Syllabus for 4 Year B. Tech Course in Computer Science and Engineering (Artificial Intelligence)

Fourth Semester

Career Advancement & Skill Development-IV-Communication Skill (TIU-UEN-S298)
Contact Hours/Week: 2–0–0 (L–T–P)
Credit: Sessional–2
Course Content
French:
The present tense of the verb faire and some professions in masculine and feminine and, their verbs in present tense and their conjunctions and text based on the same, the re verbs in present tense and the prepositions, the prepositions and the time and the daily activities, the verb pronominal and the expressions of frequency, the present tense of the verbs pouvoir, vouloir, devoir, the articles contracted ,the past tense, the near future and the future tense, the vocabulary related to vacations, food, shopping, clothes and visiting France.
The activities in the class are as follows:
Speaking about one's daily routine, saying the hour and time , speakind Nd writing about his or her shopping and recounting a vacation well spent, informing someone one's future projects and plans and activities and writing a letter from a various and cultural activities.

Mathematics-IV (TIU-UMA-T206)
Contact Hours/Week: 4–0–0 (L–T–P)
Credit: 4

Course Outcome

CO1	To understand concept of probability and apply them to real time problems
CO2	To learn discrete and continuous random variables and their applications
CO3	To introduce to probability distributions like binomial, poisson, normal etc.
CO4	To learn interpolation methods using difference calculus
CO5	To learn to perform numerical differentiation and integration
CO6	To study various numerical techniques to solve algebraic and transcendental equations, system of equations and differential equations

Course Content

Module-1: Probability
Unit 1: Classical, relative frequency and axiomatic definitions of probability, mutually exclusive events, Independent events, conditional probability, Bayes' Theorem.

Unit 2: Random Variables - Discrete and continuous random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments.
Unit 3: Distributions - Uniform, Binomial, Geometric, Poisson, Exponential, Gamma, Normal distributions. Functions of a random variable.
Unit 4: Joint and marginal distribution.

Module-2: Numerical Techniques
Unit 1: Approximations and round off errors, Truncation errors and Taylor Series.
Unit 2: Interpolation – Newton's Forward, Backward, Lagrange Interpolation methods
Unit 3: Numerical Differentiation, Numerical Integration – Trapezoidal, Simpson's $1/3^{rd}$ rules
Unit 4: Determination of roots of polynomials and transcendental equations by Bisection, Iteration, Newton-Raphson, Regula-Falsi methods.
Unit 5: Solutions of linear simultaneous linear algebraic equations by Gauss Elimination and Gauss-Seidel iteration methods.
Unit 6: Numerical solution of initial value problems by Euler, Modified Euler, Runge-Kutta and Predictor-Corrector methods.

Recommended Books:
Main Reading

1. Erwin Kreyszic-Advanced Engg. Mathematics – Eq. 5- Wiley Eastern
2. S. S. Sastry-An Introduction to Numerical Analysis.
3. Dutta and Jana- Numerical Analysis.
4. S. A. Mollah- Numerical Analysis and Computational Procedures

Computer Architecture (TIU-UCS-T220)
Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3
Prerequisite Course: Computer Organization (TIU-UCS-T207)

Course Outcome

CO1	Understand the concept of data representation, instruction formats and the operation of a digital computer
CO2	Illustrate the fixed point and floating-point arithmetic for ALU operation
CO3	Comprehend the concept of various memories, interfacing and organization of multiple processors
CO4	Implement parallel processing techniques and approaches
CO5	Understand about unconventional architectures and network topologies

Course Content

Module-1: Overview of von Neumann architecture

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

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

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

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

Taxonomy of parallel architectures; Centralized shared-memoryarchitecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers.

Module-6: Non von Neumann Architectures

Data flow Computers, Reduction computer architectures, Systolic Architectures.

Recommended Books:

Main Reading

1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.

Supplementary Reading

1. John Paul Shen and Mikko H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill.
2. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.

Graph Theory and Combinatorics (TIU-UCS-T224)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3

Course Outcome

CO1	Describe the concepts of graphs and related tools and techniques.
CO2	Identify and formulate problems which are related to graph theory and

	combinatorics.
CO3	Analyze problems and apply graph theoretic techniques for problem solving.
CO4	Describe the principles of combinatorics and apply them for problem solving.
CO5	Perform combintorial analysis of problems of diverse nature.

