TECHNO INDIA UNIVERSITY

WEST BENGAL

Syllabus

for

4-Years B.Tech.

in

Computer Science and Engineering
(Specialization in Artificial Intelligence)

Academic Year: 2024-2025

Semester 4

Mathematics IV (TIU-UMA-T206)

Program: B. Tech. in CSE, CSE AI	**Year, Semester:** 2nd Yr., 4th Sem.
Course Title: Mathematics IV	**Subject Code:** TIU-UMA-T206
Contact Hours/Week: 3–1–0 (L–T–P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. learn the concept of probability, random variables, probability distribution.
2. obtain ideas about interpolation, numerical integration, and differentiation.
3. know the methods for solving simultaneous linear algebraic equations and differential equations.

COURSE OUTCOME:

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

CO-1:	apply concept of probability and calculate the probability of random events	K4
CO-2:	apply probability distributions to solve simple problems	K3
CO-3:	have an understanding of the concept of error in numerical analysis	K2
CO-4:	develop an idea about different interpolation methods	K4
CO-5:	establish the concepts of numerical differentiation and integration	K4
CO-6:	apply various numerical methods for solving different types of problem such as simultaneous equations, algebraic, transcendental, and differential equations.	K4

COURSE CONTENT:

MODULE 1:	**PROBABILITY**	**20 Hours**
Classical, relative frequency and axiomatic definitions of probability, mutually exclusive events, independent events, conditional probability, Bayes' Theorem. Random Variables - Discrete and continuous random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments. Distributions - Uniform, Binomial, Geometric, Poisson, Exponential, Gamma, Normal distributions. Functions of a random variable.		
MODULE 2:	**NUMERICAL TECHNIQUES**	**25 Hours**
Approximations and round off errors, Truncation errors and Taylor Series. Interpolation – Newton's Forward, Backward, Lagrange Interpolation methods Numerical Differentiation, Numerical Integration – Trapezoidal, Simpson's 1/3rd rules		

Determination of roots of polynomials and transcendental equations by Bisection, Iteration, Newton-Raphson, Regula-Falsi methods. Solutions of linear simultaneous linear algebraic equations by Gauss Elimination and Gauss-Seidel iteration methods. Numerical solution of initial value problems by Euler, Modified Euler, Runge-Kutta 4th order.	
TOTAL LECTURES	**45 Hours**

Books:

1. Erwin Kreyszic-Advanced Engg. Mathematics
2. S. S. Sastry-An Introduction to Numerical Analysis.
3. Dutta and Jana- Numerical Analysis.
4. S. A. Mollah- Numerical Analysis and Computational Procedures
5. Probability and Statistics for Scientists and Engineers by Sheldon Ross

Computer Architecture (TIU-UCS-T220)

Program: B.Tech in CSE-AI	**Year, Semester:** 2nd.,4th
Course Title: Computer Architecture	**Subject Code:** TIU-UCS-T220
Contact Hours/Week: 3-0-0 (L-T-P)	**Credit:** 3

COURSE OBJECTIVES:

1. To conceptualize the basics of organizational and architectural issues of a digital computer.
2. To analyze performance issues in processor and memory design of a digital computer.
3. To understand mapping techniques of different memory organization in digital computer.
4. To analyze processor performance improvement using instruction level parallelism.

COURSE OUTCOMES:

The students will be able to:

CO 1:	Describe the fundamental working principles of Basic Computer System architecture.	K1
CO 2:	Identify the design of Pipeline Architecture.	K2
CO 3:	Explain the memory hierarchy design and perform memory mapping operations.	K3
CO 4:	Identify the design of a pipelined CPU and hypothetical parallel processor architecture. VLIW and Superscalar Architecture.	K4
CO 5:	Explain the hardware design of multiprocessors systems and its CPU-Memory interconnection architecture.	K2
CO 6:	Non-Von Neumann Architecture.	K2

COURSE CONTENT:

