School of Computer & Information Sciences

Integrated M.Tech (Computer Science & Engineering)

w.e.f 2022-23

SCHOOL OF COMPUTER & INFORMATION SCIENCES
Integrated M.Tech (CSE)

TABLE OF CONTENTS

SCHOOL OF COMPUTER & INFORMATION SCIENCES

Vision Statement:

- To invent, create and bring computing technology solutions to the common man, to the privileged and underprivileged sections of India, to bridge the digital divide and eradication of computer ignorance and digital illiteracy and to build a prosperous and technologically advanced nation.

Mission Statements:

MS-1: To pursue academic and research excellence, nationally and internationally

MS-2: To provide training, advisory, and consultancy to all the stakeholders.

MS-3: To lead the efforts in creative and newer modes of instruction delivery & supervision

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE)

Program Educational Objectives (PEOs)

PEO-1: To produce graduates with strong foundational concepts, techniques and tools to enable them to be pursue higher studies.

PEO-2: To prepare students to apply engineering knowledge to solve problems in computer science and other fields.

PEO-3: To produce graduates with strong human values and professional ethics

PEO-4: Produce graduates/Postgraduates who can contribute to the Research & Development effectively

PEO-5: To provide students a deep insight into cutting edge technologies and tools.

PEO-6: To create globally competent technocrat's with exposure to Scientific & Engineering aspects of development

PEO-7: To work collaboratively on multi-disciplinary projects and exhibit high levels of professional & ethical values

PEO-8: Create awareness of societal problems and its impact

Note: *PEO-1 to PEO-3 applies to I.MTech I-VI and PEO-4 to PEO-8 applies to I.MTech VII-X*

Mapping Program Educational Objectives (PEOs) with Mission Statements (MS)

	MS-1	MS-2	MS-3
PEO-1	3	2	
PEO-2	3		2
PEO-3	2		3
PEO-4	3	2	1
PEO-5	2	3	1
PEO-6	3	2	1
PEO-7	2	1	3
PEO-8	1	2	3

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

Name of the Academic Program: <u>Integrated M.Tech (Computer Science & Engineering) I-VI</u>

Program Outcomes (POs)

PO-1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and engineering specialization to the solution of the complex engineering problems.

PO-2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO-3: Design/Development of Solutions: Design solutions for complex engineering problems and design system component or processes that meets the specified needs with appropriate consideration for the public health and safety, and the cultural societal and environmental considerations.

PO-4: Conduct Investigation of Complex Problems: Use research based knowledge and research methods including designs of experiments, analysis, and interpretation of data and synthesis of the information to provide valid conclusions

PO-5: Model tool Usage: Create, select, and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of limitations.

PO-6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice

PO-7: Environmental and sustainability: Understand the impact of the professional engineering solutions in societal and environmental context, and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO-9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend ad write the effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and a leader in a team, to manage projects and in multidisciplinary environments

PO-12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life long learning in the broadest context of technological change.

Mapping of Program Outcomes (POs) and Program Specific Outcomes (PSOs) with Program Educational Objectives (PEOs)

	PEO-1	PEO-2	PEO-3
PO-1	2	3	1
PO-2	3	2	1
PO-3	1	3	2
PO-4	1	3	2
PO-5	2	3	1
PO-6	3	2	1
PO-7	1	2	3
PO-8	2	1	3
PO-9	1	2	3
PO-10	3	1	2
PO-11	1	2	3
PO-12	3	1	2

Mapping of Program Specific Outcomes (PSOs) where applicable.

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences
Integrated M.Tech (CSE)
Scheme (I-VI)   Monsoon Semester

I-Semester			Category	Unit
Code	**Course Title**	**Credits**		
	English-I	3-0-0	HSC	CELS
	Math-I	3-0-0	BSC	Maths
	Basket B1 (Language – I, Study of History, Study of Archaeology, ...)	4-0-0	OS	SSS
	Programming Methodology	3-0-0	ESC	SCIS
	PM Lab	0-0-2	ESC	SCIS
IE101	Probability & Statistics	4-0-0	BSC	SCIS
IE102	*IT Workshop*	2	ESC	SCIS
		21		

III-Semester				
Code	**Course Title**	**Credits**		
	Math-III : Differential Equations	3-0-0	BSC	Maths
IE203	Discrete Mathematics	3-0-0	ESC	SCIS
	Electronics Devices & Circuits (EDC)	3-0-0	ESC	Physics
IE201	Computer Based Numerical Methods (CBNM)	3-0-0	BSC	SCIS
IE202	Computer Organization & Architecture	4-0-0	PCC	SCIS
IE204	*CBNM Lab*	0-0-2	BSC	SCIS
	EDC Lab	0-0-1.5	ESC	Physics
	Basket B3 (Introduction to Public Health, AICTE MC)	3-0-0	OS	Med. Sc
		22.5		

V-Semester				
Code	**Course Title**	**Credits**		
IE301	Operating Systems (OS)	3-0-0	PCC	SCIS
IE302	Compiler Design	3-0-0	PCC	SCIS

IE303	Data Base System (DBS)	3-0-0	PCC	SCIS
IE304	Algorithms	3-1-0	PCC	SCIS
IE305	Principles of Programming Languages	3-0-0	PCC	SCIS
IE306	*DBS Lab*	2	PCC	SCIS
IE307	*OS Lab*	2	PCC	SCIS
	Compiler Design Lab	1	PCC	SCIS
		21		

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-I)

Course Code: IE352/IE353   Title of the Course: Programming Methodology/PM Lab

L-T-P: 3-0-0/0-0-3   Credits   : 3/2

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

CO-1: Create specification from problem requirements by asking questions to disambiguate the requirement statement. (Create)

CO-2: Design the solution from specification of a problem and write pseudo code of the algorithm. (Create)

CO-3: Analyze algorithms by tracing algorithms with test cases. (Analyze)

CO-4: Develop C programs using all supported features and compile them using Makefile. (Create)

CO-5: Analyze programs using debugging tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			2	1	3							
CO2			3			2					1	
CO3	2	3						1				
CO4				3					2			1
CO5		1		3	2							

.

Detailed Syllabus:

UNIT-I: Introduction to problem solving:
Problems and problem instances; Informal approach to program design: generalisation, special cases, and algorithms, breaking down a problem into functions, input and output.

UNIT-II: Introduction to the 'C' programming language:
Program structure; main() function; unnamed and named blocks; basic data types, variables, declaration and definition; initialisation and assignment; arithmetic operators and precedence; implicit and explicit type conversions; arrays; boolean variables and logical operators.

UNIT-III: Control structures:
Branching and iteration; functions and parameters; break(), return() and exit() functions; local and global variables; function prototypes.

UNIT-IV: Pointer variables and dynamic structures:
Static and dynamic (run-time) memory structures; static variables; breaking a program across multiple files; creating and linking libraries.

UNIT-V: Detecting and correcting common errors:
Debugging and debuggers; documenting programs; good programming practices; programming exercise (writing a program of at least 200 lines split across multiple files).

Reference Books:
1. Brian W. Kernighan, Dennis M. Ritchie. "The C Programming Language, 2nd Edition", Prentice-Hall India.
2. G. Michael Schneider. "Introductionto Programming and Problem Solving with PASCAL", JohnWiley and Sons.
3. Paul Deitel and Harvey Deitel . "C How to Program", Pearson Education India.
4. Stephen Kochan. "Programming in C", Pearson Education India.
5. Brian W. Kernighan and R. Pike. "The Unix Programming Environment", Prentice-Hall India.
6. Chakravarthy Bhagvati. "How to Program (An Informal Guide)", https://scis.uohyd.ac.in/~chakcs/howtoprogram.pdf

School of Computer & Information Sciences

Name of the Academic Program: <u>**Integrated M.Tech (CSE) (IMTECH-I)**</u>

Course Code   : <u>IE101</u>   Title of the Course   : <u>**Probability & Statistics**</u>

L-T-P   :. <u>4-0-0</u>   Credits   : 4

Prerequisite Course / Knowledge (If any): Sound knowledge of Mathematics at 10+2 level

Course Overview: This course introduces Engineering Methods and Statistical thinking, the notion of uncertainty and randomness, Probability & Random variables and Basic Engineering data analysis. Students will learn univariate discrete and continuous random variables including their properties. This course introduces ideas of statistical inference and its importance in real world applications. Basic statistical modelling will be carried out on Engineering data using Excel and/or R.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO1: Identify the role that statistics can play in the engineering problem-solving process (Understand)
- CO2: Calculate the probabilities of individual, joint events such as unions and intersections events. (Analyze)
- CO3: Apply Addition, Multiplication and Bayes Theorem (Apply)
- CO4: Apply Probability computation of an event for a given theoretical distribution function (Discrete as well as Continuous) . (Apply)
- CO5: Compute the four moments for a given theoretical distribution function (Discrete as well as Continuous). (Analyze)
- CO6: Analyse the given data by carrying out discretise statistics (Univariate and Bivariate). (Analyze)
- CO7: Test hypotheses on the mean, variance or standard deviation of a normal distribution. (Analyze)
- CO8: Develop Statistical Model for given engineering data (fit distributions) (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3			2	1							
CO2			2		3		1					
CO3	2				3	1						
CO4	2				1	3						
CO5				3	2	1						
CO6		1				3		2				
CO7				3	2		1					
CO8			3	2			1					

Detailed Syllabus:

UNIT- I: Role of Statistics in Engineering. Over view of Engineering Data Collection, Random experiments, sample spaces, events, probability measure on events definition, properties, examples. Conditional probability definition, properties, examples, Bayes theorem, independent events.

UNIT- II: Definition of random variables, standard discrete and continuous random variables -viz. Bernoulli, Binomial, Geometric, Poisson, Exponential, Gamma, Normal. Expectation, variance, other properties.

UNIT- III: Definition of bivariate random variables, joint distributions, covariance and correlation between two random variables, independence, distributions of sums.

UNIT- IV: Data collection methods, types of data, graphical summaries of data, numerical summaries of univariate data, bivariate summaries, measures of association.

UNIT- V: Introduction to statistical inference, population parameters, variable(s) of interest, statistic, estimators as random variables.

Reading Material

1. Douglas C. *Montgomery &* George C. Runger "Applied Statistics and Probability for Engineers " (6e) Wiley ISV Paperback – 1 January 2016   (Chapters 1 to Chapter 10)

Additional Reading

1. Ross, S. A First Course in Probability, sixth edition, Pearson Education, 2007.
2. Ramachandran, K.M. and Tsokos, C.P. Mathematical Statistics with applications, Academic Press, 2009.
3. Daniels, W.W. Biostatistics: a foundation for analysis in the health sciences, 9th edition, John Wiley & Sons, 2008 .
4. Moore, D.S. The Basic Practice of Statistics, W. H. Freeman, 2003

School of Computer & Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-I)</u>

Course Code :   Title of the Course   : IT Workshop Credits : 2

I. LINUX AND COMMANDS (3 experiments)

     1. Exploring the directory structure: /, /bin, /etc, /usr, …

     2. Programming support: text editors, compilers, debuggers, IDEs

     3. Utility software: find, grep, …

II. COMPUTER HARDWARE AND OS INSTALLATIONS (3 experiments)

     1. Opening up a computer and identifying the key parts

     2. Disk partitioning, formatting and tools

     3. Installing and upgrading Linux, dual booting

III. BASICS OF ELECTRICAL ENGINEERING (3 experiments)

     1. Single and 3-phase supplies, identifying L, N, E

     2. Switch and 3-pin plug circuit; switch and fan with regulator

     3. 2-way switches (staircase switches) and fan, bulb

IV. ELEMENTS OF ENGINEERING GRAPHICS (3 experiments)

     1. Different aspects of engineering drawings: lines and others

     2. Orthographic, perspective and isometric projections

     3. Plan, elevation and 3D drawings

V. ENGINEERING AND DATA ANALYSIS SKILLS (2 experiments)

     1. Accuracy, precision and errors

     2. Plotting data: different types of plots

14 Experiments in total.

School of Computer & Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-III)</u>

Course Code   : <u>IE203</u>   Title of the Course   : <u>Discrete Mathematics</u>

L-T-P   :. <u>3-0-0</u>   Credits   : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a mathematics course at 10+2 level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

CO1: Apply predicate and propositional logic to represent and solve problems. (Apply)

CO2: Discuss various ways of simplification and apply the same on minimizing logical circuits, (Understand)

CO3: Using principle of recursion, be able to frame a real-world situation as a recurrence relation and solve. (Apply).

