Rajasthan Technical University, Kota
Department of Computer Science& Engineering

Scheme and Syllabus for the Admission Year 2018-19

Branch- Information Technology

Scheme Code- 2017 IT

Semester	**Expected Academic Calendar**
3rd Semester	July 2019
4th Semester	January 2020
5th Semester	July 2020
6th Semester	January 2021
7th Semester	July 2021
8th Semester	January 2022

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code- 2017IT

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech. III Sem(Information Technology) 2017-18, 18-19
Theory and Practical

SEMESTER-III	SCHEME CODE- 2017IT			Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
3ITU1	ICC	Electronics Devices and Circuits	4	3	1		50	100	150
3ITU2	DCC	Data Structures and Algorithms	4	3	1		50	100	150
3ITU3	DCC	Digital Electronics	3	3	0		50	100	150
3ITU4	DCC	Software Engineering	3	3	0		50	100	150
3ITU5	DCC	Object Oriented Programming	3	3	0		50	100	150
3ITU6	DCC/IEC	Advanced Engineering Mathematics	2	2	0		50	100	150
3ITU11	DCC	Data Structures and Algorithm Lab	2			3	50	25	75
3ITU12	DCC	Object Oriented Programming Lab	2			3	50	25	75
3ITU13	DCC/IEC	Electronics Devices and Circuits Lab	1			2	50	25	75
3ITU14	DCC/IEC	Digital Electronics Lab	1			2	50	25	75
3ITU20		Extra-Curricular & Discipline	1				50		50
		Total	25	17	2	10	550	700	1250

(Parul Chauhan) 18/12/19. 18/02/19.

The above scheme & syllabus is applicable for 18-19 & adopted session also.

(Dr. Vijay Panthi) 18/12/19. Parul Chauhan.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code- 2017IT

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech. IV Sem(Information Technology) 2017-18, 18-19
Theory and Practical

SEMESTER-IV	SCHEME CODE- 2017IT			Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
4ITU1	ICC/DCC	Microprocessor and Interfaces	4	3	1		50	100	150
4ITU2	DCC	Discrete Mathematics Structure	4	3	1		50	100	150
4ITU3	DCC	Linux Shell Programming	3	3	0		50	100	150
4ITU4	DCC	Analysis of Algorithm	3	3	0		50	100	150
4ITU5	DCC	Computer Network	3	3	0		50	100	150
4ITU6	DCC/IEC	Principles of Communication	2	2	0		50	100	150
4ITU11	DCC	Linux Shell Programming Lab	2			3	50	25	75
4ITU12	DCC	Advanced Data Structures Lab	2			3	50	25	75
4ITU13	DCC/IEC	Microprocessor Lab	1			2	50	25	75
4ITU14	DCC/IEC	Communication Lab	1			2	50	25	75
4ITU20		Extra Curricular& Discipline	1				50		50
		Total	26	17	2	10	550	700	1250

18/12/19

18/12/19

The above scheme & syllabus is applicable for 18-19 academic session onw.

18/12/19.

(Dr. Vijaya Pathak)

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech.V-VIII Sem(Information Technology)
Theory and Practical

SEMESTER-V	SCHEME CODE – 2017IT			Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
5ITU1	DCC	Theory of Computation	4	3	1		50	100	150
5ITU2	DCC	Computer Architecture and Organization	4	3	1		50	100	150
5ITU3	DCC	Database Management System	3	3	0		50	100	150
5ITU4	DCC	Cyber Security Management	3	3	0		50	100	150
5ITU5.X	DEC	*Elective course	3	3	0		50	100	150
5ITU6.X	DEC	*Elective course	2	2	0		50	100	150
5ITU11	DCC	Application Development in JAVA Lab	2			3	50	25	75
5ITU12	DCC	Database Management System Lab	1			2	50	25	75
5ITU13	DCC	Network Programming Lab	1			2	50	25	75
5ITU14	DCC/IEC	Software Design Lab	1			2	50	25	75
5ITU20		Extra-Curricular & Discipline	1				50		50
		Total	25	17	2	9	550	700	1250

*Elective Course

Course code	Course
5ITU5.1	Statistics and Probability Theory
5ITU5.2	Advanced Graph Theory
5ITU5.3	Data Mining and Warehouse
5ITU6.1	GSM Communication and Mobile Data Networks
5ITU6.2	Embedded System
5ITU6.3	Human Computer Interface

9/7/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech.V-VIII Sem(Information Technology)
Theory and Practical

SEMESTER-VI	SCHEME CODE – 2017IT			Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
6ITU1	DCC	Operating System	4	3	1		50	100	150
6ITU2	DCC	Computer Graphics	4	3	1		50	100	150
6ITU3	DCC	Artificial Intelligence	3	3	0		50	100	150
6ITU4	DCC	Compiler Design	3	3	0		50	100	150
6ITU5.X	DEC	*Elective	3	3	0		50	100	150
6ITU6.X	DEC	*Elective	2	2	0		50	100	150
6ITU11	DCC	Operating System Simulation Lab	2			3	50	25	75
6ITU12	DCC	Computer Graphics Lab	2			3	50	25	75
6ITU13	DCC	Artificial Intelligence Lab	1			2	50	25	75
6ITU14	DCC/IEC	Compiler Design Lab	1			2	50	25	75
6ITU20		Extra-Curricular & Discipline	1				50		50
		Total	26	17	2	10	550	700	1250

*Elective Course

Course code	Course
6ITU5.1	Distributed System
6ITU5.2	Pattern Recognition
6ITU5.3	Data Compression Techniques
6ITU6.1	Information Theory and Coding
6ITU6.2	Software Defined Network
6ITU6.3	Agile Software Development

9/7/19

Approved
Dean, FA & UD

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech VII-VIII Sem(Information Technology)
Admitted batch 2018-19 and 2019-20
Theory and Practical

SEMESTER-VII				Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
7ITU1	DCC	Internet of Things	4	3	1		50	100	150
7ITU2	DCC	Cloud Computing	4	3	1		50	100	150
7ITU3	DCC	Big Data Analytics	3	3	0		50	100	150
7ITU4	DCC	Machine Learning	3	3	0		50	100	150
7ITU5.X	DEC	*Elective	3	3	0		50	100	150
7ITU6.X	IEC	MOOC course	4						
7ITU11	DCC	Internet of Things Lab	2			3	50	25	75
7ITU12	DCC	Big Data Analytics Lab	1			2	50	25	75
7ITU13	DCC	Machine Learning Lab	1			2	50	25	75
7ITU14	DCC	Practical Training **	4			4	150	75	225
7ITU20		Extra-Curricular & Discipline	1				50		50
		Total	30	15	2	11	600	650	1250

**Evaluation of practical training will be carried out in two parts:

1. Knowledge acquired on site (to be evaluated through presentation)
2. Applying the acquired knowledge for problem solving/development of project

Both components will be of equal weightage.

*Elective Course

Course code	Course
7ITU5.1	Information System Security
7ITU5.2	Biometrics
7ITU5.3	Digital Forensics

21/11/2020

Anil K. Mathur

Approved
Dean, FA & UD

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech VII-VIII Sem(Information Technology)
Admitted batch 2018-19 and 2019-20
Theory and Practical

OPTION-A

SEMESTER-VIII				Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
8ITU1.X	DEC	*Elective	3	3	0		50	100	150
8ITU2.X	DEC	*Elective	3	3	0		50	100	150
8ITU3.X	DEC	*Elective	3	3	0		50	100	150
8ITU4.X	IEC	MOOC Course	3						
8ITU13	DCC	Seminar	4			4	150	75	225
8ITU14	DCC	Project	12			18	350	175	525
8ITU20		Extra-Curricular & Discipline	1				50		50
		Total	29	9	0	22	700	550	1250

OPTION- B

SEMESTER-VIII				Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
8ITU13	DCC	Seminar	4			4	150	75	225
8ITU14	DCC	Project Cum Internship	21			36 hours per week	500	475	975
8ITU4.X	IEC	MOOC Course	3						
8ITU20		Extra-Curricular & Discipline	1				50		50
		Total	29			40	700	550	1250

2/11/2020

Anil K. Mathur

Approved
Dean, FA & UD

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech VII-VIII Sem(Information Technology)
Admitted batch 2018-19 and 2019-20
Theory and Practical

*Elective Course

Course code	Course
8ITU1.1	Nature Inspired Algorithms
8ITU1.2	Advanced Operating System
8ITU1.3	Software Testing
8ITU2.1	Advanced Database Management System
8ITU2.2	Real Time System
8ITU2.3	Bioinformatics
8ITU3.1	Operation Research
8ITU3.2	Project Management
8ITU3.3	Social Network Analysis

2/11/2018

Approved
Dean, FA & UD

SYLLABUS

Semester	**III**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2019-20**

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU1 ELECTRONIC DEVICES & CIRCUITS

MAX_MARKS(50+100)

Objectives:

To acquaint the students with construction, theory and characteristics of the following electronic devices:

1. p-n junction diode
2. Bipolar transistor
3. Field effect transistor
4. LED, LCD and other photo electronic devices
5. Power control / regulator devices

Syllabus:

Mobility and conductivity, charge densities in a semiconductor, Fermi Dirac distribution,carrier concentrations and fermi levels in semiconductor, Generation and recombination of charges, diffusion and continuity equation, Mass action Law, Hall effect, Junction diodes, Diode as a ckt. element, load line concept, clipping and clamping circuits, Voltage multipliers.

Transistor characteristics, Current components, Current gains: alpha and beta. Operating point. Hybrid model, h-parameter equivalent circuits. CE, CB and CC configuration. DC and AC analysis of CE,CC and CB amplifiers. Ebers-Moll model. Biasing & stabilization techniques. Thermal runaway, Thermal stability.

SMALL SIGNAL AMPLIFIERS AT LOW FREQUENCY : Analysis of BJT and FET, RC coupled amplifiers. Frequency response, midband gain, gains at low and high frequency. Miller's Theorem. Cascading Transistor amplifiers, Emitter follower. JFET, MOSFET, Equivalent circuits and biasing of JFET's & MOSFET's. Low frequency CS and CD JFET amplifiers. FET as a voltage-variable resistor. Source follower.

FEEDBACK AMPLIFIERS : Classification, Feedback concept, Transfer gain with feedback, General characteristics of negative feedback amplifiers. Analysis of voltageseries, voltage-shunt, current- series and current-shunt feedback amplifier. Stability criterion.

OSCILLATORS : Classification. Criterion for oscillation. Tuned collector, Hartley, Colpitts, RC Phase shift, Wien bridge and crystal oscillators, Astable, monostable and bistable multivibrators. Schmitt trigger.

Text/References:

1. Electronic devices & circuits theory By R.L. Boylestad, Louis Nashelsky ,Pearson education
2. Integrated Electronics By Millman Halkias, T.M.H
3. Electronic devices & circuits By David Bell, Oxford Publications

16/2/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU1	Electronics Devices and Circuits	CO 1	Understand the basic Electronic circuits and components like transistors, Diodes, OP-AMP etc.
		CO 2	Capable to evaluate various frequency bands, analog modulation techniques, principles related to the operation and concepts of Satellite and mobile Communication
		CO 3	Capable to demonstrate electronic measuring instruments, Transducers, and consumer Electronics
		CO 4	Illustrate the basic idea about components of a digital computer, its programming, different computer networks, internet and IP addressing

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU2 DATA STRUCTURES & ALGORITHMS

MAX_MARKS(50+100)

Objectives:

1. To study various data structure concepts like Stacks, Queues, Linked List, Trees and Files.
2. To overview the applications of data structures.
3. To be familiar with utilization of data structure techniques in problem solving.
4. To have a comprehensive knowledge of data structures and relevant algorithms.
5. To carry out asymptotic analysis of any algorithm

Syllabus:

Asymptotic notations: Concept of complexity of program, Big-Oh, theta, Omega- Definitions and examples, Determination of time and space complexity of simple algorithms without recursion. Representing a function in asymptotic notations viz $5n^2-6n=\theta(n^2)$

Linear Data Structures: Array as storage element, Row major & column major form of arrays, computation of address of elements of n dimensional array. Arrays as storage elements for representing polynomial of one or more degrees for addition & multiplication, sparse matrices for transposing & multiplication, stack, queue, dequeue, circular queue for insertion and deletion with condition for over and underflow, transposition of sparse matrices with algorithms of varying complexity (Includes algorithms for operations as mentioned).

Evaluation of Expression: Concept of precedence and associativity in expressions, difficulties in dealing with infix expressions, Resolving precedence of operators and association of operands, postfix & prefix expressions, conversion of expression from one form to other form using stack (with & without parenthesis), Evaluation of expression in infix, postfix & prefix forms using stack. Recursion.

Linear linked lists: singly, doubly and circularly connected linear linked lists- insertion, deletion at/ from beginning and any point in ordered or unordered lists. Comparison of arrays and linked lists as data structures.

Linked implementation of stack, queue and dequeue. Algorithms for of insertion, deletion and traversal of stack, queue, dequeue implemented using linked structures. Polynomial representation using linked lists for addition, Concepts of Head Node in linked lists.

Searching: Sequential and binary search

18/2/19

Anil K. Mathur

Sorting: Insertion, quick, heap, topological and bubble sorting algorithms for different characteristics of input data. Comparison of sorting algorithms in term of time complexity.

Non-Linear Structures: Trees definition, characteristics concept of child, sibling, parent child relationship etc, binary tree: different types of binary trees based on distribution of nodes, binary tree (threaded and unthreaded) as data structure, insertion, deletion and traversal of binary trees, constructing binary tree from traversal results. Threaded binary Tree.Time complexity of insertion, deletion and traversal in threaded and ordinary binary trees. AVL tree: Concept of balanced trees, balance factor in AVL trees, insertion into and deletion from AVL tree, balancing AVL tree after insertion and deletion. Application of trees for representation of sets.Operations on Weight Balanced Trees, 2-3-4 Trees.

Graphs: Definition, Relation between tree & graph, directed and undirected graph, representation of graphs using adjacency matrix and list. Depth first and breadth first traversal of graphs, finding connected components and spanning tree,Single source single destination shortest path algorithms. Definitions of Isomorphic Components.Circuits,Fundamental Circuits,Cut-sets.Cut-Vertices, PlanerandDualgraphs, SpanningTrees,Kuratovski'stwoGraphs.

NOTE:

1. Algorithm for any operation mentioned with a data structure or required to implement the particular data structure is included in the curriculum.

TEXT BOOK

1. Data Structures in C/C++, Tanenbaum, Pearson
2. Data Structure and Algorithms, Pai TMGH

REFERENCE BOOKS

1. An introduction to data structures with applications By Jean-Paul Tremblay, P. G. Sorenson, TMH
2. Data Structures in C/C++, Horowitz, Sawhney, Galgotia
3. Data Structures in C++, Weiss, Parson

16/2/19.

Anil K. Mathur

ion Year (AY)	2015-16	2016-17	2017-18	2018-19	
ne Code (SC)	2015IT	2016IT	2017IT	2017IT	

eme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU2	Data Structures and Algorithms	CO 1	To Understand and Examine asymptotic analysis of any algorithm.
		CO 2	To Evaluate and Analyse the implementation and application of various ADTs such as Stack, Queue etc.
		CO 3	To Design tree based data structures such as Binary Tree, BST, AVL Tree etc and Applications of it.
		CO 4	To Design and Implement graph and hashing based data structure.

Anil K. Mathus
Approved
Dean, FA & UD

3ITU3 DIGITAL ELECTRONICS

MAX_MARKS(50+100)

Objectives:

1. To identify various number systems and work with Boolean Algebra.
2. To understand various logic gates and their technologies
3. To understand the working of combinational circuits and learn their design concepts.
4. To learn working of various types of flip flops used for designing registers and counters and other sequential circuits.
5. To learn equivalence of a circuit design and Boolean algebra and use it to optimize the design

Syllabus:

Number Systems, Basic Logic Gates &Boolean Algebra: Binary Arithmetic & Radix representation of different numbers. Sign & magnitude representation, Fixed point representation, complement notation, various codes & arithmetic in different codes & their inter conversion. Features of logic algebra, postulates of Boolean algebra.Theorems of Boolean algebra.Boolean function.

Derived logic gates: Exclusive-OR, NAND, NOR gates, their block diagrams and truth tables. Logic diagrams from Boolean expressions and vica-versa. Converting logic diagrams to universal logic. Positive, negative and mixed logic.Logic gate conversion.

Minimization Techniques: Minterm, Maxterm, Karnaugh Map, K map upto 4 variables. Simplification of logic functions with K-map, conversion of truth tables in POS and SOP form. Incomplete specified functions. Variable mapping.Quinn-McKlusky minimization techniques.

Combinational Systems: Combinational logic circuit design, half and full adder, subtractor. Binary serial and parallel adders.BCD adder.Binary multiplier. Decoder: Binary to Gray decoder, BCD to decimal, BCD to 7-segment decoder. Multiplexer, demultiplexer, encoder.Octal to binary, BCD to excess-3 encoder.Diode switching matrix.Design of logic circuits by multiplexers, encoders, decoders and demultiplexers.

Sequential Systems: Latches, flip-flops, R-S, D, J-K, Master Slave flip flops. Conversions of flip-flops. Counters: Asynchronous (ripple), synchronous andsynchronous decade counter, Modulus counter, skipping state counter, counter design. Ring counter. Counter applications. Registers: buffer register, shift register.

TEXT BOOKS

1. Digital Logic and Computer Design By M. Morris Mano, Pearson.
2. Digital circuit design By S. Salivahanan, Sarivazhagan, Vikas publications.

18/2/19

Anil K. Mathews

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

REFERENCE BOOKS

1. Digital integrated electronics, By Herbert Taub, Donald L. Schilling, TMH
2. Modern Digital Electronics By R.P. Jain, TMH
3. Fundamentals of Digital circuits By A. Anandkumar, PHI

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU3	Digital Electronics	CO 1	To identify various number systems and work with Boolean algebra.
		CO 2	To understand various logic gates and their technologies
		CO 3	Student should be able to identify, analyze and design combinational circuits.
		CO 4	Develop an ability to design various synchronous and asynchronous sequential circuits.
		CO 5	To learn equivalence of a circuit design and Boolean algebra and use it to optimize the design

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU4 SOFTWARE ENGINEERING

MAX_MARKS(50+100)

OBJECTIVE-

1. Comprehend software development life cycle.
2. Prepare SRS document for a project
3. Apply software design and development techniques.
4. Identify verification and validation methods in a software engineering project.
5. Implement testing methods at each phase of SDLC.
6. Analyze and Apply project management techniques for a case study

Syllabus:-

Introduction - Evolving role of software, Software a crisis on the Horizon, Software Myths Software engineering layered technology

Software process & Software process models , The linear sequential model ,The prototyping model ,The RAID model , Evolutionary models , Component based development , The formal methods model , Fourth generation techniques

Project management concepts, Software Process and project metrics, Software project planning, Software project estimation, Risk management, RMMM plans

Project scheduling and tracking, Software quality assurance, Software configuration management,

Requirement analysis-software prototyping-Specification Review Analysis modeling, Data modeling-functional modeling

Behavioral modeling- Data dictionary Design concepts and principles, Effective modular design, design heuristics, Design model, Documentation

Software design-Software architecture, Data designing, Architectural styles, Transform mapping, Transaction mapping, Refining architectural design User interface design, Component level design

Software testing techniques-White box and black box testing, Unit testing, integrating testing, validation technique, System testing - debugging

TEXT BOOKS

1. Software Engineering By Roger S. Pressman, TMH
2. Software Engineering By Ian Sommerville

REFERENCE BOOKS

1. Software Engineering Fundamental By Ali Behforooz, Frederick J Hudson, Oxford University Press
2. Software Engineering Concepts By **Richard E. Fairley** (Mcgraw-Hill)

18/2/19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU4	Software Engineering	CO1	Understand the basic implementation model SDLC and its types
		CO2	Illustrate the process of SRS document of a project.
		CO3	Capable to apply implementation process of validation and verification methods in software project.
		CO4	Capable to implement testing techniques on SDLC phases.
		CO5	Analyse and apply project management techniques on real time project.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU5 OBJECT ORIENTED PROGRAMMING

MAX_MARKS(50+100)

Objective : The objective of this course is to provide knowledge about programming basics that might be useful to B.Tech. Computer Science and Information Technology student, in more practical manner. These programming skills often occur in practical engineering problem of Computer Science student.

Syllabus

Introduction: Review of structures in C, accessing members of structures using structure variables, pointer to structures, passing structures to functions, structures as user defined data types.

Introduction to programming paradigms- (Process oriented and Object oriented). Concept of object, class, objects as variables of class data type, difference in structures and class in terms of access to members, private and public Basics of C++: Structure of C++ programs, introduction to defining member functions within and outside a class, keyword using, declaring class, creating objects, constructors & destructor functions, Initializing member values with and without use of constructors, simple programs to access & manipulate data members, cin and cout functions. Dangers of returning reference to a private data member, constant objects and members function, composition of classes, friend functions and classes, using this pointer, creating and destroying objects dynamically using newand deleteoperators.

Static class members, members of a class, data & function members. Characteristics of OOP- Data hiding, Encapsulation, data security.

Operator overloading: Fundamentals, Restrictions, operator functions as class members v/s as friend functions. Overloading stream function, binary operators and unary operators.

Inheritance: Base classes and derived classes, protected members, relationship between base class and derived classes, constructors and destructors in derived classes, public, private and protected inheritance, relationship among objects in an inheritance hierarchy, abstract classes, virtual functions and dynamic binding, virtual destructors.

Multiple inheritance, virtual base classes, pointers to classes and class members, multiple class members. Templates, exception handling.

18/2/19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

Text/References:

1. How to Program C++, Dietel, Pearson
2. Mastering C++ By K.R.Venugopal, TMH
3. Object Oriented Programming in C++ By Robert Lafore, Pearson
4. Object Oriented Design & Modelling, Rambaugh, Pearson

18/2/19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU5	Object Oriented Programming	CO 1	Recollect the different programming paradigms and their structure.
		CO 2	Analyze dynamic memory management techniques and apply using pointers, constructors, destructors.
		CO 3	Describe the concept of function overloading, operator overloading, virtual functions and polymorphism to solve complex problems.
		CO 4	Classify inheritance with the understanding of early and late binding and design solutions.
		CO 5	Design solutions for exception handling.