MODULE 1:	**Overview of von Neumann architecture**	**5 Hours**
Instruction set architecture; The Arithmetic and Logic Unit, The Control Unit, Memory and I/O devices and their interfacing to the CPU; Measuring and reporting performance; CISC and RISC		

processors.		
MODULE 2:	**Pipelining**	**12 Hours**
Basic concepts of pipelining, data hazards, control hazards, and structural hazards;Techniques for overcoming or reducing the effects of various hazards.		
MODULE 3:	**Hierarchical Memory Technology**	**10 Hours**
Inclusion, Coherence and locality properties; Cache memoryorganizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies.		
MODULE 4:	**Instruction-level parallelism**	**10 Hours**
Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP; Superscalar, super-pipelined and VLIW processor architectures; Vector and Array processors.		
MODULE 5:	**Multiprocessor Architecture**	**6 Hours**
Taxonomy of parallel architectures; Centralized shared-memoryarchitecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers.		
MODULE 6:	**Non von Neumann Architectures**	**2 Hours**
Data flow Computers, Reduction computer architectures, Systolic Architectures.		
	TOTAL LECTURES	**45 Hours**

Books:

1. M Morris Mano, Computer System Architecture (Revised 3rd Edition), Pearson.
2. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
3. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
4. John Paul Shen and Mikko H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill.
5. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.

Graph Theory and Combinatorics (TIU-UCS-T224)

Program: B. Tech. in CSE-AI	**Year, Semester:** 4th., 7th
Course Title: Graph Theory and Combinatorics	**Subject Code:** TIU-UCS-T224
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE :

Enable the student to:

1. Introduce students to the fundamental concepts of graph theory, including different types of graphs, graph representations, and essential properties.
2. Enable students to identify and mathematically formulate problems related to graph theory.
3. Teach students to analyze and apply graph-theoretic algorithms for solving problems.
4. Develop an understanding of combinatorial principles, including permutations, combinations, and their applications in problem-solving.

COURSE OUTCOME:

The student will be able to:

CO1:	Describe the concepts of graphs and related tools and techniques.	K2
CO2:	Identify and formulate problems which are related to graph theory and combinatorics.	K3
CO3:	Analyze problems and apply graph-theoretic techniques for problem-solving.	K3
CO4:	Describe the principles of combinatorics and apply them for problem-solving.	K3
CO5:	Perform combinatorial analysis of problems of diverse nature.	K4
CO6:	Develop efficient algorithms and models using graph theory and combinatorial techniques to solve real-world problems.	K4

COURSE CONTENT:

MODULE 1:	**FUNDAMENTALS OF GRAPH THEORY**	**9 Hours**
Binary relation on set, concepts and definition of graph; undirected, directed, weighted graphs, and simple graphs; complete graphs, sparse graphs, and dense graphs. adjacency of vertices and degree of vertices; relationship between number of vertices and number of edges; incidence between vertices and edges.		
MODULE 2:	**GRAPH ALGORITHMS AND APPLICATIONS**	**10 Hours**
Operations on graph; graph isomorphism; Path connectivity and connected components in a graph, properties and computation. Euler's Tour, Hamiltonian path, Topological sorting, Traveling salesman's problem. strongly connected graph and components. representations of a graph in computer; graph exploration techniques: Breadth-first search (BFS) and Depth-first search (DFS) and their applications.		
MODULE 3:	**SPANNING TREES, SHORTEST PATHS, AND GRAPH PROPERTIES**	**9 Hours**
Spanning tree of a graph, minimum spanning tree (MST) of a weighted graph, its properties and computation (Kruskal's and Prim's algorithms). Concepts of shorted paths in a graph, their properties and computation. Concepts of planar graphs and their properties; dual graph of a planar graph; bipartite graphs; tree as a graph and its properties; graph coloring.		

MODULE 4:	COMBINATORICS AND COUNTING PRINCIPLES	9 Hours
Recapitulation of concepts from previous discrete mathematics course; the rules of sum and product; permutations, combinations, selection; The pigeonhole principle and its variants; The Binomial theorem, combinations with repetition, the Catalon numbers; The multinomial theorem. Combinatorial problems in discrete probability.		
MODULE 5:	**ADVANCED COUNTING TECHNIQUES AND RECURRENCE RELATIONS**	**6 Hours**
Principles of inclusion and exclusion, and its variants; derangements; generating functions; recurrence relations. Problem solving with these principles.		
TOTAL LECTURES		**45 Hours**

Books:

1. "Graph Theory with Applications" – J.A. Bondy and U.S.R. Murty
2. "Introduction to Graph Theory" – Douglas B. West
3. "Graph Theory" – Reinhard Diestel.
4. "Graphs, Networks, and Algorithms" – Dieter Jungnickel