CO4: Describe counting principles (Understand)

CO5: Apply counting principles in real world scenarios. (Apply)

CO6: Describe graphs and trees techniques (Understand)

CO7: Apply the graphs and trees techniques to solve the real time problems (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1		3	2			1						
CO2	1		3		2							
CO3	2	3			1							
CO4	3			2	1							
CO5	1		2	3								
CO6	3		1			2						
CO7			1		2	3						

Detailed Syllabus:

UNIT-I: Sets, Relations and Functions

Sets, relations and functions; Methods of proof; Equivalence relations; Cardinality; Countable and uncountable sets

UNIT-II: Introductory Logic

Fundamentals of Logic; Logic operators such as AND, OR etc., Truth tables; Logical inferences; Methods of proofs of an implication; First order logic; Predicate calculus Predicates and Quantifiers; Rules of inference for quantified propositions

UNIT-III: Recurrence Relations:

Recursion, Forming and solving recurrence relations by substitution method and generating functions; Method of characteristic roots; solving inhomogeneous recurrence relations

UNIT-IV: Boolean Algebra:

Partial order relations; Lattices; Boolean algebra; Combinatorial circuits; Minimization of Boolean functions using Karnaugh maps

UNIT-V: Theory of Graphs

Graphs, subgraphs, isomorphism, proofs; Types of graphs; paths and cycles; Adjacency matrices; Transitive closure; Connectivity; Directed acyclic graphs; Planar graphs and Euler's formula; Dual of a graph; Hamiltonian and Eulerian graphs; Applications like matching and colouring graphs; Graph traversals (BFS and DFS); Trees; Spanning trees.

Reference Books:

1. Kenneth H Rosen (2012), *"Discrete Mathematics and Its Applications"*, 7th Edition, McGraw Hill, NY
2. Ralph P Girimaldi(2003), *"Discrete and Combinatorial Mathematics –An Applied Introduction"*, 5th Edition, Pearson Addison Wesley, Indian Edition
3. J.R Mott, A Kandel, T.P Baker (2015), *"Discrete Mathematics for Computer Scientists and Mathematicians"*, Pearson
4. Ronald L Graham, Donald E Knuth, Oren Patashnik(1994), *"Concrete Mathematics- A Foundation of Computer Science"*, 2nd Edition, Addison Wesley .
5. Susanna S. Epp(2010), *"Discrete Mathematics with Applications"*, 4th Edition, Brooks/Cole Cengage Learning.

School of Computer & Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-III)</u>

Course Code   : <u>IE201</u>   Title of the Course   : <u>Computer Based Numerical Methods</u>

L-T-P   :. <u>3-0-0</u>   Credits   : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a basic mathematics course at 10+2 level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO-1:   Calculate the root of polynomials as well as transcendental functions using various iterative methods (Apply)
- CO-2: Compare different methods for their efficacy and accuracy in finding the roots. (Analyze)
- CO-3:   Apply various techniques to interpolate the given data. (Apply)
- CO-4: Compare interpolation techniques for their efficacy and accuracy. (Analyze)
- CO-5:   Solve system of linear equations using various iterative and direct techniques. Also compute eigen values and eigenvectors using direct methods. (Apply)
- CO-6:   Design a scheme that can calculate the derivatives of a function given a set of values of that function using methods based on interpolation and methods based on undetermined coefficients. Also they will be able to determine the order of the scheme (Create)
- CO-7:   Calculate the value of a definite integral using Trapezoidal and Simpson's 1/3 rule (Analyze)
- CO-8:   Compute numerical solution of first order differential equation with initial conditions using Euler's, Modified Euler's, Piccard's method of successive approximations, and Runge-Kutta methods (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2				3						
CO2	2	1		3								
CO3	2	1				3						
CO4	2	1		3								
CO5	2		3	1								
CO6			3	1	2							
CO7	1	2				3						
CO8	2	3	1									

Detailed Syllabus:

UNIT-I : Transcendental and Polynomial Equations

Various iterative methods for solving transcendental and polynomial equation such as Bisection method, Secant and Regula Falsi methods, Newton-Raphson method, Chebyshev method and Muller method. Also iterative methods such as Birge-Vieta and Bairstow methods and the direct Graffae's Root squaring method for polynomial equations

UNIT-II: Interpolation and Approximation

Lagrange interpolation, Iterated interpolation, Newton's divided difference interpolation, Newton-Gregory forward and backward difference interpolation, Least square approximation

UNIT-III: Differentiation and Integration

Methods based on Interpolation, Methods based on Finite Differences, Methods based on Undetermined Coefficients for Numerical differentiation. Newton-Cotes integration methods, Trapezoidal rule, Simpson's 1/3 rule, Composite integration methods for numerical integration

UNIT- IV: Linear Algebraic Equations and Eigen-value Problems

Direct methods: Gauss Elimination, Gauss-Jordan, and LD decomposition methods for solving system of linear algebraic equations. Iterative methods: Jacobi and Gauss-Seidel methods for solving a system of linear algebraic equations. Eigen-value and Eigen-vector computation of a square matrix

UNIT-V: Numerical Solution of Ordinary Differential Equations

Euler's method, Modified Euler's method, Piccard's successive approximation, and Runge-Kutta method for solving a First-order differential equation given its initial conditions

Reference Books:

1. M K Jain, S R K Iyengar, R K Jain, *Numerical Methods for Scientific and Engineering Computation*, New Age International Publications

2. James B Scarborough, *Numerical Mathematical Analysis,* Oxford & IBH Publishing Co

School of Computer and Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-III)</u>

Course Code: <u>IE204</u>   Title of the Course: <u>CBNM Lab</u>

L-T-P   : <u>0-0-3</u>   Credits   : <u>2</u>

Prerequisite Course / Knowledge (If any): Knowledge of any programming language.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO-1:   Develop iterative algorithms for finding a root of a polynomial or a transcendental function (Create)

CO-2:   Develop various interpolating algorithms (Create)

CO-3   Design and develop algorithms for solving a system of linear equations (Create)

CO-4   Develop algorithms to calculate the value of a definite integral. (Create)

CO-5   Develop methods for solving the first order differential equation with initial conditions. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1				3						
CO2	1		2			3						
CO3	2		1			3						
CO4	1		2			3						
CO5	2		1			3						

Detailed Syllabus:

- Write programs to find the root of a polynomial using Bisection, Secant Method, Newton Raphson method
- Write programs to implement interpolation using techniques such as Lagrange, Newton divided difference, Gregory-Newton forward and backward interpolation techniques
- Write programs to solve the system of linear equations using (i) Jacobi and Gauss-Seidel iterative algorithms as well as direct methods such as Gauss elimination and Gauss Jordan and (ii) direct methods for computing eigen-values and eigen-vectors of a matrix etc.
- Write programs to find the value of definite integral using Trapezoidal and Simpson's 1/3 rule.
- Write programs using Euler's, Modified Euler's, and Runge-Kutta methods to solve first order differential equations.

Reference Books:

1. M K Jain, S R K Iyengar, R K Jain, *Numerical Methods for Scientific and Engineering Computation,* New Age International Publications

2. James B Scarborough, *Numerical Mathematical Analysis,* Oxford & IBH Publishing Co

School of Computer & Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-III)</u>

Course Code   : <u>IE202</u>   Title of the Course   : <u>Computer Organization & Architecture</u>

L-T-P   :. <u>4-0-0</u>   Credits   : 4

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a programming course at any level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO1: Demonstrate arithmetic operations and assess their performance (Apply).
- CO2: Describe basic Instruction Set Architecture (ISA) (Understand)
- CO3: Explain the basic pipelining of instructions (Understand)
- CO4: Examine how the memory hierarchy has impact on performance of software. (Analyze)
- CO5: Describe Interrupt handling and DMA access for performing I/O.(Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2		3			1						
CO2	2	3			2							
CO3	3	2			1							
CO4		2		1		3						
CO5	3	2	1									

Detailed Syllabus:

UNIT - I: Computer Evolution & Arithmetic:

A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, Signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non Restoring algorithms, Floating point representations, IEEE standards, Floating point arithmetic

UNIT - II: The Central Processing Unit:

Machine Instruction characteristics, types of operands, types of operations, Addressing modes, Instruction formats, Instruction types, Processor organization, Processor as running example, Programmers model of , max/min mode, Register Organization, Instruction cycles, Read Write cycles, assembly instruction examples to explain addressing modes

UNIT – III: The Control Unit:

Single Bus Organization, Control Unit Operations: Instruction sequencing, Micro operations and Register Transfer. Hardwired Control: Design methods – State table and classical method, Design Examples - Multiplier CU. Micro-programmed Control: Basic concepts, Microinstructions and micro- program sequencing

UNIT - IV:

Memory Organization: Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM, High-Speed Memories: Cache Memory, Organization and Mapping Techniques, Replacement Algorithms, Cache Coherence, Virtual Memory: Main Memory allocation, Segmentation, Paging, Address Translation Virtual to Physical. Secondary Storage: Magnetic Disk, Tape, DAT, RAID, Optical memory, CDROM, DVD

UNIT - V: I/O Organization:

Input/ Output Systems (features and principles), Programmed I/O, Interrupt Driven I/O, Interrupt structure, Direct Memory Access (DMA), features Buses and standard Interfaces: Synchronous, Asynchronous, Parallel I/O features, Serial I/O features, PCI, SCSI, USB Ports Working mechanisms of Peripherals: Keyboard, Mouse, Scanners, Video Displays, Touch Screen panel, Dot Matrix, Desk-jet and Laser Printers.

UNIT - VI:

 Case Studies: Concepts RISC: Instruction execution characteristics,, RISC architecture and pipelining. RISC Vs CISC. ARM and Embedded Systems PowerPC, Intel X86 Evolution from 32bit to 64bit architectures. AMD Opteron

Reference Books

1. Patterson D.A. & Hennesy J.L., *Computer Organisation & Design: The Hardware/Software Interface*.
2. Computer Organization and Architecture, 10/E William Stallings ISBN-10: 0134101618 • ISBN-13: 9780134101613- See more at: http://www.pearsonhighered.com/pearsonhigheredus/educator/product/products_detail.page?isbn= 9780134101613&forced_logout=forced_logged_out#sthash.WVVJbZUb.dpuf.

School of Computer and Information Sciences

Name of the Academic Program: **Integrated M.Tech (CSE) (M.Tech-VI)**

Course Code : IE354   Title of the Course   : **Compiler Design /CD Lab**

L-T-P   : 3-0-2   Credits   : 3/1

Prerequisite Course / Knowledge (If any): Theory of Computation & Data Structures

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Discuss different phases of the compilation process (Understand)
- CO-2: Identify appropriate compiler for the given problem (Analyze)
- CO-3: Design a prototype for a small language (Create)
- CO-4: Write programs from the knowledge gained about the compilation (Apply)
- CO-5: Write the Implementation of the Lexical analyzer, parser and code generator using tools such as Lex, Bison etc (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2		1								
CO2			3	2	1							
CO3			3		2	1						
CO4			2		3		1					
CO5			2		3		1					

Detailed Syllabus:

UNIT - I Compiler Structure: Analysis- synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

UNIT - II Lexical Analysis: Interface with input program, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, implementation, regular definition, transition diagrams.

