SCHEME AND SYLLABUS

(Scheme Code- 2015CSE)

OF

COMPUTER SCIENCE AND ENGINEERING

FOR ADMISSION YEAR 2015-16

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

Scheme Code - 2015CSE

(100)

Rajasthan Technical University, Kota
Proposed Scheme for Computers Science & Engineering (2016-17)
Theory and Practical

Sem	Codes	Proposed scheme-CE-RTU	Max marks			Contact Hours/Week				Credits
			IA	TE	Total	L	T	P	Total	
III	3CSU01	Electronic Devices and Circuits	50	100	150	3	0	0	3	3
	3CSU02	Data Structures and Algorithms	50	100	150	3	0	0	3	3
	3CSU03	Digital Electronics	50	100	150	3	0	0	3	3
	3CSU04	Software Engineering	50	100	150	3	0	0	3	3
	3CSU05	Object Oriented Programming	50	100	150	3	0	0	3	3
	3CSU06	Advanced Engineering Mathematics	50	100	150	3	1	0	4	4
	3CSU07	Electronic Devices and Circuits Lab	50	25	75	0	0	2	2	1
	3CSU08	Data Structures and Algorithms Lab	50	25	75	0	0	3	3	2
	3CSU09	Digital Electronics lab	50	25	75	0	0	2	2	1
	3CSU10	Object Oriented Programming Lab	50	25	75	0	0	3	3	2
	3CSU12	Discipline and Extra-curricular activities			50	0	0	0	0	1
		Sub Total			1250	18	1	10	29	26

Note:- Approved by Deptt. BoS on 23.11.16

23/11/16
(R. S. Shaoran)

23/11/2016
(Dinesh Jain)

23/11/16.
(Dr. S. C. Jain)

23/11/16
(Harish Sharma)

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

Scheme Code - 2015CSE

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech.(Computer Science & Engineering) 2016-17
Theory and Practical

Sem	Course Code	Subject Name	Max Marks			Contact Hours/Week				Credits
			IA	TE	Total	L	T	P	Total	
IV	4CSU01	Microprocessor & Interfaces	50	100	150	3	0	0	3	3
	4CSU02	Discrete Mathematics Structure	50	100	150	3	1	0	4	4
	4CSU03	Linux Shell Programming	50	100	150	3	0	0	3	3
	4CSU04	Analysis of Algorithms	50	100	150	3	0	0	3	3
	4CSU05	Principles of Communication	50	100	150	3	0	0	3	3
	4CSU06	Computer Network	50	100	150	3	0	0	3	3
	4CSU07	Microprocessor Lab	50	25	75	0	0	3	3	2
	4CSU08	Communication Lab	50	25	75	0	0	2	2	1
	4CSU09	Linux Shell Programming Lab	50	25	75	0	0	2	2	1
	4CSU10	Advance Data Structures Lab	50	25	75	0	0	3	3	2
	4CSU12	Discipline and Extra-curricular Activities			50	0	0	0	0	1
		Sub Total			1250	18	1	10	29	26

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

Scheme Code - 2015CSE

(77)

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech.(Computer Science & Engineering) 2016-17
Theory and Practical

Sem	Course Code	Subject Name	Max Marks			Contact Hours/Week				Credits
			IA	TE	Total	L	T	P	Total	
V	5CSU01	Data base Management system	50	100	150	3	0	0	3	3
	5CSU02	Operating system	50	100	150	3	0	0	3	3
	5CSU03	Theory of computation	50	100	150	3	1	0	4	4
	5CSU04	Computer Architecture	50	100	150	3	0	0	3	3
	5CSU05	Embedded System	50	100	150	3	0	0	3	3
	5CSU06	*Elective	50	100	150	3	0	0	3	3
	5CSU07	Data base Management system Lab	50	25	75	0	0	3	3	2
	5CSU08	Java Lab	50	25	75	0	0	3	3	2
	5CSU09	Embedded System Lab	50	25	75	0	0	2	2	1
	5CSU10	CASE Lab	50	25	75	0	0	2	2	1
	5CSU12	Discipline and Extra-curricular Activities			50	0	0	0	0	1
		Sub Total			1250	18	1	10	29	26

***Proposed Subjects for Elective (5CSU06)**

5CSU06.1	Information theory and coding
5CSU06.2	Human computer interface
5CSU06.3	Software Testing and Project management

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

Scheme Code - 2015CSE

Scheme for B.Tech.(Computer Science & Engineering) 2016-17
Theory and Practical

Sem	Course Code	Subject Name	Max Marks			Contact Hours/Week				Credits
			IA	TE	Total	L	T	P	Total	
VI	6CSU01	Computer Graphics	50	100	150	3	0	0	3	3
	6CSU02	Information security system	50	100	150	3	1	0	4	4
	6CSU03	Compiler construction	50	100	150	3	0	0	3	3
	6CSU04	Data Mining and warehouse	50	100	150	3	0	0	3	3
	6CSU05	Mobile computing	50	100	150	3	0	0	3	3
	6CSU06	*Elective -2	50	100	150	3	0	0	3	3
	6CSU07	Computer Graphics lab	50	25	75	0	0	3	3	2
	6CSU08	Web Programming lab	50	25	75	0	0	2	2	1
	6CSU09	Compiler construction lab	50	25	75	0	0	2	2	1
	6CSU10	Advance java lab	50	25	75	0	0	3	3	2
	6CSU12	Discipline and Extra-curricular activities			50	0	0	0	0	1
		Sub Total			1250	18	1	10	29	26

10/01/18

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

Scheme Code - 2015CSE

Scheme for B.Tech.(Computer Science & Engineering) 2016-17
Theory and Practical

Proposed Subjects For Elective- Computer Science

5CSU06.1	Information theory and coding
5CSU06.2	Human computer interface
5CSU06.3	Software Testing and Project management
6CSU06.1	Artificial intelligence
6CSU06.2	Bioinformatics
6CSU06.3	Fuzzy logic and Application
7CSU06.1	Robotics
7CSU06.2	Cyber security
7CSU06.3	Internet of things
8CSU01.1	Machine learning
8CSU01.2	Recent trends in Computer Technology
8CSU01.3	Advance Operating System
8CSU02.1	Big data
8CSU02.2	Computer Vision
8CSU02.3	High performance computing