Object Oriented Programming (TIU-UCS-T214)

Program: B. Tech. in CSE-AI	**Year, Semester:** 2nd, 4th
Course Title: Object Oriented Programming	**Subject Code:** TIU-UCS-T214
Contact Hours/Week: 3–0–2 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. understand the real-world problem and design solutions by object oriented programming
2. analyze the nature of problems solved with object oriented techniques
3. design and implement suitable programming approach for a given application

COURSE OUTCOME :

The students will be able to:

CO-1:	Define an object-oriented approach to programming and identify potential benefits of object-oriented programming over other approaches	K2
CO-2	Understand the difference between the top-down and bottom-up program design approach	K2
CO-3:	Demonstrate the use of various OOPs concepts using C++	K3
CO-4:	Solve a computational problem by implementing the solution as a real-world entity	K4
CO-5:	Understand and apply some advanced constructs of C++ like virtual functions, operator overloading, exception handling, standard template library	K3

CO-6:	Analyze and optimize C++ programs for efficiency, maintainability, and scalability using OOP best practices.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**10 Hours**
Introduction to Object Oriented Paradigm, Need of object-oriented design, Drawbacks of Procedure Oriented Programming, Features of object-oriented languages, POP Vs OOP, Benefits & Applications of OOP, Difference between C and C++.		
MODULE 2:	**BASIC CONCEPTS OF OBJECT ORIENTATION**	**12 Hours**
Class, Object, Data abstraction, Encapsulation, Inheritance, Polymorphism, Message Passing, Dynamic binding.		
MODULE 3:	**FUNDAMENTALS OF OOPs**	**12 Hours**
Fundamentals of OOPs: Class & Objects, Constructors & Destructor. Different perspectives on inheritance, Types of inheritance, Polymorphism: Compile Time & Run time Polymorphism, Virtual functions, Virtual table construction, Overloading, Overriding, Abstract Class, Virtual Class.		
MODULE 4:	**ADVANCE OOP CONCEPTS**	**11 Hours**
Class and Function Templates, Standard Template Libraries in C++: lists, vectors, sets, maps. Exceptions Handling.		
	TOTAL LECTURES	**45 Hours**

Books:

1. Robert Lafore, Object-Oriented Programming in C++, Fourth Edition, Pearson.
2. Herbert Schildt, C++: The Complete Reference, Fourth Edition, Mc-Graw Hill Education, India, 2003
3. Bjarne Stroustrup, The C++ Programming Language, Third Edition, Pearson, 2000.
4. E. Balagurusamy, Object-Oriented Programming with C++, 8th Edition, Mc-Graw Hill Education 2021
5. Scott Meyers, Effective Modern C++, O'Reilly Media, Inc., 2014.
6. Scott Meyers, Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library, Addison-Wesley Professional Computing Series,2001.

Microprocessor & Microcontroller (TIU-UEC-T210)

Program: B. Tech. in CSE-AI	**Year, Semester:** 2nd, 4th
Course Title: Microprocessor & Microcontroller	**Subject Code:** TIU-UEC-T210
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE :

Enable the student to:

1. Learn about the fundamentals of Microprocessor and Microcontroller Systems and its Components. They should be able to write and understand the assembly language program

by applying the concept of 8085 and 8086 microprocessors' architecture, interrupts, instructions sets etc.

2. Understand understand the working principle, pin diagram, memory interfacing, instructions of 8051 microcontroller so that they can design application specific systems.
3. Design and analyse the microprocessor based system they need to understand the operations of different peripheral devices such as 8255, 8257/8237A etc. and corresponding interfacing circuits.
4. Understand measurement and control techniques of different electrical, physical quantities by designing and analysing processor based systems.