UNIT – III Syntax Analysis: CFGs, ambiguity, error detection and recovery, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LR parsers (SLR, LALR, LR).

UNIT - IV Syntax Directed Translation: Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions, Type checking

UNIT - V Run time Environments: Source language issues, storage organization, activation tree, activation record, stack allocation of activation records, parameter passing mechanisms, symbol tables, dynamic storage allocation techniques.

UNIT – VI Intermediate Code Generation: Intermediate representations, translation of declarations, assignments, control flow, boolean expressions and procedure calls. Implementation issues.

UNIT – VII Code Generation and Instruction Selection: Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGs, peep hole optimization.

UNIT - VII Code Optimization: Principal Sources of Optimization, Optimization of basic blocks, Introduction to Data flow Analysis (Reaching Definitions and Live Variable Analysis).

Text Books

1. AV Aho, MS Lam, R Sethi, JD Ullman: Compiler Design: Principles, Techniques and Tools, Pearson Education

Reference Books:

1. AW Appel, M Ginsburg: Modern Compiler Implementation in C, Cambridge University Press

2. K Cooper, L Torczon: Engineering a Compiler, Morgan Kaufmann

3. J.P. Tremblay, P.G. Sorenson: Theory and Practice of Compiler Writing, McGraw Hill

Suggested Assignments

Programming assignments based on lexical analysis, construction of predictive/operator precedence/SLR parsing table and parser, symbol table, dynamic storage allocation strategies, syntax directed translation, data flow analysis

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-V)

Course Code: IE301   Title of the Course: Operating Systems/OS Lab
L-T-P   : 3-0-3   Credits   : 3/2
Prerequisite Course / Knowledge (If any): -- None

Course Outcomes (COs)

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

- CO1: Discuss the ways system calls work.(Understand)
- CO2: Develop basic process management tasks such as scheduling, deadlock avoidance algorithms. (Create)
- CO3: Develop paging algorithm.(Create)
- CO4: Construct simple device drivers. (Create)
- CO5: Describe different file systems in existence and learn the pros and cons of the various systems. (Understand)
- CO6: Examine real world OS scheduling algorithms such as those used in Linux and Windows. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1		3		2							
CO2	3	2	1									
CO3		1	2			3						
CO4				2	1	3						
CO5	3	2	1									
CO6	1	2	3									

Detailed Syllabus:

UNIT - I: Introduction and Operating System Structures

Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT - II: Process and Thread Management

Process Concept, Process operations, Process Scheduling, Extended Process State Diagram, Process Context Switch in detail; Inter process Communication: Pipes, Named Pipes, Shared Memory; Process Synchronization: Signals, Mutexes, Semaphores, Monitors; Thread Management: thread creation, thread scheduling, thread synchronization; Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock.

UNIT - III: Memory Management

Memory allocation techniques: paging and segmentation, Swapping, structure of the page table; Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB).

UNIT - IV: File System Management

Disk management: formatting, boot block, swap-space management, RAID structure; Disk scheduling algorithms: elevator, C-SCAN; File concept, Access methods, Directory structure, file sharing, protection, file system structure; file system implementation: file system metadata storage structures such as inode, allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT - V: I/O Management

I/O devices: polling, interrupt-driven, DMA; Application I/O interface: character and block devices, network devices; clocks and timers, nonblocking and asynchronous I/O, vectored I/O; Kernel I/O interface: I/O scheduling, Buffering, Caching.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts, 9th edition, Wiley.
2. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
3. W. Richard Stevens, . Advanced Programming in Unix Environment, Pearson Education.
4. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
5. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
6. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
7. Robert Love. Linux Kernel Development, Pearson Education.
8. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-V)

Course Code: IE304   Title of the Course: Algorithms

L-T-P: 3-1-0   Credits   : 4

Prerequisite Course / Knowledge (If any): Data Structures in under graduate level, discrete mathematical structures, knowledge of sorting algorithms and basic search strategies

Course Outcomes (COs)

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

CO-1: Assess the inherent structure/hardness of a problem (Evaluate)

CO-2: Select an appropriate strategy to solve a problem (Understand)

CO-3 Design an algorithm that suits the time complexity requirements of the problem. (Create)

CO-4: Estimate the time and space complexities of an algorithm along with the necessary mathematical

proofs when necessary. (Evaluate)

CO-5: Devise algorithms by choosing appropriate data structures (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3				1						
CO2		1	2		3							
CO3	2		3	1								
CO4	1		2	3								
CO5	2		1			3						

Detailed Syllabus

UNIT-I: Analysis of Algorithms: Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substitution method, generating functions, Master's theorem etc. Warm-up to complexity analysis: Heap data structure, priority queue application, Best, worst and average case analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms. Strategies for problem solving

UNIT-II: Divide and Conquer strategy: Time complexity analysis for Merge Sort and Quick Sort Algorithms

UNIT-III: Greedy strategy: Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem (Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis

UNIT-IV: Dynamic Programming strategy: Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memorization. Applications to problems like Coin change, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree (OBST), Chained Matrix Multiplication, Traveling Salesperson Problem (TSP) etc.

UNIT-V: Backtracking and Branch & Bound strategies: State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

UNIT-VI: Theory of NP-Completeness: Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Satisfiability(SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Coloring, NP-Completeness proofs.

Reference Books:
1. Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest, PHI, 3rdEdition 2009.
2. Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
3. Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996
4. The Algorithm Design Manual- Steven S. Skiena, Springer, 2009

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-V)

Course Code: IE306   Title of the Course: Data Base Management systems lab

L-T-P: 0-0-3   Credits: 2

Prerequisite Course / Knowledge (If any): Programming Methodology, Data and File Structures, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

CO-1: Explain the fundamentals of SQL - Structured Query Language (Understand)

CO-2: Construct SQL queries to create, delete any given table structures and views including a database (Create)

CO-3: Apply set of commands to pose queries, insert new tuples, and update/delete existing tuples (Apply)

CO-4: Create nested SQL queries to retrieve /update data from/to multiple tables (Create)

CO-5: Demonstrate how to write SQL code using Triggers, Assertions, etc. (Apply)

CO-6: Demonstrate Create, Modify and delete virtual tables called views and use them wherever required (Apply)

 CO-7: Experiment PL/SQL codes using cursors, anonymous PL/SQL blocks, stored procedures, and functions (Analyze)

CO-8: Develop a real-time web application using any favorite programming language of his / her choice. The application will use the relational database management system like Oracle, MySQL or any other. The student will use the knowledge gained in the course outcomes (CO-2 to CO-7) for this purpose (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
CO1	2	1	3									
CO2			2	1		3						
CO3	2		1		3							
CO4			2	1		3						
CO5	2		1			3						
CO6	2		1			3						
CO7	1			2		3						
CO8			1		2						3	

Detailed Syllabus:

UNIT-I: Introduction to SQL. Features of SQL, DDL Statements and DML commands.

UNIT-II: Writing simple SQL queries using DDL statements and DML commands

UNIT-III: Introduction to inner, outer and natural joins. Writing nested queries and correlated nested queries to retrieve and update the data.

UNIT-IV: Writing SQL queries using EXISTS, NOT EXISTS, explicit join operation, aggregate functions, group by and having classes.

UNIT-V: Creating virtual tables (views). Using views in SQL queries.

UNIT-VI: PL/SQL programming

UNIT-VII: Mini project: Implement a real-time web application which makes use of database concepts

Reference Books:

1. R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems*, 7th Edition, Pearson Publication, US, 1168 pages.
2. Raghu Ramakrishnan, Johannes Gehrke: *Database management systems,* McGrawHill, Singapore, 1098 pages.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-V)

Course Code: IE303   Title of the Course: Data Base Management systems

L-T-P: 3-0-0   Credits   : 3

Prerequisite Course / Knowledge (If any): Programming Methodology, Data and File Structures, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Explain the fundamentals of relational database management systems (Understand)

CO-2: Explain the relational data model, ER model and relational algebra (Understand)

CO-3: Design the ER models for database applications (Create)

CO-4: Prepare SQL queries from the ER models (Apply)

CO-5: Evaluate the database design aspects by considering normalization principles (Evaluate)

CO-6: Explain concurrency, recovery, security, integrity, Indexing, Hashing, deadlock handling (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1									
CO2	3		1		2							
CO3	2	1	3									
CO4		2	2		3							
CO5		3		1		2						
CO6	3	1	2									

Detailed Syllabus:

UNIT 1: Introduction: Aims and Objectives, Technology involved and current uses of the technology.

UNIT 2: Data Models: Entity-Relationship model, Network model, Hierarchical model.

UNIT 3: Database design: Normalization principles and their uses. Secondary data storage and retrieval techniques.

UNIT 4: Query Processing: Studies on query processing strategies and cost estimation.

UNIT 5: Transaction Processing: Defining Properties and studies on recovery and concurrency. Security and Integrity.

UNIT 6: Distributed Databases: Introduction, Issues on design, concurrency, recovery, deadlock handling and coordinator selection.

Reference Books:

1. A. Silberschatz, H. F. Korth and S. Sudarshan, *Database Systems & Concepts*, 6th Edition McGrawHill Publications, 1376 pages.
2. R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems*, 7th Edition, Pearson Publication, US, 1168 pages.
3. Stefano Ceri, G. Pellagatti: *Distributed Databases Principles & Systems*, McGrawHill, India, 408 pages.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-V)

Course Code: IE305   Title of the Course: Principles of Programming Languages

L-T-P: 3-0-0   Credits   : 3

Prerequisite Course / Knowledge (If any): It is expected that students must have doe one programming language course at undergraduate level

Course Outcomes (COs)

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

CO-1: Understand the design and implementation issues of various programming Paradigms (Understand)

CO-2: Analyze the data times, data and functional abstraction mechanisms (Analyze)

CO-3: Review the sub program control mechanisms in various paradigms (Understand)

CO-4: Examine procedural and object oriented programming features (Apply)

CO-5: Examine functional and logic programming features (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3									
CO2		2		3		1						
CO3	1	2									3	
CO4	2	3	1									
CO5	2	3	1									

Detailed Syllabus:

UNIT-I: Basics of programming languages; language design and implementation issues; impact of machine architectures; what makes a good (or successful) language.

UNIT-II: Common features of programming; elementary data types; encapsulation; inheritance; sequence control.

UNIT-III: Subprogram control; storage management and run-time structures; distributed processing and network programming.

UNIT-IV: Summaries of popular procedural and object-oriented languages; FORTRAN, C; Smalltalk, C++, Java;

UNIT-V: Summaries of popular logic and functional languages; LISP, ML; Prolog; specialised languages such as Postscript, PHP.

Reference Books:
1. T.W. Pratt and M.V. Zelkowitz (2001). "*Programming Languages: Design and Implementation,*" 4th Edition, Prentice-Hall India.

