HERITAGE INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Under MAKAUT)

SYLLABUS OF 4th SEMESTER

Department of Computer Science & Engineering

A. THEORY COURSES

Course Name: Design & Analysis of Algorithms					
Course Code: CSEN2201					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

After completion of the course, students will be able to:

CSEN2201.1. Remember time complexities of various existing algorithms in different situations.
CSEN2201.2. Understand the basic principles of different paradigms of designing algorithms.
CSEN2201.3. Apply mathematical principles to solve various problems.
CSEN2201.4. Analyze the complexities of various algorithms in worst case, best case and average case.
CSEN2201.5. Assess the computational hardness of a problem and learn how some of the well-known problems are proved to be NP-hard and also design approximation algorithms for some of them.
CSEN2201.6. Create/ Design a good algorithm for a new problem given to him/ her.

2. Detailed Syllabus

Module 1 [10L]

Algorithm Analysis: Time and space complexity. Asymptotic Notations and their significance. Asymptotic Analysis. Finding time complexity of well-known algorithms like-insertion sort, heapsort, Asymptotic solution to recurrences, Substitution Method, Recursion Tree, Master Theorem.
Divide-and-Conquer Method: Basic Principle, Binary Search – Worst-case and Average Case Analysis, Merge Sort – Time Complexity Analysis, quicksort – Worst-case and Average Case Analysis, Concept of Randomized Quicksort.
Medians and Order Statistics
Lower Bound Theory: Bounds on sorting and searching techniques**.**

Module 2 [16L]

Greedy Method: Elements of the greedy strategy. Fractional Knapsack Problem, Huffman codes.
Dynamic Programming: Basic method, use, Examples: 0-1 Knapsack Problem, Matrix-chain multiplication, LCS Problem.
Graph Algorithms: Minimum cost spanning trees: Prim's and Kruskal's algorithms and their correctness proofs (Greedy Method). Shortest Path Algorithm: Dijkstra's with correctness proof. (Greedy method), Bellman Ford with correctness proof, All pair shortest path (Floyd-Warshall Algorithm) (Dynamic Programming).

Module 3 [10L]

Amortized Analysis: Aggregate, Accounting and Potential methods.
String matching algorithms: Different techniques – Naive algorithm, string matching using finite automata, and Knuth , Morris , Pratt (KMP) algorithm with their complexities
Randomized Algorithm: Skip List.

Module 4 [10L]

Disjoint Set Manipulation: UNION-FIND with union by rank, Path compression.
Network Flow: Ford Fulkerson algorithm, Max - Flow Min - Cut theorem (Statement and Illustration)
NP-completeness: P class, NP-hard class, NP-complete class. Relative hardness of problems and polynomial time reductions. Satisfiability problem, Vertex Cover Problem, Independent Sets, Clique Decision Problem.
Approximation algorithms: Necessity of approximation scheme, performance guarantee. Approximation algorithms for 0/1 knapsack, vertex cover, TSP. Polynomial time approximation schemes: 0/1 knapsack problem.

3. Textbooks

1. Introduction to Algorithms by Cormen, Leiserson, Rivest and Stein. Third Edition, 2009. Prentice Hall.
2. Algorithm Design by Jon Kleinberg and Eva Tardos. Addison Wesley, 2005.

4. Reference Books

1. Computer Algorithms: Introduction to Design and Analysis by Sarah Basee and Allen van Gelder. 3rd Edition, Addison Wesley.

Course Name: Computer Organization and Architecture					
Course Code: CSEN2202					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

After completion of the course, students will be able to:

CSEN2202.1. Understand the basic organization of computer and different instruction formats and addressing modes.
CSEN2202.2. Analyze the concept of pipelining, segment registers and pin diagram of CPU.
CSEN2202.3. Understand and analyze various issues related to memory hierarchy.
CSEN2202.4. Understand various modes of data transfer between CPU and I/O devices.
CSEN2202.5. Examine various inter connection structures of multi-processor.
CSEN2202.6. Design architecture with all the required properties to solve state-of-the-art problems.

2. Detailed Syllabus

Module 1 [10L]

Basics of Computer Organization: Basic organization of the stored program computer and operation sequence for execution of a program, Von Neumann & Harvard Architecture. RISC vs. CISC based architecture.
Fetch, decode and execute cycle, Concept of registers and storage, Instruction format, Instruction sets and addressing modes. Basics of Control Unit Design - hardwired and micro programmed control, Horizontal and Vertical micro instruction.