Approved
Dean, FA & UD

3ITU6 ADVANCED ENGINEERING MATHEMATICS

MAX_MARKS(50+100)

Objective: The objective of this course is to provide tools of Mathematics that might useful in B. Tech. Computer Science and Information Technology students in more practical manner. This course is meant to provide grounding in Statistics, Transform and Numerical Techniques that can be applied in modeling processes and decision making. These techniques often occur in practical engineering problems of computer science students.

Syllabus:

Random Variables: Discrete and Continuous random variables, Joint distribution, Probability distribution function, conditional distribution.

Mathematical Expectations: Moments, Moment Generating Functions, variance and correlation coefficients, Chebyshev's Inequality, Skewness and Kurtosis.

Binomial distribution, Normal Distribution, Poisson Distribution and their relations, Uniform Distribution, Exponential Distribution.

Correlation: Karl Pearson's coefficient, Rank correlation. Curve fitting. Line of Regression.

Laplace Transform: Definition and existence of Laplace Transform, Properties and Formulae, Unit Step Function, Dirac Delta Function, Heaviside Function, Inverse Laplace Transform, Convolution theorem, Application of Laplace Transform to Ordinary Differential Equation, Solution of Integral Equations.

Interpolation: Difference Operators- Forward, Backward, Central, Shift and Average Operators, Newton's and Gauss's forward and backward interpolation formulae for equal intervals, Stirling's Formula, Lagrange Interpolation Formula, Inverse Interpolation.

Numerical Differentiation by Newton's, Gauss's and Stirling's Formula.

Numerical Integration: Trapezoidal Rule, Simpson's 1/3 and 3/8 Rule.

Numerical Solution of ODE of First Order: Picard's method, Euler's method, Modified Euler's method, Runge-Kutta method forth order method, Milne's Method.

TEXT BOOKS:

18/2/19

Dr. K. Mathus

1. Probability and Statistics; Schaum's Outline Series, Murray Spiegel, John Schiller, R. Alu Srinivasan, *McGraw Hill Education*; 2010.
2. Advanced Engineering Mathematics, Denis Zill and Warren Wright, pg 207 to 251, *Jones & Bartlett India Private Limited*, 2011.
3. Introductory Methods of Numerical Analysis, S. S. Sastry, *PHI Lerning*, 2012.

REFERENCE:

1. Advanced Engineering Mathematics, 4th Editions, Jain and Iyengar , *Narosa Publications*.
2. Higher Engineering Mathematics, B. V. Ramana, 1st Edition, *Mc Graw Hill Education*, 2014.
3. Engineering Statistics; Montgomery, Runger and Hubele; *Wiley Publication*, 2014.
4. A First Course in Numerical Methods; Uri M Asher and Chen Greif, *SIAM Publication*, 2015.
5. Introduction to Probability and Statistics; Seymour Lipschutz and John J. Schiller; *Schaum Outline Series*; 2011.
6. Introduction to Probability and Statistics for Engineers and Scientists; *Sheldon M. Ross*; Fifth Edition, 2010.

18/4/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU6	Advanced Engineering Mathematics	CO 1	Learning different techniques and classification of Optimization problems.
		CO 2	Formulation of linear programming problems, understand different methods and apply on different type of problems.
		CO 3	Discuss Number theory and Algebraic structures (Definitions, property and Elementary operations).
		CO 4	Understand Laplace transform and its properties.
		CO 5	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU11 DATA STRUCTURES AND ALGORITHM LAB

MAX_MARKS(50+25)

Objectives:

1. To implement concepts of linear data structures like Stack and Queue.
2. To implement various Sorting and Searching Techniques.
3. To learn programming same problems using both arrays and pointers.
4. To implement concepts of non-linear data structures

List of Experiments:

1 Write a simple C program on a 32 bit compiler to understand the concept of array storage, size of a word. The program shall be written illustrating the concept of row major and column major storage. Find the address of element and verify it with the theoretical value. Program may be written for arrays upto 4-dimensions.

2 Simulate a stack, queue, circular queue and dequeue using a one dimensional array as storage element. The program should implement the basic addition, deletion and traversal operations.

3 Represent a 2-variable polynomial using array. Use this representation to implement addition of polynomials.

4 Represent a sparse matrix using array. Implement addition and transposition operations using the representation.

5 Implement singly, doubly and circularly connected linked lists illustrating operations like addition at different locations, deletion from specified locations and traversal.

6 Repeat exercises 2, 3 & 4 with linked structure.

7 Implementation of binary tree with operations like addition, deletion, traversal.

8 Depth first and breadth first traversal of graphs represented using adjacency matrix and list.

9 Implementation of binary search in arrays and on linked Binary Search Tree.

10 Implementation of insertion, quick, heap, topological and bubble sorting algorithms.

16/2/19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU11	Data Structures and Algorithms Lab	CO 1	Describe basic concepts of Function, Array and Link-list.
		CO 2	Understand how several fundamental algorithms work particularly those concerned with Stack, Queues, Trees and various Sorting algorithms.
		CO 3	Measure the performance of various algorithm used overall in the syllabus.
		CO 4	Design new algorithms or modify existing ones for new applications and able to analyze the space & time complexity.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU12 OBJECT ORIENTED PROGRAMMINGLAB

MAX_MARKS(50+25)

Objective: The objective of this course is to provide knowledge about programming basics that might be useful to B.Tech. Computer Science and Information Technology student, in more practical manner. These programming skills often occur in practical engineering problem of Computer Science student.

Syllabus:

1. To write a simple program for understanding of C++ program structure without any CLASS declaration. Program may be based on simple input output, understanding of keyword using.

2. Write a C++ program to demonstrate concept of declaration of class with public & private member, constructors, object creation using constructors, access restrictions, defining member functions within and outside a class. Scope resolution operators,
accessing an object's data members and functions through different type of object handle name of object, reference to object, pointer to object, assigning class objects to each other.

3. Program involving multiple classes (without inheritance) to accomplish a task. Demonstrate composition of class.

4. Demonstration Friend function friend classes.

5. Demonstration dynamic memory management using new &delete & static class members.

6. Demonstration of restrictions an operator overloading, operator functions as member function and/ or friend function, overloading stream insertion and stream extraction, operators, overloading operators etc.

7. Demonstrator use of protected members, public & private protected classes, multi-level inheritance etc.

8. Demonstration of exception handling.

18/9/19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU12	Object Oriented Programming Lab	CO 1	Analyze complex problems and develop solutions using objects and classes.
		CO 2	Programs to demonstrate the implementation of friend function, constructors and destructors
		CO 3	Evaluate operator overloading using unary and binary operators.
		CO 4	Analyse and implement algorithmic problems including inheritance, and polymorphism.
		CO 5	Explore the concept of Templates and implement exception handling

Anil K. Mathus
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU13 ELECTRONIC DEVICES AND CIRCUITS LAB

MAX_MARKS(50+25)

Objectives:

1. Understand the nature and scope of modern electronics.
2. Describe physical models of basic components.
3. Design and construct simple electronic circuits to accomplish a specific function, e.g., designing amplifiers, ADC converters etc.
4. Understand their capabilities and limitations and make decisions regarding their best utilization in a specific situation.

List of Experiments

1. Plot V-I characteristic of P-N junction diode & calculate cut-in voltage, reverse Saturation current and static & dynamic resistances.

2 Plot V-I characteristic of zener diode and study of zener diode as voltage regulator. Observe the effect of load changes and determine load limits of the voltage regulator.

3 Plot frequency response curve for single stage amplifier and to determine gain bandwidth product.

4 Plot drain current - drain voltage and drain current – gate bias characteristics of field effect transistor and measure of Idss & Vp

5 Application of Diode as clipper & clamper

6 Plot gain- frequency characteristic of two stages RC coupled amplifier & calculate its bandwidth and compare it with theoretical value.

7 Plot gain- frequency characteristic of emitter follower & find out its input and output resistances.

8 Plot input and output characteristics of BJT in CB, CC and CE configurations. Find their h-parameters.

9 Plot gain-frequency characteristics of BJT amplifier with and without negative feedback in the emitter circuit and determine bandwidths, gain bandwidth products and gains at 1kHz with and without negative feedback.

10 Plot and study the characteristics of small signal amplifier using FET.

Anil K. Mathur 18/2/19

Approved
Dean, FA & UD

11 Study Wein bridge oscillator and observe the effect of variation in R & C on oscillator frequency

12 Study transistor phase shift oscillator and observe the effect of variation in R & C on oscillator frequency and compare with theoretical value.

13 To plot the characteristics of UJT and UJT as relaxation.

14 To plot the characteristics of MOSFET and CMOS.

18/2/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU13	Electronics Devices and Circuits Lab	CO1	Illustrate the nature and scope of modern electronics.
		CO2	Understand the basic components of electronics devices.
		CO3	Implement the design structure of specific function e.g. designing amplifiers, ADC converters etc.
		CO4	Analyze and compare their capabilities and limitations and make decisions regarding their best utilization in a specific situation.

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

3ITU14 DIGITAL ELECTRONICS LAB

Objectives: **MAX_MARKS(50+25)**

1. Understand electrical conduction in solid state materials
2. Analyze and design dc and switching circuits containing diodes and transistors
3. Analyze and design combinational logic circuits at the transistor level
4. Develop skill with computer-based circuit simulation

List of Experiments

1. To verify the truth tables of basic logic gates: AND, OR, NOR, NAND, NOR.Also to verify the truth table of Ex-OR, Ex-NOR (For 2, 3, & 4 inputs using gates with 2, 3, & 4 inputs).

2 To verify the truth table of OR, AND, NOR, Ex-OR, Ex-NOR realized using NAND & NOR gates.

3 To realize an SOP and POS expression.

4 To realize Half adder/ Subtractor & Full Adder/ Subtractor using NAND & NOR gatesand to verify their truth tables.

5 To realize a 4-bit ripple adder/ Subtractor using basic Half adder/ Subtractor &basic Full Adder/ Subtractor.

6 To verify the truth table of 4-to-1 multiplexer and 1-to-4 demultiplexer. Realize themultiplexer using basic gates only. Also to construct and 8-to-1 multiplexer and 1-to-8demultiplexer using blocks of 4-to-1 multiplexer and 1-to-4 demultiplexer

7 Design & Realize a combinational circuit that will accept a 2421 BCD code and drive a TIL - 312 seven-segment display.

8 Using basic logic gates, realize the R-S, J-K and D-flip flops with and without clocksignal and verify their truth table

9 Construct a divide by 2,4 & 8 asynchronous counter. Construct a 4-bit binary counter and ring counter for a particular output pattern using D flip flop.

18/02/19

Anil K. Mathus

10 Perform input/output operations on parallel in/Parallel out and Serial in/Serial outregisters using clock. Also exercise loading only one of multiple values into theregister using multiplexer.

Note: As far as possible, the experiments shall be performed on bread board. However, experiment Nos. 1-4 are to be performed on bread board only.

18/2/19.

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
3ITU14	Digital Electronics lab	CO 1	Learn the basics of Logic gates and realization of logic gates
		CO 2	Construct basic combinational circuits and verify their functionalities
		CO 3	Apply the design procedures to design basic sequential circuits
		CO 4	Implementation of counters, shift registers and testing of their functionality

Anil K. Mathus
Approved
Dean, FA & UD

SYLLABUS

Semester	**IV**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2019-20**

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

4ITU1 MICROPROCESSOR AND INTERFACES

MAX_MARKS(50+100)

Objective:

1. To provide theoretical & practical introduction to microcontrollers and microprocessors.
2. Techniques of assembly language programming.
3. To understand design of hardware interfacing circuit.
4. Design considerations of microcontroller and microprocessor system.

Syllabus:

Introduction to Microprocessors, microcontroller; 8085 Microprocessor Architecture, pin description, Bus concept and organization; concept of multiplexing and de-multiplexing of buses; concept of static and dynamic RAM, type of ROM, memory map.

Software architecture registers and signals, Classification of instruction, Instruction set, addressing modes, Assembly Language Programming and Debugging, Programming Technique, instruction Format and timing.

Advance Assembly Language Programming, Counter and time delay; types of Interrupt and their uses, RST instructions and their uses, 8259 programmable interrupt controller; Macros, subroutine; Stack- implementation and uses with examples; Memory interfacing.

8085 Microprocessor interfacing:, 8255 Programmable Peripheral Interface, 8254 programmable interval timer, interfacing of Input/output device, 8279 Key board/Display interface.

Microprocessor Application: Interfacing scanned multiplexed display and liquid crystal display, Interfacing and Matrix Keyboard, MPU Design; USART 8251, RS232C and RS422A, Parallel interface- Centronics and IEEE 488.

TEXT BOOK

1. Microprocessor architecture, programming, and applications with the 8085 By Ramesh S. Gaonkar

REFERENCE BOOKS

1. Introduction to Microprocessor By Aditya P. Mathur, TMH
2. Microprocessor & Interfaceing By Douglas V. Hall, TMH
3. Microprocessor & Peripheral By A.K.Ray, K.M. Bhurchandi, TMH

18/2/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU1	Microprocessors and Interfaces	CO 1	Describe 8085 architecture and programming in assembly language.
		CO 2	Discuss different types of instruction set and addressing modes.
		CO 3	To apply concepts of interfacing memory and peripheral devices to a microprocessor.
		CO 4	To analyse different applications of microprocessor.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

4ITU2 DISCRETE MATHEMATICS STRUCTURE

MAX_MARKS(50+100)

Objectives:

1. Some fundamental mathematical concepts and terminology;
2. How to use and analyze recursive definitions;
3. How to count some different types of discrete structures;
4. Techniques for constructing mathematical proofs, illustrated by discrete mathematics examples.

Syllabus

Set Theory: Definition and types, Set operations, Partition of set, Cardinality (Inclusion-Exclusion & Addition Principles).

Relations: Definition, Binary Relation, Matrix of Relation, Digraph of Relation, Properties of Relations, Equivalence relations and partition, Equivalence Class, The Connectivity Relations, Operations on Relations, Closures of Relation, Transitive Closure-Warshall's Algorithm.

Functions: Concept, Some Special Functions (Characteristic, Floor & Ceiling Functions), Properties of Functions, The Pigeonhole &Generalized Pigeonhole Principles, Composition of Functions.

Language of Logic: Proposition, Compound Proposition, Conjunction, Disjunction, Implication, Converse, Inverse & Contrpositive, Biconditional Statements, tautology, Contradiction & Contingency, Logical Equivalences, Quantifiers, Arguments.

Methods of Proof: Direct, Indirect, Principle of Mathematical Induction.

Graph Theory: Graphs- Directed, Undirected, Simple, Adjacency & Incidence, Degree of Vertex, Subgraph, Complete graph, Cycle & Wheel Graph, Bipartite & Complete Bipartite Graph, Weighed Graph, Complete Graphs, Isomorphic Graphs, Path, Cycles & Circuits Euclerian& Hamiltonian Graphs.

Trees: Definition, Spanning Trees, Minimal Spanning Trees, Prim's Algo, Kruskal'sAlgo.

TEXT BOOKS:

1. Discrete Mathematical Analysis, Kolman et al., Pearson
2. Discrete Mathematics and its Applications, Kenneth H. Rosen, MGH

REFERENCE BOOKS:

1. Discrete Mathematical Structures, Lipschutz & Lipson, MGH
2. Discrete Mathematics with Applications, Koshy, ELSEVIER

18/2/19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU2	Discrete Mathematics Structures	CO 1	Be familiar with fundamental mathematical concepts such as sets and apply them.
		CO 2	Students analyse basics knowledge gained by mathematical relation and apply them
		CO 3	To be able to understand fundamental of functions such as (domain, co-domain, range, image, inverse image and composition) and types of functions.
		CO 4	Use mathematical propositions and proof techniques to check the truthfulness of a real life situation and to apply the notion of mathematical thinking, mathematical proofs and logics such as predicate logic, propositional logic and inference rules.
		CO 5	Use graph theory and trees to formulate the problems and solve them.

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

4ITU3 LINUX AND SHELL PROGRAMMING

MAX_MARKS(50+100)

Objective:

1. To introduce Unix/Linux kernel programming
2. Provides the knowledge of shell scripting and Linux tools and applications.
3. Student will be able to install an OS, operate advanced commands in UNIX, use an IDE for building small projects.
4. Apply the concept of shell programming to solve various problems
5. To learn basic components in constructing a shell script

Syllabus:

Introduction: Logging in, changing password (*passwd* command only), *man, xman, info* commands to access on line help. Simple commands like *ls, cp, mv, grep, head, tail, sort, uniq, diff, echo, date, which, whereis, whatis, who, finger w* (option and variations included).

Directory commands, access permissions, changing access permissions for files and directories, hard & symbolic links.Environment and path setting.

vi editor: Creating and editing files, features of vi, insertion deletion, searching, substitution operations, yank, put, delete commands, reading & writing files, *exrc* file for setting parameters, advance editing techniques. Vim (improved vi).

Programming utilities: Compiling & linking C, C++ programs, *make* utility, debugging C programs using *gdb*, system call.

Introduction to X-window system: x-window as client/ server system, concept of window manager, remote computing & local displays, *xinitrc* file, customize X work environment and applications, customizing the *fvwm* window manager

Shell: Meaning and purpose of shell, Introduction to types of shell. The command line, standard input and standard output, redirection, pipes, filters special characters for searching files and pathnames.

Bourne Again Shell: shell script-writing and executing, command separation & grouping, redirection, directory stack manipulation, processes, parameters & variables, keyword variables.

Shell Programming: Control structures, the *Here* document, expanding *NULL* or *USET* variables, Built-ins, functions, history, aliases, job control, filename substitution. source code management-RCS and CVS. *awk* utility.

TEXT BOOK

1. A practical Guide to Linux, Sobell, Pearson.

18/2/19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

REFERENCE BOOKS

1. A Practical Guide to Linux Commands, Editors, and Shell Programming, Sobell, Pearson.
2. A Practical Guide to Fedora and Red Hat Enterprise Linux, Sobell, 5e, Pearson
3. Harley Hahn: Guide to Unix & Linux, TMH
4. Blum, Bresnahan, Linux Command and Shell Scripting Bible, Wiley India, 2nd Ed.

18/2/19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU3	Linux Shell Programming	CO 1	Understanding of Unix/Linux operating system, and its basic commands to operate.
		CO 2	Able to understand the vi editor and its basic commands for creating and editing files in Unix/Linux environment and analysis of C environment
		CO 3	Understanding of X-window system and its environment.
		CO 4	Able to understand the shell with its basic features and working of BASH
		CO 5	Apply the concept of shell programming to solve various problems

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT

Scheme Code - 2017IT

4ITU4 ANALYSIS OF ALGORITHM

MAX_MARKS(50+100)

Objectives:

1. To understand various complexity order notations.
2. To study mathematical background for algorithm analysis and implementation of various strategies like divide and conquer, Greedy method, Dynamic programming, Branch and bound , Backtracking and number theoretic algorithm.
3. To study different string matching algorithms.
4. To study various problem classes like P, NP, NP- Hard etc.

Syllabus:

BACKGROUND: Review of Algorithm, Complexity Order Notations: definitions and calculating complexity.

DIVIDE AND CONQUER METHOD: Binary Search, Merge Sort, Quick sort and Strassen's matrix multiplication algorithms.

GREEDY METHOD: Knapsack Problem, Job Sequencing, Optimal Merge Patterns and Minimal Spanning Trees.

DYNAMIC PROGRAMMING: Matrix Chain Multiplication. Longest Common

Subsequence and 0/1 Knapsack Problem.

BRANCH AND BOUND: Traveling Salesman Problem and Lower Bound Theory. Backtracking Algorithms and queens problem.

PATTERN MATCHING ALGORITHMS: Naïve and Rabin Karp string matching algorithms, KMP Matcher and Boyer Moore Algorithms.

ASSIGNMENT PROBLEMS: Formulation of Assignment and Quadratic Assignment Problem.

RANDOMIZED ALGORITHMS- Las Vegas algorithms, Monte Carlo algorithms, randomized algorithm for Min-Cut, randomized algorithm for 2- SAT. Problem definition of Multicommodity flow, Flow shop scheduling and Network capacity assignment problems.

NUMBER THEORITIC ALGORITHM: Number theoretic notions, Division theorem, GCD, recursion, Modular arithmetic, Solving Modular Linear equation, Chinese Remainder Theorem, power of an element, Computation of Discrete Logarithms, primality Testing and Integer Factorization.

PROBLEM CLASSES NP, NP-HARD AND NP-COMPLETE: Definitions of P, NP-Hard and NP-Complete Problems. Decision Problems. Cook's Theorem. Proving NP-Complete

18/2/19

Anil K. Mathur

Problems - Satisfiability problem and Vertex Cover Problem. Approximation Algorithms for Vertex Cover andSet Cover Problem.

TEXTBOOKS:

1. Cormen, Leiserson, Rivest: Introduction to Algorithms, Prentice Hall of India.
2. Horowitz and Sahani: Fundamental of Computer algorithms.
3. Aho A.V , J.D Ulman: Design and analysis of Algorithms, Addison Wesley

REFERENCES:

1. Michael Gooddrich& Roberto Tammassia, "Algorithm design foundation, analysis and internet examples", Secondedition ,wiley student edition.
2. Ellis horowitz ,sartajSahni , s. Rajsekaran. "Fundamentals of computer algorithms" University Press.