2. Robert W. Sebesta. (2009) "*Concepts of Programming Languages,*" 10th Edition, Pearson Publishing.

School of Computer & Information Sciences
Integrated M.Tech (CSE)
Scheme (I-VI) Winter Semester

II-Semester				
Code	**Course Title**	**Credits**		
	English-II	3-0-0	HSC	CELS/Eng
	Waves and Oscillations,	3-0-0	ESC	Physics
	Math-II	3-0-0	BSC	Maths
IE152	Basket B2 (Level II - Sanskrit /Hindi/Telugu/Urdu)	4-0-0	HSC	Humanities
IE151	Data & File Structures	3-0-0	ESC	SCIS
	Data Structures Lab	2	ESC	SCIS
IE153	*Environmental Science(Mandatory)*	2	OS	Cent. For Earth...
	Physics Lab	1.5	ESC	Physics
		21.5		
IV-Semester				
Code	**Course Title**	**Credits**		
IE251	Computer Based Optimization Techniques	4-0-0	PCC	SCIS
IE252	Internet Technologies (IT)	3-0-0	PCC	SCIS
IE253	Object Oriented Design	3-0-0	PCC	SCIS
IE254	Theory of Computation	3-0-0	PCC	SCIS
IE255	Signals & Systems	4-0-0	ESC	SCIS
IE256	*Java Lab*	2	PCC	SCIS
IE257	*IT Lab*	2	PCC	SCIS
		21		
VI-Semester				
Code	**Course Title**	**Credits**		
IE351	Software Engineering	3-0-0	PCC	SCIS
IE352	Computer Networks(CN)	3-0-0	PCC	SCIS
IE353	Computer Graphics	4-0-0	PCC	SCIS

IE354	Computer Ethics	3-0-0	BSC	SCIS
	Elective-I	3 or 4	PEC	SCIS
IE355	*Software Engineering Lab*	2	PCC	SCIS
IE356	*CN Lab*	2	PCC	SCIS
	Internship (Summer)	3	PROJ	Industry
		23/24		

School of Computer and Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-II)</u>

Course Code: IE252

L-T-P: 3-0-0

Prerequisite Course / Knowledge (If any): Programming Language course (PM)

Title of the Course: <u>Data & File Structures</u>

Credits   : 3

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO-1:   Discuss which data structures are used for static and dynamic allocations. (Understand)
- CO-2:   Solve the problem where in elements can be traversed in either direction and select the suitable data structure for this idea using C/Java Programming Language (Apply)
- CO-3:   Analyze the time taken to solve the given problem by using C/Java programming language (Analyze)
- CO-4:   Assess the solution in terms of efficiency, modularity and well-documented programs in C/Java under Linux environment (Evaluate)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			2	3	1							
CO2					3	2					1	
CO3				3	2		1					
CO4	1			2	3							

Detailed Syllabus:

UNIT-I

Introduction to data structures and data types: Primitive and Non-Primitive types, Arrays- Sparse matrix, Stacks, Queues, Circular queues, Priority queues, Dequeues, Conversions and Evaluations of expressions, Polynomial representation using arrays, Time complexity analysis of algorithms with respect to data structure operations

Unit-II

Linked Lists: Linked stacks and queues, Circular and Doubly linked lists, Polynomial representation using linked lists.

Unit-III

Trees and Graphs: Binary Trees, Tree Traversal, Binary Search trees and basic operations, Heaps, AVL Trees, height balanced trees. , Graphs – Representation of the graphs, Graph Traversals.

Unit-IV

Sorting and Searching Mechanism: Selection sort, Bubble sort, Insertion sort, Merge sort, Quick sort, Heap sort. Linear Search, Binary Search, Hash Tables.

UNIT – V

File structures: Concepts of Double Buffering and Block Buffering, Indexing, B-tree needs, properties, creations and Uses, B+ trees

Suggested reading::

1.   Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press,   1978.
2.   Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.
3.   Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole   Pub. Co. 1987.
4.   Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata   McGraw Hill.
8.   Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in   C; Prentice - Hall of India, 1999.
9.   The *C Programming* Language by Brian W.Kernighan , Dennis M. Ritchie
10.   Michael J. Folk and Bill Zoellick, "File Structures" (Second Edition).
11.   Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.
12.   Schaum"s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.
13.   Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

School of Computer and Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-II)</u>

Course Code: IE257   Title of the Course: <u>DFS Lab</u>

L-T-P: 0-0-3   Credits   : 2

Prerequisite Course / Knowledge (If any): Programming Language course (PM)

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO-1: Solve a given problem by choosing appropriate data structures (Apply).

- CO-2: Select suitable data structure for given idea and propose an appropriate solution (Understand)

- CO-3: Analyze the time taken to solve a given problem (Analyze)

- CO-4: Assess the solution in terms of performance and standard programming principles under Linux environment (Evaluate)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1					3	2					1	
CO2	1	2		3								
CO3				3	2		1					
CO4	1			2	3							

Detailed Syllabus:

UNIT-I: Implementation of Stacks and different types of Queues data structures using arrays, Conversions and Evaluations of an expressions.

Suggested Assignments:

Large integer arithmetic. Arithmetic operations are to be performed on very large integers of N digits (where 0 < N < 20). Write C program that performs the operations of multiplication and division on such large integers.

Two stacks using single array. Implement two stacks using a single array such that neither overflows unless total number of elements in both the stacks is equal to the size of the array.

Deque. Deque is a queue which allows insertions and deletions at both ends. Write a C program that implements deque using both *arrays* and *linked list*. Each implementation should provide support for basic operations such as (i) Insertion of an element at both ends (ii) Deletion of an element from both ends (iii) Displaying all elements of deque.

Evaluation of arithmetic expression. Write a C program that reads an input arithmetic expression in (INFIX, PREFIX, POSTFIX) notations and outputs its result.

Arithmetic expression notation conversion. Write a C program that reads an input arithmetic expression in infix notation (fully parenthesized) and converts it into an output arithmetic expression in postfix notation.

UNIT-II: Implementation of basic Linked List operations such as addition, updation, deletion, searching and traversal of all elements of the list

Suggested Assignments:

Implement Stack data structure (using linked list) operations. Write C program to perform the basic operations on the stack.

Implement Queue data structure (using linked list) operations. Write C program to perform the basic operations on the queue.

Doubly Linked List operations. Linked list (doubly) is an important data structure for dynamic allocation wherein elements can be traversed by either direction. Write C program to perform the basic operations on the linked list.

Union and Intersection. Given two linked lists of numbers, write a program that finds a resultant linked list which is union of the two input linked lists and another resultant linked list which is intersection of the two input linked lists.

Detecting cyclic linked list. Write a C program that detects whether a given linked list is cyclic or not, if yes, then return the node where the cycle begins.

UNIT-III: Implementation of Trees and Graphs of basic operations. Implementation of Adjacency Matrix and List Representation. Breadth and Depth First Search

Suggested Assignments

Tree traversals. Write C program to display a tree using all the methods of traversals: (i) Inorder traversal, (ii) Preorder traversal, (iii) Postorder traversal

Binary Search Tree. Binary Search Tree is an important data structure for dynamic allocation and optimized searching. Write C program to perform the basic operations on binary search

tree (BST): (i) Adding, (ii) Updating (iii) Deleting (iv) Search for an element (v) Displaying all elements (in-order).

Building heap. Using a C program, build a *max-heap*, given N random integers. Display the heap thus formed in its in-order form.

Heap sort. Using the heap data structure, sort the given N random integers.

AVL Trees. Using a C program, perform the following operations on the *AVL tree* data structure: (i) Inserting, (ii) Deleting, (iii) Update, (iv) Searching, (v) Displaying an element

Topological Sorting. In a university curriculum, often each course has a set of pre- requisites. Given a set of courses along with their respective set of pre-requisites, prepare a curriculum such that no course appears before its pre-requisite.

Using Linked List and BST: Construction of a city database using a linked list and binary search tree and the appropriateness of these structures under various demands for the data.

UNIT-IV: Sorting and Searching Mechanism: Linear Search, Binary Search, Implementation of Bubble, Insertion, Quick, Selection Sort.

Suggested Assignments

Write C programs to perform both linear and binary search on a given random set of integers. The following points should perform by the program

- o Take as input an integer, N, which would decide number of integers to be processed and another input an integer, X ($0 < X < N+1$), which is the key to be searched
- o Randomly generate N integers whose values are between 1 to N, multiple entries are allowed
- o Output all the indexes (positions) of key in given set of random integers
- o Count number of comparisons in the linear and binary searching process, please note comparisons involved in sorting process (in case of binary search) are not to be included
- o Output the result in following table:-

Input size (N)	Number of Comparisons	
	Linear Search	**Binary Search**
10		
30		
50		
70		
100		

Merge Sort. Logging activity files of two users are given as input, merge them into a single file. Assume that the format of logging activity file is two column with first column representing the date-time record and second column the event description. Merging is to be done with respect to the date-time record field.

UNIT-V: File Structures: Implementation of B Trees and B+ Trees

Suggested Assignments

Improved file copying. Modify the file copy program to avoid over writing the existing target file, instead if target file has some contents, then target file is appended by contents of source file.

File handling API. Write a program to read and write a file using following combinations of functions:-

    a. fgetc( ) and fputc( )
    b. fprintf( ) and fscanf( )
    c. fgets( ) and fputs( )
    d. fread( ) and fwrite( )

Suggested readings::

1.   Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.
2.   Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.
3.   Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.
4.   Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.
5.   Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.
6.   The *C Programming* Language by Brian W.Kernighan , Dennis M. Ritchie
7.   Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.
8.   Schaum"s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.
9.   Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.
10.   Michael J. Folk and Bill Zoellick, "File Structures" (Second Edition).
11.   Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.
12.   Schaum"s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.
13.   Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

School of Computer and Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-IV)</u>

Course Code: IE255   Title of the Course: <u>Signals and Systems (S&S)</u>

L-T-P: <u>4-0-0</u>   Credits   : 4

Prerequisite Course / Knowledge (If any): The course is aimed for students who already have knowledge of Mathematics: Calculus, Differential Equations, Linear Algebra and Completed Programming Methodology

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO-1: Classify Signals, Systems and identify LTI systems (Understand)
- CO-2: Discuss Discrete-time systems and LTI systems (Understand)
- CO-3: Develop (Derive) Fourier series for continuous time signals (Create)
- CO-4: Develop (Derive) Fourier transform for different time domain signals, Discrete Time (DT) or Continuous time (CT) systems(Create)
- CO-5: Develop (Derive) Convolution Sum and Convolution for DT and CT (Create)
- CO-6: Analyze DT systems and their realization using Z-transforms (Analyze)
- CO-7: Describe probability concepts and applications to Random Signals and noise with statistical properties (mean, variance, auto-correlation) of random variables (Understand)
- CO-8: Generate signals using MATLAB and process the same using various types of filters and systems.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2		1								
CO2	3		2			1						
CO3				1	2					3		
CO4				1	2					3		
CO5				1	2					3		
CO6		3		2							1	
CO7	3	2		1								
CO8			2		3		1					

Detailed Syllabus:

UNIT -I: Introduction to signals: Brief introduction to signals and their applications. Analog and digital signals. Continuous and discrete signals. Types of signals: Constant, Step, Ramp, Impulse, Dirac, periodic, exponential. Operations on Signals: Magnitude Scaling, Time shifting, Sampling.

UNIT -II: Introduction to systems: Properties: Memory less, stability, Linearity, Causality, Time invariance. Linear Time invariant systems, Causality and Causal Systems, system stability Integration of systems: Cascading, parallel, feedback.

UNIT -III: Mathematical Concepts: Review of probability, statistics and differential equations. Introduction to Random Variables, Probability and Cumulative distribution curves, Correlation, Covariance, Convolution of signals

UNIT-IV: Signal Processing Techniques: Fourier Transforms, Fourier series, Z-Transforms and Laplace Transforms and their applications

UNIT-V: Digital signal Processing: Types of Filters, Signal responses to filters, Finite Impulse Response (FIR) and Infinite Impulse Response (I IR). Nyquist criterion, Fast Fourier Transform, Discrete Fourier Transforms, decimation-in-time (DIT) FFT, Inverse Fourier transforms, Types of window functions and frequency responses.

Reading Material

Text Books

1. "Signals and Systems" Simon Haykin and Barry Van Veen, 2nd Ed., John Wiley & Sons, ISBN: 978-0-471-16474-6 October 2002

2. "Linear Systems and Signals", B. P. Lathi, 2nd Ed. Oxford University Press, 2005

3. "Probability, Random Variables & Random Signal Principles," P.Z. Peebles, McGraw Hill Education; 4th edition, ISBN-10: 9780070474284, July 2017.