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

Scheme Code - 2015CSE

Theory and Practical

Sem	Course Code	Subject Name	Max Marks			Contact Hours/Week				Credits
			IA	TE	Total	L	T	P	Total	
VII	7CSU01	Cloud Computing	50	100	150	3	0	0	4	4
	7CSU02	Real time system	50	100	150	3	1	0	3	3
	7CSU03	Soft Computing	50	100	150	3	0	0	3	3
	7CSU04	Digital Image Processing	50	100	150	3	0	0	3	3
	7CSU05	Distributed System	50	100	150	3	0	0	3	3
	7CSU06	*Elective -3	50	100	150	3	0	0	3	3
	7CSU07	DIP lab	50	25	75	0	0	3	3	2
	7CSU08	Android lab	50	25	75	0	0	3	3	2
	7CSU09	Minor Project	50	25	75	0	0	2	2	1
	7CSU10	Seminar	50	25	75	0	0	2	2	1
	7CSU12	Discipline and Extra-curricular activities			50	0	0	0	0	1
		Sub Total			1250	18	1	10	29	26

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

Scheme Code - 2015CSE

Rajasthan Technical University, Kota
Department of Computer Science& Engineering
Scheme for B.Tech. VIII Sem(Computer Science & Engineering) of session 2018-19
(The scheme is applicable for the students admitted in session 2015-16 and 2016-17)
Theory and Practical

SEMESTER-VIII				Hrs./Week			IA	End Term exam	Total
Course code	Type of Course	Course	Credits	L	T	P			
8CSU01	DCC	Machine Learning	4	3	1		50	100	150
8CSU02	DCC	Big Data using Hadoop	4	3	1		50	100	150
8CSU03.X	DEC	*Elective	3	3	0		50	100	150
8CSU07	DCC	Machine Learning Lab	1			2	50	25	75
8CSU08	DCC	Big Data using Hadoop Lab	1			2	50	25	75
8CSU09	DCC	SEMINAR	4			4	150	75	225
8CSU10	DCC	PROJECT	8			12	250	125	375
8CSU12		Extra Curricular& Discipline	1				50		50
		Total	26	9	2	20	700	550	1250

***Proposed Subjects for Elective**

Course code	Course
8CSU03.1	Computer Vision
8CSU03.2	Advance Operating System
8CSU03.3	Wireless Sensor Networks

03/01/19 (Harish Sharma)

03/01/19

(Nirmala Sharma)

3/1/19 (Parul Chauhan)

(Rajesh Dadhich)

(Dr. Vikas Pareek)

SYLLABUS

Semester	**III**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2016-17**

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

Scheme Code - 2015CSE

3CS301 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 BOOKS:

1. Electronic devices & circuits theory By R.L. Boylestad, Louis Nashelsky,Pearson Education
2. Integrated Electronics By Millman Halkias, T.M.H

REFERENCE BOOKS:

1. Electronic devices & circuits By David Bell, Oxford Publications

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

Scheme Code - 2015CSE

3CSU02 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.

Linear Data Structures: Array and its storage representation, sparse matrices stack, queue, dequeue, circular queue for insertion and deletion.

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, various implementations of Linked list.

Searching: Sequential and binary search

Sorting: Insertion, quick, heap, topological and bubble sorting algorithms.

Non-Linear Structures: Definition of tree, binary tree, tree traversal binary search tree, B-tree , B+ tree, AVL tree, Threaded binary tree.

Graphs: Definition its various representations Depth first and breadth first traversal of graphs, spanning tree, Single source single destination shortest path algorithms.

TEXT BOOKS

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

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

Scheme Code - 2015CSE

3CSU03 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.

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

Scheme Code - 2015CSE

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

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

Scheme Code - 2015CSE

3CSU04 SOFTWARE ENGINEERING

MAX_MARKS(50+100)

Objectives:

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)

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

Scheme Code - 2015CSE

3CSU05 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 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 new and delete operators.

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.

TEXT BOOKS:

1. How to Program C++, Dietel, Pearson
2. Mastering C++ By K.R. Venugopal, TMH

REFERENCE BOOKS

1. Object Oriented Programming in C++ By Robert Lafore, Pearson
2. Object Oriented Design & Modelling, Rambaugh, Pearson

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

Scheme Code - 2015CSE

3CSU06 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:

1. Probability and Statistics; Schaum's Outline Series, Murray Spiegel, John Schiller, R. AluSrinivasan, *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 BOOKS:

1. Advanced Engineering Mathematics, 4th Editions, Jain and Iyengar ,*Narosa Publications*.
2. Higher Engineering Mathematics, B. V. Ramana, 1st Edition, *McGraw 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.

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

Scheme Code - 2015CSE

3CSU07 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.

8Plot 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.

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

23/11/16 23/11

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.

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

Scheme Code - 2015CSE

3CSU08 DATA STRUCTURES AND ALGORITHMS 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.

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

Scheme Code - 2015CSE

3CSU09 DIGITAL ELECTRONICS LAB

MAX_MARKS(50+25)

Objectives:

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 gates and 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
10 Perform input/output operations on parallel in/Parallel out and Serial in/Serial out registers using clock. Also exercise loading only one of multiple values into the register 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.

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

Scheme Code - 2015CSE

3CSU10 OBJECT ORIENTED PROGRAMMING LAB

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.

SYLLABUS

Semester	**IV**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2016-17**

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

Scheme Code - 2015CSE

4CSUC1 MICROPROCESSOR AND INTERFACES

Maximum Marks (50+100)

Objective:

The objective of this course is to provide a theoretical & practical introduction to microcontrollers and microprocessors, assembly language programming techniques, design of hardware interfacing circuit, microcontroller and microprocessor system design considerations.

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/Reference books

1. Microprocessor architecture, programming, and applications with the 8085 By Ramesh S. Gaonkar
2. Introduction to Microprocessor By Aditya P. Mathur, TATA MAC GRAW HILL
3. Microprocessor &Interfaceing By Douglas V. Hall, TATA MAC GRAW HILL
4. Microprocessor & Peripheral By A.K.Ray, K.M. Bhurchandi, TATA MAC GRAW HILL

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

Scheme Code - 2015CSE

(60)

4CSU02 DISCRETE MATHEMATICS STRUCTURE

Maximum Marks (50+100)

Objectives:

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

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 & CeilingFunctions), 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& Hamiltoniar Graphs.

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

Text/Reference books:

1. Discrete Mathematical Analysis, Kolman et al., Pearson Education
2. Discrete Mathematics and its Applications, Kenneth H. Rosen, MAC GRAW HILL
3. Discrete Mathematical Structures, Lipschutz& Lipson, MAC GRAW HILL
4. Discrete Mathematics with Applications, Koshy, ELSEVIER

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

Scheme Code - 2015CSE

4CSU03 LINUX AND SHELL PROGRAMMING

Maximum Marks (50+100)

Objective:

The objective of this course is to introduce Unix/Linux kernel programming techniques that useful in B.Tech.Computer science and Information Technology students in more practical manner.This course also provide the knowledge of shell scripting and Linux tools and applications.