Module 2 [11L]

Memory and I/O Organization: Memory system overview, Cache memory organizations, Techniques for reducing cache misses, Hierarchical memory technology: Inclusion, Coherence and locality properties, Virtual Memory, Memory mapped IO. Introduction to I/O interfaces. Interrupts, Interrupt hardware, Enabling and Disabling interrupts, Concept of handshaking, Polled I/O, Priorities, Daisy Chaining. Vectored interrupts; Direct memory access, DMA control.

Module 3 [10L]

Pipelined Architecture: Brief Introduction, Performance Measures - speed up, Efficiency, performance - cost ratio etc.
Static pipelines - reservation tables, scheduling of static pipelines, definitions - minimum average latency, minimum achievable latency, greedy strategy etc. Theoretical results on latency bounds without proof.
Vector Processing: Vector registers; Vector Functional Units; Vector Load / Store; Vectorization; Vector operations: gather / scatter; Masking; Vector chaining.

Module 4 [9L]

SIMD Architectures: Brief introduction, various concepts illustrated by studying detailed SIMD algorithms, viz., Matrix multiplication, Sorting on Linear array.
Interconnection Networks: Detailed study of Interconnection Network - Boolean cube, Mesh, Shuffle-exchange, Banyan, Omega, Butterfly, Generalized Hypercube, Delta etc.

3. Textbooks

1. Computer Organization, 5th Edition, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, MGH.
2. Computer System Architecture, 3rd Edition, Morris M. Mano, Pearson.
3. Computer Organization and Design: The Hardware/Software interface, David A. Patterson and John L. Hennessy, 3rd Edition, Elsevier, 2005.
4. Advanced Computer Architecture and Parallel processing, Hwang & Briggs, MH.
5. Advanced Computer Architecture: Parallelism, Scalability, Programmability, Kai Hwang, McGraw-Hill.

4. Reference Books

1. Onur Mutlu's lecture materials on Computer Architecture from CMU web site: https://users.ece.cmu.edu/~omutlu/.
2. NPTEL materials on Computer Organization.

Course Name: Operating Systems					
Course Code: CSEN2203					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	3	**0**	**0**	**3**	**3**

1. Course Outcomes

After completion of the course, students will be able to:

CSEN2203.1. Develop knowledge about the importance of computer system resources and the role of operating system in their management policies and algorithms.

CSEN2203.2. Understand processes and its management policies and scheduling of processes by CPU.

CSEN2203.3. Acquire an understanding of the need of process synchronization, evaluate the requirement for process synchronization and coordination handled by operating system.

CSEN2203.4. Analyze the memory management and its allocation policies and compare different memory management approaches.

CSEN2203.5. Understand the impact and co-relation of different scheduling algorithm in secondary storage and different structure of file system and able to design the system with improved performance.

CSEN2203.6. Identify the different activities and impact of threat, virus, worm and able to protect system from them.

2. Detailed Syllabus

Module 1 [9L]

Introduction: Operating system functions, OS Architecture (Monolithic, Microkernel, Layered, Hybrid), Different types of O.S. (batch, multi-programmed, time-sharing, real-time, distributed, parallel), Evaluation of OS.

System Structure: Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, Operating system structure (simple, layered, virtual machine), O/S services, System calls.

Protection & Security: Goals of protection, Domain of protection, Access matrix and its representation, Threats and system security.

Processes and Threads: 7 state process model, Process scheduling, Operations on processes, Inter-process communication, Threads overview, Benefits of threads, User and kernel threads.

Module 2 [11L]

CPU Scheduling: Scheduling criteria, Preemptive & non-preemptive scheduling, Scheduling algorithms (FCFS, SJF, RR, Priority, Multi-level queue, Multi-level feedback queue), Comparative study of the algorithms, Multi-processor scheduling.

Process Synchronization: Background, Critical section problem, Software solution – Peterson and Bakery algorithm, Synchronization hardware, Semaphores, Classical problems of synchronization.

Deadlocks: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

Module 3 [9L]

Primary Memory: Background, Physical address, Logical address, Virtual address, Contiguous memory allocation (Fixed and Variable partition), Non-contiguous memory allocation techniques (Paging, Segmentation, Segmentation with Paging), Virtual memory, Demand Paging, Performance, Page replacement algorithms (FCFS, LRU, optimal), Thrashing.