5. "Signals and Systems", Alan S. Willsky, S. Hamid Nawab, Alan V. Oppenheim ISBN: 9789332550230, 9332550239, 2nd Edition, Pearson India, 2015

Reference Books:

1."Communication Systems", Simon Haykin and Michael Moher, Wiley; Fifth edition, June 2009

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-IV)

Course Code: IE253/IE256   Title of the Course: Object Oriented Design (OOD)/

                                                                                Java Lab

L-T-P: 3-0-0/0-0-3   Credits   : 3/2

Prerequisite Course / Knowledge (If any): Basic Linear Algebra and Calculus

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO-1: Describe the object oriented design concepts. (Understand)
- CO-2: Analyse a given computational problem. (Analyze)
- CO-3: Design classes for a given Computational problem (Create)
- CO-4: Apply the UML concepts to model a problem. (Apply)
- CO-5: Create Java programs for the object oriented design of the given problem. (Create)
- CO-6: Create Java programs which require to use advanced features of Java such as Exception handling, Interfaces, GUI package etc. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs)
and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1									
CO2		1	2								3	
CO3		2	3			1						
CO4	2					3				1		
CO5			1							2	3	
CO6			1							2	3	

Detailed Syllabus:

UNIT -I: Object Oriented Fundamentals and Modeling: Need for OOP paradigm, What is object orientation and OO Development, Modelling, Abstraction, Three models of OOD, Object and class concepts, Links and Association Concepts, Generalization and Inheritance, N-ary associations, Aggregation, Abstract classes, multiple inheritance, metadata, Reification, Constraints, Derived data, packages.

UNIT -II: Java Basics: History of Java, java data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT -III: State Modelling and Interaction Modelling: Events, states, Transitions and Conditions, State Diagram, Nested state diagram, Concurrency Use-Case model, Sequence model, Activity model, procedural sequence model, Relation between class, state model and interaction model.

UNIT -IV: Hierarchical abstractions, Generalization and Aggregation, Base class object, subclass, subtype, substitutability, forms of inheritance-specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism-method overriding, abstract classes, the Object class Exception handling in Java: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling, Exploring java.util.

UNIT -V: System Design, Class Design and Implementation Modelling: Overview of system design, performance estimation, reuse plan, Subsystems, Management of data storage and global resources, software control strategy and boundary conditions. Overview of class design, Realizing use-cases, designing algorithms, refactoring and design optimization, Overview of implementation, fine tuning of classes, generalization, and realizing associations.

Java Lab Exercises:

1. Basic features of Java such as data types, control structures, loops and arrays (2 Lab Sessions)
2. Working with classes, constructors, methods, objects. (2 Lab Sessions)
3. Using Inheritance, Polymorphism, Interfaces and abstract classes (3 Lab Sessions). Exercises can include a case study depicting OO application design using polymorphism and inheritance. (For example) Developing a Solitaire Application (Chapter 8 of "Introduction to Object Oriented Programming by Timothy Budd")
4. Exception Handling (1 or 2 Lab Session)
5. Exploring Java IO Package (2 or 3 Lab Sessions)
6. Java GUI Programming such as Applets (2 Lab Sessions)
7. Miscellaneous Topics such as generic classes, collection framework and java.util packages (1 or 2 Lab Sessions)

Reference Books:

1. Herbert Schildt, Java: The complete reference, McGraw hill.
2. Paul J. Deitel and Harvey M. Deitel , Java: How to Program, Prentice Hall.
3. T. Budd, Understanding OOP with Java, Pearson Education.
4. Michael Blaha and James Rambaugh, Object Oriented Modelling and Design with UML, 2nd edition, Eastern Economy Edition.
5. Herbert Schildt, Java: A Beginner's Guide, McGraw Hill Education (India) Private Limited.
6. Bruce Eckle, Thinking in Java, Prentice Hall.
7. Joshua Bloch, Effective Java, Createspace Independent Pub.
8. Kathy Siera, Head First Java, O'Reilly Media

School of Computer and Information Sciences

Name of the Academic Program: <u>Integrated M.Tech (CSE) (IMTECH-IV)</u>

Course Code: IE251   Title of the Course: <u>Computer Based Optimization Techniques</u>

L-T-P: <u>4-0-0</u>   Credits   : 4

Prerequisite Course / Knowledge (If any): Basic Linear Algebra and Calculus

Course Outcomes (COs)

After completion of this course successfully, the students will be able to......

- CO-1: Formulate optimization problems as Linear Programming Problem (Create)
- CO-2: Solve Linear Programming Problem using graphical method. (Apply)
- CO-3: Solve LP using Simplex method and its variants (Apply)
- CO-4: Solve Special classes of LP (Apply)
- CO-5: Solve Integer Linear Programming Problems using Cutting Plane and Branch and Bound method (Apply)
- CO-6: Discuss the Optimality Principle (Understand)
- CO-7: Apply Dynamic Programming Technique to solve problems(Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1				3		2				1		
CO2			2	1		3						
CO3			2	1		3						
CO4			2	1		3						
CO5			3	2					1			
CO6	3	2	1									
CO7					1	3					2	

Detailed Syllabus:

This course consists of four modules

UNIT-I: Linear Programming: Mathematical formulation of Linear Programming problem, Canonical and standard forms of Linear Programming problem, Solution by Graphical and Simplex methods, Revised Simplex method, Two phase & Big M –method, Duality, Primal –Dual relationship, Dual Simplex method.

UNIT-II: Special Types of Linear Programming Problem: Transportation problem, Northwest corner method, Stepping stone method, Unbalanced transportation problem, Assignment problem, The Hungarian method

UNIT-III: Integer Programming: Integer Linear Programming problem, Mixed Integer Linear Programming problem, Cutting Plane method, Branch and Bound Technique

UNIT-IV: Dynamic Programming: Bellman's Principle of optimality, General theory of solving multistage decision problems using Dynamic Programming, Application of General Theory to specific problems such as the Travelling Salesman problem.

Suggested Reading

1. F S Hillier and G J Lieberman, Introduction to Operations Research, 7th edition, McGraw Hill, 2000

2. H A Taha, Operations Research –An Introduction, 8th Edition, Pearson Prentice Hall, 2007

Additional Reading

1. G Hadley, Linear Programming, Narosa Publishing
2. Harvir Singh Kasana and K D Kumar, Introductory Operations Research: Theory and Applications, Springer Science & Business Media, 2004

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-IV)

Course Code: IE254

Title of the Course: Theory of Computation (TOC)

L-T-P: 3-0-0

Credits   : 3

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO-1: Review various models of computation, their capabilities and limitations (Understand)
- CO-2: Outline the suitability of different models of computation in various application scenarios. (Analyze)
- CO-3: Categorize decidable and undecidable problems (Create)
- CO-4: Develop programs simulating different models of computation (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1		2								
CO2	1	3		2								
CO3	1		1	3								
CO4						1				3	2	

Detailed Syllabus:

UNIT-1: Finite State Automata and Regular Languages (RL)

Preliminaries on Alphabets and Languages, Finite State Automata and Regular Languages (RL): Definition and examples; Regular expressions (RE), Non-deterministic finite automata (NFA), λ-NFA and Deterministic Finite automata (DFA); Equivalence of RE, NFA and DFA; Conversions from RE to (λ -) NFA to DFA to RE; Minimal DFA; Moore machine, Melay machine; Closure properties of RL; Pumping lemma for RL.

UNIT-II: Push Down Automata and Context-Free Languages (CFL)
CFL: Definition and examples; Grammar formalism for regular languages, Context free grammar (CFG), Derivation trees, Ambiguity, Normal forms; Push down automata(PDA) (deterministic and non-deterministic); Equivalence of CFG and PDA; CYK Algorithm; Closure properties of CFL; Pumping lemma for CFL.

UNIT-III: Turing Machine (TM)
Definition of TM; Examples; Variants of TM: Multi-tape and other variants of TM; Post Machine, Two-Stack PDA; Nondeterministic TM; Equivalence; Church-Turing Thesis; Universal Turing Machine

UNIT-IV: Decidability and Undecidability
Definition of decidability; decidable problems concerning RL, CFL; Recursive and recursively enumerable languages; Undecidability; The Halting problem; Cantor's diagonalization argument; Examples of undecidable problems: Post's correspondence problem; Chomsky Hierarchy

Suggested Reading

1. J. Hopcroft, R. Motwani and J. Ullman, Introduction to Automata Theory, Languages and Computation. 3rd Edition, Pearson, 2014.
2. D.I.A. Cohen, Introduction to Computer Theory. 2nd Edition, Wiley India, 1991.
3. J.C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw Hill, 2003

Additional Reading

1. H. Lewis and C. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall, 1998.
2. M Sipser, Introduction to Theory of Computation, Thomson Learning, 2014.
3. Peter Linz, An introduction to formal languages and automata, Jones and Barlett Publishers, 2016.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-IV)

Course Code: IE302/IE307   Title of the Course: Internet Technologies/IT Lab
L-T-P: 3-0-0/0-0-3   Credits   : 3/2

Prerequisite Course / Knowledge (If any): -- None

Course Outcomes (COs)

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

CO-1: Apply protocols related to network application layer for internet applications such as decentralized communications and remote data sharing. (Apply)

CO-2: Analyze a web page and identify its elements and attributes, create web pages using scripting languages , cascading styles sheets, and build dynamic webpages using JavaScript. (Create)

CO-3: Develop interactive web applications using server side and database technologies (Create)

CO-4: Create schemas and documents using markup languages, design and develop Lightweight data-interchange format s for exchange of data between client and server applications. (Create)

C0-5: Analyze and apply search engine services for web applications (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1		2			3						
CO2		3	1		2							
CO3			2		1	3						
CO4	2		1		3							
CO5	1				3	2						

Detailed Syllabus:

UNIT- I: Application Layer Protocols: HTTP, Proxy Servers, SMTP, POP, IMAP, SSH, FTP, Peer-to-Peer protocols such as BitTorrent, Distributed Hash Tables.

UNIT-II: Client-Side Technologies: HTML, CSS, PHP, JavaScript, XML, Document Object Model (DOM), Dynamic Content, Cookies.

UNIT-III: Connection to Server and Server-Side Technologies: Databases (MySQL/MongoDB) and JDBC, Servlets, JSP, NodeJS..

UNIT- IV: Advanced Client-Side Technologies: Asynchronous JavaScript and XML (AJAX), JQuery, JavaScript Object Notation (JSON), Google Web Toolkit, [Ruby on Rails1], GoJS, Firebug.

UNIT-V: Advanced Web Applications: Search Engines and their algorithms, Google Maps and building your own Google Maps, Keyhole Markup Language (KML) on Google Earth.