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.

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

Scheme Code - 2015CSE

Text/Reference books:

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

4CSU04 ANALYSIS OF ALGORITHMS

Maximum Marks (50+100)

Objectives- To teach various problem solving strategies. To teach mathematical background for algorithm analysis and implementation of various strategies like divide and conquer, Greedy method, Dynamic programming, backtracking, branch and bound. To teach different string matching algorithms.

Contents of the subject

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

Text:

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 Ullman: 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.

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

Scheme Code - 2015CSE

4CSU05 PRINCIPLES OF COMMUNICATION

Maximum 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, directsequence spread spectrum (DSSS) with coherent BPSK, processing gain, probability of error, frequency-hop spread spectrum (FHSS). Application of spread spectrum: CDMA

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

Scheme Code - 2015CSE

Text/Reference Books:

1. Principal of Communication system By Taub Schilling, TATA MAC GRAW HILL
2. Fundamentals of communication system By Proakis&Salehi, Pearson Education education
3. Communication system by Simon Haykin, John Wiley
4. Communication system (Analog and Digital) By R.P. Singh, S.D.Spare, TATA MAC GRAW HILL
5. Modern Digital & Analog Communication By B. P. Lathi oxford university

(59)

4CSU06 COMPUTER NETWORKS

Maximum Marks (50+100)

Objective:

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:

Network Reference Models (OSI/ISO and TCP/IP) Issues and Challenges in Physical Layer.

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 Coding: Error Detection, Two Dimensional Parity Checks, and Internet Checksum. Polynomial Codes, Standardized polynomial codes, error detecting capability of a polynomial codes. Linear codes, performance of linear codes, error detection & correction using linear codes, Data Link Control: HDLC & PPP including frame structures.

MAC sub layer: Channel Allocation Problem, Pure and slotted Aloha, CSMA, CSMA/CD, collision free multiple access, Throughput analysis of pure and slotted Aloha, Ethernet Performance.

Network layer-design issue, routing algorithms: Distance vector, link state, hierarchical, Broadcast routing, Congestion control: congestion prevention policies, congestion control in Datagram subnets, load shedding, jitter control, Leaky bucket and token bucket algorithms.

Internetworking: Differences in networks, Tunneling, Internetwork routing, Fragmentation Network layer in the Internet: IPv4 classful and classless addressing, subnetting Network layer protocols(only working and purpose; packet headers etc. not included), Differences in IPV6 over IPV4.

(50)

Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and demultiplexing, crash recovery, introduction to UDP protocol, Principles of Reliable Data Transfer

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

Application Layer: Domain Name System (DNS), SMTP, HTTP, File Transfer Protocol (FTP), Introduction to Network security.

Text/Reference Books:

1. Tanenbaum;ComputerNetwork,4thEd.,Pearson Education
2. Stallings,Dataandcomputercommunication,8th ed.Pearson Education
3. Kurose;ComputerNetworking,3rdEd.,Pearson Education.
4. Peterson,Davie;ComputerNetworks,4rdEd.,ELSEVIER
5. AlbertoLeon-Garcia,IndraWidjaja, COMMUNICATIONNETWORKS, 2nd ed., TATA MAC GRAW HILL
6. Analysis of Computer and Communication Networks, ISBN: 0387744363, Fayez

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

Scheme Code - 2015CSE

4CSU07 MICROPROCESSOR LAB

Maximum Marks (50+25)

Objective:

The objective of this lab is to introduce the basics of microprocessor programming, learn use of microprocessor in simple applications. this lab also provide a theoretical & practical introduction to microcontrollers and microprocessors, assembly language programming techniques, design of hardware interfacing circuit, microcontroller and microprocessor system design considerations.

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.

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

Scheme Code - 2015CSE

(56)

4CSU08 COMMUNICATION LAB

Maximum Marks (50+25)

Objectives:

To understand basic analog and digital communication system theory and design, with an emphasis on wireless communications methods.

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).

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

Scheme Code - 2015CSE

(57)

4CSU09 LINUX SHELL PROGRAMMINGLAB

Maximum Marks (50+25)

Objective:

The objective of this lab is to learn Unix/Linux Shell command file and directory structure of Unix file system. This lab also introduces the fundamentals of shell scripting/programming. Students will be able to use Linux environment efficiently and solve problems using bash for shell scripting.

List of Experiments

5. Use of Basic Unix Shell Commands: ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.
6. Commands related to inode, I/O redirection and piping, process control commands, mails.
7. 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 administrat.
8. 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.
9. 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.
10. Write a shell script to change data format .Show the time taken in execution of this script
11. Write a shell script to print files names in a directory showing date of creation & serial number of the file.
12. Write a shell script to count lines, words and characters in its input (do not use wc).
13. Write a shell script to print end of a Glossary file in reverse order using Array. (Use awk tail)
14. Design programs for different scheduling algorithm of OS and compare their performance.

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

Scheme Code - 2015CSE

54

4C U010 ADVANCE DATA STRUCTURE LAB

Maximum 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 algorithmsfor 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:

SYLLABUS

Semester	**V**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2017-18**

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

Scheme Code - 2015CSE

5CSU01 DATABASE MANAGEMENT SYSTEM

Maximum Marks(50+100)

Objective:

1. The knowledge acquired here will be utilized in the laboratory course for developing database applications.
2. To cover the concepts of Relational data model and Relational Algebra
3. To provide concept and need of database system
4. To make students design logical database with the help of E-R model.

Syllabus:

Introduction to database systems: Overview and History of DBMS. File System v/s DBMS .Advantage of DBMS Describing and Storing Data in a DBMS. Queries in DBMS. Structure of a DBMS.

Entity Relationship 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, Class Hierarchies, Aggregation, Conceptual Data Base, and Design with ER Model- Entity v/s Attribute, Entity vs Relationship Binary vs Ternary Relationship and Aggregation v/s ternary Relationship Conceptual Design for a Large Enterprise.

Relationship Algebra and Calculus: Relationship Algebra Selection and Projection, Set Operations, Renaming, Joints, Division, Relation Calculus, Expressive Power of Algebra and Calculus.

SQL queries programming and Triggers: The Forms of a Basic SQL Query, Union, and Intersection and Except, Nested Queries, Correlated Nested Queries, Set-Comparison Operations, Aggregate Operators, Null Values and Embedded SQL, Dynamic SQL, ODBC and JDBC, Triggers and Active Databases.