Module 4 [7L]

Secondary Storage: Disk structure, Disk performance, Disk scheduling (FCFS, SSTF, SCAN, C-SCAN), Boot block, Bad blocks.

File Systems: File concept, Access methods, Directory structure, File system structure, Allocation methods (Contiguous, Linked, Indexed), Free-space management (Bit vector, Linked list, Grouping), Directory Implementation (Linear list, Hash table), Efficiency and Performance.

I/O Management: PC Bus Structure, I/O connections, Data transfer techniques (Programmed, Interrupt driven, DMA), Bus arbitration (Daisy chain, Polling, Independent request), Blocking and non-blocking I/O, Kernel I/O subsystem (Scheduling, Buffering, Caching, Spooling and device reservation, Error handling).

3. Textbooks

1. Operating System Concepts, 10E, Silberschatz A., Galvin P. B., Gagne G., Wiley Publications.
2. Operating Systems Internals and Design Principles, 9E, Stalling W., Pearson Education.

4. Reference Books

1. Operating System: Concept & Design, Milenkovie M., McGraw Hill.
2. Operating System Design & Implementation, Tanenbaum A.S., Prentice Hall NJ.
3. Operating System Concepts, Silberschatz A., Peterson J. L., Wiley Publications.
4. Operating Systems A Concept Based Approach, Dhamdhere D.M., McGraw Hill.

Course Name: Algebraic Structures					
Course Code: MATH2201					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

After completion of the course, students will be able to:

MATH2201.1. Describe the basic foundation of computer related concepts like sets, Posets, lattice and Boolean Algebra.

MATH2201.2. Analyse sets with binary operations and identify their structures of algebraic nature such as groups, rings and fields.

MATH2201.3. Give examples of groups, rings, subgroups, cyclic groups, homomorphism and isomorphism, integral domains, skew-fields and fields.

MATH2201.4. Compare even permutations and odd permutations, abelian and non-abelian groups, normal and non-normal subgroups and units and zero divisors in rings.

MATH2201.5. Adapt algebraic thinking to design programming languages.

MATH2201.6. Identify the application of finite group theory in cryptography and coding theory.

2. Detailed Syllabus

Module 1 [10L]

Sets, Relations and Functions: Basic operations on sets, Venn diagrams. Binary relations defined on sets, equivalence relations and equivalence classes, order, relation and lattices, partially ordered sets, Hasse diagrams, maximal, minimal, greatest and least elements in a partially ordered set, lattices and their properties, principle of duality, distributive and complemented lattices.

Module 2 [10L]

Group Theory I: Cartesian product, Binary operation, Composition Table. Group, Elementary theorems on groups, Quasi-group and Klein's 4 group. Permutations, Product of permutations, Group property of permutations, Cyclic permutation, Transposition, Even and Odd permutations, Proposition regarding permutations, Alternating Groups.

Module 3 [10L]

Group Theory II: Order of an element of a group, Properties of the order of an element of a group , Subgroups, some basic theorems on subgroups, Cyclic group, Cosets, Lagrange's theorem, Fermat's Little Theorem(statement only). Normal subgroup, some basic theorems on Normal subgroup.

Module 4 [6L]

Morphisms, Rings and Fields: Homomorphism and Isomorphism of groups, some basic theorems. Rings, some elementary properties of a ring, Ring with unity, Characteristic of a ring, Ring with zero divisors, Sub-ring, Integral domain, Field, Division Ring or Skew Field. (Emphasis should be given on examples and elementary properties).

3. Textbooks

1. Higher Algebra, S.K.Mapa, Sarat Book Distributors.
2. Advanced Higher Algebra, J.G.Chakravorty and P.R.Ghosh, U.N. Dhur and Sons.

4. Reference Books

1. A First course in Abstract Algebra, J.B.Fraleigh, Narosa.
2. Algebra, M. Artin, Pearson.

Course Name: Microprocessors & Microcontrollers					
Course Code: AEIE2205					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	2	**0**	**0**	**2**	**2**

1. Course Outcomes

After completion of the course, students will be able to:

AEIE2205.1. Understand the architecture of 8-bit microprocessor (8085A).

AEIE2205.2. Develop the skill in program writing of 8-bit microprocessor (8085A).

AEIE2205.3. Understand the architecture and develop the skill in program writing of 16-bit microprocessor (8086).

AEIE2205.4. Understand the architecture and develop the skill in program writing of microprocessor 8051 and PIC16F877.