Reference Books:

1. Core Servlets and Java Server Pages (JSP), by Marty Hall, Prentice Hall, 2nd edition (2003).
2. Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX by Elliotte Rusty Harold, Addison-Wesley Pub Co; 1st edition, 2002. (Available online at http://cafeconleche.org/books/xmljava/
3. Glee Harrah Cady, Pat McGregor: Mastering the Internet, BPB, Sybex 1996.
4. Alan Simpson: HTML Publishing Bible, IDG Books, Comdex Computer Publishing, A Division of Pusthak Mahal, 1996.
5. Bryan Pfaffenberger: Publish on the Web, AP Professional, 1996.
6. Clayton Walnum: Java by Example, Que 1996.
7. Marty Hall: The Core Web Programming, Prentice-Hall, 1998.
8. J. Niederst: Web Design in a Nutshell, O'Reilly Associates,

Name of the Academic Program: **Integrated M.Tech (CSE) (M.Tech-VI)**

Course Code   : IE352/IE356   Title of the Course   : **Computer Networks**

L-T-P   : 3-0-0/0-0-3   Credits   : 3/2

Prerequisite Course / Knowledge (If any): C Programming, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Distinguish between multiplexing techniques (Understand)
- CO-2: Evaluate the different types of switched networks (Analyze)
- CO-3: Explain the functionalities media access for data-link and network protocols. (Understand)
- CO-4: Apply IP addressing and routing algorithms to design networks by subnetting/supernetting (Apply)
- CO-5: Describe the essential principles such as reliable data transfer, flow control, congestion control of a transport layer protocol (Understand)
- CO-6: Predict the topology given the routing protocol messages (Apply)
- CO-7: Analyze and capture network traffic using simulation tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2		1								
CO2		2		3						1		
CO3	3	2								1		
CO4	1		2			3						
CO5	3	2									1	
CO6				1		2					3	
CO7		1		2	3							

Detailed Syllabus:

UNIT - I: Physical Layer: Modulation Techniques: Amplitude, Frequency and Phase, ADSL, Multiplexing Techniques: Frequency division multiplexing, Time division multiplexing, wave length division multiplexing, Differential PCM, Switching Techniques: Circuit, message and packet switching.

UNIT - II: Data Link Layer: PPP, PPPoE, MAC Layer: Ethernet (incl. manchester encoding), Switched Ethernet, VLANs, Spanning Tree Protocol.

UNIT - III: Network Layer: Data Plane: Internet Protocol Addressing: CIDR, Internet Protocol Datagram (including fragmentation and reassembly, routing options), IP Forwarding Algorithm, ARP, ICMP (including ICMP Redirect, ICMP Path MTU discovery, ICMP Destination Unreachable options).

UNIT - IV: Transport Layer: UDP, TCP sliding window protocol, TCP connection establishment, TCP reliability including cumulative and delayed acknowledgements, Nagle algorithm, Karn's algorithm for RTT and RTO estimation, TCP AIMD Congestion Control Algorithm, TCP half-close connections including TCP keepalive timer and probe timer, TCP Fast Retransmit and Fast Recovery.

UNIT - V: Network Layer: Control Plane: Distance Vector Algorithm and Routing Information Protocols V1 and V2, Link State Algorithm and Open Shortest Path First Protocol (OSPF).

UNIT - VI: Application Layer: Domain Naming System (DNS) and Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP.

Reference Books:
1. James F. Kurose and Keith W. Ross. Computer Networking: A top-down approach, 6th edition, Pearson Education.
2. Douglas Comer. Computer Networks and Internets Sixth Edition, 2014. ISBN 0133587932/9780133587937, Pearson Education.
3. Douglas Comer. Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 6th edition, 2013. ISBN-10: 0-13-608530-X ISBN-13: 9780136085300, Pearson Education.
4. Kevin R. Fall and W.Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, 2/E, 2012, ISBN-10: 0321336313 ISBN-13: 9780321336316,Pearson Education.
5. Radia Perlman. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2/E, 2000, ISBN-10: 0201634481 ISBN-13: 9780201634488.Pearson Education.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (M.Tech-VI)

Course Code: Title of the Course: **Computer Networks Lab**

L-T-P   : 0-0-3   Credits   : 2

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO1: Analyze network traffic after capture through any of the tools such as tcpdump/wireshark

CO2: Troubleshoot network problems through analysis of network traffic

CO3: Implement core network functions such as forwarding, reassembly of IP fragments, TCP sliding window protocol (Create)

CO4: Configure network elements such as routers in Linux and Quagga/Cisco ios

CO5: Learn basic network set up in GNS3

Detailed Syllabus:

1. Implement the IP fragmentation and reassembly algorithm.

2. Implement the IP forwarding algorithm.

3. Implement the simplest sliding window protocol of TCP.

4. Connect two systems using a switch and configure private IP addresses to the systems and ping them from each other. Using Wireshark, capture packetsand analyze all the header information in the packets captured.

5. Convert a system with two network interface cards (NICs) into a router by configuring each NIC in a different LAN and enabling forwarding. Use two switches to connect one NIC each of the router to these two switches. Connecttwo other systems, one each to each switch. Now, we have two VLANs. Ping from one system to the other through the router after configuring the required default routes in the hosts and static routes in the router.

6. Install Telnet Server on one of the systems connected by a switch and telnet to it from the other system. Using Wireshark, capture the packets and analyze the TCP 3-way Handshake for connection establishment and tear down.

7. Connect two hosts using a switch in GNS3 and configure them to belong to the same LAN and ping each other. Capture traffic on the hosts through wireshark and analyze the traffic.

Reference Books:

- "Hands-on Networking" by Douglas Comer, Pearson Education, ISBN: 9780131486966, 9780131486966, Edition: 2004.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (I.M.Tech-VI)

Course Code   : IE355   Title of the Course: **Software Engineering Lab**

L-T-P   : 0-0-3   Credits   : 2

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate level.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Create user stories (Create).

- CO-2: Develop test plans for test first development (Create).

- CO-3: Design & develop the stories (Create).

- CO-4: Create the documentation (Create).

- CO-5: Develop Software requirements specification document (Create).

- CO-6: Apply object oriented and structured paradigm (Apply).

- CO-7: Generate test reports (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			3		2						1	
CO2			3		2						1	
CO3			2		3						1	
CO4			3		2						1	
CO5		2			3						1	
CO6					2	3					1	
CO7				2	3						1	

Detailed Syllabus:

For a given case study/problem statement, the following deliverables are to be realized

- Define stories
- Identify tasks and develop test plan for stories/task (with the help of specifications)
- Design and develop increments
- Test the increments and release the increment
- Apply object oriented and structured modelling
- Implement the case study for plan driven approach by writing use case specification, designing the system and implementing the same.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

Name of the Academic Program: Integrated M.Tech (CSE) (M.Tech-VI)

Course Code   : IE351   Title of the Course   : **Software Engineering**

L-T-P   : 3-0-0   Credits   : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Explain the models of software development process (Understand)
- CO-2: Evaluate the appropriateness of different models of software development for their application in various domains (Evaluate).
- CO-3: Apply the requirements engineering to software systems. (Apply)
- CO-4: Describe Software Architectures (understand).
- CO-5: Assess the applicability of software architectures for various combinations of non-functional requirements (Evaluate level).
- CO-6: Apply object oriented and structured and structured paradigms to design software systems (Apply).
- CO-7: Apply testing strategy to test software applications (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2				1						
CO2	2			3	1							
CO3						3				1	2	
CO4	3	2		1								
CO5				3	1		2					
CO6	1		3		2							
CO7	1		3								2	

Detailed Syllabus:

UNIT-I: Introduction to Software Engineering

Need of software engineering, systems engineering, challenges in software engineering, Software process models, quality characteristics of software systems, Ethics in Software Engineering.

UNIT-II: Requirements Engineering

Requirements engineering process, requirements specification, structured and object oriented analysis

UNIT-III: Software Design

Architectural design, detailed design, Structured and object oriented design, user interface design

UNIT-IV: Software Testing

Verification, Validation, testing techniques, Testing Process

UNIT-V: Tools and Evolution

CASE Tools, Reverse engineering, Reengineering and Configuration management.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition.
2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition
3. Grady Booch, James Rumbaugh, Ivor Jacobson(2005), "*The Unified Modeling Language User Guide*", 2nd Edition, Addison Wesley Professional.US

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (M.Tech-VI)

Course Code   : IE353   Title of the Course   : **Computer Graphics**

L-T-P   : 3-0-2   Credits   : 3/1

Prerequisite Course / Knowledge (If any): Theory of Computation & Data Structures

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Discuss the features of OpenGL programming. (Understand)
- CO-2: Model reasonably complex scenes with multiple objects. (Analyze)
- CO-3: Demonstrate simple animations with translation, rotation etc. or with fractal surfaces changing over time. (Apply)
- CO-4: Develop basic screensavers. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2		1								
CO2					3				2	1		
CO3			1	2	3							
CO4			3		2		1					

Detailed Syllabus:

UNIT - I: Introduction: History, Advantages, Applications, Graphics I/O Devices, Raster Graphics, Graphics Packages and Libraries, Line and Circle Drawing Algorithms, Scan Conversion, Polygon Filling.

UNIT - II: Geometric Transformations and Clipping: 2D Transformations, Homogeneous Coordinate System, 3D Transformations, Plane Geometric. Projections, Viewing Transformations, Line and Polygon Clipping..

UNIT - III: Curves and Surfaces: Parametric Representation of Curves, Cubic Splines, Bezier Curves, Bsplines, Parametric Surfaces, Surfaces of Revolution, Sweep Surfaces, Quadric Surfaces, Fractal Curves and Surfaces.

UNIT - IV: Realism in 3D Graphics: Hidden Line and Hidden Surface Removal Algorithms, Illumination Models, Phong and Gouraud Shading.

Reading Material

Suggested Reading

1. D. F. Rogers: Procedural Elements for Computer Graphics, Tata McGraw Hill

2. D. F. Rogers and J.A. Adams: Mathematical Elements for Computer Graphics, Tata McGraw Hill

3. J. D. Foley, A. VanDam, S.K. Feiner. and J.F. Hughes: Computer Graphics: Principle and Practice, Pearson Education.

4. Z. Xiang and R. Plastock: Computer Graphics, Tata McGraw Hilledition, Pearson Education.

Additional Reading/References

1. E. Angel: OpenGL - A Primer, Pearson Education

2. D. Shreiner, G. Sellers,J. Kessenich, B. Licea-Kane: OpenGL Programming Guide, Pearson Education.

3. T. McReynolds and D. Blythe: Advanced Graphics Programming Using OpenGL, ElsevierNehe OpenGL Tutorials @ http://nehe.gamedev.net..

Name of the Academic Program: Integrated M.Tech (CSE) (M.Tech-VI)

Course Code   : IE354   Title of the Course   : **Computer Ethics**
L-T-P   : 3-0-0   Credits   : 3

Credits: 3

UNIT - I − Morality and Ethics:

Moral code of behaviour, Purpose of Law, Penal code, Morality and the law; Ethical theories, Functional definition of ethics, Ethical reasoning and decision making, codes of ethics, technology and values.

UNIT - II − Ethics and the Professions:

Professional code of conduct, Professional decision making and ethics, Whistleblowing, harassment and discrimination, ethical and moral implications of profession.

UNIT - III − Anonymity, Privacy and Civil Liberties:

Anonymity in the Internet, its advantages and disadvantages, Types and value of Privacy, Privacy protection and civil liberties, implications for privacy of the Aadhaar database [4], Workplace privacy and surveillance, Ethics for privacy, Ethical and legal basis for privacy. Case studies of Cambridge Analytica [1] [2], SECC in India [3].

UNIT - IV − Intellectual Property Rights, Computer Crime and Ethics:

Computer Products and services, foundations of Intellectual Property Rights (IPR), Ownership, IPR crimes, Protecting computer software under IPR, Transnational issues. Digital evidence, preserving digital evidence, analysis of digital evidence, ethical implications and responsibilities in computer forensic investigations.

UNIT - V − Social Implications of Information Technology:

Advances in Artificial Intelligence, AI and ethics, bias in AI algorithms [7], [8], [9], [10], Online social networks, Ethical issues in online social networks, security and crimes in online social networks; Internet of Things (IoT) and Location tracking [5], ethics of tracking, ethics of personalised search/ advertisements in online social networks and search engines, self-driving vehicles and ethical/legal issues [12] [13], surveillance and ethical issues [11], use of biometrics for authentication, ethical implications of biometric technologies [14] [15], fake news and implications of its spread via social media platforms, digital divide and its implications, sexual predators and pornography on the Internet and its implications.