COURSE OUTCOME :

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

CO-1:	Understand the fundamentals of microprocessors and microcontrollers	K2
CO-2:	Develop assembly language programming skills for 8085 microprocessor	K2
CO-3:	Analyze and design interfacing techniques for peripheral devices	K4
CO-4:	Demonstrate knowledge of 8086 microprocessor architecture and programming	K2
CO-5:	Demonstrate knowledge of 8051 microcontroller architecture and programming	K2
CO-6:	Apply microprocessor and microcontroller concepts to measurement and control applications	K3

COURSE CONTENT :

MODULE 1:	**INTRODUCTION TO 8085 (8 BIT PROCESSOR)**	**15 Hours**
Introduction to 8085 (8 bit processor): Evolution of microprocessors and microcontrollers, CPU architecture-register organization, pin description and features, addressing modes, Instruction set and Assembly Language Programming. Instruction cycle, machine cycle, Timing diagram, stack and subroutine operation.		
MODULE 2:	**HARDWARE INTERFACING AND PERIPHERALS**	**10 Hours**
Hardware Interfacing and Peripherals: Interfacing memory, IO devices (IO mapped IO& Memory mapped IO). Interrupts, Interfacing 8255 peripheral, DMA controller (8237).		
MODULE 3:	**INTRODUCTION TO 8086 (16 BIT PROCESSOR)**	**10 Hours**
Introduction to 8086 (16 bit processor): Architecture, register organization, pin description and features. Instruction Set and Addressing modes, interfacing memory, IO devices, Interrupts.		
MODULE 4:	**INTRODUCTION TO 8051 (8 BIT MICROCONTROLLER)**	**10 Hours**
Introduction to 8051 (8 bit microcontroller): Architecture, Pin description and features, Special function registers, I/O pin ports and circuits, Internal, External memories, addressing modes, Assembly Language Programming.		
	TOTAL LECTURES	**45 Hours**

Books:

1. R. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085”, Penram.
2. B.Ram, “Fundamentals of Microprocessors and Microcontrollers” , Dhanpat Rai Publication.
3. D. V. Hall, “Microprocessors and Interfacing”, McGraw Hill
4. K. M. Bhurchandi and A. K. Ray, “Advanced Microprocessors and Peripherals”, Tata McGraw Hill
5. M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, “The 8051 Microcontroller and Embedded Systems”, Pearson
6. K. Ayala, “The 8051 Microcontroller”, Delmar Cengage Learning

Numerical Lab (TIU-UCS-L216)

Program: B.Tech. in CSE-AI	**Year, Semester:** 2nd, 4th.
Course Title: Numerical Lab	**Subject Code:** TIU-UCS-L216
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. apply numerical techniques to approximate functions and find the area under them
2. use numerical methods to find approximate roots of equation, find solutions to systems of equations, solve differential equations etc.
3. use computational tools to implement the numerical methods

COURSE OUTCOME :

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

CO-1:	approximate a function by a linear polynomial	K4
CO-2:	calculate the approximate the area under a curve	K4
CO-3:	use numerical techniques to find the approximate solution of alegebraic and transcendental equations	K4
CO-4:	find the exact solution of a system of equations by direct methods	K4
CO-5:	find the approximate solution of a system of equations by indirect methods	K4
CO-6:	find the numerical solution for initial value problems (IVPs)	K4

COURSE CONTENT :

MODULE 1:	**FORWARD AND BACKWARD DIFFERENCE TABLE**	**9 Hours**
Introduction to finite differences and their applications, Forward and Backward Difference Operators, Construction of Difference Tables, Higher-order differences and accuracy analysis		
MODULE 2:	**INTERPOLATION METHODS**	**9 Hours**
Newton’s Forward and Backward Interpolation Formulas, Lagrange’s Interpolation Method, Applications of interpolation in numerical computations		

MODULE 3:	**NUMERICAL INTEGRATION**	**6 Hours**
Introduction to Numerical Integration, Trapezoidal Rule and its implementation, Simpson's 1/3rd Rule and applications, Error analysis in numerical integration		
MODULE 4:	**NUMERICAL SOLUTION OF ALGEBRAIC AND TRANSCENDENTAL EQUATIONS**	**6 Hours**
Bisection Method: Concept, Algorithm, and Implementation, Newton-Raphson Method: Convergence, Limitations, and Applications		
MODULE 5:	**NUMERICAL SOLUTION OF A SYSTEM OF EQUATIONS**	**6 Hours**
Introduction to solving systems of linear equations, Gauss Elimination Method: Direct approach to solving equations, Gauss-Seidel Iterative Method: Convergence and Stopping Criteria		
MODULE 6:	**NUMERICAL SOLUTION OF INITIAL VALUE PROBLEMS**	**9 Hours**
Euler's Method: Concept and Implementation, Modified Euler's Method for higher accuracy, Runge-Kutta Methods: 2nd and 4th Order Techniques, Applications in solving Ordinary Differential Equations (ODEs).		
TOTAL LAB HOURS		**45 Hours**