Schema refinement and Normal forms: Introductions to Schema Refinement, Functional Dependencies, Boyce-Codd Normal Forms, Third Normal Form, Normalization-Decomposition into BCNF Decomposition into 3-NF.

Transaction Processing: Introduction-Transaction State, Transaction properties, Concurrent Executions. Need of Serializability, Conflict vs. View Serializability, Testing for Serializability, Recoverable Schedules, Cascadeless Schedules.

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.

Text/Reference Books:

1. H.f. Korth and Silberschatz: Database Systems Concepts, McGraw Hill
2. Almasri and S.B. Navathe: Fundamentals of Database Systems.
3. Ramakrishnan and Gehrke: Database Management System, McGraw Hill
4. C.J. Date: Data Base Design, Addison Wesley
5. Hansen and Hansen : DBM and Design, PHI

5CSU02 OPERATING SYSTEM

Maximum Marks(50+100)

OBJECTIVES:

1. To understand the structure and functions of OS
2. To learn about Processes, Threads and Scheduling algorithms
3. To understand the principles of concurrency and Deadlocks
4. To learn various memory management schemes
5. To study I/O management and File systems

Introduction and need of operating system, layered architecture/logical structure of operating system, Type of OS, operating system as resource manager and virtual machine, OS services, BIOS, System Calls/Monitor Calls, Firmware- BIOS, Boot Strap Loader.

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.

Interprocess communication- Introduction to message passing, Race condition, critical section problem, mutual exclusion with busy waiting- disabling interrupts, lock variables, strict alteration, 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.

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.

File System- concepts, naming, attributes, operations, types, structure, file organization & access(Sequential, Direct ,Index Sequential) methods, memory mapped files, directory structures one level, two level, hierarchical/tree, acyclic graph, general graph, file system mounting, file sharing, path name, directory operations, overview of file system 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.

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

Text/Reference Books:

1. A. Silberschatz and Peter B Galvin: Operating System Principals, 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.

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

Scheme Code - 2015CSE

5CSU03 THEORY OF COMPUTATION

Maximum Marks(50+100)

Objectives:

1. **Understand various computing models like Finite State Machine, Pushdown Automata and Turing machine.**
2. **Be aware of Decidability and undesirability of various problems.**
3. **Familiar with types of grammars.**

Syllabus:

Finite Automata & Regular Expression: Basic Concepts of finite state system, Deterministic and non-deterministic finite automation and designing regular expressions, relationship between regular expression & Finite automata minimization of finite automation mealy & Moore Machines.

Regular Sets of Regular Grammars: Basic Definition of Formal Language and Grammars. Regular Sets and Regular Grammars, closure proportion of regular sets, Pumping lemma for regular sets, decision Algorithms for regular sets.

Context Free Languages& Pushdown Automata: Context Free Grammars – Derivations and Languages – Relationship between derivation and derivation trees, ambiguity ,simplification of CEG – Greibach Normal form – Chomsky normal forms – Problems related to CNF and GNF Pushdown Automata: Definitions ,Moves ,Deterministic pushdown automata ,Pushdown automata and CFL , pumping lemma for CFL -Applications of pumping Lemma.

Turing Machines: Turing machines – Computable Languages and functions , Turing Machine constructions ,Storage in finite control , checking of symbols, subroutines ,two way infinite tape. Undecidability: Properties of recursive and Recursively enumerable languages, Universal Turing Machines as an undecidable problem , Universal Languages – Rice's Theorems.

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

Scheme Code - 2015CSE

Linear bounded Automata Context Sensitive Language: Chomsky Hierarchy of Languages and automata, Basic Definition & descriptions of Theory & Organization of Linear bounded Automata Properties of context-sensitive languages.

Text/Reference Books:

1. Hopcroft, Motwani and Ullman, "Introduction to Automata Theory, Formal Languages and Computation," Narosa.
2. Cohen, "Introduction to Computer Theory", Addison Wesley.
3. Papadimitriou, "Introduction to Theory of Computing", Prentice Hall.
4. John C.Martin "Introduction to Languages and The Theory of Computation", third edition, TMH.

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

Scheme Code - 2015CSE

5CSU04 Computer Architecture and Organization

Maximum Marks(50+100)

Objective: The objective of this course is to become familiar in how computer systems work and its basic principles, how to analyze the system performance, concepts behind advanced pipelining techniques, the current state of art in memory system design, how I/O devices are being accessed and its principles to provide the knowledge on instruction Level Parallelism.

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 Unit: General register organization, stack organization, Instruction formats, Data transfer and manipulation, program control. RISC, CISC characteristics.

Pipeline and Vector processing: Pipeline structure, speedup, efficiency, throughput and bottlenecks. Arithmetic pipeline and Instruction pipeline.

Computer Arithmetic: Adder, Ripple carry Adder, carry look Ahead Adder, Multiplication: Add and Shift, Array multiplier and Booth Multiplier, Division: restoring and Non-restoring Techniques. Floating Point Arithmetic: Floating point representation, Add, Subtract, Multiplication, Division.

Memory Organization: RAM, ROM, Memory Hierarchy, Organization, Associative memory, Cache memory, and Virtual memory: Paging and Segmentation.

Input-Output Organization: Input-Output Interface, Modes of Transfer, Priority Interrupt, DMA, IOP processor.

TEXT / REFERENCE BOOKS

1. Computer Organization and Architecture - William Stallings (Pearson Education Asia)
2. Computer Organization and Architecture -John P. Hayes (McGraw -Hill)
3. Computer Organization - V. Carl. Hamacher (McGraw-Hill)

5CSU05 Embedded System

Maximum Marks(50+100)

Objective: The objective of this course is to provide a theoretical & practical introduction to embedded computing, Processors, Platforms and their synthesis.

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

Embedded Computing Platform: CPU Bus – Bus protocols, DMA, system bus configurations, ARM bus; Timers and counters, A/D and D/A converters, Keyboards, LEDs, displays and touch screens; Design examples.

Embedded Software Analysis and Design: Software design pattern for Embedded Systems; Model programs – data flow graphs and control/data flow graphs; Assembly and linking; Compilation techniques; Analysis and optimization of execution time, energy, power and program size.

Embedded System Accelerators: Processor accelerators, accelerated system design

Recommended Book:

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

3. Embedded System Design by Steve Heath published by Elsevier Inc

4. Embedded System design: A unified hardware/software Introduction by Frank Vahid & Tony Givagi published by John Wiley & Sons Inc.

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

Scheme Code - 2015CSE

5CSU06.1 Information theory and Coding

Maximum Marks(50+100)

Objective : The objective of this course is to provide knowledge about the properties of codes and their fitness for a specific application that might be useful for the purpose of designing efficient and reliable data transmission methods.