Course Content

Module 1:
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:
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 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:
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:
Principles of inclusion and exclusion, and its variants; derangements; generating functions; recurrence relations. Problem solving with these principles.

Recommended Books:

Main Reading

1. Narsingh Deo. Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall, Inc.

2. Douglas B. West. Introduction to Graph Theory. Second Edition. Pearson Education Inc., 2002.
3. John M. Harris, Jeffry L. Hist, Michael J. Mossinghoff. Combinatorics and Graph Theory. Second Edition, Springer, 2008.
4. Robert A. Beeler. How to Count: An Introduction to Combinatorics and Its Applications. First Edition, Springer, 2015.

Supplementary Reading

1. Reinhard Diestel. Graph Theory. Fifth Edition, Springer, 2017.
2. Chen Chuan-Chong, Koh Khee-Meng. Principles and Techniques of Combinatorics. World Scientific Publishing Co. Pte. Ltd., 1992.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. Third Edition, The MIT Press, 2009
4. Michael T. Goodrich and Roberto Tamassia. Algorithm Design: Foundations, Analysis, and Internet Examples. Second Edition, 2006, John Wiley and Sons. Inc.

Object Oriented Programming (TIU-UCS-T214)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3
Prerequisite Course:
Introduction to Computing (TIU-UCS-T105); Problem Solving Techniques (TIU-UCS-T106)

Course Outcome

CO1	Define an object-oriented approach to programming and identify potential benefits of object-oriented programming over other approaches
CO2	Understand the difference between the top-down and bottom-up program design approach
CO3	Demonstrate the use of various OOPs concepts using C++
CO4	Solve a computational problem by implementing the solution as a real-world entity
CO5	Understand and apply some advanced constructs of C++ like virtual function, operator overloading, exception handling, standard template library

Course Content

Module-1:
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. Class, Object, Data abstraction, Encapsulation, Inheritance, Polymorphism, Message Passing, Dynamic binding.

Module-3:
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:
Class and Function Templates, Standard Template Libraries in C++: lists, vectors, sets, maps. Exceptions Handling.

Recommended Books:

Main Reading

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.

Supplementary Reading

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

Microprocessor and Microcontroller (TIU-UEC-T210)

Contact Hours/Week: 3–0–0 (L–T–P)
Credit: Theory–3

Course Outcome

CO1	Do understand the 8085, 8086 processor operations
CO2	Do assembly language programming.
CO3	Do interfacing design of peripherals like 8255, 8237, memory etc.
CO4	Develop systems using different microcontrollers.

Course Content

Module-1: Introduction to 8085 (8bit 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: Interfacing memory, IO devices (IO mapped IO & Memory mapped IO). Interrupts, Interfacing 8255 peripheral, DMA controller (8237).

Module-3: Introduction to 8086 (16bit processor): Architecture, register organization, pin description
and features. Instruction Set and Addressing modes, interfacing memory, IO devices, Interrupts.

Module-4: Introduction to 8051 (8bit microcontroller): Architecture, Pin description and features, Special function registers, I/O pin ports and circuits, Internal, External memories, addressing modes, Assembly Language Programming.

Recommended Books:

Main Reading

1. R. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085”, Penram
2. D. A. Patterson and J H Hennessy, "Computer Organization and Design The hardware and software interface. Morgan Kaufman Publishers.
3. P. K. Ghosh and P. R. Sridhar, “0000 to 8085: Introduction to Microprocessors for Engineers and Scientists”, PHI Learning
4. D. V. Hall, “Microprocessors and Interfacing”, McGraw Hill
5. K. M. Bhurchandi and A. K. Ray, “Advanced Microprocessors and Peripherals”, Tata McGraw Hill
6. M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, “The 8051 Microcontroller and Embedded Systems”, Pearson
7. K. Ayala, “The 8051 Microcontroller”, Delmar Cengage Learning
8. Krishna Kant, “Microprocessor and Microcontrollers”, Prentice Hall of India

Raj Kamal, “8051 Microcontrollers”, Tata McGraw Hill