Books:

1. Balagurusamy, E. (2017). Numerical Methods (1st ed.). McGraw-Hill Education.
2. Veerarajan, T., & Ramachandran, T. (2006). Numerical Methods with Programs in C and C++ (1st ed.). Tata McGraw-Hill.
3. Chapra, S. C. (2018). Applied Numerical Methods with MATLAB for Engineers and Scientists (4th ed.). McGraw-Hill Education.
4. Pradeep, N., & Govindarajan, G. (2008). Numerical Methods and Computer Programming (1st ed.). New Age International Publishers.
5. Grewal, B. S. (2019). Numerical Methods in Engineering and Science with Programs in C and C++ (10th ed.). Khanna Publishers.
6. Rajaraman, V. (2012). Computer Oriented Numerical Methods (3rd ed.). PHI Learning Pvt. Ltd.

Computer Architecture Lab (TIU-UCS-L220)

Program: B.Tech. in CSE-AI	**Year, Semester: 2**nd, 4th.
Course Title: Computer Architecture Lab	**Subject Code:** TIU-UCS-L220
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. Understand the architectural principles of computer systems and their organization.

2. Implement and analyze fundamental hardware components.
3. Develop memory and arithmetic logic unit (ALU) designs for computing applications.

COURSE OUTCOME :

Upon completion of the course, the student will be able to:

CO-1	Explain the fundamental aspects of computer architecture design and analysis.	K3
CO-2	Demonstrate a comprehensive understanding of basic hardware concepts with Integrated Circuits	K3
CO-3	Illustrate the operations of different gates and their applications	K3
CO-4	Implement hardware designs and apply them in various domains.	K4
CO-5	Design the memory unit of a computer system.	K4
CO-6	Develop the arithmetic logic unit of a computer system.	K3

COURSE CONTENT:

MODULE 1:	**BASIC LOGIC FUNDAMENTALS**	**9 Hours**
Boolean algebra, Logic gates, Truth tables, K-map simplification, Implementation of logic functions using gates		
MODULE 2:	**COMBINATIONAL CIRCUIT DESIGN**	**9 Hours**
Design and implementation of Adders, Subtractors, Multiplexers, Decoders, and Encoders using logic gates		
MODULE 3:	**SEQUENTIAL CIRCUITS AND STATE MACHINES**	**6 Hours**
Flip-Flops (SR, D, JK, T), Registers, Counters (Synchronous and Asynchronous), Finite State Machines		
MODULE 4:	**MEMORY ORGANIZATION AND ADDRESSING**	**6 Hours**
RAM and ROM architectures, Cache memory design, Memory hierarchy, Address decoding		
MODULE 5:	**ARITHMETIC LOGIC UNIT (ALU) AND PROCESSOR DESIGN**	**6 Hours**
Design of an ALU, Arithmetic operations (addition, subtraction, multiplication), Logical operations, Bitwise manipulations		
MODULE 6:	**CPU DESIGN AND PERFORMANCE OPTIMIZATION**	**9 Hours**
Instruction set architecture, Execution cycle, Control unit design, Pipelining, Performance evaluation metrics		
	TOTAL LAB HOURS	**45 Hours**

Books:

1. M. Morris Mano & Michael D. Ciletti - Digital Design (5th ed.), Pearson

2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky - Computer Organization and Embedded Systems (6th ed.), McGraw-Hill
3. William Stallings - Computer Organization and Architecture (10th ed.), Pearson
4. David A. Patterson & John L. Hennessy - Computer Organization and Design (5th ed.), Elsevier

Object Oriented Programming Lab (TIU-UCS-L214)

Program: B.Tech. in CSE-AI	**Year, Semester:** 2nd, 4th.
Course Title: Object Oriented Programming Lab	**Subject Code:** TIU-UCS-L214
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. Understand the mathematical foundations of various numerical methods and their significance in solving engineering and scientific problems.
2. Apply numerical techniques to solve algebraic equations, system equations, interpolation, differentiation, and integration using computational tools.