AEIE2205.5. Understand the architecture and operation of programmable peripheral device 8255A.

2. Detailed Syllabus

Module 1 [6L]

Introduction to 8-bit microprocessor: 8085 microprocessor internal architecture, 8085 pin configuration, Software instruction set, timing diagram of the instructions.

Module 2 [7L]

Addressing modes and Assembly language programming: Interrupts of 8085 processor: classification of interrupts, Programming using interrupts. Counter and Time delay, Support IC chips 8255- Block diagram, pin configuration, mode of operation, control word(s) format and Interfacing with Microprocessors.

Module 3 [7L]

Introduction to 8086/8088 Architecture: Architecture, memory segmentation, pin configuration, clock generator, instruction set, addressing modes and assembly language programming of 8086/8088, interrupts.

Module 4 [6L]

Introduction to microcontrollers: Intel MCS-51 family features, 8051 architecture, pin configuration, I/O ports and memory organization; Instruction set and basic assembly language programming, interrupts and returns; Interrupts, timer/counter and serial communication.

Brief introduction to PIC microcontroller (16F877): Architecture, pin details, memory layout etc.

3. Textbooks

1. Microprocessor architecture, programming and applications with 8085/8085A, Ramesh S. Gaonkar, Wiley eastern Ltd.
2. Fundamental of Microprocessor and Microcontrollers, B. Ram, Dhanpat Rai Publications.
3. Microprocessors and Microcontrollers, N. Senthil Kumar, M. Saravanan, S. Jeevanathan, Oxford Publications.
4. 8085 Microprocessor and its Applications, A. Nagoor Kani, Third Edition, TMH Education Pvt. Ltd.

4. Reference Books

1. The 8051 Microcontroller and Embedded. Systems. Using Assembly and C. Muhammad Ali Mazidi, Janice Gillispie Mazidi. Rolin D. McKinlay, Second Edition, Pearson Publication.
2. Advanced Microprocessors and Peripherals, A.K.Ray, K.Bhurchandi, TMH Education Pvt. Ltd.
3. PIC Microcontroller and Embedded. Systems. Using Assembly and C. Muhammad Ali Mazidi, Janice Gillispie Mazidi. Rolin D. McKinlay, Pearson Publication.
4. Design with PIC Microcontroller, John Peatman, Pearson Publication.

Course Name: Environmental Sciences (Mandatory)					
Course Code: EVSC2016					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	2	**0**	**0**	**2**	**0**

1. Course Outcomes

After completion of the course, students will be able to:

EVSC2016.1. Understand the natural environment and its relationships with human activities.
EVSC2016.2. Characterize and analyze human impacts on the environment.
EVSC2016.3. Integrate facts, concepts, and methods from multiple disciplines and apply to environmental problems.
EVSC2016.4. Educate engineers who can work in a multi-disciplinary environment to anticipate and address evolving challenges of the 21st century.
EVSC2016.5. Understand and implement scientific research strategies, including collection, management, evaluation, and interpretation of environmental data.
EVSC2016.6. Design and evaluate strategies, technologies, and methods for sustainable management of environmental systems and for the remediation or restoration of degraded environments.

2. Detailed Syllabus

Module 1 [6L]

Socio Environmental Impact: Basic ideas of environment and its component
Population growth: exponential and logistic; resources; sustainable development.
Concept of green chemistry: green catalyst, green solvents
Environmental disaster and social issue: environmental impact assessment, environmental audit, environmental laws and protection act of India.

Module 2 [6L]
Air Pollution: Structures of the atmosphere, global temperature models, Greenhouse effect, global warming; acid rain: causes, effects and control. Lapse rate and atmospheric stability; pollutants and contaminants; smog; depletion of ozone layer; standards and control measures of air pollution.

Module 3 [6L]
Water Pollution: Hydrosphere; pollutants of water: origin and effects; oxygen demanding waste; thermal pollution; pesticides; salts. Biochemical effects of heavy metals; eutrophication: source, effect and control. Water quality parameters: DO, BOD, COD. Water treatment: surface water and wastewater.

Module 4 [6L]
Land Pollution: Land pollution: sources and control; solid waste: classification, recovery, recycling, treatment and disposal.
Noise Pollution: Noise: definition and classification; noise frequency, noise pressure, noise intensity, loudness of noise, noise threshold limit value; noise pollution effects and control.