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.

Text/References Book

1. Digital Communication, Simon Haykin

5CSU06.2 Human Computer Interface

Maximum Marks(50+100)

Objective : The objective of this course is to give an introduction to the key areas, approaches and developments in the field. The main objective is to get student to think constructively and analytically about how to design and evaluate interactive technologies. Basically, the course will introduce them to key areas, theoretical frameworks, approaches and major developments in HCI.

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.

Text/References:

1. Human Computer Interaction; Alan Dix et.al, 3rd ed., Pearson

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

Scheme Code - 2015CSE

(39)

5CSU06.3 Software Testing and Project Management

Maximum Marks(50+100)

Objective:

To introduce software project management and to describe its distinctive characteristics

- To discuss project planning and the planning process
- To show how graphical schedule representations are used by project management
- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

Syllabus:

Methods of verification, validation, level of validation, principle of testing, static testing, structural testing, Regression Testing ,Integration Testing, Scenario testing, Defect bash. Internationalization testing, localization testing, ad-hoc testing-overview, buddy testing, pair testing, iterative testing agile and extreme testing, Usability and Accessibility Testing. Test Planning, Test Management, Test reporting, Test Project matrices

The management spectrum, 4Ps, W5HHprinciple, critical practices, Metrics in the process and project Domains, software measurements, metrics for software quality

Software project objective, Software scope, resources, software project estimation, Decomposition techniques, empirical estimation models, estimation for Agile development and web engineering projects, the make/buy decision.
Project Scheduling, Relationship between people and effort, defining a task set and task network, scheduling, earned value analysis

Project Execution and Closure: The Review Process, Planning, Group Review Meeting, Rework and Follow-up, Guidelines for Reviews in Projects, Data Collection, Analysis and Control Guidelines, NAH Syndrome.

Project Monitoring and Control: Project Tracking, Activities Tracking, Defect Tracking, Issues Tracking, Status Reports, Milestone Analysis, Actual Versus Estimated Analysis of Effort and Schedule, Monitoring Quality, Risk-Related Monitoring.

Text Books/References:

1. Software Engineering By Roger S. Pressman, TMH
2. Software Engineering Fundamental By Ali Behforooz, Frederick J Hudson, Oxford University Press
3. Software Engineering By Ian Sommerville
4. Software Engineering Concepts By Richard E. Fairley (Mcgraw-Hill)

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

Scheme Code - 2015CSE

137

5CSU07 Database Management Lab

Maximum Marks(50+25)

Objectives: At the end of the semester, the students should have clearly understood and implemented the following:

1. Stating a database design & application problem.
2. Preparing ER diagram
3. Finding the data fields to be used in the database.
4. Selecting fields for keys.
5. Normalizing the database including analysis of functional dependencies.
6. Installing and configuring the database server and the front end tools.
7. Designing database and writing applications for manipulation of data for a standalone and shared data base including concepts like concurrency control, transaction roll back, logging, report generation etc.
8. Get acquainted with SQL. In order to achieve the above objectives, it is expected that each students will chose one problem. The implementation shall being with the statement of the objectives to be achieved, preparing ER diagram, designing of database, normalization and finally manipulation of the database including generation of reports, views etc. The problem may first be implemented for a standalone system to be used by a single user. All the above steps may then be followed for development of a database application to be used by multiple users in a client server environment with access control. The application shall NOT use web techniques. One exercise may be assigned on creation of table, manipulation of data and report generation using SQL.

Suggested Tool:

For standalone environment, Visual FoxPro or any similar database having both the database and manipulation language may be used. For multi-user application, MYSql is suggested. However, any other database may also be used.

For front end, VB.Net, Java, VB Script or any other convenient but currently used by industry may be chosen. Indicative List of exercise:

1. Student information system for your college.

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

Scheme Code - 2015CSE

(36)

2. Student grievance registration and redressal system.
3. A video library management system for a shop.
4. Inventory management system for a hardware/ sanitary item shop.
5. Inventory management system for your college.
6. Guarantee management system for the equipments in your college.

(34)

5CSU08 Java Programming Lab

Maximum Marks(50+25)

Objectives: At the end of the semester, the students should have clearly understood and implemented the following:

1. Develop an in depth understanding of programming in Java: data types, variables, operators, operator precedence, Decision and control statements, arrays, switch statement, Iteration Statements, Jump Statements, Using break, Using continue, return.
2. Write Object Oriented programs in Java: Objects, Classes constructors, returning and passing objects as parameter, Inheritance, Access Control, Using super, final with inheritance Overloading and overriding methods, Abstract classes, Extended classes.
3. Develop understanding to developing packages & Interfaces in Java: Package, concept of CLASSPATH, access modifiers, importing package, Defining and implementing interfaces.
4. Develop understanding to developing Strings and exception handling: String constructors, special string operations, character extraction, searching and comparing strings, string Buffer class. Exception handling fundamentals, Exception types, uncaught exceptions, try, catch and multiple catch statements. Usage of throw, throws and finally.
5. Develop applications involving file handling: I/O streams, File I/O.
6. Develop applications involving concurrency: Processes and Threads, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Joins, and Synchronization.
7. Develop applications involving Applet: Applet Fundamentals, using paint method and drawing polygons.

It is expected that each laboratory assignments to given to the students with an aim to in order to achieve the above objectives

Indicative List of exercises:

1. Programs to demonstrate basic concepts e.g. operators, classes, constructors, control & iteration statements, recursion etc. such as complex arithmetic, matrix arithmetic, tower of Hanoi problem etc.

32

2. Development of programs/projects to demonstrate concepts like inheritance, exception handling, packages, interfaces etc. such as application for electricity department, library management, ticket reservation system, payroll system etc.
3. Development of a project to demonstrate various file handling concepts.
4. Development of a project to demonstrate various applet concepts.

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

Scheme Code - 2015CSE

30

5CSU09 Embedded System Lab

Maximum Marks (50+25)

Objectives:
Upon successful completion of the course, students will be able to design simple embedded systems and develop related software. Students also learn to work in a team environment and communicate the results as written reports and oral presentations.

Suggested Microcontroller Platform: Texas Instruments MSP430, ARM 9, 68HC12, 8051.

It is assumed that there are 14 weeks in the semester and about 5 to 6 experiments will be carried out. More experiments are provided to bring in variation.

Experiment #0
Get familiar with the microcontroller kit and the development software. Try the sample programs that are supplied to get familiar with the Microcontroller.