COURSE OUTCOME :

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

CO-1	Able to understand the foundational principles of object-oriented programming and its application in solving computational problems using C++.	K2
CO-2	Able to comprehend and construct algorithms and flowcharts for problem-solving in object-oriented programming contexts.	K3
CO-3	Able to apply object-oriented programming concepts like classes, inheritance, and polymorphism in software development using C++.	K3
CO-4	Able to design, implement, and test object-oriented programs to solve real-world scenarios using appropriate programming techniques.	K4
CO-5	Able to analyze and optimize object-oriented solutions for complex problems encountered in industrial applications and software development projects.	K4
CO-6	Evaluate and debug object-oriented programs to ensure robustness, efficiency, and adherence to best practices in software development using C++.	K4

COURSE CONTENT :

MODULE 1:	**C++ BASICS AND STRUCTURES**	**4 Hours**
Introduction to C++, Data types, Control Flow and loops, Mathematical Operations using loops and conditions, Introduction of Structure in C++, Implementing Structure in Real World Scenarios, Using Structure for practical applications		

MODULE 2:	**CLASSES AND OBJECTS, DATA STRUCTURES USING OBJECTS**	**7 Hours**
Introduction to Classes and Objects, Object-Oriented Program Design, Real-World Example: Toy-Train Ticket Counter, implementing a Vector Class, Introduction to Data Structures Using Objects, Implementing a Linked List and Stack, Practical Application and Optimization		
MODULE 3:	**CONSTRUCTORS, FUNCTIONS, ENUM AND STRING**	**9 Hours**
Introduction to Constructors and Destructors, different types of Constructors, Introduction to function, Different use cases of Function implementation, Enum and String, String Handling in C++		
MODULE 4:	**FRIEND FUNCTION, INLINE FUNCTION, POLYMORPHISM, VIRTUAL FUNCTION**	**9 Hours**
Implementation of Friend class and Friend Function, Introduction to Inline Function with implementation of inline mathematical operations, Implementation of Polymorphism, Run time and Compile time polymorphism, Function Overloading, Operator Overloading, Function Overriding, Runtime Polymorphism using Virtual Functions		
MODULE 5:	**INHERITANCE, ABSTRACT CLASSES, AND FILE HANDLING**	**8 Hours**
Implementation of Inheritance, Different types of Inheritance and its use cases,Abstract Classes and Pure Virtual Functions, Introduction to File Handling in C++, Reading and Writing Files		
MODULE 6:	**TEMPLATES, EXCEPTIONS, STANDARD TEMPLATE LIBRARY(STL) AND MULTI-FILE PROGRAMS: MINI PROJECT**	**8 Hours**
Introduction to Templates -Concept of generic programming, Function templates, Class templates, Exception Handling, Standard Template Library (STL), Introduction to Multi-File Programs and implementing a Mini Project		
	TOTAL LAB HOURS	**45 Hours**

Books:

1. *Bjarne Stroustrup(2013).* The C++ Programming Language(4th ed.) Addison-Wesley.
2. Herbert Schildt (2014). *C++: The Complete Reference* (4th ed.). McGraw-Hill.
3. *Bjarne Stroustrup(2014).* Programming: Principles and Practice Using C++(2nd Ed.) Addison-Wesley.
4. Paul Deitel & Harvey Deitel(2016). C++ How to Program(10th ed.). Pearson.
5. Stanley B. Lippman, Josée Lajoie, & Barbara E. Moo (2012). *C++ Primer* (5th ed.). Addison-Wesley.
6. Matt Weisfeld (2019). *The Object-Oriented Thought Process* (5th ed.). Addison-Wesley.

Microprocessor and Microcontroller Lab (TIU-UEC-L218)

Program: B.Tech. in ECE	**Year, Semester:** 2nd, 4th.

Course Title: Microprocessor and Microcontroller Lab	**Subject Code:** TIU-UEC-L218
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

5. Learn how to write, execute and debugassemble language programs (ALPs) for microprocessors.
6. Understand data manipulation, arithmetic, bitwise and logical operations for microprocessors.
7. Understand about memory addressing and data storage for microprocessors.
8. Understand the uses of looping, branching and conditional and unconditional jumps in ALPs to control execution flow for microprocessors.