3. Textbooks
1. Basic Environmental Engineering and Elementary Biology, Gour Krishna Das Mahapatra, Vikas Publishing House P. Ltd.
2. Environmental Chemistry, A. K. De, New Age International.
3. Environmental Chemistry with Green Chemistry, A. K. Das, Books and Allied P. Ltd.

4. Reference Books
1. Environmental Science, S. C. Santra, New Central Book Agency P. Ltd.
2. Fundamentals of Environment & Ecology, D. De, D. De, S. Chand & Company Ltd.

B. LABORATORY COURSES

Course Name: Design & Analysis of Algorithms Lab					
Course Code: CSEN2251					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**3**	**3**	**1.5**

1. Course Outcomes
After completion of the course, students will be able to:
CSEN2251.1. Understand and Apply different types of algorithm designing paradigms like divide and conquer, greedy, dynamic programming etc.
CSEN2251.2. Realize and Apply underlying mathematical principles of algorithms in the corresponding implemented program.
CSEN2251.3. Analyse and Evaluate the performance of various algorithms by observing the actual running time and main memory consumption of the corresponding implemented programs for best case, worst case and average case input data.
CSEN2251.4. Create / Design a good algorithm for solving real life computing problems, by using various design techniques and data structures, learnt in this course.

2. Detailed Syllabus
A tentative list (non-exhaustive) of the practical topics is given below:

1. **Divide and Conquer:** Implement Quick Sort and **randomized version** of quick sort using Divide and Conquer approach. Check the running time for each of the n! combinations or input sequences of a particular set of integers to observe the best, worst and average cases.
2. **Divide and Conquer:** Implement Merge Sort using Divide and Conquer approach. Check the running time for each of the n! combinations or input sequences of a particular set of integers to observe the best, worst and average cases.
3. Implement Heapsort algorithm. Check the running time for each of the n! combination or input sequences of a particular set of integers to observe the best, worst and average cases.
4. **Dynamic Programming:** Find the minimum number of scalar multiplications needed for chain of Matrices.
5. **Dynamic Programming:** Implement Bellman Ford Algorithm to solve Single Source shortest Path problem of a graph.
6. **Dynamic Programming:** Implement Floyd-Warshall Algorithm to solve all pair shortest path for a graph.
7. **Dynamic Programming:** Solve 0/1 Knapsack problem using dynamic problem.
8. **Dynamic Programming:** Solve Longest Common Subsequence problem using dynamic problem.
9. **Greedy method:** Implement Dijkstra's algorithm to find Minimum Spanning Tree of a graph by using minimum priority Queue or minimum heap data structure.
10. **Greedy method:** Implement Prim's algorithm to find Minimum Spanning Tree of a graph by using minimum priority Queue or minimum heap data structure.

11. **Greedy method:** Implement Kruskal's algorithm to find Minimum Spanning Tree of a graph by implementing and using various operations of Disjoint-set Forest data structure.
12. **Greedy method:** Implement Huffman coding using greedy approach.
13. **Realization of Amortized Analysis:** Implement a Queue using Stacks.
14. Implement KMP algorithm for string matching
15. Implement Ford-Fulkerson algorithm to get maximum flow in a given flow network.
16. **Randomized Algorithm:** Implement Skip-List)**.**

3. Textbooks
1. Introduction to Algorithms, Cormen, Leiserson, Rivest and Stein. Third Edition, 2009. Prentice Hall.
2. Algorithm Design, Jon Kleinberg and Eva Tardos. Addison Wesley, 2005.

4. Reference Books
1. Computer Algorithms: Introduction to Design and Analysis, Sarah Basee and Allen van Gelder. 3rd Edition, Addison Wesley.

Course Name: Computer Architecture Lab					
Course Code: CSEN2252					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**2**	**2**	**1**

1. Course Outcomes
After completion of the course, students will be able to:

CSEN2252.1. Understand the role of simulator environment in digital hardware design

CSEN2252.2. Design various combinational and sequential logic circuits using simulator

CSEN2252.3. Realize / Implement simulator-based circuits performing various arithmetic and logic operations

CSEN2252.4. Interpret informal system requirements into formal testbenches

CSEN2252.5. Realize various types of memory / storage elements via simulator

CSEN2252.6. Create / Design a standard open-source CPU in RTL and / or in microarchitecture using logic building blocks.