13/2/19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course code	Course name	Course outcome	Details
4ITU4	Analysis of Algorithms	CO1	Analyze worst-case running times of algorithms using asymptotic analysis.(K4)
		CO2	The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to formulate a given problem.(K5)
		CO3	Analysis of the algorithm will be used to compare the efficiency of the algorithm over the naive techniques.(K6)
		CO4	Students will be able to identify a proper pattern matching algorithm for given problem.(K4)
		CO5	Apply to randomization as a tool for developing algorithms.(K3)
		CO6	Identify the computational issues of problem solving in computing.(K2)

Anil K. Mathus
Approved
Dean, FA & UD

4ITU5 COMPUTER NETWORK

MAX_MARKS(50+100)

Objectives:

1. To make students learn architecture of data communication networks.
2. Build an understanding of the fundamental concepts of computer networking.
3. Explain how communication works in data networks and the Internet.
4. Recognize the different internetworking devices and their functions.
5. Analyze the services and features of the various layers of data networks.
6. Design, calculate, and apply subnet masks and addresses to fulfill networking requirements.
7. Analyze the features and operations of various application layer protocols such as Http, DNS, and SMTP.

Syllabus:

ISOOSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP. Data Link Layer: Functions of data link layer and design issues, Flow Control: Flow control in lossless and lossy channels using stop-and-wait, sliding window protocols. Performance of protocols used for flow control, Error Control: Error Detection, Two Dimensional Parity Checks, and Internet Checksum. Polynomial Codes, Bit oriented protocols.

MAC sub layer:MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Channel Allocation Problem, Pure and slotted Aloha, CSMA, CSMA/CD, collision free multiple access, Throughput analysis of pure and slotted Aloha, High Speed LAN: Fast Ethernet, Gigabit Ethernet, FDDI, Performance Measuring Metrics.

Network layer-design issue, routing algorithms: Distance vector, link state, hierarchical, broadcast routing, Congestion Control Algorithms: General Principles of Congestion control, Prevention Policies. IPv4 classful and classless addressing, sub netting, comparative study of IPv4 & IPv6.

Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, Carrying Unicast/Multicast Real-Time Traffic. Transport Layer in the Internet: Introduction to TCP, TCP service Model, TCP Header and segment structure, TCP connection establishment and release, transmission policy, timer management, Transactional TCP. Mobile TCP, TCP Congestion Control.

Session layer: Authentication, Authorization, Session layer protocol, Presentation layer: Data conversion, Character code translation, Compression, Encryption and Decryption, Presentation layer protocol, Application Layer:DNS, SMTP, WWW, HTTP,FTP.

18/2/19

Anil K. Mathur

TEXT BOOKS

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks" Pearson Education.
2. Dimitri Bertsekas, Robert Gallager, "Data Networks", PHI Publication, Second Edition.

REFERENCE BOOKS

1. Kaveh Pahlavan, Prashant Krishnamurthy, "Networking Fundamentals", Wiley Publication.
2. William Stallings, Data and computer communication, Pearson Education.
3. Uyless Black, "Computer Networks", PHI Publication, Second Edition.
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw Hill.

13/2/19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU5	Computer Network	CO 1	Able to differentiate between OSI and TCP/IP models and identify the responsibility of each layer.
		CO 2	Design, calculate, and apply subnet masks and addresses to fulfill networking requirements.
		CO 3	Acquire knowledge about routing algorithms, the congestion control algorithms and apply knowledge of Implementation of the routing protocols
		CO 4	Analyze the features and operations of various application layer protocols such as Http,DNS,SMTP,etc

Anil K. Mathur
Approved
Dean, FA & UD

4ITU6 PRINCIPLES OF COMMUNICATION

MAX_MARKS(50+100)

Objectives:

1. To introduce communication and to demonstrate the importance of communication in a variety of contexts including that of the manager of innovation and change.
2. To evaluate and discuss the characteristics of good communication and how to improve our communication.

Syllabus:

ANALOG MODULATION: Concept of frequency translation. Amplitude Modulation: Description of full AM, DSBSC, SSB and VSB in time and frequency domains, methods of generation & demodulation, frequency division multiplexing (FDM). Angle Modulation: Phase and frequency modulation. Descriptions of FM signal in time and frequency domains, methods of generation & demodulation, pre- emphasis & de-emphasis, PLL.

PULSE ANALOG MODULATION: Ideal sampling, Sampling theorem, aliasing, interpolation, natural and flat top sampling in time and frequency domains. Introduction to PAM, PWM, PPM modulation schemes. Time division multiplexing (TDM)

PCM & DELTA MODULATION SYSTEMS: Uniform and Non-uniform quantization. PCM and delta modulation, Signal to quantization noise ratio in PCM and delta modulation.DPCM, ADM, T1 Carrier System, Matched filter detection.Error probability in PCM system.

DIGITAL MODULATION: Baseband transmission: Line coding (RZ, NRZ), inter symbol interference (ISI), pulse shaping, Nyquist criterion for distortion free base band transmission, raised cosine spectrum. Pass band transmission: Geometric interpretation of signals, orthogonalization. ASK, PSK, FSK, QPSK and MSK modulation techniques, coherent detection and calculation of error probabilities.

SPREAD-SPECTRUM MODULATION: Introduction, Pseudo-Noise sequences, directsequencespread spectrum (DSSS) with coherent BPSK, processing gain, probability of error, frequency-hop spread spectrum (FHSS). Application of spread spectrum: CDMA

TEXTBOOKS:

1. Principal of Communication system By Taub Schilling, T.M.H
2. Fundamentals of communication system By Proakis&Salehi, Pearson education

REFERENCES:

1. Communication system by Simon Haykin, John Wiley
2. Communication system (Analog and Digital) By R.P. Singh, S.D.Spare, T.M.H
3. Modern Digital & Analog Communication By B. P. Lathi oxford university

18/2/19 Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU6	Principle of Communication	CO 1	Illustrate analog modulation techniques.
		CO 2	Analyse pulse modulation and sampling techniques for analog communication.
		CO 3	To be able to understand PCM and DELTA Modulation system.
		CO 4	Explain digital Modulation techniques.
		CO 5	Define various techniques used in spread-spectrum modulation.

Approved
Dean, FA & UD

4ITU11 LINUX SHELL PROGRAMMING LAB

MAX_MARKS(50+25)

Objective:

1. To introduce Unix/Linux kernel programming and be able to install OS.
2. Learn and use basic shell commands to perform operations on files and directories.
3. Learn the concepts of File handling and regular expressions.
4. To learn about shell scripts and using operators to solve various problems.
5. Use of if-then-else and various flow control statements in shell scripts.

List of Experiments

1. Use of Basic Unix Shell Commands: ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.
2. Commands related to inode, I/O redirection and piping, process control commands, mails.
3. Shell Programming: Shell script exercises based on following
   (i) Interactive shell scripts (ii) Positional parameters (iii) Arithmetic
   (iv) if-then-fi, if-then-else-fi, nested if-else (v) Logical operators
   (vi) else + if equals elif, case structure (vii) while, until, for loops, use of break
   (viii) Metacharacters (ix) System administration: disk management and daily administration
4. Write a shell script to create a file in $USER /class/batch directory. Follow the instructions
   (i) Input a page profile to yourself, copy it into other existing file;
   (ii) Start printing file at certain line
   (iii) Print all the difference between two file, copy the two files at $USER/CSC/2007 directory.
   (iv) Print lines matching certain word pattern.
5. Write shell script for-
   (i) Showing the count of users logged in,
   (ii) Printing Column list of files in your home directory
   (iii) Listing your job with below normal priority
   (iv) Continue running your job after logging out.
6. Write a shell script to change data format .Show the time taken in execution of this script
7. Write a shell script to print files names in a directory showing date of creation & serial number of the file.
8. Write a shell script to count lines, words and characters in its input (do not use wc).
9. Write a shell script to print end of a Glossary file in reverse order using Array. (Use awk tail)
10. Write a shell script to check whether Ram logged in, Continue checking further after every 30 seconds till success.

18/2/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU11	Linux Shell Programming Lab	CO1	Understanding the installation procedure of the Linux operating system, hands on with simple commands and Installation of Linux packages into the operating system.
		CO2	Able to apply the concepts of file handling, and regular expression using shell programming.
		CO3	Capable to apply and implement grep, awk script.
		CO4	Ability to develop and evaluate shell script program that handle processes.
		CO5	Apply the concepts of debugging and execute shell programming.

Approved
Dean, FA & UD

4ITU12 ADVANCED DATA STRUCTURES LAB

MAX_MARKS(50+25)

Objectives: Upon successful completion of this course, students should be able to:

1. Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains;
2. Apply the algorithms and design techniques to solve problems;
3. Analyze the complexities of various problems in different domains.

Suggested Tools: For implementation and estimation of running time on various sizes of input(s) or output(s) as the case may be, Linux platform is suggested.

EXERCISES:

A. It is expected that teachers will assign algorithms to the students for estimation of time & space complexity. Algorithms reported in various research journals may be chosen by the teachers.

B. Problem on designing algorithms to meet complexity constraints may beassigned. For example, a problem on design, analysis and implementation for transposing a sparse matrix requiring not more than one pass from the original matrix may be assigned.

C. A guide to such problems is given below:

1. Exploring a Binary Heap: Consider a binary heap containing n numbers (the root stores the greatest number). You are given a positive integer $k < n$ and a number x. You have to determine whether the k th largest element of the heap is greater than x or not. Your algorithm must take O(k) time. You may use O(k) extra storage.

2. Merging two search trees: You are given two height balanced binary search trees T and T', storing m and n elements respectively. Every element of tree T is smaller than every element of tree T'. Every node u also stores height of the subtree rooted at it. Using this extra information how can you merge the two trees in time O(log m + log n) (preserving both the height balance and the order)?

3. Complete binary tree as an efficient data-structure:

You are given an array of size n (n being a power of two). All the entries of the array are initialized to zero. You have to perform a sequence of the following online operations :

1. (i) Add(i,x) which adds x to the entry A[i].

2. (ii) Report sum(i,j) = sum of the entries in the array from indices i to j for any $0 < i < j <= n$.

18/2/19

Approved
Dean, FA & UD

It can be seen easily that we can perform the first operation in O(1) time whereas the second operation may cost O(n) in worst case. Your objective is to perform these operations efficiently. Give a data- structure which will guarantee O(log n) time per operation.

4. Problems on Amortized Analysis

a. Delete-min in constant time!!! Consider a binary heap of size n, the root storing the smallest element. We know that the cost of insertion of an element in the heap is O(log n) and the cost ofdeleting the smallest element is also O(log n). Suggest a valid potential function so that the amortized cost of insertion is O(logn) whereas amortized cost of deleting the smallest element is O(1).

b. Implementing a queue by two stack

c. Show how to implement a queue with two ordinary stacks so that the amortized cost of each Enqueue and each Dequeue operation is O(1).

5. Computing a spanning tree having smallest value of largest edge weight: Describe an efficient algorithm that, given an undirected graph G, determines a spanning tree of G whose largest edge weight is minimum over all spanning trees of G.
6. Shortest Path Problems:

i. From a subset of vertices to another subset of vertices

a. Given a directed graph G(V,E), where edges have nonnegative weights. S and D are two disjoint subsets of the set of vertices. Give an O(|V| log |V| + |E|) time algorithm to find the shortest path among the set of paths possible from any node in S to any node in D.

ii. Paths in Directed Acyclic Graph

a. Counting the number of paths Given two nodes u,v in a directed acyclic graph G(V,E). Give an O(|E|) time algorithm to count all the paths from u to v.

b. Path passing through a subset of nodesGiven two nodes u,v and a set of vertices w 1 , w 2 ,...,w k in a directed acyclic graph G(V,E). Give an O(|E|) time algorithm to output a path(if exists) from u to v which passes through each of the nodes w 1 ,...,w k . If there is no such path then your algorithm must report that "no such path exists".

7. Searching for a friend: You are standing at a crossing from where there emerge four roads extending to infinity. Your friend is somewhere on one of the four roads. You do not know on which road he is and how far he is from you. You have to walk to your friend and the total distance traveled by you must be at most a constant times the actual distance of your friend from you. In

terminology of algorithms, you should traverse O(d) distance, where d is the distance of your friend from you.

Anil K. Mathews
18/7/19

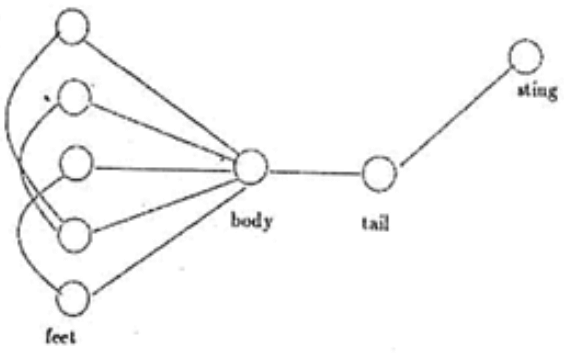
8. A simple problem on sorted array: Design an O(n)-time algorithm that, given a real number x and a sorted array S of n numbers, determines whether or not there exist two elements in S whose sum is exactly x .

9. Finding the decimal dominant in linear time: You are given n real numbers in an array. A number in the array is called a decimal dominant if it occurs more than n/10 times in the array. Give an O(n) time algorithm to determine if the given array has a decimal dominant.

10. Finding the first one: You are given an array of infinite length containing zeros followed by ones. How fast can you locate the first one in the array?

11. Searching for the Celebrity: Celebrity is a person whom everybody knows but he knows nobody. You have gone to a party. There are total n persons in the party. Your job is to find the celebrity in the party. You can ask questions of the form Does Mr. X know Mr. Y ?. You will get a binary answer for each such question asked. Find the celebrity by asking only O(n) questions

12. Checking the Scorpion: An n-vertex graph is a scorpion if it has a vertex of degree 1(the sting) connected to a vertex of degree two (the tail) connected to a vertex of degree n-2 (the body) connected to the other n-3 (the feet). Some of the feet may be connected to other feet. Design an algorithm that decides whether a given adjacency matrix represents a scorpion by examining only O(n) entries.



13. Endless list: You are having a pointer to the head of singly linked list. The list either terminates at null pointer or it loops back to some previous location(not necessarily to the head of the list). You have to determine whether the list loops back or ends at a null location in time proportional to the length of the list. You can use at most a constant amount of extra storage.

14. Nearest Common Ancestor: Given a rooted tree of size n. You receive a series of online queries:

"Give nearest common ancestor of u, v ". Your objective is to preprocess the tree in O(n) time to get a data structure of size O(n) so that you can answer any such query in O(log n) time.

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU12	Advanced Data Structure Lab	CO 1	Able to prove the correctness and analyse the running time of the basic algorithms for the classic problems in various domains
		CO 2	Capable to apply the algorithms and design techniques to solve classic problems
		CO 3	Analyse the complexities of various problems in different domains
		CO 4	Capable to create the efficient algorithms for real-life problems

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

4ITU13 MICROPROCESSOR LAB

MAX_MARKS(50+25)

Objective:

1. To introduce the basics of microprocessor programming.
2. Learn use of microprocessor in simple applications.
3. Theoretical & practical introduction to microcontrollers and microprocessors, assembly language programming techniques.
4. To understand design of hardware interfacing circuit.
5. Design considerations of microcontroller and microprocessor system.

List of Experiments

1. Add the contents of memory locations XX00 &XX01 & place the result in memory location XX02.
2. Add the 16 bit numbers stored in memory location & store the result in another memory location.
3. Transfer a block of data from memory location XX00 to another memory location XX00 in forward & reverse order.
4. Write a program to Swap two blocks of data stored in memory.
5. Write a program to find the square of a number.
6. Write a main program & a conversion subroutine to convert Binary to its equivalent BCD.
7. Write a program to find largest & smallest number from a given array.
8. Write a program to Sort an array in ascending & descending order.
9. Write a program to multiply two 8 bit numbers whose result is 16 bit.
10. Write a program of division of two 8 bit numbers.
11. Generate square wave from SOD pin of 8085 & observe on CRO.
12. Write a program to perform traffic light control operation.
13. Write a program to control the speed ofa motor.

18/2/19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU13	Microprocessor Lab	CO 1	To describe architecture and instruction set of 8085 microprocessor.
		CO 2	To develop experience with Assembly Language Programming.
		CO 3	To demonstrate the students with interfacing of various peripheral devices with 8085 microprocessor.
		CO 4	Design and implement programs on 8085 microprocessor.

Approved
Dean, FA & UD

4ITU14 COMMUNICATION LAB

Objectives: **MAX_MARKS(50+25)**

1. To understand basic analog and digital communication system theory and design, with an emphasis on wireless communications methods.
2. To provide learning experiences that enables students to Analyze and design basic electronic circuits, to carry out AM and FM modulation experiments using discrete electronic components.
3. Become proficient with computer skills for the analysis and design of circuits.
4. Develop understanding about performance of analog communication systems.

List of Experiments

1. Harmonic analysis of a square wave of modulated waveform Observe the amplitude modulated waveform and measures modulation index. Demodulation of the AM signal
2. To modulate a high frequency carrier with sinusoidal signal to obtain FM signal. Demodulation of the FM signal
3. To observe the following in a transmission line demonstrator kit :
   i. The propagation of pulse in non-reflecting Transmission line.
   ii. The effect of losses in Transmission line.
   iii. The resonance characteristics of al half wavelength long x-mission line.
4. To study and observe the operation of a super heterodyne receiver
5. To modulate a pulse carrier with sinusoidal signal to obtain PWM signal and demodulate it.
6. To modulate a pulse carrier with sinusoidal signal to obtain PPM signal and demodulate it.
7. To observe pulse amplitude modulated waveform and its demodulation.
8. To observe the operation of a PCM encoder and decoder. To consider reason for using digital signal x-missions of analog signals.
9. Produce ASK signals, with and without carrier suppression. Examine the different processes required for demodulation in the two cases
10. To observe the FSK wave forms and demodulate the FSK signals based on the properties of (a) tuned circuits (b) on PI.L.
11. To study & observe the amplitude response of automatic gain controller (AGC).

18/2/19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code (SC)	2015IT	2016IT	2017IT	2017IT	

Scheme Code - 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
4ITU14	Communication Lab	CO 1	To analyse digital communications with a software to understand how each component works together.
		CO 2	To analyse, design and implement AM and FM modulation experiments using discrete electronic components.
		CO 3	To understand the basic of MATLAB and PSPICE which are used to simulate the circuit operations.
		CO 4	To illustrate the concepts of communication techniques which are useful for sending information from transmitter to receiver
		CO 5	To demonstrate and compare different analog modulation schemes.

Approved
Dean, FA & UD

SYLLABUS

Semester	**V**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2020-21**

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU1 THEORY OF COMPUTATION

Credit: 4 (3L+1T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. To be able to construct finite state machines and the equivalent regular expressions. Be able to prove the equivalence of languages described by finite state machines and regular expressions.
2. Design context free grammars to generate strings from a context free language and convert them into normal forms.
3. To be able to construct pushdown automata and the equivalent context free grammars and to able to prove the equivalence of languages described by pushdown automata and context free grammars.
4. To be able to construct Turing machines and to prove the equivalence of languages described by Turing machines and Post machines.
5. Distinguish between computability and non-computability and Decidability and un decidability.

Syllabus:

Finite Automata & Regular Expression: Finite Automata & Regular Expression: Basic machine, Finite state machine, , Transition graph, Transition matrix, Deterministic and non-deterministic finite automation, Equivalence of DFA and NDFA, Decision properties, minimization of finite automata, Mealy & Moore machines, Alphabet, words, Operations, Regular sets, relationship and conversion between Finite automata and regular expression and vice versa, designing regular expressions, closure properties of regular sets, Pumping lemma and regular sets, Application of pumping lemma, Power of the languages

Context Free Languages: Context Free Grammars (CFG), Derivations and Languages, Relationship between derivation and derivation trees, leftmost and rightmost derivation, sentential forms, parsing and ambiguity, simplification of CFG, normal forms, Greibach and Chomsky Normal form , Problems related to CNF and GNF including membership problem

Nondeterministic PDA, Definitions, PDA and CFL, CFG for PDA, Deterministic PDA, and Deterministic PDA and Deterministic CFL, pumping lemma for CFL's, Closure Properties and Decision properties for CFL, Deciding properties of CFL

Turing Machines: Introduction, Definition of Turing Machine, TM as language Acceptors and Transducers, Computable Languages and functions, Universal TM & Other modification, multiple tracks Turing Machine

Hierarchy of Formal languages: Recursive & recursively enumerable languages, Properties of RL and REL, Introduction of Context sensitive grammars and languages, The Chomsky Hierarchy

9.7.19

1

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Tractable and Untractable Problems: P, NP, NP complete and NP hard problems, Undecidability, examples of these problems like vertex cover problem, Hamiltonian path problem, traveling sales man problem.

REFERENCE BOOKS:

1. Aho, Hopcroft and Ullman, Introduction to Automata Theory, Formal Languages and Computation, Narosa
2. An introduction to Formal Languages and Automata, Peter Linz
3. Cohen, Introduction to Computer Theory, Addison Wesley.
4. Papadimitriou, Introduction to Theory of Computing, Prentice Hall.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU1	THEORY OF COMPUTATION	CO1	Outline the concept of Finite Automata and Regular Expression.
		CO2	Illustrate the design of Context Free Grammar for any language set.
		CO3	Demonstrate the push down automaton model for the given language.
		CO4	Make use of Turing machine concept to solve the simple problems.
		CO5	Explain decidability or undecidability of various problems.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU2 COMPUTER ARCHITECTURE AND ORGANIZATION

Credit: 4 (3L+1T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To make students understand the basic structure, operation and hardware-software interface of a digital computer.
2. To familiarize the students with arithmetic and logic and implementation of fixed point and floating point arithmetic operations.
3. To familiarize the students with hierarchical memory system including cache memory and virtual memory.
4. To expose the students with various I/O devices and standard I/O interfaces.

Prerequisites: Fundamentals of Computer, Digital Logic Circuits

Syllabus:

Introduction to Computer Architecture and Organization: Von Neuman Architecture, Flynn Classification. Register Transfer and Micro operations: Register transfer language, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Bus and memory transfers. Computer Organization and Design: Instruction cycle, computer registers, common bus system, computer instructions, addressing modes, design of a basic computer

Central Processing : General register organization, stack organization, Instruction formats, Data transfer and manipulation, program control. RISC, CISC characteristics.
Input-Output Organization: Input-Output Interface, Modes of Transfer, Priority Interrupt, DMA, IOP processor.