Experiment #1
a) Blink an LED which is connected to your microcontroller using the built-in timer in the microcontroller. Assume that the LED should be on for x milliseconds and off for y milliseconds; assume that these values are stored in memory locations X and Y. We should be able to change the value of x and y and rerun the program.
b) Consider an alternate way to program this application. Here, the microcontroller turns the LED on and waits in a busy loop to implement a delay of x milliseconds. Then it turns the LED off and waits in a busy loop to implement a delay of y milliseconds. How do you compare these two solutions?

Experiment #2
Assume that in Experiment #1, the values of x and y have been chosen to be 200 and 500 respectively. When the LED blinking program runs, pressing a key on the keyboard should generate an interrupt to the microcontroller. If the key that has been pressed is a numeric key, the value of x and y must be interchanged by the interrupt service routine. If the key that has been pressed is not a numeric key, then the LED must be turned off for 2 seconds before resuming the blinking.

Experiment #3
If your microcontroller kit has an LCD interface, write a program to display a character string on the LCD. Assume that the string is stored at a location STRING and consists of alphanumeric characters. The string is nullterminated. Modify your program to scroll the displayed string from left to right.

Experiment #4

Modern microcontrollers usually have an in-built Digital-to-Analog and Analogto- Digital converter. Use the built-in DAC to generate voltage waveforms such as (a) pulse train (b) triangular waveform (c) sinusoidal waveform. Observe these waveforms on an oscilloscope.

Experiment #5

Your microcontroller may have a built-in temperature sensor. If not, interface an external temperature sensor to the microcontroller. Write a program to take several measurements of temperature at regular intervals and display the average temperature on the LCD display. Test if the readings change when the ambient temperature changes.

Experiment #6

Your microcontroller may have a built-in ADC. Build a voltmeter that can measure stable voltages in a certain range. The measured value must be displayed on the LCD display. Measure the same voltage using a multimeter and record the error in measurement. Tabulate the error for several values of the voltage.

Experiment #7

Build a simple security device based on the microcontroller kit. Interface an external motion sensor to the microcontroller. An alarm must be generated if motion is sensed in a specified region. There must be a provision to record the time at which the intrusion was detected. Similarly, there must be a provision to turn the alarm off by pressing a key.

Experiment #8

A voltage waveform v(t) is available as an input to the microcontroller. We must continuously check the waveform and record the maximum value of the waveform and display the maximum value on the LCD display. Test the program by using a DC supply to generate v(t) and varying the DC value.

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

Scheme Code - 2015CSE

(20)

5CSU10 CASE Lab

Maximum Marks(50+25)

Objectives:

The students shall be able to use following modules of UML for system description, implementation and finally for product development

- Capture a business process model.

- The User Interaction or Use Case Model - describes the boundary and interaction between the system and users. Corresponds in some respects to a requirements model.

- The Interaction or Communication Model - describes how objects in the system will interact with each other to get work done.

- The State or Dynamic Model - State charts describe the states or conditions that classes assume over time. Activity graphs describe the workflows the system will implement.

- The Logical or Class Model - describes the classes and objects that will make up the system.

- The Physical Component Model - describes the software (and sometimes hardware components) that make up the system.

- The Physical Deployment Model - describes the physical architecture and the deployment of components on that hardware architecture.

The students are expected to use the UML models, prepare necessary documents using UML and implement a system. Some hardware products like digital clock, digital camera, washing machine controller, air conditioner controller, an elctronic fan regulator, an elementary mobile phone etc. may also be chosen.

The students shall be assigned one problem on software based systems and another involving software as well as hardware.

SYLLABUS

Semester	**VI**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2017-18**

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

Scheme Code - 2015CSE

6CSU01 Computer Graphics

MAX_MARKS(50+100)

Objectives:

This course provides an introduction to the principles of computer graphics. In particular, the course will consider methods for modeling 3-dimensional objects and efficiently generating photorealistic renderings on color raster graphics devices. The emphasis of the course will be placed on understanding how the various elements that underlie computer graphics (algebra, geometry, algorithms and data structures, optics, and photometry) interact in the design of graphics software systems

Syllabus:

Introduction to Raster scan displays, Storage tube displays, refreshing, flicking, interlacing, color monitors, display processors, resolution, Introduction to Interactive. Computer Graphics: Picture analysis, Overview of programmer's model of interactive graphics, Fundamental problems in geometry. Scan Conversion: point, line, circle, ellipse polygon, Aliasing, and introduction to Anti Aliasing (No anti aliasing algorithm).

2D & 3D Co-ordinate system: Homogeneous Co-ordinates, Translation, Rotation, Scaling, Reflection, Inverse transformation, Composite transformation. Polygon Representation, Flood Filling, Boundary filling. Point Clipping, Cohen-Sutherland Line Clipping Algorithm, Polygon Clipping algorithms.

Hidden Lines & Surfaces: Image and Object space, Depth Buffer Methods, Hidden Facets removal, Scan line algorithm, Area based algorithms. 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. J. Foley, A. Van Dam, S. Feiner, J. Hughes: Computer Graphics- Principles and Practice, Pearson
2. Hearn and Baker: Computer Graphics, PHI

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

Scheme Code - 2015CSE

6CSU02 Information security system

MAX_MARKS(50+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 ciphers principals, Shannon's theory of confusion and diffusion, fiestal structure, data encryption standard(DES), differential and linear cryptanalysis of DES, block cipher modes of operations, triple DES.

AES, RC6, random number generation. S-box theory: Boolean Function, S-box design criteria, Bent functions, Propagation and nonlinearity, construction of balanced functions, S-box design.

Public Key Cryptosystems: Principles of Public Key Cryptosystems, RSA Algorithm, security analysis of RSA, Exponentiation in Modular Arithmetic. Key Management in Public Key Cryptosystems: Distribution of Public Keys, Distribution of Secret keys using Public Key Cryptosystems. X.509 Discrete Logarithms, Diffie-Hellman Key Exchange.

Message Authentication and Hash Function: Authentication requirements, authentication functions, message authentication code, hash functions, birthday attacks, security of hash functions and MAC, MD5 message digest algorithm, Secure hash algorithm(SHA). Digital Signatures: Digital Signatures, authentication protocols, digital signature standards (DSS), proof of digital signature algorithm. Remote user Authentication using symmetric and Asymmetric Authentication

Pretty Good Privacy. IP Security: Overview, IP Security Architecture, Authentication Header, Encapsulation Security Payload in Transport and Tunnel mode with multiple security associations (Key Management not Included). Strong Password Protocols: Lamport's Hash, Encrypted Key Exchange.

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

Scheme Code - 2015CSE

TEXT BOOKS

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, 2nd Ed., PHI/Pearson.
3. Atul Kahate, Cryptography & Network Security, TMH, 2nd Ed.