TEXTBOOK

Ethical and Social Issues in the Information Age, Joseph Migga Kizza, 5th ed. 2013

Edition, ISBN-10: 1447149890, ISBN-13: 978-1447149897

https://www.utc.edu/center-academic-excellence-cyber-defense/course-listing/ cpsc3610.php

References

[1] https://theprint.in/politics/exclusive-inside-story-cambridge-analytica-actually-india/44012/

[2] https://thewire.in/tech/what-we-know-and-what-we-dont-about-what-cambridge-analytica-did-in-india

[3] https://www.huffingtonpost.in/entry/aadhaar-national-social-registry-database-modi-in-5e6f4d3cc5b6dda30fcd3462

[4] https://www.huffingtonpost.in/entry/one-year-after-aadhaar-judgement-we-are-still-not-listening-in-5d8bb628e4b0ac3cdda24659

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6596313/

[6] https://yjolt.org/sites/default/files/21 yale j.l. tech. 106 0.pdf

[7] https://hbr.org/2019/10/what-do-we-do-about-the-biases-in-ai

[8] https://www.nytimes.com/2019/11/19/technology/artificial-intelligence-bias.html

[9] https://aibusiness.com/three-notable-examples-of-ai-bias/

[10] https://thewire.in/tech/artificial-intelligence-has-a-gender-bias-problem-just-ask-siri

[11] https://www.theguardian.com/books/2019/feb/02/age-of-surveillance-capitalism-shoshana-zuboff-review

[12] https://www.nature.com/articles/d41586-018-07135-0

[13] https://www.newyorker.com/science/elements/a-study-on-driverless-car-ethics-offers-a-troubling-look-into-our-values

[14] https://towardsdatascience.com/how-ethical-is-facial-recognition-technology-8104db2cb81b

[15] https://www.technologyreview.com/f/615068/facial-recognition-european-union-temporary-ban-privacy-ethics-regulation/

School of Computer & Information Sciences
Integrated M.Tech (CSE)
Scheme (VII-IX)
Monsoon Semester

VII-Semester				
Code	**Course Title**	**Credits**		
	Network Programming	3-0-0	PCC	SCIS
	Essentials of AI	3-0-0	PCC	SCIS
	Distributed Computing	3-0-0	PCC	SCIS
	Elective –II	3 or 4	PEC	SCIS
	Elective –III	3 or 4	PEC	SCIS
	NP Lab	2	PCC	SCIS
	EAI Lab	2	PCC	SCIS
		19/21		

IX-Semester				
Code	**Course Title**	**Credits**		
	Project	6	PROJ	SCIS

Name of the Academic Program: Integrated M.Tech (CSE) (M.Tech-VII)

Course Code   : IE354   Title of the Course   : **Distributed Computing**

L-T-P: 3-0-0   Credits   : 3

Unit-I

1. Introduction.

Distributed computing Concepts, Basic network concepts, Basic operating system concepts.

2. Inter process Communication.

Basic model, Primitives (operations): connect, send, receive, disconnect, Connection-oriented/connectionless, Data marshalling: data flattening, data representation, serialization, Event synchronization, Event diagram, sequence diagram.

3. Distributed Computing Paradigms.

Message passing, client server, P2P, message system, RPC, Distributed Object Paradigm, Object Space, Mobile agent, Network services, Groupware paradigms, SOA, Overview & comparison of each paradigm.

Unit-II

4. The Socket API.

The basic model, Stream-mode (connection-oriented) socket, Datagram socket (connectionless) socket, Java socket API, Using socket to implement a client.

Using socket to implement a server, A simple middleware using sockets, Secure sockets and the Java secure socket extension API.

5. The Client-server Paradigm.

The daytime protocol and a sample client-server suite, The echo protocol and a sample client-server suite, Connection-oriented client-server, Connectionless client-server, Iterative server and concurrent server, Stateful server and stateless server.

6. Group Communications.

Unicast versus multicast, Basic model of group communications, The Java multicast API.

Sample multicast sender program, Sample multicast listener program, Multicast and message ordering, Reliable multicast/broadcast.

Unit-III

7. Distributed objects.

Message passing versus distributed objects, The basic model, Remote procedure call, Remote method invocation, CORBA Historical Perspective

8. Advanced Remote Method Invocations (RMI).

RMI stub downloading, Security policy, Callback.

9. Internet applications.

Basic components and protocols: HTTP, HTML, XML, MIME, web server, browser, web forms, Web document types: static, dynamic, executable, active, CGI: background; interaction and passing of data among browser, web server, and script(s), HTTP Session state information: hidden tags, cookies, session objects, Client-side programming: Applets, JavaScript, Server-side programming: common gateway Interface (CGI), servlets, server pages, Applets, Servlets; session data maintenance, Introduction to Web services and the Simple Object Access Protocol (SOAP), REST Protocol.

Text books:

1. Distributed Computing: Principles and Applications, M L Liu, Pearson Publisher.

2. Distributed Systems Concept and Design, George Coulouris, Jean Dollimore and Tim Kindberg, Pearson Publisher.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-VII)

Course Code: IE 402

Title of the Course: Essentials of Artificial Intelligence/EAI Lab

L-T-P: 3-0-3

Credits : 3/2

Prerequisite Course / Knowledge (If any):

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO-1: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation and learning. (Apply)

CO-2: Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models. (Apply)

CO-3: Demonstrate proficiency in applying scientific method to models of machine learning. (Apply)

CO-4: Apply selected basic AI techniques; judge applicability of more advanced techniques. (Apply)

CO-5: Design and develop a system that act intelligently and learns from experience (Create)

CO-6: Demonstrate an ability to share in discussion of AI, its current scope and limitations and societal implications. (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			2	3	1				
CO2			3		1	2			
CO3	2		3			1			
CO4			3	2	1				
CO5	3	1				2			
CO6	2	1				3			

Detailed Syllabus:

UNIT-I: Introduction to AI: Introduction to artificial intelligence, History of AI, Agents & Environment, The Structure of Agents, Agent programs, simple reflex agents, Model based agents, Goal based agents, utility based agents, Learning agents. Proposing and evaluating AI applications

Case study: Google Duplex.

1) *Russell & Norvig, "Chapter 1: Introduction" in Artificial Intelligence: A Modern Approach, Third Edition*

2) *Chui, et. al, "Notes From the AI Frontier," McKinsey Global Institute, April 2018, file://localhost/available at https::/www.mckinsey.com:~:media:mckinsey:featured_insights:artificial_i ntelligence:notes_from_the_ai_frontier applications and_value of deep learning:mgi_notes-from-ai-frontier_discussionpaper.ashx*

3) *Amadeo, R., June 27, 2018, "Talking to Google Duplex: Google's human-like phone AI feels revolutionaryfile://localhost/"* https::/arstechnica.com:gadgets:2018:06:google-duplex-is-calling-we-talk-to-therevolutionary-but-limited-phone-ai: Unit 2:

UNIT-II: Search & Planning: Problem spaces and search, Knowledge and rationality, Uninformed search Strategies (BFS, Uniform-cost search, DFS, Depth-limited search, iterative deepening, Bidirectional search), Heuristic search strategies (Hill climbing, Simulated Annealing, , A*, memory bounded heuristic search), Heuristic Functions, Local search algorithms, Searching with partial observations, searching with non-deterministic actions, Online search, Search and optimization (gradient descent) , Adversarial search (minmax, Alhha-Beta pruning, stochastic games, partially observable games sate-of-the art game progrma), Planning and scheduling,

Case studies: Playing chess, manufacturing scheduling.

1) "Chapters 3, 4 : Solving Problems by Searching," "Chapter 5: Adversarial search", "Chapter 10.2-10.5: Planning", "Chapter 11: Planning and Acting in the Real World" in Russell & Norvig, Artificial Intelligence: A Modern Approach, Third Edition

UNIT-III: Knowledge Representation & Reasoning: Logic and inference, Propositional theorem proving, Propositional Model checking, Agents based on Propositional Logic, First Order logic, Knowledge Engineering in first-order logic, unification & lifting, Forward chaining & Backward chaining Resolution, logic programming, Ontologies, Bayesian reasoning, Temporal reasoning, ase study: Medical diagnosis

1) Readings in Russell & Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Propositional logic: Chapter 7, First-order logic: Chapter 8, Sections 8.1.2 - 8.2, 8.4, 9.1-9.5, Knowledge representation: Sections 12.1-12.5, 12.7, Quantifying uncertainty: Ch. 13, Probabilistic reasoning: Ch. 14.1-14.2 ○ (Optional) Probabilistic reasoning over time: Ch. 15

UNIT-IV: Machine Learning & Supervised Methods: What is machine learning? Supervised vs. unsupervised learning, The theory of learning, Regression --linear, logistic, ridge. Classification − decision trees, SVM, random forests, Model performance evaluation, non-parametric models, Ensemble learning.

UNIT-V: Machine Learning: Unsupervised Methods: Dimensionality reduction: PCA, Clustering-- k-means, hierarchical clustering. Semi-supervised methods (GANs), Reinforcement learning, Choosing among machine learning techniques.

1) *Chapter 18 in Russell & Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition*

2) Chapter 5.8: "Unsupervised Machine Learning," in Goodfellow, I.,Bengio, Y. and Courville A., Deep Learning, 2016.

3) Russell & Norvig, Chapter 21 "Reinforcement Learning" in Artificial Intelligence: A Modern Approach, 3rd Edition

UNIT-VI: Natural Language Understanding: Intro to natural language understanding, Language Models, Information retrieval, Information Extraction, Phase Structure grammars, Syntactic Analysis, Augmented grammars and semantic interpretation, Machine translation, Sentiment analysis, Hidden Markov Models, Chatbots, Natural language generation, Speech synthesis, Case study: Google Duplex (revisited)

1) Russell & Norvig, "Chapter 22: Natural Language Processing" in Artificial Intelligence: A Modern Approach, Third Edition.
2) Collobert et al. "Natural Language Processing (Almost) from Scratch," Journal of Machine Learning Research, 2011 available at https://arxiv.org/pdf/1103.0398.pdf
3) (Optional) G Golderg, Y. Neural Network Methods for Natural Language Processing Synthesis Lectures on Human Language Technologies, April 2017, freely available monograph at https://doi.org/10.2200/S00762ED1V01Y201703HLT037
4) (Optional) Feldman, R, "Sentiment Analysis Tutorial, IJCAI-13, 2013, http://ijcai13.org/files/tutorial_slides/tf4.pdf
5) Russell & Norvig, "Chapter 15.3: Hidden Markov Models" and "Chapter 22: Natural Language for Communication" in Artificial Intelligence: A Modern Approach, Third Edition.

. UNIT-VII: AI in the Enterprise, Ethical & Legal Considerations in AI: Privacy, Bias, AI and the future of work, Appropriate uses of AI, Infrastructure for AI: Parallel & distributed computing for scalability, Resolving technical tradeoffs.

. Case Study: Uber & Facebook

1) Beyer, D, "AI and Machine learning in industry," 2017, download from http://www.oreilly.com/data/free/ai-machine-learning-in-industry.csp?cmp=tw-data-free-article-lgen_tw_free_ebook_as ⬚
2) Jerome, J, "Why AI may be the next big privacy trend,"
3) https://iapp.org/news/a/why-artificial-intelligence-may-be-the-next-big-privacy- trend/, 2016
4) Burt, A . "How will the GDPR impact machine learning?", May 16, 2018, "https://www.oreilly.com/ideas/how-will-the-gdpr-impact-machine learning
5) Vanian, J "Unmasking A.I.'s Bias Problem," Fortune, June 25, 2018, http://fortune.com/longform/ai-bias-problem/ NSTC, "Preparing for the Future of AI," October 2016, Brynjolfsson, E and Mitchell, T. "What can machine learning do?
Workforce implications," *Science* 22 Dec 2017: Vol. 358, Issue 6370, pp. 1530- 1534 DOI: 10.1126/science.aap8062
6) Courtland, R. "Bias detectives: the researchers striving to make algorithms fair," *Nature*, June 2018, https://www.nature.com/magazine- assets/d41586-018-05469-3/d41586-018-05469-3.pdf
7) Zheng, H. Wang, Y, and Molino, P. "COTA: Improving Uber Customer Care with NLP & Machine Learning," January 2018, https://eng.uber.com/cota/ ⬚○
8) Hermann and Del Balso, 2017, "Meet Michelangelo: Uber's Machine Learning Platform," https://eng.uber.com/michelangelo/ ⬚
9) National Science and Technology Council, "Preparing for the future of AI," October 2016, https://obamawhitehouse.archives.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/p

Reference Books:

1. Artificial Intelligence: A Modern Approach, Stuart Russel & Peter Norvig, Third Edition.
2. Essentials of Artificial Intelligence, Matt Ginsbeg.