Data Representation and Arithmetic Algorithms: Number representation: Binary Data representation, two's complement representation and Floating-point representation. IEEE 754 floating point number representation. Integer Data computation: Addition, Subtraction. Multiplication: Signed multiplication, Booth's algorithm. Division of integers: Restoring and non-restoring division Floating point arithmetic: Addition, subtraction

Memory Organization: Introduction to Memory and Memory parameters. Classifications of primary and secondary memories. Types of RAM and ROM, Allocationpolicies, Memory hierarchy and characteristics. • Cache memory: Concept, architecture (L1, L2, L3), mapping techniques. Cache Coherency,Interleavedand Associative memory. Virtual Memory: Concept, Segmentation and Paging, Page replacement policies.

9.7.19 3

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Introduction to parallel processing systems: Introduction to parallel processing concepts , pipeline processing , instruction pipelining, pipeline stages , pipeline hazards, Arithmetic pipeline and Instruction pipeline.

TEXT BOOKS:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, Tata McGraw-Hill.
2. John P. Hayes, "Computer Architecture and Organization", Third Edition.
3. William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.
4. B. Govindarajulu, "Computer Architecture and Organization: Design Principles and Applications", Second Edition, Tata McGraw-Hill.

REFERENCE BOOKS:

1. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization",First Edition, WileyIndia.
2. "Computer Organization" by ISRD Group, Tata McGraw-Hill.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU2	COMPUTER ARCHITECTURE AND ORGANISATION	CO1	Identify the components of instruction set such as opcode, operand and format
		CO2	Conversion of fractional numbers in to IEEE scientific format
		CO3	Implement 32 bit multiplication using iterative method
		CO4	Use of pipelining method to improve the performance of a instruction set
		CO5	Understand the design of direct mapped and associative cache
		CO6	To understand the concepts of parallel processing

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU3 DATABASE MANAGEMENT SYSTEM

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To provide concept and need of Database System.
2. To cover the concepts of ER-data model and Relational data model.
3. To make students design database with the help of E-R model and Normalization.
4. To know the fundamental concepts of transaction processing- concurrency control techniques and recovery procedure.

Syllabus:

Introduction to Database Systems: Overview of DBMS, Database approach v/s Traditional file accessing approach, Advantages, of database systems, Data models, Schemas and instances, Data independence, Data Base Language and interfaces, Overall Database Structure, Functions of DBA.

ER Data Model: Overview of Data Design Entities, Attributes and Entity Sets, Relationship and Relationship Sets. Features of the ER Model- Key Constraints, Participation Constraints, Weak Entities, Design with ER Model, Concept of Generalization, Aggregation and Specialization. Various other data models like object oriented data Model, Network data model, and Relational data model, Comparison between the three types of models, Transforming ER diagram into the tables.

Relational Data models: Domains, Tuples, Attributes, Relations, Characteristics of relations, Keys, Key attributes of relation, Relational database, Schemas, Integrity constraints. Referential integrity, Relationship Algebra, Selection, Projection, Set Operations, Renaming, Joins, Division, Relation Calculus, Expressive Power of Algebra and Calculus.

Schema refinement and Normal forms: Introductions to Schema Refinement, Functional Dependencies, Normal forms (First, Second, Third and BCNF normal form), Dependency preservation and lossless join, Problems with null valued and dangling tuples.

Transaction Processing: Introduction to Transaction Processing, Transaction State, Transaction Properties, Serial and Non-serial Schedules, Characterizing Schedules Based on Recoverability and Serializability, Need of Serializability, Conflict vs. View Serializability, Testing for Serializability.

Concurrency Control: Implementation of Concurrency: Lock-based protocols, Timestamp-based protocols, Validation-based protocols, Deadlock handling,
Database Failure and Recovery: Database Failures, Recovery Schemes: Shadow Paging and Log-based Recovery, Recovery with Concurrent transactions.

5 9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Elmasi, R. and Navathe, S.B., "Fundamentals of Database Systems", 4th Ed., 2005, Pearson Education.
2. A Silberschatz, H Korth, S Sudarshan, "Database System and Concepts", fifth Edition McGraw-Hill , 2005
3. Ramakrishnan, R. and Gekhre, J., "Database Management Systems", 3rd Ed., 2003, McGraw-Hill.

REFERENCE BOOKS:

1. Databases Illuminated 3nd Ed., Catherine Ricardo and Susan Urban, Jones and Bartlett, 2017
2. Date, C. J., "Introduction to Database Systems", 2002, Pearson Education.

COURSE OUTCOMES:

Course code	Course name	Course outcome	Details
5ITU3	DATABASE MANAGEMENT SYSTEM	CO1	Describe data models and schemas in DBMS.
		CO2	Apply logical database design principles, including E-R diagrams and database normalization.
		CO3	Construct simple and moderately advanced database queries using relational Algebra.
		CO4	To understand the concept of Transaction and Concurrency Control.
		CO5	To understand the concept of Failure and Recovery.

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU4 CYBER SECURITY MANAGEMENT

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. Introduce details of cyber security, cyber crime to learn how to avoid becoming victims of cyber crimes.
2. To understand about security attack, services and mechanism.
3. To gain a fundamental understanding of cyber crime and law.
4. To acquire knowledge of vulnerability, security tools and method.

Prerequisites: Basic fundamental knowledge of Networking, Web Application

Syllabus:

Introduction of Cyber Crime, Challenges of cyber crime, Classifications of Cybercrimes: E-Mail Spoofing/Security, Spamming, Internet Time Theft, Salami attack/Technique, Risk in Social Networking.

Web jacking, Online Frauds, Software Piracy, Computer Network Intrusions, Password Sniffing, Identity Theft, cyber terrorism, Virtual Crime, Perception of cyber criminals: hackers, insurgents and extremist group etc. Web servers hacking, session hijacking.

Cyber Crime and Criminal justice: Concept of Cyber Crime and the IT Act 2000, Hacking, Teenage Web Vandals, Cyber Fraud and Cheating, Defamation, Harassment and E-mail Abuse, Other IT Act Offences, Monetary Penalties, jurisdiction and Cyber Crimes, Nature of Criminality, Strategies to tackle Cyber Crime and Trends, Digital Forensics.

Overview of vulnerability scanning: OpenSSL, DVWA, Webgoat, Metasploit. Network Sniffers and Injection Tools:Tcpdump and Windump,Wireshark. Network Address Translation (NAT) and Port Forwarding, Network Defense tools: Firewalls, Use of Firewall, VPN, DNS, NMAP.

Proxy Servers and Anonymizers, Password Cracking, Key loggers and Spyware, virus and worms, Trojan Horses, Backdoors, Ransomware, DoS and DDoS Attacks , Buffer and Overflow, Attack on Wireless Networks, Phishing : Method of Phishing, Phishing Techniques, Cyber Insurance, Crytocurrencies, Introduction to Blockchain.

Case Study: 1. Banking Related Frauds, Credit Card Related Frauds 2.Cyber defamation: A Young Couple Impacted.

9.2.19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and S Belpure, Publication Wiley.
2. AntiHaCker Tool Kit (Indian Edition) by Mike Shema, Publication McGraw Hill.

REFERENCE BOOKS:

1. Principles of Cyber crime, Jonathan Clough Cambridge University Press.
2. Cyber Law Simplified, VivekSood, Pub: TMH
3. Information Warfare: Corporate attack and defense in digital world, William Hutchinson, Mathew Warren, Elsevier.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU4	CYBER SECURITY MANAGEMENT	CO1	Gain a fundamental knowledge of what Cyber Security is and Apply knowledge of computer science to provide security.
		CO2	Classify different type of attack and how to identify and prevent.
		CO3	Identify issues to protect digital assets in compliance with cyber laws.
		CO4	Determine the vulnerability to detects and classifies system weaknesses in networks, application and predicts the effectiveness of countermeasures
		CO5	Acquire knowledge about network security tools and authentication applications and apply legal and ethical aspects to manage and audit digital assets.

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU5.1 STATISTICS AND PROBABILITY THEORY

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To understand concepts of Statistics and Probability and Analyze discrete & continuous distributions.
2. Understand Correlation & Regression analysis and Markov process.

Syllabus:

Introduction & Discrete random variables: Sample space, events, algebra of events, Bernoulli's trials, Probability & Baye's theorem. Random variable & their event space, probability generating function, expectations, moments, computations of mean time to failure, Bernoulli & Poisson processes.

Discrete & continuous distributions: Probability distribution & probability densities: Binomial, Poisson, normal rectangular and exponential distribution & their PDF's, moments and MGF's for above distributions.

Correlation & Regression: Linear regression, Rank correlation, Method of least squares Fitting of straight lines & second degree parabola. Linear regression and correlation analysis.

Queuing Theory: Pure birth, pure death and birth-death processes. Mathematical models for M/M/1, M/M/N, M/M/S and M/M/S/N queues.

Discrete Parameter Markov chains: M/G/1 Queuing model, Discrete parameter birth-death process.

TEXT BOOKS:

1. Statistics Probability and Random process , Dr. K.C.Jain & Dr. M.L.Rawat

REFERENCE BOOKS:

1. Hogg, R.V. & Craig, A.T., Introduction to Mathematical Statistics, 5th Ed., Prentice-Hall, Inc., Englewood Cliffs, N.J., 1995.
2. Mood, A.M., Graybill, F.A. and Boes, D.C., Introduction to the Theory of Statistics, 3rd Ed. McGraw Hill, Inc., New York, 1974

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU5.1	STATISTICS AND PROBABILITY THEORY	CO1	Find the probability of the given data in every decision making process and analyse the given data on applying Bayes' Theorem
		CO2	Construct probabilistic models for observed phenomena through certain discrete & continuous distributions
		CO3	Construct probabilistic models for observed phenomena through certain continuous distributions which play an important role in many engineering applications.
		CO4	Analyze sample data and interpret the same for population using correlation & regression.
		CO5	Correlate two variables and fit the curves for prediction using data.
		CO6	Analyze Queuing Theory mathematical models

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU5.2 ADVANCED GRAPH THEORY

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Demonstrate the knowledge of fundamental concepts in Graph theory.
2. Apply models of Graph theory to solve problems of connectivity and uncertainty.

Syllabus:

Introduction : Finite and Infinite graphs, Incidence and Degree, Isolated Vertex, Pendant Vertex and Null Graph, Isomorphism, Subgraphs, Walk,Path, and Circuits, Connected Graphs, Disconnected Graphs and Components, Euler Graphs, Operation on Graphs, Hamiltonian Paths and Circuits.

Cuts-sets and Cut- Vertices: Cut-Sets, Properties of Cut-sets, Cut-Sets in a Graph, Connecivity and Separability, Network Fows, 1- Isomorphism, 2- Isomorphism.
Planar and Dual Graph: Combinatorial Vs Geometric Graphs, Planar Graphs, Different representation of Planar Graphs, Detection of Planarity, Thickness and Crossing, Kuratowski's Theorem, Wagner's Theorem.

Matrix Representation of Graphs: Incidence Matrix, Submatrics of A(G), Circuit Matrix, Cut-Set Matrix,Path Matrix, Adjacency Matrix.
Coloring, Covering and Partitioning: Chromatic Number, Chromatic Partitioning, Chromatic Polynomial, Matching, Covering. Maximum Matching,Hall's matching condition,Min-Max theorms, Independent sets and Covers.

Connectivity and Paths: Cuts and Connectivity, k-Connected Graphs, Network Flow Ford-Fulkerson Labeling Algorithm, Max-Flow Min-cut Theorem, Menger's Proof using Max-Flow Min-Cut Theorem.

Perfect Graph: The Perfect graph theorem, Classes of perfect graph. Matroid: Properties of Matroid, The dual of Matroid,Matroid Minors and Planer graph,Matroid Intersection and Union

TEXT BOOKS:

1. N. Deo: Graph Theory with Application to Engineering and Computer Science, PHI.
2. D.B. West, Introduction to Graph Theory, Prentice Hall.

REFERENCE BOOKS:

1. J.A.Bondy and U.S.R.Murty: Graph Theory, Springer.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU5.2	ADVANCED GRAPH THEORY	CO1	Demonstrate the knowledge of fundamental concepts in Graph theory.
		CO2	Apply models of Graph theory to solve problems of connectivity and uncertainty.
		CO3	Analyzing graphs and random phenomena occurring in real life situations using Graph theory.
		CO4	Interpret the models of Graph theory for real life and engineering problems.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU5.3 DATA MINING AND WAREHOUSE

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To enable students to understand and implement classical algorithms in data mining and data warehousing.
2. Students will learn how to analyze the data, identify the problems, and choose the relevant algorithms to apply.
3. Students will be able to assess the strengths and weaknesses of the algorithms and analyze their behavior on real datasets.

Prerequisites: Linear Algebra, Probability and Statistics, Programming

Syllabus:

Introduction: Types of Data, Data Mining Functionalities, Interestingness of Patterns, Classification of Data Mining Systems, Data Mining Task Primitives, Integration of a Data Mining System with a Data Warehouse, Issues, Data Preprocessing.

Frequent Pattern Miming: Mining Frequent Patterns, Associations and Correlations, Mining Methods, Mining various Kinds of Association Rules, Correlation Analysis, Constraint Based Association Mining Classification and Prediction, Basic Concepts, Decision Tree Induction, Bayesian Classification Rule Based Classification, Classification by Back propagation, Support Vector Machines Associative Classification , Lazy Learners, Other Classification Methods, Prediction.

Cluster Analysis: Types of Data, Categorization of Major Clustering Methods, K-means Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid Based Methods Model-Based Clustering Methods, Clustering High Dimensional Data, Constraint Based Cluster Analysis, Outlier Analysis, Data Mining Applications.

Data Warehouse: Data warehousing Components, Building a Data warehouse, Mapping the Data Warehouse to a Multiprocessor Architecture, DBMS Schemas for Decision Support, Data Extraction, Cleanup, and Transformation Tools, Metadata.

Reporting and Query tools and Applications, Tool Categories, The Need for Applications, Cognos Impromptu, Online Analytical Processing (OLAP), Multidimensional Data Model OLAP Guidelines, Multidimensional versus Multirelational OLAP, Categories of Tools, OLAP Tools and the Internet.

9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

CASE STUDY: Use your knowledge gained in the subject to explain how online retailers track and process our web browsing data to show us specific advertisements.

TEXT BOOKS:

1. Alex Berson and Stephen J.Smith, "Data Warehousing, Data Mining and OLAP", Tata McGraw Hill Edition, Thirteenth Reprint 2008.
2. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Third Edition, Elsevier, 2012.

REFERENCE BOOKS:

1. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Person Education, 2007.
2. K.P. Soman, Shyam Diwakar and V. Aja, "Insight into Data Mining Theory and Practice", Eastern Economy Edition, Prentice Hall of India, 2006.
3. G. K. Gupta, "Introduction to Data Mining with Case Studies", Eastern Economy Edition, Prentice Hall of India, 2006.
4. Daniel T.Larose, "Data Mining Methods and Models", Wiley-Interscience, 2006.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU5.3	DATA MINING AND WAREHOUSE	CO1	Understand the functionality of the various data mining components.
		CO2	Appreciate the strengths and limitations of various data mining models.
		CO3	Compare and contrast the various classifiers.
		CO4	Understand various clustering methodologies.
		CO5	Describe and utilise a range of techniques for designing data warehousing and data mining systems for real-world applications.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU6.1 GSM COMMUNICATION AND MOBILE DATA NETWORKS

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To impart knowledge of wireless communication including challenges in wireless communication.
2. To cover various design and realization aspects for a cellular network
3. To recollect the journey of cellular networks and explain future developments.

Syllabus:

Wireless Channels: Large scale path loss – Path loss models: Free Space and Two-Ray models - Link Budget design – Small scale fading- Parameters of mobile multipath channels – Time dispersion parameters-Coherence bandwidth – Doppler spread & Coherence time, Fading due to Multipath time delay spread – flat fading – frequency selective fading – Fading due to Doppler spread – fast fading – slow fading.

Cellular Architecture: Multiple Access techniques - FDMA, TDMA, CDMA – Capacity calculations–Cellular concept- Frequency reuse - channel assignment- hand off- interference & system capacity- trunking & grade of service – Coverage and capacity improvement.

Digital Signalling For Fading Channels: Structure of a wireless communication link, Principles of Offset-QPSK, p/4-DQPSK, Minimum Shift Keying, Gaussian Minimum Shift Keying, Error performance in fading channels, OFDM principle – Cyclic prefix, Windowing, PAPR.

Adaptive equalization, Linear and Non-Linear equalization, Zero forcing and LMS Algorithms, Rake Receiver, Diversity Combining Techniques, MIMO systems, MIMO Capacity Calculation

Wireless Systems and Standards: GSM Services and Features, Second and Third Generation Cellular Networks, 3G CDMA, WLAN, WLL, Bluetooth and Personal Area Networks, Concept of 4G and 5G Cellular Networks

9.7.19

Anil K. Mathus
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU6.1	GSM COMMUNICATION AND MOBILE DATA NETWORKS	CO1	Identify and estimate the challenges of wireless communication
		CO2	Differentiate among design paradigms and components of building a cellular network
		CO3	Develop mathematical basis for realizing cellular communication
		CO4	Ability to Manipulate and fine tune system components for better capacity extraction
		CO5	Describe evolution of Cellular communication systems and developments including explanation of future possibilities

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU6.2 EMBEDDED SYSTEM

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Discuss the major components that constitute an embedded system.
2. Implement small programs to solve well-defined problems on an embedded platform.
3. To provide experience to integrate hardware and software for microcontroller application systems.

Prerequisites: Microprocessor And Interfaces

Syllabus:

Embedded Computing Requirements: Characteristics and applications of embedded systems; Components of Embedded Systems; challenges in Embedded System Design and design process; Formalism for system design.

Embedded Computing Platform: CPU Bus- Bus protocols, DMA, system bus configurations; Memory Devices- Memory Device Organization; Random-Access Memories; Read-Only Memories; Timers and counters, A/D and D/A converters, Keyboards, LEDs, displays and touch screens; Design examples.

Embedded Processors: RISC vs. CISC architectures; ARM processor – processor architecture and memory organization, instruction set, data operations and flow control; ARM bus; Parallelism within instructions; Input and output devices, supervisor mode, exception and traps; Memory system, pipelining and superscalar execution.

Embedded Software Analysis and Design: Components for Embedded Programs; Model programs – data flow graphs and control/data flow graphs; Assembly, Linking and Loading; Basic Compilation techniques; Program Optimization- Expression Simplification; Dead Code Elimination; Procedure Inlining; Loop Transformations; Register Allocation; Scheduling; Instruction Selection

Embedded System Accelerators: Processor accelerators, accelerated system architecture, Co-Processor, accelerated system design, Performance Analysis, Accelerator CPU interface, Case Studies for Embedded System Accelerators- Setup Box, CD/DVD player, Router.

TEXT BOOKS:

1. Computer as Components by Wayne Wolf published by Elsevier Inc
2. ARM System Developer's Guide by Andrew S. Loss published by Elsevier Inc

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOKS:

1. Embedded System Design by Steve Heath published by Elsevier Inc
2. Embedded System design: A unified hardware/software Introduction by Frank Vahid & Tony.
3. Givagi published by John Wiley & Sons Inc.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU6.2	EMBEDDED SYSTEM	CO1	Define Embedded System and its Components
		CO2	Illustrate bus configuration and memory organization
		CO3	Convert C program into assembly language using ARM instruction set.
		CO4	Identify correct optimization technique for assembly language program
		CO5	Examine case studies related to Embedded System Accelerators.

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU6.3 HUMAN COMPUTER INTERFACE

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Describe and apply user-centered design methods to conduct formative and summative evaluations
2. Explain and apply core theories and models from the field of HCI.
3. Design and implement useful, usable, and engaging graphical computer interfaces.
4. Discuss and critique research in the field of HCI.
5. Describe special considerations in designing user interfaces for wellness.

Syllabus:

The Human: input-output channels, Human memory, thinking, emotions, individual differences, psychology and the design of interactive systems.

The Computer: Text entry devices with focus on the design of key boards, positioning, pointing and drawing, display devices. The Interaction: Models of interaction, ergonomics, interaction styles, elements of WIMP interfaces, interactivity, experience, engagement and fun. Paradigms for Interaction.

Design Process: The process of design, user focus, scenarios, navigation design screen design and layout, iteration & prototyping. Usability Engineering

Design rules: Principles to support usability, standards, guidelines, rules and heuristics, HCI patterns.

Evaluation Techniques: Definition and goals of evaluation, evaluation through expert analysis and user participation, choosing an evaluation method.
User support, requirement, approaches, adaptive help systems, designing user support systems

Cognitive methods: Goals and task hierarchies, linguistic models, challenges of display based systems, physical and device models, cognitive architectures.

Communications and collaborations models: Face to Face communication, conversations, Text based communication, group working.
Task Analysis: Differences between task analysis and other techniques, task decomposition, knowledge based analysis, ER based analysis, sources of information and data collection, use of task analysis.

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. "Human-Computer Interaction 3/E", Dix, Prentice Hall.
2. "Smart Things: Ubiquitous Computing User Experience Design, Mike Kuniavsky".
3. "The UX Book: Process and Guidelines for Ensuring a Quality User Experience", Rex Harston and Pardha Pyla.

REFERENCE BOOKS:

1. Design of Everyday Things", Donald Norman
2. "Designing for the Digital Age: How to Create Human-Centered Products and Services", Kim Goodwin and Alan Cooper
3. "Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications", Third Edition, Julie Jacko

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU6.3	HUMAN COMPUTER INTERFACE	CO1	Explain the capabilities of both humans and computers from the viewpoint of human information processing.
		CO2	Describe typical human–computer interaction (HCI) models, styles, and various historic HCI paradigms
		CO3	Identify the use HCI design principles, standards and guidelines
		CO4	Analyze user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.
		CO5	Discuss HCI issues in groupware, ubiquitous computing, virtual reality, multimedia, and Word Wide Web-related environments

9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU11 APPLICATION DEVELOPMENT IN JAVA LAB

Credit: 2 (0L+0T+3P) **Max. Marks: 75(IA: 50, ETE: 25)**

Objectives:

1. To know about IDE tool and Java feature.
2. To design java program using file handling, exception handling, and multithreading.
3. To develop a web based application using java feature.