2. Detailed Syllabus
Programming using VHDL
1. All Logic Gates (Data flow and Behavioral model)
2. Half adder and half subtractor (Data flow and Behavioral Model)
3. Combinatorial Designs (Data flow and Behavioral Model)
   a. 2:1 Multiplexer
   b. 4:1 Multiplexer
   c. 3:8 Decoder
   d. Comparator
4. Full adder and full subtractor (Data flow, Behavioral and Structural Model)
5. Sequential design of flip flops (SR, JK, D, T)
6. ALU design
7. Ripple carry adder (Structural Model)
8. Adder subtractor composite unit (Structural Model)
9. 4 bit synchronous and asynchronous counters.
10. Small projects like stepper motor.

3. Textbooks
1. VHDL: Programming by Example, Douglas L. Perry, Fourth Edition, McGraw Hill.

4. Reference Books
1. Introduction to Logic Circuits & Logic Design with VHDL, LaMeres, Brock J, Springer.

Course Name: Operating Systems Lab					
Course Code: CSEN2253					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**3**	**3**	**1.5**

1. Course Outcomes

After completion of the course, students will be able to:

CSEN2253.1. Understand and implement basic services and functionalities of the operating system using system calls.
CSEN2253.2. Will be able to describe and create user defined processes.
CSEN2253.3. Understand the benefits of thread over process and implement them.
CSEN2253.4. Synchronization programs using multithreading concepts.
CSEN2253.5. Use modern operating system calls and synchronization libraries in software to implement process synchronization.
CSEN2253.6. Implementation of Inter-process communication using PIPE.

2. Detailed Syllabus

1. **Shell programming:** Creating a script, making a script executable, shell syntax (variables, Conditions, control structures, functions and commands).
2. **Process:** starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.
3. **Signal:** signal handling, sending signals, signal interface, signal sets.
4. **Semaphore:** programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).
5. **POSIX Threads:** programming with pthread functions (viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)
6. **Inter-process communication:** pipes (use functions pipe, popen, pclose), named pipes (FIFOs, accessing FIFO).

3. Textbooks

1. Your Unix: The Ultimate Guide, Sumitabha Das, MH

4. Reference Books

1. Beginning Linux Programming, Neil Matthew, Richard Stones, Wrox.

Course Name: Microprocessors & Microcontroller Lab					
Course Code: AEIE2255					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**2**	**2**	**1**

1. Course Outcomes

After completion of the course, students will be able to:

AEIE2255.1. Understand and apply different instructions of 8085 microprocessor.
AEIE2255.2. Understand and apply different instructions of 8086 microprocessor.
AEIE2255.3. Understand and apply different instructions of 8051 microcontroller.
AEIE2255.4. Interface 8085A microprocessor with different input and output devices (e.g., LEDs, seven segments displays ADC, DAC, and stepper motor etc.).
AEIE2255.5. Interface 8086A microprocessor/ 8051 microcontroller with different input and output devices (e.g., LEDs, seven segments displays ADC, DAC, and stepper motor etc).

2. Detailed Syllabus

1. Familiarization with 8085A trainer kit components with the process of storing and viewing the contents of memory as well as registers. Repeat the above all using 8085A Simulator.
2. Study of programs using basic instruction set (data transfer, load/store, arithmetic, logical) of 8085A microprocessor.
3. Programming using 8085A trainer kit/simulator for:
   a) Copying and Shifting block of memory
   b) Packing and unpacking of BCD numbers
   c) Addition/Subtraction of two 8-bit Hex numbers
   d) Addition of 16-bit Hex numbers.
   e) BCD Addition
   f) Binary to ASCII conversion
   g) String Matching and Sorting.

4. Familiarization of 8086 microprocessor trainer kit/simulator using data transfer, load/store, arithmetic and logical instructions.
5. Write assembly language programs (ALP) using 8086 microprocessor trainer kit /simulator on the following:
   a) Finding the largest/ smallest number from an array
   b) Arranging numbers in ascending/descending order
   c) Shifting a block of data from one memory location to another
   d) Addition of a series of BCD numbers
   e) String matching

6. Interfacing of 8085A through 8255A PPI/ 8051 Microcontroller with switches and LEDs to perform
   a) Display operation
   b) Blinking operation and
   c) Scrolling operation
7. Interfacing with seven segment displays through 8-bit latch (e.g., 74LS373) using- a) 8085A trainer kit, b) 8086A trainer kit through 8255A PPI.
8. Interfacing of ADC, DAC, and Stepper motor with 8085A/8086 microprocessor trainer kit.

3. Textbooks and References

Assignment Sets to be provided.