysis and elicitation (2) o Perform an analysis of a proposed computer-based system design project, including identification of need, information gathering, problem definition, feasibility considerations, and economic considerations. o Articulate a range of functional and non-functional requirements that might be applicable to the design of computer-based systems for a range of applications and discuss how requirements can change as a system design project evolves. o Discuss how trade-offs between different system requirements might be necessary for a proposed computer-based system design. o Describe the strengths and weaknesses of different approaches to requirements elicitation and capture. o Apply one or more techniques for elicitation and analysis to produce a set of requirements for a medium-size computer-based system. o Describe some quality factors for measuring the ability of a system design to meet requirements o Conduct a review of a computer-based system requirements document using best practices to determine the document's quality. 3. System specifications (2) o Discuss the relationship and differences between system specifications and requirements. o Articulate some typical functional and non-functional specifications for the design of a computer-based system and the importance of specifications to the design process. o Discuss one or more approaches for deriving system specifications from a requirements document. o Discuss how trade-offs between different system specifications might be necessary to meet system requirements. o Assess the quality of a given specification, considering such factors as completeness, consistency, simplicity, verifiability, basis for design, specification in the event of failure, and degraded modes of operation.

- Given a set of requirements, create a high-quality specification for a computer-based system of medium complexity.
- Create a test plan, based on the specification, considering the role of independence in relation to test, safety cases, and limitations of such tests.
- System architectural design and evaluation

4. System architectural design and evaluation (4)
   - Describe concepts and principles of system architecture design, such as top-down design, subdivision into systems and subsystems, modularity and reuse, the hardware/software interface, and trade-offs between various design options.
   - Describe strengths and weaknesses of various systems-level architectural design methods, including procedural and functional methods.
   - Describe design methods to meet system specifications and achieve performance measures, including dependability and safety.
   - Given a system specification, select an appropriate design methodology (e.g., structured design or modular design) and create an architectural design for a medium-size computer-based system.
   - Demonstrate ability to model, simulate, and prototype a range of computer-based system architectures.
   - Using appropriate guidelines, conduct the review of one or more computer-based system designs to evaluate design quality based on key design principles and concepts.
   - To include in different sessions

5. Concurrent hardware and software design (3)
   - Recognize the potential of hardware-software co-design in circumstances in which this approach is pertinent.
   - Discuss how particular design constraints can make the coordinated development of both hardware and software important, such as in the design of low-power systems, real-time systems, or systems with high-performance requirements.
   - Apply hardware-software co-design principles in situations of modest complexity.
   - Discuss challenges to effective hardware-software co-design, such as demands of hard real-time features.
   - Demonstrate ability to co-design to achieve specific technical objectives, such as low power, real-time operation, and high performance.
   - Select and apply computer-aided tools to support hardware and software co-design.

6. System integration, testing and validation (3)
   - Recognize the range of testing and validation methods appropriate for each stage of the system life cycle, including review of hardware models and software code; white box, black box, and regression testing; stress testing; and interface testing.
   - Describe the role of various system validation tools and show how tools can support efficient and effective development.
   - Discuss approaches to testing and validation at the unit level and at the integration and system levels.
   - Create a test plan and generate test cases for a computer-based system of medium complexity, selecting an appropriate combination of tests for ensuring system quality.
   - Demonstrate the application of the different types and levels of testing (unit, integration, systems, and acceptance) on computer-based systems of medium size.

	○ Undertake, as part of a team activity, an inspection of a medium-size computer-based system design. ○ Discuss methods used for manufacturing test and inspection, and acceptance testing. ○ Discuss methods for specialized testing: security, dependability/fault tolerance, and usability. 7. Maintainability, sustainability, manufacturability (2) ○ Describe the need for, and characteristics of, maintainable software, hardware, and system designs. ○ Discuss the inevitability of maintenance in certain systems, such as diagnosis, defect removal, hardware and/or software upgrades, and enhancement. ○ Describe how to apply principles of maintainable design to a computer-based system of modest complexity. ○ Identify issues associated with system evolution and explain their impact on the system life cycle. ○ Explain configuration management and version control in engineering systems—the need for it, the issues associated with it, the nature of the information to be held, legal requirements, and planning for possible disasters. ○ Develop a plan for reengineering a medium-size product in response to a change request. ○ Identify and exploit opportunities for component reuse in a variety of contexts. ○ Discuss how design decisions can affect future generations, including impact on the environment and energy resources, and disposal of systems and components at end of life. ○ Discuss design for manufacturability, part selection and standardization, manufacturing cost, and product lead-time for delivery. ○ Explain the importance and influence of standards, guidelines, legislation, regulations, and professional issues on the development of computer-based systems. ○ Describe tradeoffs that occur in following regulatory standards and regulations. **Design project** **Progress i.** Prepare a project plan for a computer-system design project that includes estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management—this could be done in the context of a class project or assignment. **Progress ii.** Perform an analysis of a proposed computer-based system design project, including identification of need, information gathering, problem definition, feasibility considerations, and economic considerations. **Progress iii.** Conduct a review of a computer-based system requirements document using best practices to determine the document's quality. Design, prototype, and conduct a usability test of a simple 2D GUI, using a provided GUI-builder, and, in doing so, create an appropriate usability test plan. **Progress iv.** Create a test plan and generate test cases for a computer-based system of medium complexity, selecting an appropriate combination of tests for ensuring system quality. **Progress v.** Model reliability, availability, and maintainability of designed computer-based systems. Perform a risk analysis of a medium-size computer-based system.