List of Experiments

1. Java program to display Welcome message, demonstrate Command line arguments and Scanner(I/O Streams).
2. Write a Java program to demonstrate BufferedReader(I/O Streams) and Arrays.
3. A program to illustrate the concept of class with Constructor overloading, Method overloading, Method overriding and Dynamic Polymorphism Problem Definition.
4. Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
5. A program to illustrate the concept of Single inheritance and Multilevel inheritance Problem Definition.
6. Write a Java program to implement the concept of exception handling Problem Definition.
7. Program to illustrate the concept of multithreading that creates three threads. First thread displays —Good Morningl every second, the second thread displays —Hellol every two seconds and the third thread displays —Welcomel every three seconds.
8. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable and reads a file and displays the file on the screen, with a line number before each line.
9. Develop an applet that displays lines, rectangles, ovals, square etc and illustrate GUI Components using AWT/SWING.
10. Write a java program that connects to a database using JDBC and does add, deletes, modify and retrieve operations.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Mini Project:

1. Design a Desktop application like ERP,CRM using AWT/SWING.
2. Design a web based application like hotel management system or flight booking using servlet, jsp and jdbc.

REFERENCE BOOKS:

1. Herbert Scheldt, —The Complete Reference Java, 7th Edition, Tata McGraw Hill, 2006.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002. 3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 5th Edition,
3. McGraw Hill Publishing, 2010.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education PHI

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU11	APPLICATION DEVELOPMENT IN JAVA LAB	CO 1	Use java functionality based feature like (inheritance, overloading polymorphism etc.).
		CO 2	Develop a web application using servlet, jsp and jdbc.
		CO 3	Design a program for file handling and multithreading.
		CO 4	Use of IDE tools like Netbens.

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU12 DATABASE MANAGEMENT SYSTEM LAB

Credit: 1 (0L+0T+2P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. Implementing databases and applications software primarily in the relational model.
2. Using querying languages, primarily SQL, and other database supporting software.
3. The design methodology for databases and verifying their structural correctness.
4. Working in group settings to design and implement database projects.

Prerequisites: RDBMS

List of Experiments:

1. Design a Database and create required tables. For e.g. Employee , Department, College Database.
2. Apply the constraints like Primary Key , Foreign key, NOT NULL to the tables.
3. Write a SQL statement for implementing ALTER,UPDATE and DELETE.
4. Write the queries to implement the joins.
5. Write the query for implementing the following functions: MAX(),MIN(),AVG() and COUNT().
6. Write the query to implement the concept of Integrity constraints.
7. Write the query to create the views.
8. Perform the queries for triggers.
9. Perform the following operation for demonstrating the insertion , updation and deletion using referential integrity constraints.
10. Write the query for creating the users and their role.

Database Designing Project:

For better understanding students (group of 3-4 students) should design data base for of the following projects.

9.7.19

Anil K. Mathur
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Project 1: Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts.

Description

a) The university keeps track of each student's name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student's permanent address and to the student's last name. Both Social Security number and student number have unique values for each student.
b) Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department.
c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course.
d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.
e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3,or 4).

1- Draw an ER diagram for the schema (**Make sure to use correct notation for specifying cardinality ratios, total/partial participations, key constraints.**)
2- Design the relational schema for this application.
3-Create tables in SQL for all the relations along with constraints.

Project 2: Consider a MAIL_ORDER database in which employees take orders for parts from customers.

Description

The data requirements are summarized as follows:

a) The mail order company has employees, each identified by a unique employee number, first and last name, and Zip Code.
b) Each customer of the company is identified by a unique customer number, first and last name, and Zip Code.
c) Each part sold by the company is identified by a unique part number, a part name, price, and quantity in stock.
d) Each order placed by a customer is taken by an employee and is given a unique order number. Each order contains specified quantities of one or more parts. Each order has a date of receipt as well as an expected ship date. The actual ship date is also recorded.

9.2.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

1- Draw an ER diagram for the schema **(Make sure to use correct notation for specifying cardinality ratios, total/partial participations, key constraints.)**

2- Design the relational schema for this application.

3-Create tables in SQL for all the relations along with constraints.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU12	DATABASE MANAGEMENT SYSTEM LAB	CO1	Create and populate a RDBMS using SQL.
		CO2	Create queries in SQL to retrieve information from a database.
		CO3	Design and verify structural correctness of data base,
		CO4	Design conceptual models of a database using ER diagram or Normalization for real life applications.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU13 NETWORKING PROGRAMMING LAB

Credit: 1 (0L+0T+2P) **Max. Marks: 75(IA: 50, ETE: 25)**

Objectives:

1. Analyses the requirements of a networked programming environment and identify the issues to be solved;
2. Understand the use of TCP/UDP Sockets
3. Experiment with protocol analyzers to understand and analyze the operation of the different TCP/IP protocols
4. Install and configure a network simulator.

Prerequisites: Basic fundamental knowledge of Networking and Programming

List of Experiments:

1. To study the usage of various basic tools (crimping, krone etc.) used in establishing a LAN.
2. To learn the usage of connecters and cables (cabling standards) used in networks.
3. To familiarize with switch, hub, routers & bridges used in networks and configuration of router, hub, switch etc. using real devices or simulators.
4. To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.
5. Running and using services/commands like ping, trace route, nslookup, arp, telnet, etc
6. Installation of Wire shark, tcpdump, etc and observe data transferred in client server communication using UDP/TCP and identify the UDP/TCP datagram.
7. Implementation of specific network topology (Number of nodes and physical layer configuration) with respect to simulation using tools like Packet Tracer, NetSim:
8. Socket Programming with C/Java:
   1. TCP Client, TCP Server 2. UDP Client, UDP Server
9. Implementation of File access using RPC/RMI.
10. A case study to design and configure any organization network example College network or campus network, using any packet tracer or network topology design software based on infrastructure requirements, servers and clients, traffic consideration and application requirements.

TEXT BOOKS

1. Douglas E. Comer, Hands-on Networking with Internet Technologies, Pearson Education.
2. W. Richard Stevens, Unix Network Programming, Prentice Hall/Pearson Education, 2009.

9.7.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOKS

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks" Pearson Education.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU13	NETWORKING PROGRAMMING LAB	CO1	Get familiar with the basic network commands, connectors, tools, and demonstrate their use in different network scenarios
		CO2	To write, execute and debug a program which uses Socket API.
		CO3	Analyze network traffic using network monitoring tools.
		CO4	Configurations of LAN, Networking devices.
		CO5	Design and configure a network for any organization network using simulator.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

5ITU14 SOFTWARE DESIGN LAB

Credit: 1 (0L+0T+2P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. Explain the role of a system analyst.
2. Identify the important parts of SRS document.
3. Identify the functional requirements from any given problem description.
4. Develop real time project according to new methodology.
5. Identify the important properties of a good SRS document.

Prerequisites: Software Engineering

List of Experiments

1. Prepare the SRS document. You should identify the appropriate requirements for each problem; Draw the Use Case diagrams, Domain Models, and Class Diagrams using Rational Rose. ; Draw the Sequence Diagrams and Collaboration Diagrams for each Use Case, using Rational Rose; Draw the State Chart Diagrams and Activity Diagrams using Rational Rose, wherever necessary; Develop the corresponding software using Java with an interactive GUI and appropriate Database ;

2. Develop software to automate the bookkeeping activities of a 5 star hotel ; The local newspaper and magazine delivery agency wants to automate the various clerical activities associated with its business.

3. Develop a software for this ; A small automobile spare parts shop sells spare parts for vehicles of several makes and models. Each spare part is typically manufactured by several small industries. To streamline the sales and supply ordering, the shop owner wants to automate the activities associated with his business.

4. Develop a software for this; Develop software for the automation of the dispensary of RTU Kota
5. Develop software for automating various activities of the Estate Office of RTU Kota ;
6. Develop a word processing software with some limited number of facilities such as making bold, italics, underline, cut, copy and paste etc ;

7. Develop a graphics editor software package, using which one can create / modify several common types of graphics entities ;
8. Develop software for automating various activities of the department offices of RTU Kota ;

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. R. S. Pressman, Software Engineering A Practitioner's Approach, McGraw Hill Publications, 2006.

REFERENCE BOOKS:

1. R. Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2nd Ed, 2006.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
5ITU14	SOFTWARE DESIGN LAB	CO1	Practice the concept of software development methodology
		CO2	Obtain new skill of preparation of SRS document according to real time project.
		CO3	Describe advanced and emerging in project development models
		CO4	Obtain skills of design UML models of real time case study

9.7.19

Anil K. Mathur
Approved
Dean, FA & UD

SYLLABUS

Semester	**VI**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2020-21**

Anil K. Mathus
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU1 OPERATING SYSTEM

Credit 4: (3L+1T+0P) **Max. Marks 150(IA:50 ETE:100)**

Objectives:

1. Understanding of basics OS and its types and overview of various modules of OS.
2. Construct Process model and understanding of the IPC and Process management.
3. Knowledge of various process scheduling and Deadlocks avoidance strategies.
4. Study of various algorithms of memory allocation schemes, page replacement, file organization and disk scheduling.

Syllabus:

Introduction and need of operating system, layered architecture/logical structure of operating system, Type of OS, operating system as a resource manager and virtual machine, OS services,

Process management- Process model, creation, termination, states & transitions, hierarchy, context switching, process implementation, process control block, Basic System calls- Linux & Windows. Threads- processes versus threads, threading, concepts, models, kernel & user level threads, thread usage, benefits, multithreading models.
Inter-process communication- Introduction to message passing, Race condition, critical section problem, mutual exclusion with busy waiting- disabling interrupts, lock variables, strict alternation, Peterson's solution, TSL instructions, busy waiting, sleep and wakeup calls, semaphore, monitors, classical IPC problems.

Process scheduling- Basic concepts, classification, CPU and I/O bound, CPU scheduler- short, medium, long-term, dispatcher, scheduling:- preemptive and non-preemptive, Static and Dynamic Priority, Co-operative & Non-cooperative, Criteria/Goals/Performance Metrics, scheduling algorithms- FCFS, SJFS, shortest remaining time, Round robin, Priority scheduling, multilevel queue scheduling, multilevel feedback queue scheduling, Fair share scheduling.
Deadlock- System model, resource types, deadlock problem, deadlock characterization, methods for deadlock handling, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Memory management- concepts, functions, logical and physical address space, address binding, degree of multiprogramming, swapping, static & dynamic loading- creating a load module, loading, static & dynamic linking, shared libraries, memory allocation schemes- first fit, next fit, best fit, worst fit, quick fit. Free space management- bitmap, link list/free list, buddy's system, memory protection and sharing, relocation and address translation.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Virtual Memory- concept, virtual address space, paging scheme, pure segmentation and segmentation with paging scheme hardware support and implementation details, memory fragmentation, demand paging, pre-paging, working set model, page fault frequency, thrashing, page replacement algorithms- optimal, NRU, FIFO, second chance, LRU, LRU- approximation clock, WS clock; Belady's anomaly, distance string; design issues for paging system- local versus global allocation policies, load control, page size, separate instruction and data spaces, shared pages, cleaning policy, TLB (translation look aside buffer) reach, inverted page table, I/O interlock, program structure, page fault handling, Basic idea of MM in Linux & windows.

Input/Output subsystems- concepts, functions/goals, input/output devices- block and character, spooling, disk structure & operation, disk attachment, disk storage capacity, disk scheduling algorithm- FCFS, SSTF, scan scheduling, C-scan schedule.

TEXT/REFERENCE BOOKS:

1. A. Silberschatz and Peter B Galvin: Operating System Principles, Wiley India Pvt. Ltd.
2. Achyut S Godbole: Operating Systems, Tata McGraw Hill
3. Tanenbaum: Modern Operating System, Prentice Hall.
4. DM Dhamdhere: Operating Systems – A Concepts Based Approach, Tata McGraw Hill
5. Charles Crowly: Operating System A Design – Oriented Approach, Tata McGraw Hill.

COURSE OUTCOMES:

Course code	Course name	Course outcome	Details
6ITU1	OPERATING SYSTEM	CO1	Identify the basics OS and its types and overview of various modules of OS.
		CO2	Construct Process model and define Process management.
		CO3	Apply of various process scheduling and Deadlocks avoidance strategies.
		CO4	Recognise various algorithms of memory allocation schemes, page replacement algorithms.
		CO5	Describe the file organization and disk scheduling algorithms.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU2 COMPUTER GRAPHICS

Credit: 4 (3L+1T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Gain knowledge about graphics hardware devices and software used.
2. Understand the two dimensional and three dimensional graphics and their transformations.
3. Be familiar with understand clipping techniques.
4. Determine projections and visible surface detection techniques for display.
5. Appreciate illumination and color models.

Prerequisites: Linear algebra , Data Structures and Algorithms.

Syllabus:

Introduction and Line Generation: Types of computer graphics, Graphic Displays, Random scan displays, Raster scan displays, Frame buffer and video controller, Scan Conversion of Point, Line, Circle, Ellipse and Polygon, Introduction to Aliasing and Anti Aliasing technique.

Transformations: Basic transformation, Matrix representations and homogeneous coordinates, Composite transformations, Reflections and shearing. Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Point Clipping; 2-D Line clipping: Cohen Sutherland, Liang Barsky and Cyrus–Beck line clipping algorithm; 2-D Polygon clipping: Sutherland Hodgeman, Weiler and Atherton polygon clipping.

3-D Geometric Primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections.

Hidden Lines & Surfaces: Back Face Detection algorithm, Depth buffer method, A- buffer method, Scan line method, Curves and Splines: Parametric and Non parametric Representations, Bezier curve, BSpline Curves.

Rendering: Basic illumination model, Diffuse reflection, Specular reflection, Phong shading, Gourand shading, Ray tracing, Color models like RGB, YIQ, CMY, HSV.

TEXT BOOKS:

1. Donald Hearn and Pauline Baker M, "Computer Graphics", Prentice Hall, New Delhi, 2007.
2. Theory and Problems of Computer Graphics, Zhigang Xiang, Roy, plastock, Schaum's outline series.
3. J. Foley, A. Van Dam, S. Feiner, J. Hughes: Computer Graphics- Principles and Practice,Pearson.

9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOKS:

1. Jeffrey McConnell, "Computer Graphics: Theory into Practice", Jones and Bartlett Publishers, 2006.
2. Hill F S Jr., "Computer Graphics", Maxwell Macmillan" , 1990.

COURSE OUTCOMES:

Course code	Course name	Course outcome	Details
6ITU2	COMPUTER GRAPHICS	CO1	Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
		CO2	Design two and three dimensional graphics.
		CO3	Apply two dimensional and three dimensional transformations.
		CO4	Determine projections and visible surface detection techniques for display.
		CO5	Understand and apply the concept of illumination and color models.

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU3 ARTIFICIAL INTELLIGENCE

Credit: 3 (3L+0T+0P) **Max. Marks:150 (IA:50, ETE:100)**

Objectives:

1. To have an overview of various AI system and applications
2. To know about basic concepts of Problem solving method and game playing.
3. To obtain a thorough knowledge of various knowledge representation schemes
4. To know about various learning methods and natural language processing.

Syllabus:

Introduction to AI and Intelligent agent, Different Approach of AI
Problem Solving: Solving Problems by Searching, Uninformed search, BFS, DFS, Iterative deepening, Bi directional search, Informed search techniques: heuristic, Greedy search, A* search, AO* search, Hill climbing, constraint satisfaction problems.

Game Playing: minimax, alpha-beta pruning, tic-tac-toi, jug problem, chess problem, tiles problem

Knowledge representation and Reasoning: Building a Knowledge Base: Propositional logic, first order logic, Theorem Proving in First Order Logic, Resolution, refutation, deduction, Frame, Semantic network script, Knowledge bases and inference. Monotonic and nonmonotonic reasoning. Planning, partial order planning.

Learning: Overview of different forms of learning, Supervised base learning: Decision Trees, Naive Bayes, Unsupervised based learning. Neural Networks, Fuzzy logic.

Introduction to Natural Language Processing, Different issue involved in NLP, Expert System, Computer Vision.

TEXT BOOKS:

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Nils J. Nilsson, Artificial Intelligence: A New Sythesis, Morgan-Kaufmann.

REFERENCE BOOKS:

1. Artificial Intelligence for Humans by Jeff Heaton.
2. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 2013.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU3	ARTIFICIAL INTELLIGENCE	CO1	Define different approaches of AI, NLP and related issue.
		CO2	Explain and compare different searching algorithm.
		CO3	Solve different gaming problem using minimax, alpha-beta pruning or searching algorithm.
		CO4	Formulate and solve problems with uncertain information using Bayes Navies or Decision Tree approaches.
		CO5	Represent English sentence domains into logic based techniques.

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU4 COMPILER DESIGN

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To learn the process of translating a modern high-level language to executable code and to introduce phases of compiler and its use.
2. To extend the knowledge of parser by parsing LL parser and LR parser.
3. To apply the code generation algorithms to get the machine code for the optimized code.
4. Understand design/implementation issues involved with storage allocation and binding, control flow, parameter passing, symbol table.
5. To understand the machine dependent code and to apply the optimization techniques to have a better code for code generation.

Prerequisites: Basic knowledge of programming in C, basics of different types of Automata and their theory

Syllabus:

Compiler, Translator, Interpreter definition, Phase of compiler, Bootstrapping, Review of Finite automata lexical analyzer, Input, Recognition of tokens, Idea about LEX: A lexical analyzer generator, Error handling.

Review of CFG Ambiguity of grammars, Introduction to parsing. Top down parsing, LL grammars & passers error handling of LL parser, Recursive descent parsing predictive parsers, Bottom up parsing, Shift reduce parsing, LR parsers, Construction of SLR, Conical LR & LALR parsing tables, parsing with ambiguous grammar. Operator precedence parsing, Introduction of automatic parser generator: YACC error handling in LR parsers.

Syntax directed definitions; Construction of syntax trees, S-Attributed Definition, L-attributed definitions, Top down translation. Intermediate code forms using postfix notation, DAG, Three address code, TAC for various control structures, Representing TAC using triples and quadruples, Boolean expression and control structures.

Storage organization, Storage allocation, Strategies, Activation records, Accessing local and non-local names in a block structured language, Parameters passing, Symbol table organization, Data structures used in symbol tables.

Definition of basic block control flow graphs, DAG representation of basic block, Advantages of DAG, Sources of optimization, Loop optimization, Idea about global data flow analysis, Loop invariant computation, Peephole optimization, Issues in the design of code generator, A simple code generator, Code generation from DAG.

9.7.19

7

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Aho, Ullman and Sethi: Compilers, Addison Wesley.
2. Holub, Compiler Design in C, PHI

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU4	COMPILER DESIGN	CO1	To Analyze the principles, algorithms, and data structures involved in the design and construction of compilers and parsers by using theory of computation.
		CO2	To understand the various phases of converting high level language to low level language such as lexical, syntax, and semantic analysis, code generation and optimization phases of compilation.
		CO3	To be able to create lexical, semantic rules and grammar for a programming language.
		CO4	To be able to apply optimization while doing simple programming.
		CO5	To be able to understand the requirement of heap and stack memory allocation system in programming.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU5.1 DISTRIBUTED SYSTEM

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. Identify potential applications of distributed system in practice.
2. Describe different distributed system model.
3. Apply appropriate distributed system model on specific problem.

Syllabus:

Distributed Systems: Features of distributed systems, nodes of a distributed system, Distributed computation paradigms, Model of distributed systems, Types of Operating systems: Operating System, Network Operating Systems, Distributed Operating Systems and Cooperative Autonomous Systems, design issues in distributed operating systems. Systems Concepts and Architectures: Goals, Transparency, Services, Architecture Models, Distributed Computing Environment (DCE). Theoretical issues in distributed systems: Notions of time and state, states and events in a distributed system, time, clocks and event precedence, recording the state of distributed systems.

Concurrent Processes and Programming: Processes and Threads, Graph Models for Process Representation, Client/Server Model, Time Services, Language Mechanisms for Synchronization, Object Model Resource Servers, Characteristics of Concurrent Programming Languages (Language not included). Inter-process Communication and Coordination: Message Passing, Request/Reply and Transaction Communication, Name and Directory services, RPC and RMI case studies.

Distributed Process Scheduling: A System Performance Model, Static Process Scheduling with Communication, Dynamic Load Sharing and Balancing, Distributed Process Implementation.

Distributed File Systems: Transparencies and Characteristics of DFS, DFS Design and implementation, Transaction Service and Concurrency Control, Data and File Replication.
Case studies: Sun network file systems, General Parallel file System and Window's file systems. Andrew and Coda File Systems.

Distributed Shared Memory: Non-Uniform Memory Access Architectures, Memory Consistency Models, Multiprocessor Cache Systems, Distributed Shared Memory, Implementation of DSM systems.

Models of Distributed Computation: Preliminaries, Causality, Distributed Snapshots, Modeling Distributed Computation, Failures in a Distributed System, Distributed Mutual Exclusion, Election, Distributed Deadlock handling, Distributed termination detection.

9.2.19

9

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Distributed Agreement: Concept of Faults, failure and recovery, Byzantine Faults, Adversaries, Byzantine Agreement, Impossibility of Consensus and Randomized Distributed Agreement. Replicated Data Management: concepts and issues, Database Techniques, Atomic Multicast, and Update Propagation.
CORBA case study: Introduction, Architecture, CORBA RMI, CORBA Services.