REFERENCE BOOKS

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

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

Scheme Code - 2015CSE

6CSU03 COMPILER CONSTRUCTION

MAX_MARKS(50+100)

Objectives:

The aim of this module is to show how to apply the theory of language translation introduced in the prerequisite courses to build compilers and interpreters. It covers the building of translators both from scratch and using compiler generators. In the process, the module also identifies and explores the main issues of the design of translators. The construction of a compiler/interpreter for a small language is a necessary component of this module, so students can obtain the necessary skills

Syllabus:

Compiler, Translator, Interpreter definition, Phase of compiler introduction to one pass & Multipass compilers, Bootstrapping, Review of Finite automata lexical analyzer, Input, buffering, Recognition of tokens, Idea about LEX: A lexical analyzer generator, Error handling.

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

Syntax directed definitions; Construction of syntax trees, L-attributed definitions, Top down translation. Specification of a type checker, Intermediate code forms using postfix notation and three address code, Representing TAC using triples and quadruples, Translation of assignment statement. Boolean e xpression 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 design of code generator, A simple code generator, Code generation from DAG.

TEXT BOOKS

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

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

Scheme Code - 2015CSE

6CSU04 DATA MINING AND WAREHOUSE

MAX_MARKS(50+100)

Objectives:

1. To introduce the basic concepts of Data Warehouse and Data Mining techniques.
2. Examine the types of the data to be mined and apply preprocessing methods on raw data.
3. Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.

Syllabus:

Overview, Motivation(for Data Mining),Data Mining-Definition & Functionalities, Data Processing, Form of Data Preprocessing, Data Cleaning: Missing Values, Noisy Data, (Binning, Clustering, Regression, Computer and Human inspection), Inconsistent Data, Data Integration and Transformation. Data Reduction:-Data Cube Aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Clustering, Discretization and Concept hierarchy generation.

Concept Description: Definition, Data Generalization, Analytical Characterization, Analysis of attribute relevance, Mining Class comparisons, Statistical measures in large Databases. Measuring Central Tendency, Measuring Dispersion of Data, Graph Displays of Basic Statistical class Description, Mining Association Rules in Large Databases, Association rule mining, mining Single-Dimensional Boolean Association rules from Transactional Databases– Apriori Algorithm, Mining Multilevel Association rules from Transaction Databases and Mining Multi-Dimensional Association rules from Relational Databases.

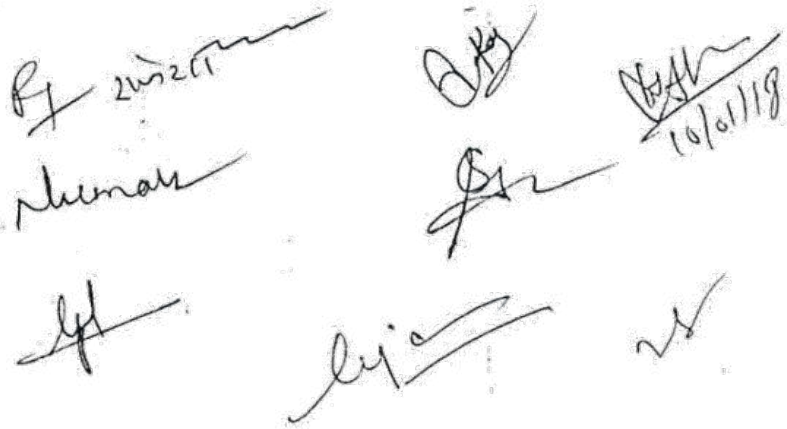
What is Classification & Prediction, Issues regarding Classification and prediction, Decision tree, Bayesian Classification, Classification by Back propagation, Multilayer feed-forward Neural Network, Back propagation Algorithm, Classification methods K-nearest neighbour classifiers, Genetic Algorithm. Cluster Analysis: Data types in cluster analysis, Categories of clustering methods, Partitioning methods. Hierarchical Clustering- CURE and Chameleon. Density Based Methods-DBSCAN, OPTICS. Grid Based Methods- STING, CLIQUE. Model Based Method –Statistical Approach, Neural Network approach, Outlier Analysis.

Data Warehousing: Overview, Definition, Delivery Process, Difference between Database System and Data Warehouse, Multi Dimensional Data Model, Data Cubes, Stars, Snow Flakes, Fact Constellations, Concept hierarchy, Process Architecture, 3 Tier Architecture, Data Mining.

Aggregation, Historical information, Query Facility, OLAP function and Tools. OLAP Servers, ROLAP, MOLAP, HOLAP, Data Mining interface, Security, Backup and Recovery, Tuning Data Warehouse, Testing Data Warehouse.

TEXT BOOKS

1. Data Warehousing in the Real World – Anahory and Murray, Pearson Education.
2. Data Mining – Concepts and Techniques – Jiawai Han and Micheline Kamber.
3. Building the Data Warehouse – WH Inmon, Wiley.



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

Scheme Code - 2015CSE

6CSU05 MOBILE COMPUTING

MAX_MARKS(50+100)

Objective : This course will give you an understanding of mobile computer systems particularly in the context of wireless network systems such as 2G/3G/4G mobile telephony, data networks, and other wireless networks and infrastructure. The course emphasises how to interface hardware to mobile computing devices, and programming those devices..

Syllabus:

Mobile computing: Definitions, adaptability issues (transparency, Environmental Constraints, application aware adaptation), mechanisms for adaptation and incorporating adaptations. Mobility management: mobility management, location management principle and techniques, PCS location management Scheme.

Data dissemination and management: challenges, Data dissemination, bandwidth allocation for publishing, broadcast disk scheduling, mobile cache maintenance schemes, Mobile Web Caching. Introduction to mobile middleware.

Middleware for application development: adaptation, Mobile agents. Service Discovery Middleware: Service Discovery & standardization Methods (universally Unique Identifiers, Textual Description & using interfaces), unicast Discovery, Multicast Discovery & advertisement, service catalogs, Garbage Collection, Eventing.

Mobile IP, Mobile TCP, Database systems in mobile environments, World Wide Web and mobility

Ad Hoc networks, localization, MAC issues, Routing protocols, global state routing (GSR), Destination sequenced distance vector routing (DSDV), Dynamic source routing (DSR), Ad Hoc on demand distance vector routing (AODV), Temporary ordered routing algorithm (TORA), QoS in Ad Hoc Networks, applications.