COURSE OUTCOME :

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

CO-1	Understand microprocessor architecture, memory organization, and register operations and assembly language programming.	K2
CO-2	Perform data manipulation and bitwise operations for complement, shifting, masking, and mask-off of data.	K3
CO-3	Implement arithmetic and logical operations to perform addition, subtraction, multiplication and division.	K3
CO-4	Apply data conversion and encoding techniques for efficient data representation.	K3
CO-5	Develop and implement sorting and searching algorithms.	K4
CO-6	Learn how to store and retrieve data from specific memory locations, and debug and optimize assembly language programs for efficient execution.	K4

COURSE CONTENT :

MODULE 1:	**DATA MANIPULATION AND BITWISE OPERATIONS**	**6 Hours**
1's and 2's complement of input data, shifting or rotating of accumulator contents.		
MODULE 2:	**ARITHMETIC OPERATIONS**	**9 Hours**
Addition and subtraction of input data using control flow (loop), use of different arithmetic and branch control instructions.		
MODULE 3:	**DATA ENCODING AND MASKING**	**6 Hours**
Pack (mask) and unpack (mask off) of input data, checking even or odd input data, use of logical and branch control instructions.		
MODULE 4:	**DATA CONVERSION**	**9 Hours**
BCD to Binary conversion and vice-versa: use of arithmetic, logical and branch control instructions.		
MODULE 5:	**SORTING AND SEARCHING ALGORITHMS**	**9 Hours**

Searching and sorting of data from an array: handling of multiple memory locations for different data, use of branch control instructions.		
MODULE 6:	**ADVANCED ARITHMETIC OPERATIONS**	**9 Hours**
Multiplication and division of input data using control flow (loop), use of different arithmetic and branch control instructions, repetitive addition and subtraction methods.		
TOTAL LAB HOURS		**48 Hours**

Books:

1. R. Gaonkar, "Microprocessor Architecture, Programming, and Applications with the 8085",Penram
2. A. Nagoor Kani,"Microprocessor (8085) And its Applications" 2005, McGraw-Hill Education
3. D. V. Hall, "Microprocessors and Interfacing", McGraw Hill
4. M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, "The 8051 Microcontroller and Embedded Systems", Pearson

Career Advancement & Skill Development-IV (TIU-UEN-S298)

Program:B.Tech CSE-AI	**Year, Semester:** 2nd, 4th
Course Title: CAREER ADVANCEMENT & SKILL DEVELOPMENT-IV	**Subject Code:** TIU-UEN-S298
Contact Hours/Week: 2-0-0 (L-T-P)	**Credit:** 2

COURSE OBJECTIVE :

Enable the student to:

1. Strengthen and expand their communication skills in French.
2. Develop greater fluency in listening, speaking, reading, and writing.
3. Understand and apply more complex grammatical structures and vocabulary.
4. Express ideas in past and future tenses in conversations and written texts.

COURSE OUTCOME :

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

CO-1:	Recognize and use an expanded range of everyday expressions.	K1
CO-2:	Understand and respond to conversations involving personal experiences.	K2
CO-3:	Use the passé composé and imparfait to talk about past events.	K3
CO-4:	Appraise short narratives, dialogues, and informational texts.	K4
CO-5:	Compose personal letters and messages.	K5
CO-6:	Generate and sustain conversations about familiar topics.	K5

COURSE CONTENT :

MODULE 1:	**EXPRESSING PAST EVENTS**	**6 Hours**
Introduction to passé compose, Using auxiliary verbs "être" and "avoir", Talking about past experiences		

MODULE 2:	**DESCRIBING ROUTINES AND HABITS IN THE PAST**	**6 Hours**
Introduction to imparfait, Differences between passé composé and imparfait, Describing past habits and ongoing actions		
MODULE 3:	**FUTURE PLANS AND INTENTIONS**	**6 Hours**
Expressing future intentions with "aller + infinitive", Talking about upcoming events and travel plans, Making appointments and invitations		
MODULE 4:	**SHOPPING, SERVICES, AND TRANSACTIONS**	**6 Hours**
Asking for help and making purchases, Using expressions of quantity and price, Talking about preferences and needs		
MODULE 5:	**SOCIAL INTERACTIONS AND EVERYDAY SCENARIOS**	**6 Hours**
Making polite requests and giving advice, Expressing opinions and emotions, Writing informal letters and messages		
	TOTAL LECTURES	**30 Hours**

Books:

Tech French - French for Science and Technology, Goyal Publishers, 2011