TEXT/REFERENCE BOOKS:

1. Distributed operating systems and algorithm analysis by Randy Chow and T. Johnson, Pearson
2. Operating Systems A concept based approach by DM Dhamdhere, TMH.
3. Distributed Systems- concepts and Design, Coulouris G., Dollimore J, and Kindberg T., Pearson

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU5.1	DISTRIBUTED SYSTEM	CO1	Become aware of a variety of distributed systems
		CO2	Proper knowledge of programming, synchronization models.
		CO3	Knowledge of distributed process scheduling.
		CO4	Ability to understand the different types of memory models.
		CO5	Be capable to knowing various type of services in real time environment.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU5.2 PATTERN RECOGNITION

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To develop a basic understanding about key application areas of Pattern Recognition, Classification and Clustering algorithms.
2. Develop skills to apply feature selection and feature extraction method on real life application.

Syllabus:

Pattern Classifier: Overview of Pattern recognition; Discriminant functions; Supervised learning; Parametric estimation; Maximum Likelihood Estimation; Bayesian parameter Estimation; Problems with Bayes approach; Pattern classification by distance functions; Minimum distance pattern classifier.

Clustering: Basics of Clustering; Clustering vs. Classification; Supervised vs. unsupervised; similarity / dissimilarity measures; clustering criteria; Different distance functions and similarity measures; Minimum within cluster distance criterion; K-means algorithm; Hierarchical clustering, K Mediods, DBSCAN.

Feature Extraction and Structural Pattern Recognition: Principal component analysis, Independent component analysis, Linear discriminant analysis, Feature selection through functional approximation.

Hidden Markov Models and Support Vector Machine: State Machines; Hidden Markov Models: Training, Classification; Support Vector Machine; Feature Selection.

Recent Advances: Structural Pattern Recognition; Fuzzy Pattern Classifiers; Pattern Classification using Genetic Algorithms.

TEXT BOOKS:

1. Pattern Recognition - An Algorithmic Approach (M.Narasimha Murty, V.Susheela Devi)

REFERENCE BOOKS:

1. Pattern Classification by Richard O. Duda, David G. Stork, Peter E.Hart
2. Christopher M. Bishop Pattern Recognition and Machine Learning

9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU5.2	PATTERN RECOGNITION	CO 1	Understand the Broad Areas of Pattern Recognition.
		CO 2	Develop understanding about Pattern Classifier and Clustering
		CO 3	Understand feature selection and feature extraction method
		CO 4	Understand hidden Markov model and various classification algorithm

9.2.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU5.3 DATA COMPRESSION TECHNIQUES

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To discuss the theoretical underpinnings of data compression and cover many fundamental algorithms.

Prerequisites:

The prerequisites for this subject include knowledge of elementary mathematics and basic algorithmic. You should review the main topics in mathematics, such as sets, basic probability theory, basic computation on matrices and simple trigonometric functions (e.g. sin(x) and cos(x)), and topics in algorithms, such as data structures, storage and efficiency.

Syllabus:

Compression Techniques: Lossless, lossy, measure of performance, modeling & coding. **Lossless compression:** Derivation of average information, data models, uniquely decodable codes with tests, prefix codes, Kraft-Mc Millan inequality. **Huffman coding:** Algorithms, minimum variance Huffman codes, optimality, length extended codes, adaptive coding, Rice codes, using Huffman codes for lossless image compression.

Arithmetic coding with application to lossless compression.
Dictionary Techniques: LZ77, LZ78, LZW.
Predictive coding: Burrows-Wheeler Transform and move-to-front coding, JPEG-LS.
Facsimile Encoding: Run length, T.4 and T.6

Lossy coding- Mathematical preliminaries: Distortion criteria, conditional entropy, average mutual information, differential entropy, rate distortion theory, probability and linear system models.
Scalar quantization: The quantization problem, uniform quantizer, Forward adaptive quantization, non uniform quantization, formal adopting quantization, companded quantization
Vector quantization: Introduction, advantages, The Linde-Ruzo-Grey algorithm, lattice vector quantization.

Differential encoding: Introduction, Basic algorithm, Adaptive DPCM, Delta modulation, speech and image coding using delta modulation.
Sampling in frequency and time domain, z-transform, DCT, DST, DWHT, quantization and coding of transform coefficient.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Sub band coding: Introduction, Filters, Basic algorithm, Design of Filter banks, G.722, MPEG. Wavelet based compression: Introduction, wavelets multiresolution analysis and the scaling function implementation using filters.

TEXT BOOKS:

1. Sayood, K, Data Compression, Morgan Kauffman, 2006.

REFERENCE BOOKS:

1. Saloman, Handbook of Data Compression
2. Drew & Li, Fundamentals of Multimedia, PHI, 2006
3. Halsall, Multimedia Communications, Pearson Edu Asia, 2004
4. Parekh Ranjan, Principles of Multimedia, TMH, 2006

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU5.3	DATA COMPRESSION TECHNIQUES	CO1	Explain the evolution and fundamental concepts of Data Compression and Coding techniques.
		CO2	Apply various coding techniques for compression of any raw data.
		CO3	Differentiate between Lossy and Lossless compression.
		CO4	Understand the scalar quantization and vector quantization
		CO5	Determine Differential, Sub band Coding and Wavelet based Compression.

9.7.19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU6.1 INFORMATION THEORY AND CODING

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To equip students with a basic understanding of the fundamental concept of entropy and information as they are used in communications.
2. To enhance knowledge of probabilities, entropy, measures of information.
3. To guide the student through the implications and consequences of fundamental theories and laws of information theory and coding theory with reference to the application in modern communication and computer systems

Prerequisites: Understanding of basic probability.

Syllabus:

Introduction to information theory: Uncertainty, Information and Entropy, Information measures for continuous random variables, source coding theorem. Discrete Memory less channels, Mutual information, Conditional entropy.

Source coding schemes for data compaction: Prefix code, Huffman code, Shanon-Fane code & Hempel-Ziv coding channel capacity. Channel coding theorem. Shannon limit.

Linear Block Code: Introduction to error connecting codes, coding & decoding of linear block code, minimum distance consideration, conversion of non-systematic form of matrices into systematic form.

Cyclic Code: Code Algebra, Basic properties of Galois fields (GF) polynomial operations over Galois fields, generating cyclic code by generating polynomial, parity check polynomial. Encoder.& decoder for cyclic codes.

Convolutional Code: Convolutional encoders of different rates. Code Tree, Trllis and state diagram. Maximum likelihood decoding of convolutional code: The viterbi Algorithm fee distance of a convolutional code.

CASE STUDY: Read about the several works done by Claude Shannon in the field of Information Theory.

TEXT BOOKS:

1. Simon Haykin. 2009. Communication Systems (5th ed.). Wiley Publishing.
2. R Bose, "Information Theory, Coding and Cryptography", TMH 2007.

9.7.19

15

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOKS:

1. Fred Halsall, "Multidedia Communications: Applications, Networks, Protocols and Standards", Perason Education Asia, 2002.
2. K Sayood, "Introduction to Data Compression" 3/e, Elsevier 2006.
3. Amitabha Bhattacharya, "Digital Communication", TMH 2006.
4. S Gravano, "Introduction to Error Control Codes", Oxford University Press 2007.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU6.1	INFORMATION THEORY AND CODING	CO1	Calculate channel performance using Information theory.
		CO2	Implement various error control code properties.
		CO3	Apply linear block codes for error detection and correction.
		CO4	Design BCH & RS codes for Channel performance improvement against burst errors.
		CO5	Apply convolution codes for performance analysis & cyclic codes for error detection and correction.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU6.2 SOFTWARE DEFINED NETWORK

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Differentiate between traditional networks and software defined networks
2. Understand advanced and emerging networking technologies
3. Obtain skills to do advanced networking research and programming
4. Learn how to use software programs to perform varying and complex networking tasks.

Prerequisites: Network programming

Syllabus:

Introducing SDN: SDN Origins and Evolution – Introduction – Why SDN? - Centralized and Distributed Control and Data Planes - The Genesis of SDN

SDN Abstractions: How SDN Works - The Openflow Protocol - SDN Controllers: Introduction – General Concepts - VMware - Nicira - VMware/Nicira - OpenFlow-Related - Mininet - NOX/POX - Trema - Ryu - Big Switch Networks/Floodlight - Layer 3 Centric - Plexxi - Cisco OnePK

Programming SDN'S: Network Programmability - Network Function Virtualization - NetApp Development, Network Slicing

SDN Applications And Use Cases: SDN in the Data Center - SDN in Other Environments - SDN Applications - SDN Use Cases - The Open Network Operating System

SDN'S Future And Perspectives: SDN Open Source - SDN Futures - Final Thoughts and Conclusions

TEXT BOOKS:

1. Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black, Morgan Kaufmann Publications, 201

REFERENCE BOOKS:

1. SDN - Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013
2. Software Defined Networking with OpenFlow By SiamakAzodolmolky, Packt Publishing, 2013.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU6.2	SOFTWARE DEFINED NETWORK	CO1	Obtain skills to do advanced networking research and programming
		CO2	Learn how to use software programs to perform varying and complex networking tasks
		CO3	Apply SDN concept on Real Time Application
		CO4	Implement the SDN concept using Network programming

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU6.3 AGILE SOFTWARE DEVELOPMENT

Credit: 2 (2L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Describe the background and driving forces for taking an Agile approach to software development
2. Describe the business value of adopting Agile approaches and development practices
3. Use Test Driven Development with unit tests
4. Apply design principles, refactoring, version control and continuous integration to achieve Agility
5. Perform testing activities within an Agile project

Prerequisites: Software Engineering

Syllabus:

Agile Programming: Introduction, Flavors of Agile Development, Agile Manifesto, Refactoring Techniques, Limitations of The Agile Process.

Extreme Programming (XP): Introduction, XP Equation, XP Values, Assuming Sufficiency- Sufficient time and resources, Constant change of cost, Developer effectiveness, Freedom to experiment. Extreme Programming Practices- Introduction, Coding Practices, Developer Practices, Business Practices.

XP Events: Introduction, Iteration Planning- Stories and tasks, Estimates and schedules, First iteration, Iteration, Releasing. Extreme Programming Practices- Introduction, Story Cards, Task Cards, Bullpens.

Roles in Extreme Programming: Introduction, Customer's Roles, Developer's Roles, Supplementary Roles. Coding XP Style- Introduction, Balance Functionality with Simplicity, Implement Only the Needed Features, Eliminate Repetition. Adopting XP- Introduction, Before Commencing XP, Eliminating Fear and Working Together, Starting Feedback, Including Managers and Customers.

Agile Modeling with XP- Introduction, Agile Modeling: Principles, Comparing XP and Agile Modeling, Scrum Methodology- The roles of Scrum, Advantages of Scrum. Dynamic Systems Development Methodology- Introduction, Overview of DSDM, the Principles of DSDM, Phases of DSDM, Core Techniques Used in DSDM.

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall
2. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson

REFERENCE BOOKS:

1. R. S. Pressman, Software Engineering A Practitioner's Approach, McGraw Hill Publications, 2006.
2. Lisa Crispin, Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison Wesley
3. Alistair Cockburn, Agile Software Development: The Cooperative Game, Addison Wesley
4. Mike Cohn, User Stories Applied: For Agile Software, Addison Wesley

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU6.3	AGILE SOFTWARE DEVELOPMENT	CO1	Describe the background and driving forces for taking an Agile approach to software development
		CO2	Apply Extreme Programming Concept on real time project.
		CO3	Demonstrate the agile methodology and XP Concepts.
		CO4	Extend knowledge of scrum and XP concepts.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU11 OPERATING SYSTEM SIMULATION LAB

Credit: 2 (0L+0T+3P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. Practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in Linux environment.

List of Experiments

1. Implement the Producer-Consumer problem using semaphores(Using UNIX system calls)
2. Given the list of processes, their CPU burst times and arrival times. Display/print the Gantt chart for Priority and Round robin.
3. Threading & Synchronization Applications list of processes, their CPU burst times and arrival times. Display/print the Gantt chart for FCFS and SJF.
4. Implement Banker's algorithm for Deadlock Avoidance 8. Implement an Algorithm for Deadlock Detection
5. Implement some Memory management schemes like FIRST FIT, BEST FIT & WORST FIT
6. Implement the all page replacement algorithms a) FIFO b) LRU c) LFU
7. Implement file allocation techniques Linked list.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU11	OPERATING SYSTEM SIMULATION LAB	CO1	Use modern operating system calls and synchronization libraries in software/ hardware interfaces.
		CO2	Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
		CO3	Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.
		CO4	Implement memory management schemes and page replacement schemes.
		CO5	Simulate file allocation and organization techniques.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU12 COMPUTER GRAPHICS LAB

Credit: 2 (0L+0T+3P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. Understand the need of developing graphics application.
2. Learn algorithmic development of graphics primitives like: line, circle, polygon etc.
3. Learn the representation and transformation of graphical images and pictures.
4. Create a visual scene using computer graphics algorithms concepts.

List of Experiments

1. Draw different types of squares and shape given in fig. 1by implementing line drawing algorithms using DDA and Bresenham's method.

Figure 1

2. Draw semi-circle, circle and arcs (30°, 60°, 90°) using Midpoint algorithm.
3. Perform 2D geometric transformations- Translation, Rotation, Scaling, Reflection, shearing on the shapes generated in Experiment 1 and 2.
4. Perform composite 2D transformations on the shapes generated in Experiment 1 and 2.
5. Draw a polygon using any line drawing algorithm and perform line clipping using Cyrus-Beck, Cohen-Sutherland and Liang-Barsky algorithms against a selected window.
6. Perform 3D geometric transformation on the shape given in fig. 1.
7. Create a visual scene by using all the above implemented algorithms.

Note- Implement each algorithm in a function and later on use that function.

TEXT BOOKS:

1. Interactive Computer Graphics A Top-Down Approach with OpenGL, Edward Angel, Pearson, 5th Edition, 2009.

9.2.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU12	COMPUTER GRAPHICS LAB	CO1	Demonstrate the concepts of line and circle drawing.
		CO2	Apply transformations on various objects like, line, circle and polygon.
		CO3	Make use of clipping algorithm to render an object in a selected window.
		CO4	Develop a visual scene using computer graphics concepts.

9.7.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU13 ARTIFICIAL INTELLIGENCE LAB

Credit: 1(0L+0T+2P) **Max. Marks 75(IA:50,ETE:25)**

Objectives:

1. To know about prolog language for knowledge representation.
2. To design searching algorithm and game playing algorithm for specific problem.
3. To apply learning algorithm for real world problem.

List of Experiments

1. Find out path in a graph from given node to goal node using breadth first search.
2. Find out path in a graph from given node to goal node using depth first search.
3. Apply A^* algorithm on weighted graph (distance problem) and find out optimal path from given node to goal node using heuristic function.
4. Write down Prologs response(s) to the query p(X) for the following program.

```
p(X) :- q(X,Y), Z is X + Y, r(Z).
p(X) :- r(X), X > 3.
q(X,X) :- r(X).
q(1,2).
r(2).
r(3).
r(4).
```

5. In Prolog, assume that you have a number of facts of the form:

```
mother(M, C).        /* M is the mother of C */
father(F, C).        /* F is the father of C */
male(X).             /* X is male */
female(X).           /* X is female */
```

Write Prolog rules for

```
grandmother(G, X).   /* G is a grandmother of X */
sister(S, X).        /* S is a sister of X */
```

6. Write a program to solve 8-Puzzle problem using Prolog.
7. Apply Naïve Bayes classifier on labelled data(Jerman base credit card fraud detection) and find out accuracy.Apply Decision tree algorithm on labelled data(Jerman base credit card fraud detection) and find out accuracy.
8. Compare Naïve Bayes and Decision Tree algorithm on same labelled data using
9. ROC curve.
10. Design Alpha beta pruning algorithm for tic tac toi problem.

9.7.19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOK:

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. An Introduction to Programming in Prolog by Patrick Saint-Dizier

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU13	ARTIFICIAL INTELLIGENCE LAB	CO1	Design an algorithm for searching problem.
		CO2	Implement decision tree and Naïve Bayes algorithms to solve uncertain information data problems.
		CO3	Write English sentences into first order logic using Prolog.
		CO4	Develop correct inference rule for English sentence using Prolog.

9.2.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

6ITU14 COMPILER DESIGN LAB

Credit: 1(0L+0T+2P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. To implement Lexical Analyzer using Lex tool & Syntax Analyzer or parser using YACC Tool
2. To implement Symbol Table.
3. To implement front end of the compiler by means of generating Intermediate codes.

List of Experiments

1. To identify whether given string is keyword or not?
2. Count total no. of keywords in a file. [Taking file from user]
3. Count total no of operators in a file. [Taking file from user]
4. Count total occurrence of each character in a given file. [Taking file from user]
5. Write a C program to insert, delete and display the entries in Symbol Table.
6. Write a LEX program to identify:
    1. Valid mobile number
    2. Valid url
    3. Valid identifier
    4. Valid date (dd/mm/yyyy)
    5. Valid time (hh:mm:ss)
7. Write a lex program to count blank spaces,words,lines in a given file.
8. Write a lex program to count the no. of vowels and consonants in a C file.
9. Write a YACC program to recognize strings aaab,abbb using a^nb^n, where b>=0.
10. Write a YACC program to evaluate an arithmetic expression involving operators +,-,* and /.
11. Write a YACC program to check validity of a strings abcd, aabbcd using grammar a^nb^nc^md^m, where n , m>0
12. Write a C program to find first of any grammar.

9.7.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
6ITU14	COMPILER DESIGN LAB	CO1	Identify the tokens from a file and classify them.
		CO2	Make use of appropriate data structure to implement various operations of symbol table.
		CO3	Apply the concept of regular expressions and construct solution for lex programs
		CO4	Show implementation of bottom up parsers for various problems by YACC Tool.
		CO5	To construct Program to calculate first and follow by applying theoretical concepts of its calculation.

9.2.17

Anil K. Mathur

Approved
Dean, FA & UD

SYLLABUS

Semester	**VII**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2021-22**

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU1 INTERNET OF THINGS

Credit: 4(3L+1T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. Discuss the terminology, technology and its applications.
2. Provide an understanding of the technologies and the standards relating to the Internet of Things
3. Introduce the concept of M2M (machine to machine) with necessary protocols
4. Identify domain specific IOTs

Prerequisite: Fundamentals of computer network, wireless sensor network, communication & internet technology, web technology, information security.

Syllabus:

Introduction to IoT: Definition and characteristics of IoT, Design of IOT: Physical design of IOT, Logical Design of IOT- Functional Blocks, communication models, communication APIs, IOT enabling Technologies- Wireless Sensor Networks, Cloud computing, big data analytics, embedded systems. IOT Levels and deployment templates.

IoT Hardware and Software: Sensor and actuator, Humidity sensors, Ultrasonic sensor, Temperature Sensor, Arduino, Raspberry Pi, LiteOS, RIoTOS, Contiki OS, Tiny OS.

Architecture and Reference Model: Introduction, Reference Model and architecture, Representational State Transfer (REST) architectural style, Uniform Resource Identifiers (URIs). Challenges in IoT- Design challenges, Development challenges, Security challenges, Other challenges.

IOT and M2M: M2M, Difference and similarities between IOT and M2M, Software defined networks, network function virtualization, difference between SDN and NFV for IoT.

Case study of IoT Applications: Domain specific IOTs- Home automation, Cities, environment, Energy, Retail, Logistics, Agriculture, Industry, Health and Lifestyles.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXTBOOK:

1. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT , 2014.

REFERENCE BOOKS:

1. Jan Holler , Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence",1st Edition, Academic Press, 2014.
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759
3. Michael Margolis, "Arduino Cookbook", 2nd edition, O'Reilly, 2012.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU1	INTERNET OF THINGS	CO 1	Define IOT and its logical and physical design
		CO 2	Review enabling technologies of IOT, hardware, software components, and architecture of IOT
		CO 3	Discover challenges of IOT and its dissimilarity with M2M
		CO 4	Use SDN and NFV to virtualise IOT
		CO 5	Examine case studies related to domain specific IOTs

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU2 CLOUD COMPUTING

Credit: 4 (3L+1T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. Understand the current trend and basics of cloud computing
2. Learn cloud enabling technologies and its applications
3. Explore different cloud mechanisms and get exposure to advanced clouds
4. Analyze the cost metrics, handle the security threats and construct different cloud delivery design models

Prerequisites: Computer Architecture and Organization.

Syllabus:

Understanding Cloud Computing: Concepts of cloud computing, Cloud origins and influences, basic concepts and terminology, Cloud computing leverages the Internet, Elasticity and scalability,goals and benefits, risks, challenges, and Limjtations. Defining Cloud Computing, Roles and boundaries, cloud characteristics.

Cloud Service delivery models: Infrastructure as Services(IaaS),Software as a Services(SaaS), Plateform as a Services(PaaS), Identitiy as a Services(IDaaS),Compliance as a Services(CaaS).
Cloud deployment scenarios: Cloud deployment models, Public clouds, Hybrid clouds, Community, Virtual private clouds, Vertical and special purpose, Migration paths for cloud, Selection criteria for cloud deployment.

Cloud Enabling Technology: Introduction to Virtualization, Virtualization, Characteristics of virtualization; Benefits of virtualization, Virtualization in cloud computing, Hypervisors, multitenant technology, Types of tenancy, Virtual machine monitors, Virtual machines. Use of Virtualization Technology, Load Balancing, benfits of virtualization, Hypervisor,. Full virtualization and paravirtualization, Hardware support for virtualization.
Case study: Xen, KVM, VMware, VMM based on para-virtualization.

Cloud Administration and Security Management in cloud computing: Cloud security reference model, How security gets integrated , Cloud security, Understanding security risks, Principal security dangers to cloud computing, Cloud Security, Data Security, Disaster Recovery and Planning, Cloud Disaster Management, Identity and Access Management, Availability management in SaaS, IaaS, PaaS, future of Security in Cloud computing.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Cost Metrics and Pricing Models: Business Cost Metrics, Cloud Usage Cost Metrics, Cost Management Considerations.

Case study: IBM Smart Cloud, Amazon Web Services, Google Cloud platform, Windows Azure platform, A comparison of Cloud Computing Platforms, Common building Blocks.