EEX7340 AI Techniques and Agent Technology

Level	7
Course Code	EEX7340
Course Title	AI Techniques and Agent Technology
Credit value	3
Core/Optional	Optional (Computer)
Course Aim/s	To equip students with concepts of knowledge representation and reasoning in AI and develop solution mechanisms for real world problems using different applications of AI.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Apply suitable knowledge representation mechanism for different scenarios CLO2: Evaluate the applicability of different reasoning mechanisms in a given situation. CLO3: Select an appropriate searching method to design an algorithm CLO4: Implement a solution in Prolog to solve a given problem CLO5: Build an expert system for an identified scenario CLO6: Design and implement simple agent based solutions for real world problems
Content (Main topics, subtopics)	**Outline Syllabus:** Unit 01 Introduction and KRR mechanisms Session 1: Introduction to AI –History, development, different thought schools, Session 2: Knowledge Representation – rule-based systems, frames, semantic nets, scripts Session 3: Propositional logic for knowledge representation – logical operators, truth tables, representing natural language sentences in propositional logic Session 4: Predicate logic for knowledge representation – quantifiers, representing natural language sentences in predicate logic Session 5: Resolution in predicate logic – conversion of predicate logic to CNF, principle of resolution, unification Session 6: Foundations of reasoning mechanisms – deductive, abductive, inductive, case based, analogical Session 7: Basic search methods – data driven/goal driven, depth first, breadth first etc, and applications Session 8: Heuristic search methods – hill climbing, A*, best first, beam search Session 9: Constraint satisfaction- forward checking, identifying most constrained variable, heuristic repair Unit 02 Introduction to AI languages Session 10: Introduction to AI languages – Prolog, Lisp, Java Session 11: Basic operations in prolog –facts, queries, writing rules, recursions Session 12: Data structures and file handling- data objects, lists, operators, backtracking, implementations Unit 03 AI Techniques Session 13: Expert Systems – representation, development methodologies and tools, inference engine, uncertainty handling, knowledgebase Session 14: Game Playing – minimax algorithms, Alpha-beta cutoffs, reference on specific games Session 15: Intelligent Agents – introduction, properties, classifications, architectures Graphs - Representation of graphs

	Session 16: Multi Agent Systems – introductions, properties, development Session 17: Multi Agent system development – introduction to different tool kits, JaDE, MadKit, practical developments of mini real world Session 18: Current trends in AI – emerging areas, major break throughs, future directions **Laboratory work:** 1. Practice to use basics of prolog language for given problems 2. demonstrate the use of expert system development tool kits for given scenarios 3. demonstrate the use of Agent development tool kits for given scenarios **Mini Project:** Select appropriate AI techniques and tools to solve a given problem

EEX7436 Processor Design

Level	7
Course Code	EEX7436
Course Title	Processor Design
Credit value	4
Core/Optional	Core (Computer), Elective (Electronic and Communication)
Course Aim/s	To provide design principles, techniques and hands-on experience on design simulation and construction of digital hardware and processors.
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Design an instruction set by analyzing the requirements of a targeted application. CLO2: Design main components of a processor necessary for the data path design. CLO3: Construct the processor by interconnecting previously designed components. CLO4: Evaluate the processor implemented in VHDL for its cost and performance. CLO5: Perform the implementation of the processor on FPGAs using hardware/software tools for testing.
Content (Main topics, sub topics)	**Outline Syllabus:** • Fundamentals of Computer design: Task of a computer designer, Technology, Cost, Performance, Design methodologies, design levels, design cycle, ethics • Processor design: Instruction set design, ALU design, and arithmetic processors. Control design. • VLSI design: Behavioural design & system specification; VHDL, modelling simulation. Logic design, schematic capture, synthesis; Implementation in PLD's. • Security & safety • Performance & cost analyzing • Relevant tools, standards and/or engineering constraints:

	○ Describe and contrast two hardware description languages, such as VHDL and Verilog, and identify tools to simulate computer system at different levels of design abstraction: system, ISP, register-transfer (RTL), and gate level. ○ Discuss the effect of a processor's arithmetic unit on its overall performance. • Computer arithmetic: ○ Discuss the effect of a processor's arithmetic unit on its overall performance. • Processor organization: ○ Design a data path and a hard-wired control unit for a simple instruction set architecture. ○ Design arithmetic units for multiplication, division, and floating-point arithmetic. **Laboratory Work:** Synthesize digital circuits on FPGA 1. implement digital circuits based on digital components using VHDL and implement them on FPGAs 2. implement circuits to access memory in FPGA 3. implement the designed processor/digital system on FPGA **Design Project:** Design, implement and analyse an Application Specific Processor/Digital system

EEY4181 Group Project (Computer Engineering)

Level	4
Course Code	EEY4181
Course Title	Group Project (Computer Engineering)
Credit value	1
Core/Optional	Core
Course Aim/s	Aim of this course is to instil in learners the ability to investigate problems and/or issues by following a methodical and a shared approach, to discover the required areas of knowledge in providing sustainable solutions.
Course Learning Outcomes (CLO):	CLO1: To apply concepts and principles in a related area of study; analyse information and suggest solutions to selected problems. CLO2: To communicate successfully, the results of analysis and arguments to specialist and non-specialist audiences. CLO3: To exercise responsibilities as an individual and as a team. CLO4: To display qualities and transferable skills as well as subject specific skills necessary for employment, carry out further training and to manage their own learning.
Content	The content is based on prior learning and information researched.

EEY7881 Engineering Research Project (Computer Engineering)

Level	7
Course Code	EEY7881
Course Title	Engineering Research Project (Computer Engineering)
Credit value	8
Core/Optional	Core
Course Aim/s	Aim of this course is to carry out an industry-based project during the final year, in order to demonstrate learners' exposure to professional engineering practice. It should also demand individual analysis and judgement, assessed independently from the work of others. Learners are encouraged to undertake Projects in their main discipline or to undertake projects of inter-disciplinary nature.
Course Learning Outcomes (CLO):	CLO1: To engage with selected information in the research literature to construct new knowledge related to the Research Question, the learner plans to investigate. CLO2: To plan out the investigation of a complex engineering problem using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions. CLO3: To display qualities and transferable skills as well as subject specific skills necessary to communicate successfully, to a specialist audience. CLO4: To construct/create/make/build knowledge based on established pre-knowledge and techniques, using the Scientific Method. CLO5: To apply suitable/recognised methods, tools and procedures when executing the work. CLO6: To demonstrate individual analysis and judgement based on reflective learning (Concrete Experience, Reflective Observations, Abstract Conceptualisation, and Active Experimentation). CLO7:To verify accuracy and relevance of the proposed methodology and research findings. CLO8: To design and develop suitable concepts and models (graphical, mathematical, statistical, prototype) to analyse, interpret and communicate research findings. CLO9: To disseminate research findings in a professional and ethical manner.
Content	The content is based on prior learning and information researched.

EEM6201 Professional Practice

Level	6
Course Code	EEM6236
Course Title	Professional Practice
Credit value	2
Core/Optional	Core
Course Aim/s	To apply knowledge and skills within a practical environment considering professional, ethical, legal, security, environmental, social and cultural issues
Course Learning Outcomes (CLO):	At the completion of this course student will be able to: CLO1: Describe professional, ethical, legal, security, environmental, social and cultural issues and responsibilities. CLO2: Analyse the local and global impact of computing on individuals, organizations, and society. CLO3: Apply current technical concepts and practices in the core information technologies.

Content **(Main topics, sub topics)**	**Outline Syllabus:** Session 1: Basics of engineering solutions and societal effects Session 2: Advanced concepts of engineering solutions and societal effects Session 3: Philosophical frameworks and cultural issues Session 4: Cultural issues of professional practice Session 5: Intellectual property Session 6: Legal practices related to professional practice Session 7: Business and management practices Session 8: Professional practice Session 9: Ethical responsibilities Session 10: Social and global trends in technology Session 11: Contemporary issues