School of Computer and Information Sciences

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-VII)

Course Code: IE401/IE404   Title of the Course: Network Programming/NP Lab

L-T-P: 3-0-0/0-0-3   Credits: 3/2

Prerequisite Course / Knowledge (If any): C programming, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO-1: Review the basics of multi-processing, multi-threading and signal handling to develop network programs. (Understand)

CO-2: Discuss different types of sockets and the associated function and system calls (Understand)

CO-3: Write UDP client and server programs for different network applications. (Apply)

CO-4: Write TCP client and server programs for different network applications. (Apply)

CO-5: Use UNIX socket system calls to manage multiple I/O streams and also socket options. (Apply)

CO-6: Describe mechanisms for logging messages and demonstration of server programs. (Understand)

CO-7: Practice debugging client server applications using network debugging tools. (Cognitive level: Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1		3		2			1		
CO2			2	3			1		
CO3	2	1		3					
CO4	2	1		3					
CO5	1		3	2					
CO6		1	3		2				
CO7			3	2			1		

Detailed Syllabus:

UNIT - I: To review Unix System Programming and OS concepts relevant for Network Programming: Process Control (fork, vfork, wait, exec system calls, user ids and related system calls), Reliable Signal Implementation (signal, sigaction, sigprocmask, sigsuspend, sigpending system calls, handling SIGCHLD) and Pthreads.

UNIT - II: Overview of TCP/IP Protocol Suite and Internet Applications: Overview of TCP/IP Protocols, TCP (State Transitions), UDP, IP and Popular Internet Applications and their protocol usage.

UNIT - III: For introducing fundamentals of sockets and common functions used in any Client/Server application: Socket, Introduction, role as an application programming interface, Address Structure and address management functions for IPv4, IPv6, Common functions, Byte Ordering, byte manipulation functions, readn, writen, readline functions.

UNIT - IV: Basics of TCP Client/Server application development: Client/Server Paradigm, Iterative TCP Server, Socket, Bind, Listen, Accept, read, write, close system calls, TCP Client, Socket, Bind, Connect, read, write, close system calls, Concurrent TCP Server, Using fork, Using pthreads and Example of TCP Client/Server Application.

UNIT - V: Basics of UDP Client/Server application development, UDP Server and UDP Client, recvfrom, sendto system calls, connect system call in UDP Client and asynchronous errors.

UNIT - VI: Miscellaneous topics for building more sophisticated client/server applications: Data representation issues, I/O Multiplexing, select or poll system call, Design of TCP, UDP Servers using select or poll system call, Socket Options, getsockopt, setsockopt system calls, Few examples of SOCKET, TCP, UDP, IP options and their role in client/server applications, Using fcntl and DNS related functions.

UNIT - VII: Understanding the mechanisms for logging messages, making a server as a daemon server and developing server programs to be used by inetd super server: syslogd server and syslog system call, Daemon server using daemon_init function, inetd super server.

UNIT – VIII: Debugging client server applications and understanding an implementation of application protocol: Tools for debugging TCP, UDP applications, tcpdump, tcpflow, netstat, ethereal, detailed analysis of implementation of an application layer protocol's client and server programs (like HTTP Server, HTTP Client).

Reference Books:
1. Richard Stevens: "Advanced Programming in Unix Environment", Pearson Education Asia.
2. Richard Stevens: "Unix Network Programming Volume I (Networking APIs: Sockets and XTI)", Pearson Education Asia.
1. Douglas E. Comer and David L. Stevens: "Internetworking with TCP/IP Volume III: Client-Server Programming and Applications, Linux/POSIX Sockets Version", Prentice Hall.

Course Code : IE356   **Title of the Course: NP Lab**

L-T-P : 0-0-3   **Credits: 2**

Prerequisite Course / Knowledge (If any): C Programming, Operating Systems

Suggested Assignments:

   1. Write a multithreaded client/server program using FIFOs.

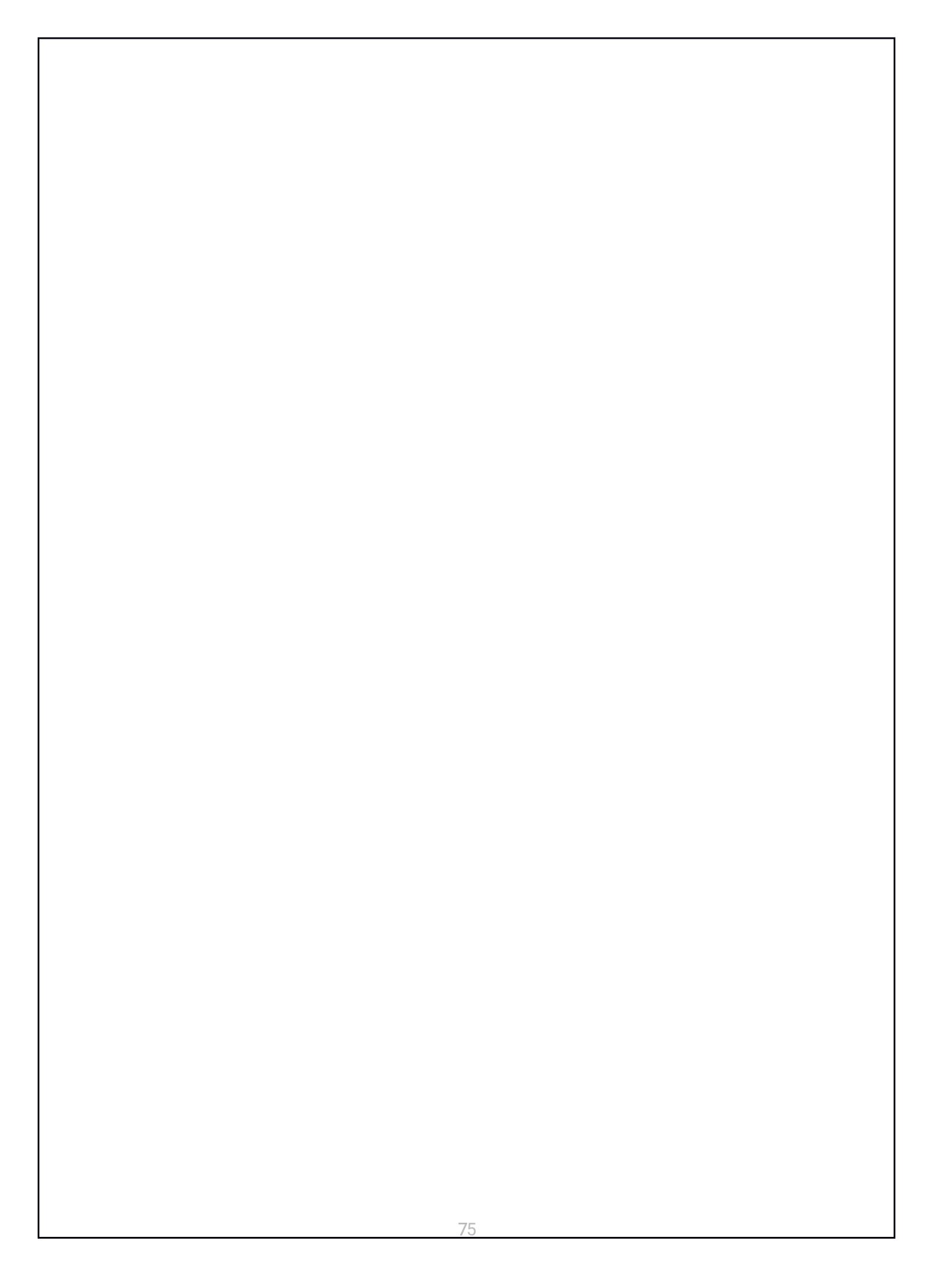   2. Write a concurrent server program using UDP

3. Write a concurrent server program using TCP

4. Write an iterative server program using the *select()* system call using TCP

5. Implement a protocol from any application level RFC using either UDP or TCP

School of Computer & Information Sciences
Integrated M.Tech (CSE)
Scheme (VIII-X)
Winter Semester

	VIII-Semester			
Code	**Course Title**	**Credits**		
IE451	Software Project Management	3	PROJ	SCIS
	Elective –IV	3/4	PEC	SCIS
	Elective –V	3/4	PEC	SCIS
	Elective –VI	3/4	PEC	SCIS
	Basket B4 (Universal Human Values, AICTE MC)	3	MC	SCIS
	Elective –VII and VIII OR Project (exiting option)	6/8	PEC/PROJ	SCIS
		21/26		

	X-Semester			
Code	**Course Title**	**Credits**		
	Project	12	PROJ	SCIS

Name of the Academic Program: Integrated M.Tech (CSE) (IMTECH-VII)

Course Code: IE401/IE404

Title of the Course: Software Project Management

L-T-P: 3-0-0

Credits   : 3

Total Minimum credits of all Semesters.

Semester	Minimum Credits
i	21
II	21.5
III	22.5
IV	21
V	21
VI	23
VII	19
VIII	21
IX	6
X	12
Total	188

SCHOOL OF COMPUTER & INFORMATION SCIENCES

- I.MTech (Computer Science & Engineering) is a 10 semester course with an exit after 4 years [B.Tech(Computer Science & Engineering)].
- As per the recommendations of NEP Committee of UoH, a 4 year exit option is given to a student completing 144-160 credits. As per AICTE the total number of credits for a 4 year program should be 160 (Approximately 20 Credits per Semester)

Definition of Credit:
1 hour lecture per week – 1 credit
1 hour tutorial per week – 1 credit
2 hours practical per week – 1 credit
Note: Our lab credits has to be changed to 1 or 1.5 credits

S.No	Category	Credit Break up (upto VIII Sem)
1	Humanities and Social Sciences including Management Courses(HSC)	10
2	Basic Sciences Courses(BSC)	21
3	Engineering Science courses including workshop, drawing, basics of electrical/mechanical/computer etc (ESC)	28
4	Professional Core Courses(PCC)	69
5	Professional Elective Courses relevant to chosen specialization/branch (PEC)	18
6	Open Subjects-Electives from other technical and/or emerging subjects (OS)	9
7	Project work, seminar and internship in industry or elsewhere(PROJ)	14
8	Mandatory Courses [Induction Program, Indian Constitution, Essence of Indian Knowledge Tradition](MC) Mandatory Course on Universal Human Values (UHV-II)	3
		172

- Multiple Entry: 10% of students in addition to consequential vacancies can be admitted in 2nd year of I.MTech who qualifies certain recognized entrance examinations JEE Mains, the rank for admission is to be within 10 percent of the rank of the cutoff for that category. Course prerequisites should be satisfied, in terms of Programming Methodology, Data and File Structures. Students with backlogs cannot be admitted. Interviews may be conducted.
- As suggested by AICTE, few of the existing courses are casted as PC and PEC as given in the scheme.
- Project to Exit in VIII sem is calculated as 8 credits