TEXT BOOKS:

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood ,Cloud Computing: Concepts, Technology & Architecture, PHI Publications,2013
2. Gautam Shroff, Enterprise Cloud Computing: Technology, Architecture,applications, Cambridge University Press, 2010.
3. Ronald Krutz Russell Dean Vines, A Comprehensive guide to secure cloud computing, Wiley, 2010.

REFERENCE BOOKS:

1. Borko Furht, Armando Escalante (Editors), Handbook of Cloud Computing, Springer, 2010.
2. John W. Rittinghouse, James F.Ransome, Cloud Computing: Implementation, Management and Security, CRC Press, 2012.
3. Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski, Cloud Computing: Principles and paradigms, 2011
4. Barrie Sosinsky, Cloud Computing Bible, Wiley, 2011.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU2	CLOUD COMPUTING	CO1	Outline the concept of cloud computing, characteristics, risk and challenges involved in it.
		CO2	Explain cloud service and deployment models and apply map reduce programming model.
		CO3	Illustrate concept of virtualization along with its types
		CO4	Assess the cloud storage system, cloud security, the risk involved, its impact and develop cloud application.
		CO5	Identify the different cost metric, Analyze the concept of Google cloud platform and describe the components of Amazon web services.

18.10.19

4

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU3 BIG DATA ANALYTICS

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. To provide concept and challenge of big data (3 V's: volume, velocity, and variety).
2. Provide HDFS Concepts and Interfacing with HDFS.
3. Imparting the architectural concepts of Hadoop and introducing map reduce paradigm.
4. To teach students in applying skills and tools to manage and analyze the big data.
5. Exposure in PIG & HIVE in Hadoop echo system.

Syllabus:

Introduction to Big Data: Big data features and challenges, Problems with Traditional Large-Scale System , Sources of Big Data, 3 V's of Big Data, Types of Data.

Working with Big Data: Google File System.Hadoop Distributed File System (HDFS) - Building blocks of Hadoop (Namenode. Data node.Secondary Namenode.Job Tracker. Task Tracker), Introducing and Configuring Hadoop cluster (Local. Pseudo-distributed mode, Fully Distributed mode). Configuring XML files.

Writing MapReduce Programs: A Weather Dataset. Understanding Hadoop API for MapReduce Framework (Old and New). Basic programs of HadoopMapReduce: Driver code. Mapper code, Reducer code. Record Reader, Combiner. Partitioner.

Hadoop I/O: The Writable Interface. Writable Comparable and comparators. Writable Classes: Writable wrappers for Java primitives. Text.Bytes Writable.Null Writable, Object Writable and Generic Writable.Writable collections.Implementing a Custom Writable: Implementing a Raw Comparator for speed, Custom comparators.

Pig: Hadoop Programming Made Easier Admiring the Pig Architecture, Going with the Pig Latin Application Flow. Working through the ABCs of Pig Latin.Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin.

Applying Structure to Hadoop Data with Hive: Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive.Examining the Hive Clients.Working with Hive Data Types.Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analyzing Data.

18.10.19

5

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Srinath Perera Thilina Gunarathne,"Hadoop Mapreduce Cookbook" ,Paperback, 2013.
2. Tom White,"Hadoop: The Definitive Guide",3rd Edition,O'Reilly,2012.
3. Chuck Lam,"Hadoop in Action",Paperback, 2011.
4. Matteo Golfarelli and Stefano Rizzi,"Data Warehouse Design: Modern Principles and Methodologies ",1st Edition,McGraw Hill,2009.

REFERENCE BOOKS:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press ,2013.
3. ArvindSathi, "BigDataAnalytics: Disruptive Technologies for Changing the Game", MC Press, 2012.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU3	BIG DATA ANALYTICS	CO1	Understand the concept and challenge of big data and why existing technology is inadequate to analyze the big data.
		CO2	Access and Process Data on Distributed File System
		CO3	Manage Job Execution in Hadoop Environment
		CO4	Collect, manage, store, query, and analyze various form of big data
		CO5	Create Hive and Pig scripts for Business Application

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU4 MACHINE LEARNING

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. Identify potential applications of machine learning in practice.
2. Describe different machine learning model.
3. Compare and evaluate machine learning models based on mathematical analysis.
4. Apply appropriate machine learning model on specific problem.

Syllabus:

Introduction to Machine Learning, types of learning, application, supervised learning: Linear Regression Model, Logistic Regression, Naive Bayes classifier, Decision Tree, K nearest neighbor, Support Vector Machine, Random forest algorithm

Unsupervised learning algorithm: Grouping Unlabelled items using k-means clustering, Hierarchical Clustering, Probabilistic Clustering, Association rule mining, Apriori Algorithm, f-p growth algorithm, Gaussian mixture model.

Introduction to Statistical Learning Theory, Feature extraction: Principal component analysis, Singular value decomposition. Feature selection – feature ranking and subset selection, filter, wrapper and embedded methods, Evaluating Machine Learning algorithms and Model Selection.

Semi supervised learning, Reinforcement learning: Markov decision process (MDP), Bellman equations, policy evaluation using Monte Carlo, Policy iteration and Value iteration, Q-Learning, State-Action-Reward-State-Action (SARSA), Model-based Reinforcement Learning.

Recommended system, Collaborative filtering, Content-based filtering, Artificial neural network, Perceptron, Multilayer network, Back propagation, Introduction to Deep learning.

TEXTBOOKS:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 2013.
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin
3. Jason Bell, —Machine learning – Hands on for Developers and Technical Professionals, First Edition, Wiley, 2014

18.10.19

7

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU4	MACHINE LEARNING	CO1	Become aware of variety of machine learning algorithms and explain how these algorithms are different from traditional algorithms.
		CO2	Develop machine learning solutions to classification, regression, and clustering problems; and be able to interpret the results of the algorithms.
		CO3	To compare different methods for performance evaluation of machine learning and select appropriate model according to problem.
		CO4	Ability to apply semi supervised machine learning techniques and associated computing techniques.
		CO5	Be capable of performing experiments in Machine Learning using real-world data.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU5.1 INFORMATION SYSTEM SECURITY

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA: 50, ETE: 100)**

Objectives:

1. To provide an understanding of principal concepts, major issues, technologies and basic approaches in information security.
2. Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.
3. Gain familiarity with prevalent network and distributed system attacks, defenses against them and forensics to investigate the aftermath
4. Develop a basic understanding of cryptography, how it has evolved and some key encryption techniques used today.
5. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

Syllabus:

Introduction to security attacks: services and mechanism, classical encryption techniques-substitution ciphers and transposition ciphers, cryptanalysis, stream and block ciphers.

Modern block ciphers: Block Cipher structure, Data Encryption standard (DES) with example, strength of DES, Design principles of block cipher, AES with structure, its transformation functions, key expansion, example and implementation.
Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode.

Public Key Cryptosystems with Applications: Requirements and Cryptanalysis, RSA cryptosystem, Rabin cryptosystem, Elgamal cryptosystem, Elliptic curve cryptosystem.
Cryptographic Hash Functions, their applications: Simple hash functions, its requirements and security, Hash functions based on Cipher Block Chaining, Secure Hash Algorithm (SHA).
Message Authentication Codes, its requirements and security, MACs based on Hash Functions, Macs based on Block Ciphers.Digital Signature, its properties, requirements and security, various digital signature schemes (Elgamal and Schnorr), NIST digital Signature algorithm.

18.10.19

SC- 2017IT

Key management and distribution: symmetric key distribution using symmetric and asymmetric encryptions, distribution of public keys, X.509 certificates, Public key infrastructure. Remote user authentication with symmetric and asymmetric encryption, Kerberos
Web Security threats and approaches, SSL architecture and protocol, Transport layer security, HTTPS and SSH.

TEXTBOOKS:

1. Stalling Williams: Cryptography and Network Security: Principles and Practices, 4th Edition, Pearson Education, 2006.
2. Kaufman Charlie et.al; Network Security: Private Communication in a Public World, 2ndEd., PHI/Pearson

REFERENCE BOOKS:

1. Pieprzyk Josef and et.al; Fundamentals of Computer Security, Springer-Verlag, 2008.
2. Trappe & Washington, Introduction to Cryptography, 2nd Ed. Pearson.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU5.1	INFORMATION SYSTEM SECURITY	CO1	Describe major issues, basic approaches in information security and Identify common network security attacks.
		CO2	Illustrate various Public Key Cryptosystems.
		CO3	Understand different Authentication requirements and Mechanisms.
		CO4	To Explain IP Security and summarise its Architecture.

18/10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU5.2 BIOMETRICS

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Introducing area of Biometrics such as identification of an individual based on his/her physiological characteristics, like a fingerprint, face, voice or behavior like handwriting or keystroke patterns.
2. Introducing a case study

Prerequisites: Statistics, image processing, Programming SkillsIT

Syllabus:

The Basics of Biometrics: Overview of field and applications. Development of biometric authentication. Basic terms, biometric data, biometric characteristics, biometric features, biometric templates and references. Expected properties of biometric identifiers. Basics in biometric errors estimation. Enrollment, verification and identification.

Face Recognition: Introduction to the face processing pipeline: acquisition, face detection, alignment, feature extraction, matching. Classic subspace methods. Hand-tuned feature descriptors. Deep learning architectures for face representation learning. Distance, similarity and learning-based matching. Face recognition in video. Describable visual attributes. Face pair matching, verification, and identification. Data sets for evaluation. Face image quality. Considerations for social media, mobile authentication, surveillance and other real-world applications.

Fingerprint Recognition: Fingerprint capture, sensor types, latent fingerprints. Fingerprint image preprocessing, segmentation, binary and skeletal images. Fingerprint singularities, detection of loops, deltas, whirls and cores, using singularities in fingerprints classification. Galton's details, base and complex minutiae, detection of minutiae. Fingerprint recognition, minutiae- and correlation-based methods. Fingerprints in forensics and biometrics, similarities and differences.

Iris Recognition: Eye and iris morphogenesis, genetic penetrance. Principles of iris image capture, iris sensors. Iris image preprocessing, segmentation, formatting and filtering. Daugman's method, iris code, statistical properties of the iris code. Other iris coding methods, wavelet analysis.

Multi-Biometric Fusion: Levels of fusion: sensor, feature, rank, decision. Score normalization and fusion rules. Quality-based fusion and failure prediction.

18.10.19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Biometric System Security: Secure transfer of biometric data. Secure storage, use of smart cards, principles of match-off-card and match-on-card techniques. Biometrics in the cloud. Points of attack. Privacy models.
Spoofing: Static and dynamic liveness features. What we want to detect (subversive actions) vs. what we can detect (suspicious actions). Liveness detection in biometrics. Selected liveness detection techniques, frequency analysis for paper printouts detection, pupil dynamics and blood pulse analyses for detection of sophisticated eye and finger spoofing trials.

Template Protection: Overview of principles from cryptography that help us secure fuzzy data. Template protection strategies: feature protection, key-binding, key-generating, hybrids. Overview of fuzzy vaults, fuzzy commitment, fuzzy extractors and revocable biotokens. Biocryptographic infrastructures for secure template management.

CASE STUDY: In 2010, the India's Aadhaar program was launched by the Unique Identification Authority of India (UIDAI), aimed to provide each Indian resident with a unique identification number to enable easier, more efficient and secured access to citizen services and make sure the welfare benefits go directly to the right person.The Unique Identification Authority of India (UIDAI) was the world largest biometric project in history run by Indian government to register its 1.2 billion people with biometrics data. Comprehend use of different types of biometrics in process of enrollment and authentication used in Aadhaar.

TEXT BOOKS:

1. Anil K. Jain, Arun A. Ross, and Karthik Nandakumar. 2011. Introduction to Biometrics. Springer Publishing Company, Incorporated.
2. Samir Nanavathi, Michel Thieme, and Raj Nanavathi : "Biometrics -Identity verification in a network", 1st Edition, Wiley Eastern, 2002.

REFERENCE BOOKS:

1. Digital Image Processing using MATLAB, By: Rafael C. Gonzalez, Richard Eugene Woods, 2nd Edition, Tata McGraw-Hill Education 2010.
2. Guide to Biometrics, By: Ruud M. Bolle, Sharath Pankanti, Nalini K. Ratha, Andrew W. Senior, Jonathan H. Connell, Springer 2009.
3. Pattern Classification, By: Richard O. Duda, David G.Stork, Peter E. Hart, Wiley 2007.
4. John Chirillo and Scott Blaul : "Implementing Biometric Security", 1st Edition, Wiley Eastern Publication, 2005.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU5.2	BIOMETRICS	CO1	Describe the principles of the three core biometric modalities (face, fingerprint and iris), and know how to deploy them in authentication scenarios.
		CO2	Calculate distributions of within- and between-class matching scores, and calculate various error estimates based on these distributions.
		CO3	Identify the privacy and security concerns surrounding biometric systems, and know how to address them in such a way that balances both.
		CO4	Deploy statistical methods in biometric system evaluation.
		CO5	Itemize the most up-to-date examples of real biometric applications in human authentication.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU05.3 DIGITAL FORENSIC

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To provide an understanding Computer forensics fundamentals.
2. To identify methods for data recovery.
3. To analyse various computer forensics technologies.
4. To provide memory and network forensics systems.
5. To identify legal issue involved with digital forensic.

Prerequisites: Computer Network and Operating System

Syllabus:

Introduction to Computer Forensics: computer crimes, evidence, extraction, preservation, etc. Overview of hardware and operating systems: structure of storage media/devices; windows/Macintosh/ Linux -- registry, boot process, file systems, file metadata.

Data recovery: identifying hidden data, Encryption/Decryption, Steganography, recovering deleted files.
Digital evidence controls: uncovering attacks that evade detection by Event Viewer, Task Manager, and other Windows GUI tools, data acquisition, disk imaging, recovering swap files, temporary &cache files

Computer Forensic tools: Encase, Helix, FTK, Autopsy, Sleuth kit Forensic Browser, FIRE, Found stone Forensic ToolKit, WinHex, Linux dd and other open source tools.
Network Forensic: Collecting and analyzing network-based evidence, reconstructing web browsing, email activity, and windows registry changes, intrusion detection, tracking offenders, etc.

Memory Forensic: Image acquisition, Memory Image Analysis using Volatility, Detecting code injection etc.
Network Forensic: Introduction, Mobile Network Technology, Investigations, Collecting Evidence, where to seek Digital Data for further Investigations, Interpretation of Digital Evidence on Mobile Network.

Software Reverse Engineering: defend against software targets for viruses, worms and other malware, improving third-party software library, identifying hostile codes-buffer overflow, provision of unexpected inputs, etc.
Computer crime and Legal issues: Intellectual property, privacy issues, Criminal Justice system for forensic, audit/investigative situations and digital crime scene, investigative procedure/standards for extraction, preservation, and deposition of legal evidence in a court of law.

14

18.10.19

Anil K. Mathus

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Guide to Computer Forensics and Investigations (4 th edition). By B. Nelson, A. Phillips, F. Enfinger, C. Steuart. ISBN 0-619-21706-5, Thomson, 2009.
2. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory (1st Edition). By Michael Hale Ligh, Andrew Case, Aaron Walters.
3. Computer Forensics and Cyber Crime: An Introduction (3rd Edition) by Marjie T. Britz, 2013.

REFERENCE BOOKS:

1. Computer Forensics: Hard Disk and Operating Systems, EC Council, September 17, 2009
2. Computer Forensics Investigation Procedures and response, EC-Council Press, 2010
3. Computer Forensics: Principles and Practices by Linda Volonino, Reynaldo Anzaldua, and Jana Godwin (Paperback - Aug 31, 2006)
4. File System Forensic Analysis. By Brian Carrier. Addison-Wesley Professional, March 27, 2005.
5. NIST Computer Forensic Tool Testing Program (www.cftt.nist.gov/)
6. Computer Forensics: Investigating Data and Image Files (Ec-Council Press Series: Computer Forensics) by EC-Council (Paperback - Sep 16, 2009)
7. Digital Evidence and Computer Crime, Third Edition: Forensic Science, Computers, and the Internet by Eoghan Casey, 2011

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU05.3	DIGITAL FORENSIC	CO1	Define the basics of computer forensics.
		CO2	Explain the concepts of data recovery.
		CO3	Make use of various computer forensic tools.
		CO4	Discover different types of forensic for a given scenario.
		CO5	Interpret the legality for different aspects of digital forensic.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU11 INTERNET OF THINGS LAB

Credit: 2(0L+0T+3P) **Max. Marks: 75(IA:50, ETE:25)**

Objectives:

1. To introduce the Python Scripting Language which is used in many IoT device.
2. To introduce the Raspberry PI platform, that is widely used in IoT app
3. Implement python commands on Raspberry pi
4. Use python commands on sensors to perform an action.

List of Experiments:

1. Start Raspberry Pi and try various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.
2. Run some python programs on Pi like:
   a) Read your name and print Hello message with name
   b) Read two numbers and print their sum, difference, product and division.
   c) Word and character count of a given string.
   d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.
   e) Print a name 'n' times, where name and n are read from standard input, using for and while loops.
   f) Handle Divided by Zero Exception.
   Print current time for 10 times with an interval of 10 seconds.
   g) Read a file line by line and print the word count of each line.
3. Light an LED through Python program
4. Get input from two switches and switch on corresponding LEDs
5. Flash an LED at a given on time and off time cycle, where the two times are taken from a file.
6. Flash an LED based on cron output (acts as an alarm)
7. Switch on a relay at a given time using cron, where the relay's contact terminals are connected to a load.
8. Get the status of a bulb at a remote place (on the LAN) through web.

The student should have hands on experience in using various sensors like temperature, humidity, smoke, light, etc. and should be able to use control web camera, network, and relays connected to the Pi.

18.10.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU11	INTERNET OF THINGS LAB	CO1	Use python scripting language with IOT device
		CO2	Apply python commands on sensors using raspberry pi/arduino board
		CO3	Understand the functioning of raspberry pi/arduino board
		CO4	Learn controlling a device using web

18.10.19

Anil K. Mathur
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU12 BIG DATA ANALYTICS LAB

Credit: 1(0L+0T+2P) **Max. Marks:75(IA:50,ETE:25)**

Objectives:

1. To configure different modes of Hodoop Eco System.
2. To Provide an overview of Apache Hadoop.
3. Provide HDFS Concepts and Interfacing with HDFS.
4. To Understand Map Reduce Jobs.
5. Apply analytics on Structured, Unstructured Data for real world application.

List of Experiments:

1. Implement the following Data structures in Java

   i) Linked Lists ii) Stacks iii) Queues iv) Set v) Map

2. Perform setting up and Installing Hadoop in its three operating modes: Standalone, Pseudo distributed, Fully distributed.

3. Implement the following file management tasks in Hadoop:

   - Adding files and directories
   - Retrieving files
   - Deleting files Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.

4. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

5. Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.

6. Implement Matrix Multiplication with Hadoop Map Reduce

7. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.

8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes.

9. Solve some real life big data problems.

18

18.10.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU12	BIG DATA ANALYTICS LAB	CO1	Understand different modes of Hodoop Eco System.
		CO2	Access and Process data on Distributed File System.
		CO3	Configure and Evaluate Pig and Hive scripts.
		CO4	Create and Analyze real world big data application.

18.10.19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

7ITU13 MACHINE LEARNING LAB

Credit: 1(0L+0T+2P) **Max. Marks:75(IA:50, ETE:25)**

Objectives:

1. To know about dataframe and feature extraction and selection methods.
2. To design recommended system for uncertain data.
3. To apply learning algorithm for real world.

List Of Experiment:

1. Develop programs to implement DataFrame, Statistical Learning, Feature extraction & Feature selection.
2. Exercises to solve the real-world problems using the supervised machine learning model: Linear Regression Model, Naive Bayes classifier Decision Tree, K nearest neighbor, Logistic Regression, Support Vector Machine, Random forest algorithm.
3. Exercises to solve the real-world problems using the Unsupervised clustering methods: k-means clustering, Hierarchical Clustering, Probabilistic clustering.
4. Develop application involving Market basket analysis using: Apriori Algorithm and f-p growth algorithm.
5. Develop programs to implement Recommendation Systems like movie, shopping etc…

REFERENCES:

1. Building machine learning systems with python by Willi Richert
2. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 2013.
3. Jason Bell, —Machine learning – Hands on for Developers and Technical Professionals, First Edition, Wiley, 2014

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
7ITU13	MACHINE LEARNING LAB	CO 1	Use the feature extraction & feature selection method from data.
		CO 2	Develop the skills in applying appropriate supervised learning algorithm in uncertain data.
		CO 3	Design unsupervised learning algorithms for solving real world problems.
		CO 4	Implement recommended system for movie data.

18.6.19

20

Anil K. Mathur

Approved
Dean, FA & UD

SYLLABUS

Semester	**VIII**
Branch	**IT**
Admission Year	**2018-19**
Academic Year	**2021-22**

Anil K. Mathus
Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU1.1 NATURE INSPIRED ALGORITHMS

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To understand the various concepts of nature inspired algorithms.
2. Detailed understanding of the Evolutionary Algorithms.
3. Knowledge enhancement on the swarm intelligence based algorithms.
4. Practical knowledge of the Discrete Nature Inspired Algorithms and Local Search Techniques.
5. Implementation and Applications of the NIA for engineering optimization problem.

Syllabus:

Introduction to Nature Inspired Algorithms: Overview of Computational Intelligence, Biologically inspired computing: nature as source of inspiration for the design of algorithms; Overview of the Nature Inspired Algorithms, Evolutionary Computation, Swarm Intelligence based algorithms.

Evolutionary Computation Theory and Paradigms: History, overview, Genetic Algorithm, Differential Algorithm, Evolutionary Programming, Evolutionary Strategies. An overview of Evolutionary Algorithms, etc.

Swarm Intelligence Based Algorithms: Basic Particle Swarm Optimization, Global Best PSO, Local Best PSO, gbest versus, lbest PSO, Basic PSO Parameters, Artificial Bee Colony Algorithms, ANT Colony Optimization, Spider Monkey Optimization Algorithm, Gravitational Search Algorithm, Bio-Geography Based Optimization etc.

Discrete Nature Inspired Algorithms and Local Search Techniques: Discrete versions of the PSO, ABC, BBO, SMO Local Search Algorithms, Performance Evaluation of memetic algorithms, Parameterization and Balancing Local and Global Search, Memetic Algorithms in Discrete Optimization, Memetic Algorithms in Constrained Optimization, Multiobjective: Memetic Algorithms.

Step by step procedure of the Nature Inspired Algorithms: Applications and implementation of Nature Inspired Algorithms to solve engineering optimization problems for example Knapsack Problem, Quadratic Assignment Problem, Robot Path Planning Problem, Job Shop Scheduling Problem etc.

18.10.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Engelbrecht, Andries P. Computational intelligence: an introduction. John Wiley & Sons, 2007.
2. Smolinski, Tomasz G., Mariofanna G. Milanova, and Aboul-Ella Hassanien, eds. *Applications of computational intelligence in biology: current trends and open problems*. Vol. 122. Springer, 2008.
3. Clerc, Maurice. *Particle swarm optimization*. Vol. 93. John Wiley & Sons, 2010.
4. Hariri, S., and M. Parashar. "Handbook of bioinspired algorithms and applications, chapter the foundations.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU1.1	NATURE INSPIRED ALGORITHMS	CO1	Define the basic concept of natural phenomenon for developing the optimization algorithms
		CO2	Classify the evolutionary algorithms as per the optimization problem
		CO3	Apply the swarm intelligence based algorithms to solve the engineering optimization problems
		CO4	Analyse the discrete variants of the nature inspired algorithms
		CO5	Develop the step by step learning mechanism of the nature inspired algorithms

18.10.'19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU1.2 ADVANCED OPERATING SYSTEM

Credit: 3 (3L+0T+0P) **Max. Marks: 150 (IA:50, ETE:100)**

Objectives:

1. To make students understand the basic structure and functions of operating system .
2. To familiarize the students with architecture of distributed operating system.
3. To familiarize the students with concepts of multiprocessor operating systems.
4. To expose the students various real time operating systems.

Prerequisites: Fundamentals of operating systems

Syllabus:

Introduction: Functions of operating systems, Design approaches: layered ,kernel based and virtual machine approach, why advanced operating systems, types of advanced operating systems

Distributed Operating Systems: Architecture of distributed operating systems, system architecture types, issues in distributed operating systems, inherent limitation of distributed systems, distributed mutual exclusion: classification of mutual exclusion algorithms, Lamport's token based algorithm, Suzuki-Kasami's Broadcast algorithm, Raymond's Tree based algorithm, Distributed deadlock detection, Distributed file systems, Distributed shared memory, Distributed scheduling

Multiprocessor Operating Systems: Introduction, structure of multiprocessor operating system, operating system design issues, threads, the test and set instruction, the swap instruction, implementation of the process wait , processor scheduling, reliability and fault tolerance.

Real Time Operating System: Introduction to Real time systems and Real Time Operating Systems, Characteristics of Real Time operating Systems, Classification of Real Time Operating Systems, Services, structure, goal and feature of RTOS, architecture of RTOS, micro kernels and monolithic kernels, tasks in RTOS, Performance measures, estimating program runtimes, task assignment, scheduling in RTOS, rate monotonic scheduling, priority inversion, task management, inter task communication, applications of various RTOS.

Data base operating Systems: Introduction to database operating systems, concurrency control: theoretical aspect, distributed database system, concurrency control algorithms Application development using Android. Introduction to cloud OS.

18.10.19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. M Singhal and NG Sivaratri, Advanced Concepts in Operating Systems, Tata McGraw Hill Inc., 2001
2. A.S. Tanenbaum, Distributed Operating system, Pearson Education Asia, 2001.
3. A.S. Tanenbaum, Modern Operating system, Prentice Hall, 3rd edition.
4. Real Time Operating System, Barr M.
5. Real-Time Systems, Jane Liu, Pearson Ed. Asia
6. Real -Time Systems, Krishna and Shin, McGraw Hill International.

REFERENCE BOOKS:

1. SILBERSCHATZ and P. GALVIN, Operating System Concepts, VI edition, Addison Wesley 2004.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU1.2	ADVANCED OPERATING SYSTEM	CO1	To understand the architecture of various types of operating systems
		CO2	To know the scheduling policies about distributed operating systems
		CO3	To familiarize with various deadlock management policies of different types of operating systems
		CO4	Memory management policies of different operating systems
		CO5	Understand the design of mobile operating systems

18-10-19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU1.3 SOFTWARE TESTING

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Learn concepts of graph theory for testing approaches.
2. Differentiate between black box, white box, and grey box testing
3. Understand advanced and emerging in project management technologies
4. Obtain skills to a generation of test cases basis on UML models
5. Learn how to use object-orient testing approaches for testing software.

Prerequisites: Software Engineering

Syllabus:

Introduction: Basic concepts, discrete mathematics for testers, Graph theory for testers,

Black box testing: Boundary value testing, Equivalence class testing,

White box testing: statement coverage, Branch coverage, condition coverage, path coverage, Mc Cabe's cyclomatic complexity; Decision Table based testing, Data flow based testing, Integration testing, System testing, Interaction testing, Performance testing, Mutation testing, Regression testing, error seeding ,

Object-Oriented testing: issues in object oriented testing , Test case design by object oriented software, Fault based testing, test cases and class hierarchy, Scenario based Test design, Testing surface structure and deep structure,

Model Based Testing: Tests case derived from behaviour models, Test case generation using UML diagrams, GUI testing, object oriented system testing.

TEXT BOOKS:

1. C. J. Paul, Software testing: A craftsmen's approach, CRC Press, 2nd Ed, 2002.
2. R. Gopalswamy, Software testing, Pearson, 2005.

REFERENCE BOOKS:

1. R. S. Pressman, Software Engineering A Practitioner's Approach, McGraw Hill Publications, 2006.
2. R. Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2nd Ed, 2006.

18-10-19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU1.3	SOFTWARE TESTING	CO1	Recall concepts of graph theory for testing approaches.
		CO2	Differentiate between black box, white box, and grey box testing.
		CO3	Describe advanced and emerging in project management technologies.
		CO4	Obtain skills to a generation of test cases basis on UML models
		CO5	Formulate how to use object-orient testing approaches for testing software.

18-10-19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU2.1 ADVANCED DATABASE MANAGEMENT SYSTEM

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To understand the concept of Distributed Database Systems (DDBMS), including the architecture and design of DDBMS.
2. To apply various fragmentation techniques for a given problem
3. To understand the steps of query processing and how optimization techniques are applies to Distributed Database
4. To understand Transaction Management & Compare various approaches to concurrency control in Distributed database
5. To understand the concept of NOSQL.

Prerequisites: Overview of DBMS, RDBMS and Computer Networks.

Syllabus:

Distributed Databases: An Overview: Distributed Data Processing, What is a Distributed Database System, Features of Distributed versus Centralized Databases, Why Distributed Databases, Distributed Database Management Systems, Design Issues, Distributed DBMS Architecture.

Distributed Database Design: Top-Down Design Process, Distribution Design Issues: Reasons for Fragmentation, Fragmentation Alternatives, Degree of Fragmentation, Correctness Rules of Fragmentation, Allocation Alternatives, Fragmentation: Horizontal Fragmentation, Vertical Fragmentation, Hybrid Fragmentation, Allocation of Resources, Data Directory.

Query Processing: Query Processing Problem, Objectives of Query Processing ,Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing, Query Decomposition, Localization of Distributed Data, Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.

Distributed Concurrency Control: Serializability Theory, Taxonomy of Concurrency Control Mechanisms, Locking-Based Concurrency Control Algorithms, Timestamp-Based Concurrency Control Algorithms, Optimistic Concurrency Control Algorithms, Deadlock Management.
Cassandra(NoSQL): Introduction to Cassandra, Problems in the RDBMS, NoSQL, Cassandra, Cassandra Data Model, Cassandra architecture, Components of Cassandra, Reading and Writing Data.

18.10.19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. M. Tamer Özsu; and Patrick Valduriez," Principles of Distributed Database Systems",3 Ed, 2011, Springer
2. Giuseppe Pelagatti and Stefano Ceri," Distributed Databases: Principles and Systems"1 Ed,2008, McGraw Hill Education.
3. Elmasi, R. and Navathe, S.B., "Fundamentals of Database Systems", 4th Ed., 2005, Pearson Education.
4. A Silberschatz, H Korth, S Sudarshan, "Database System and Concepts", 5 Ed, 2005, McGraw-Hill.
5. Eben Hewitt," Cassandra: The Definitive Guide",2011, O'Reilly.

REFERENCE BOOKS:

1. Saeed K. Rahimi and Frank S. Haug," Distributed Database Management Systems: A Practical Approach",2010, Wiley-IEEE Press.
2. Nishant Neeraj," Mastering Apache Cassandra", 2013, Packt Publishing.

COURSE OUTCOMES:

Course code	Course name	Course Outcome	Details
8ITU2.1	ADVANCED DATABASE MANAGEMENT SYSTEM	CO1	Learn the concept, architecture and design of DDBMS
		CO2	Apply various fragmentation techniques for a given problem
		CO3	Apply query processing and optimization techniques for a Distributed Database
		CO4	Analysis various approaches to concurrency control in Distributed database
		CO5	Create and analysis of NOSQL database using Cassandra

18-10-19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU2.2 REAL TIME SYSTEM

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To make students understand the basic concepts of real time system and their applications.
2. To familiarize the students with types of real time tasks and their handling mechanisms.
3. To familiarize the students with concepts related to real time scheduling.
4. To make students understand about effective use of resources in real time environment..

Syllabus:

Introduction: Definition, Typical Real Time Applications, concept of tasks, types of tasks and real time systems, block diagram of RTS, and tasks parameters -Release Times, execution time, period, Deadlines, and Timing Constraints etc. RTS requirements.

Reference Models for Real Time Systems: processors and Resources, Temporal Parameters of Real-Time Workload, Periodic and Aperiodic Task Model, Precedence Constrains and Data Dependency, Other Types of Dependencies, Functional Parameters, Resource Parameters.
Real Time Scheduling: classification of Real Time Scheduling, scheduling criteria, performance metrics, schedulability analysis, Introduction to Clock Driven scheduling, Weighted Round Robin Approach and Priority Driven Approach. Dynamic Versus Static systems, Offline Versus Online Scheduling.

Periodic tasks scheduling: Clock Driven Scheduling – definition, notations and assumption, scheduler concepts, general scheduling structure,cyclic executives.
Priority Driven Scheduling; notations and assumption, fixed priority verses dynamic priority, fixed priority scheduling algorithms (RM and DM) and their schedulability analysis, concept of schedulability tests – Inexact and exact schedulability tests for RM and DM, Optimality of the RM and DM algorithms, practical factors.

Aperiodic task scheduling: assumption and approaches, server based and non-server based fixed priority scheduling algorithms– polling server, deferrable server, simple sporadic server, priority exchange, extended priority exchange, slack stealing.

Resources Access Control: Assumptions on Resources and their usage, Effect of Resource Contention and Resource Access Control (RAC), Non-preemptive Critical Sections, priority inversion problem, need of new resource synchronization primitives/protocols for RTS, Basic Priority-Inheritance and Priority-Ceiling Protocols, Stack Based Priority-Ceiling Protocol, Use of Priority- Ceiling Protocol in Dynamic Priority Systems, Preemption Ceiling Protocol, Access Control in Multiple- Unit Resources, Controlling Concurrent Accesses to Data Objects.

18-10-19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT/REFERENCE BOOKS:

1. J.W.S.Liu: Real-Time Systems, Pearson Education Asia
2. C.M. Krisna & K. G. Shim- Real time systems- TMH
3. P.D. Laurence, K.Mauch: Real Time Microcomputer System Design, An Introduction TMH

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU2.2	REAL TIME SYSTEM	CO1	Identify the components of real time system
		CO2	Difference between hard real time system Vs Soft real time system
		CO3	To know the difference between the periodic task scheduling and aperiodic task scheduling
		CO4	Use of scheduling policies for various types of real time tasks
		CO5	To understand the concepts of resource access control

18-10-19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU2.3 BIOINFORMATICS

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To give introduction to the basic practical techniques of bioinformatics.
2. To understand applications of bioinformatics.
3. Emphasis on biological databases to problem solving in real research problems.

Syllabus:

Major Bioinformatics Resources: NCBI, EBI, ExPaSy.
Pairwise sequence alignments: Sequence similarity, identity, and homology. Global and local alignment, Dot plots for sequence comparison, Dynamic programming, BLAST and PSI-Blast, Application of Blast tool, Concept of Scoring matrix (PAM and BLOSUM).

Multiple sequence alignments: Progressive Alignment Algorithm (ClustalW), Application of multiple sequence alignment.
Phylogenetic analysis: Definition and description of phylogenetic trees, a primer on computational phylogenetic analysis. Computational gene prediction methods, analysis of codon usage bias, computational prediction and analysis of regulatory sites.
Schematic representations and structure visualization of proteins structure, Protein DataBank.

TEXT BOOKS:

1. Claverie, J.M. and Notredame C. 2003 Bioinformatics for Dummies. Wiley Editor.
2. Letovsky, S.I. 1999 Bioinformatics. Kluwer Academic Publishers.
3. Baldi, P. and Brunak, S. 2001 Bioinformatics: The machine learning approach, The MIT Press.

REFERENCE BOOKS:

1. Setubal, J. and Meidanis, J. 1996 Introduction to Computational Molecular Biology. PWS Publishing Co., Boston.
2. Lesk, A.M. 2005, 2nd edition, Introduction to Bioinformatics. Oxford University Press.
3. Fogel, G.B. and Corne, D.W., Evolutionary Computation in Bioinformatics.
4. Mount, D.W., Bioinformatics: 2001, Sequence and Genome Analysis. CSHL Press Durbin R., Eddy S., Krogh A. and Mithchison G. 2007 Biological Sequence Analysis, Cambridge University Press.

18-10-19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU2.3	BIOINFORMATICS	CO 1	Give students an introduction to the basic practical techniques of bioinformatics.
		CO 2	Application of bioinformatics and biological databases to problem solving in real research problems.
		CO 3	Know about Multiple sequence alignments and phylogenetic analysis.
		CO 4	Analysis of codon usage bias, computational prediction and analysis of regulatory sites.

12-10-19

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU3.1 OPERATION RESEARCH

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Discuss models, objectives and phases of operation research
2. Provide an understanding of different methods to solve optimization problem
3. Describe different sequencing and network optimization models
4. Describe queening theory with its different dimensions
5. Discuss different types of simulation models with its advantages and disadvantages

Syllabus:

Introduction to Operations Research: Basics definition, scope, objectives, phases, models and limitations of Operations Research.
Linear Programming Problem – Formulation of LPP, Graphical solution of LPP. Simplex Method, Artificial variables, two-phase method, degeneracy and unbound solutions. Duality, Dual-Simplex, Sensitivity analysis.

Transportation Problem: Formulation, solution, unbalanced Transportation problem. Finding basic feasible solutions – Northwest corner rule, least cost method and Vogel's approximation method. Optimality test: the stepping stone method and MODI method.
Assignment model: Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Traveling salesman problem and assignment problem.

Sequencing models: Solution of Sequencing Problem – Processing n Jobs through 2 Machines – Processing n Jobs through 3 Machines – Processing 2 Jobs through m machines – Processing n Jobs through m Machines.
Network Optimization Models: Prototype example, terminology of networks, Shortest-path problem, minimum spanning tree problem, maximum flow problem, minimum cost flow problem, network simplex method

Queuing Theory: Introduction, single channel - poission arrivals - exponential service times with infinite population & finite population, Multi-channel - poisson arrivals - Exponential service times with infinite population

Simulation: Introduction, definition, rationale, simulation models, Monte-Carlo Simulation, Generation of /random-numbers for Monte-Carlo, application of simulation, advantages of simulation, limitations of simulations

18-10-19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

TEXT BOOKS:

1. Operation Research by Fredrick S. Hillier Geralad J. Liberman, Bodhibrata Nag, Preetam Basu
2. Operation Research by Hamdy A. Taha.

REFERENCE BOOKS:

1. Operation Research: Principles and Practice, Ravindran, Phillips
2. Operation Research by Er. Prem Kumar Gupta, D.S Hira

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU3.1	OPERATION RESEARCH	CO 1	Define operation research and Linear Programming problem
		CO 2	Identify different methods to solve transportation problem
		CO 3	Illustrate sequential and network optimization models and recognize their limitations.
		CO 4	Analyse simulation models.

18-10-19

Anil K. Mathur

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU3.2 PROJECT MANAGEMENT

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. To provide basic concept of project management.
2. To cover the concepts of PERT and CPM for developing project network.
3. To cover the concept of risk management.
4. To know the fundamental software quality and organizational issues.

Prerequisites: Software Engineering, Software Testing

Syllabus:

Basics of Project Management: Introduction, Need for Project Management, Project Management Knowledge Areas and Processes, The Project Life Cycle, The Project Manager (PM), Phases of Project Management Life Cycle, Project Management Processes, Impact of Delays in Project Completions, Essentials of Project Management Philosophy, Project Management Principles

PERT and CPM: Introduction, Development of Project Network, Time Estimation, Determination of the Critical Path, PERT Model, Measures of variability, CPM Model, Network Cost System

Project Risk Management: Introduction, Risk, Risk Management, Role of Risk Management in Overall Project Management, Steps in Risk Management, Risk Identification, Risk Analysis, Reducing Risks

Software Quality: S/W quality engineering, defining quality requirements, quality standards, practices & conventions, ISO 9000, ISO 9001, S/W quality matrices,

Managerial and organization issues: defect prevention, reviews & audits, SEI capability maturity model, six sigma. Special topics in process and quality management, Agile and Extreme Programming..

TEXT BOOKS:

1. R. H. Thayer, Software Engineering Project management, IEEE CS Press, 2nd Ed, 1988.
2. R. Pressman, Software Engineering A Practitioner's approach, McGraw Hill, 4th Ed, 2005.

18-10-19

15

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

REFERENCE BOOKS:

1. B. Hughes, M. Cotterell, Software Project Management, McGraw Hill, 4th ed, 2005.
2. R. Walker, Software Project Management, Pearson, 2003,

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU3.2	PROJECT MANAGEMENT	CO1	Describe the introduction of project management methods.
		CO2	Apply CPM and PERT concept on real time project.
		CO3	Describe the software quality concept according to real time case study.
		CO4	Apply risk analysis and risk management concept on advanced project.

18-10-19

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

8ITU3.3 SOCIAL NETWORK ANALYSIS

Credit: 3 (3L+0T+0P) **Max. Marks: 150(IA:50, ETE:100)**

Objectives:

1. Understand the concept of semantic web and related applications.
2. Learn knowledge representation using ontology.
3. Understand human behaviour in social web and related communities
4. Learn visualization of social networks.

Prerequisite: Basic knowledge in the fields of graph theory and artificial intelligence/ database applications will be necessary for this course.

Syllabus:

Introduction to Semantic Web: Limitations of current Web, Development of Semantic Web, Emergence of the Social Web, Social Network analysis: Development of Social Network Analysis, Key concepts and measures in network analysis, Electronic sources for network analysis: Electronic discussion networks, Blogs and online communities, Web based networks, Applications of Social Network Analysis.

Modelling, aggregating and knowledge representation: Ontology and their role in the Semantic Web: Ontology-based knowledge representation, Ontology languages for the Semantic Web: Resource Description Framework, Web Ontology Language, Modelling and aggregating social network data: State-of-the-art in network data representation, Ontological representation of social individuals, Ontological representation of social relationships, Aggregating and reasoning with social network data, Advanced representations.

Extraction and mining communities in web social networks: Extracting evolution of Web Community from a Series of Web Archive, Detecting communities in social networks, Definition of community, Evaluating communities, Methods for community detection and mining, Applications of community mining algorithms, Tools for detecting communities social network infrastructures and communities, Decentralized online social networks, Multi-Relational characterization of dynamic social network communities.

Predicting human behaviour and privacy issues: Understanding and predicting human behaviour for social communities, User data management, Inference and Distribution, Enabling new human experiences, Reality mining, Context Awareness, Privacy in online social networks, Trust in online environment, Trust models based on subjective logic, Trust network analysis, Trust transitivity analysis, Combining trust and reputation, Trust derivation based on trust comparisons, Attack spectrum and countermeasures.

18-10-19

Anil K. Mathur

Approved
Dean, FA & UD

Admission Year (AY)	2015-16	2016-17	2017-18	2018-19	
Scheme Code(SC)	2015IT	2016IT	2017IT	2017IT	

SC- 2017IT

Visualization and applications of social networks: Graph theory, Centrality, Clustering, Node-Edge Diagrams, Matrix representation, Visualizing online social networks, Visualizing social networks with matrix-based representations, Matrix and Node-Link Diagrams, Hybrid representations, Applications, Cover networks, Community welfare, Collaboration networks, Co-Citation networks.

Case study: Understanding classrooms using social network analysis.

TEXT BOOKS:

1. Peter Mika, "Social Networks and the Semantic Web", , First Edition, Springer 2007.
2. Borko Furht, "Handbook of Social Network Technologies and Applications", 1st Edition, Springer, 2010.

REFERENCE BOOKS:

1. Guandong Xu ,Yanchun Zhang and Lin Li, "Web Mining and Social Networking, Techniques and applications", First Edition Springer, 2011.
2. Dion Goh and Schubert Foo, "Social Information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively", IGI Global Snippet, 2008.

COURSE OUTCOMES:

Course Code	Course Name	Course Outcome	Details
8ITU3.3	SOCIAL NETWORK ANALYSIS	CO 1	Outline the need for semantic web and its analysis
		CO 2	Describe ontology and Ontology-based knowledge representation
		CO 3	Identify extraction and mining communities in web social networks
		CO 4	Predict human behavior in social web and related communities
		CO 5	Discuss Visualization and applications of social networks

18-10-19

Approved
Dean, FA & UD