TEXT BOOKS:

1. Frank Adelstein, Sandeep Gupta, Golden Richard III, Loren Schwiebert, Fundamentals of Mobile and Pervasive Computing, TMH.
2. Principles of mobile computing Hansmann & Merk., Springer
3. Mobile communications Jochen Schiller , Pearson

REFERENCE BOOKS:

1. 802.11 wireless networks Matthew S.Gast, O'REILLY.
2. Wireless LANs: Davis & McGuffin, McGraw Hill
3. Mobile Communications Handbook by Jerry D. Gybson

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

Scheme Code - 2015CSE

6CSU06.1 ARTIFICIAL INTELLIGENCE

(25)

MAX_MARKS(50+100)

Objective: The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches. Develop a basic understanding of the building blocks ofAI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning.

Syllabus:

Meaning and definition of artificial intelligence, Various types of production systems, Characteristics of production systems, Study and comparison of breadth first search and depth first search. Techniques, other Search Techniques like hill Climbing, Best first Search. A* algorithm, AO* algorithms etc, and various types of control strategies.

Knowledge Representation, Problems in representing knowledge, knowledge representation using propositional and predicate logic, comparison of propositional and predicate logic, Resolution, refutation, deduction, theorem proving, inferencing, monotonic and nonmonotonic reasoning.

Probabilistic reasoning, Baye's theorem, semantic networks scripts schemas, frames, conceptual dependency and fuzzy logic, forward and backward reasoning.

Game playing techniques like minimax procedure, alpha-beta cut-offs etc, planning, Study of the block world problem in robotics, Introduction to understanding and natural languages processing.

Introduction to learning, Various techniques used in learning, introduction to neural networks, applications of neural networks, common sense, reasoning, some example of expert systems.

TEXT BOOKS:

1. Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
2. Introduction to AI & Expert System: Dan W. Patterson, PHI.
3. Artificial Intelligence by Luger (Pearson Education)
4. Russel & Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall

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

Scheme Code - 2015CSE

6CSU06.2 BIOINFORMATICES

MAX_MARKS(50+100)

Objectives:

The objective is to give students an introduction to the basic practical techniques of bioinformatics. Emphasis will be given to the application of bioinformatics and 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.

REFRENCES:

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.
4. Setubal, J. and Meidanis, J. 1996 Introduction to Computational Molecular Biology. PWS Publishing Co., Boston.
5. Lesk, A.M. 2005, 2nd edition, Introduction to Bioinformatics. Oxford University Press. 6. Fogel, G.B. and Corne, D.W., Evolutionary Computation in Bioinformatics.
6. 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.

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

Scheme Code - 2015CSE

6CSU06.3 FUZZY LOGIC AND APPLICATIONS

MAX_MARKS(50+100)

Objectives:

The objective of this course is to teach the students the need of fuzzy sets, arithmetic operations on fuzzy sets, fuzzy relations, possibility theory, fuzzy logic, and its applications

Pre-requisites: Basic knowledge of algebra and analysis at the level of undergraduate studies

Syllabus:

Classical sets vs Fuzzy Sets , Need for fuzzy sets , Definition and Mathematical representations , Level Sets , Fuzzy functions , Zadeh's Extension Principle

Operations on [0,1] – Fuzzy negation, triangular norms, t-conorms, fuzzy implications, Aggregation Operations, Fuzzy Functional Equations

Fuzzy Binary and n-ary relations – composition of fuzzy relations – Fuzzy Equivalence Relations – Fuzzy Compatibility Relations – Fuzzy Relational Equations

Fuzzy Measures – Evidence Theory – Necessity and Belief Measures – Probability Measures vs Possibility Measures

Fuzzy Decision Making - Fuzzy Relational Inference – Compositional Rule of Inference - Efficiency of Inference - Hierarchical

Fuzzy If-Then Rule Base – Inference Engine – Takagi-Sugeno Fuzzy Systems - Function Approximation

References:

1. George J Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic : Theory and Applications, Prentice Hall NJ,1995.
2. H.J. Zimmermann, Fuzzy Set Theory and its Applications, Allied Publishers, New Delhi, 1991.
3. Kevin M Passino and Stephen Yurkovich, Fuzzy Control, Addison Wesley Longman, 1998.
4. Michal Baczynski and Balasubramaniam Jayaram, Fuzzy Implications, Springer Verlag, Heidelberg, 2008.

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

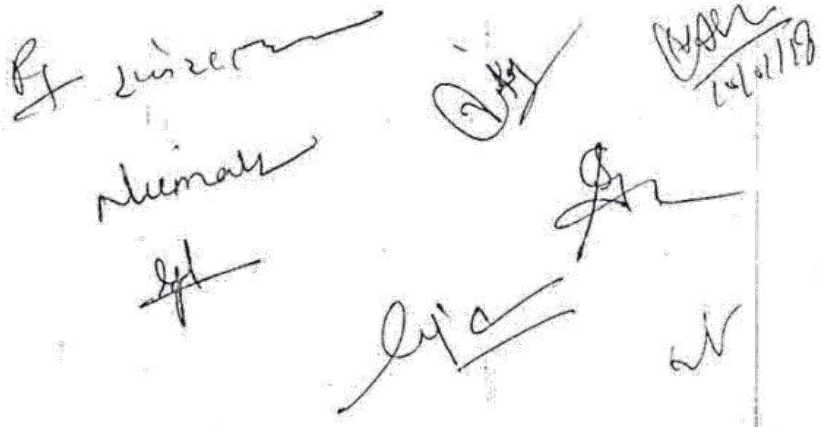
Scheme Code - 2015CSE

6CSU07 Computer graphics & multimedia lab

MAX_MARKS(50+25)

List of Experiment

1 Implementation of Line, Circle and ellipse attributes.
2 Two Dimensional transformations - Translation, Rotation, Scaling, Reflection, Shear.
3 Composite 2D Transformations.
4 Cohen Sutherland 2D line clipping and Windowing.
5 Sutherland – Hodgeman Polygon clipping Algorithm.
6 Three dimensional transformations - Translation, Rotation, Scaling.
7 Composite 3D transformations.
8 Drawing three dimensional objects and Scenes.
9 Generating Fractal images.
10 To plot a point (pixel) on the screen.
11 To draw a straight line using DDA Algorithm.
12 Implementation of mid-point circle generating Algorithm.
13 Implementation of ellipse generating Algorithm.
14 To translate an object with translation parameters in X and Y directions.
15 To scale an object with scaling factors along X and Y directions.
16 To rotate an object with a certain angle about origin.
17 Perform the rotation of an object with certain angle about an arbitrary point.



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

Scheme Code - 2015CSE

6CSU08 Web Programming lab

MAX_MARKS(50+25)

List of Experiment

1 . Creation of HTML Files.
2. Working with Client Side Scripting : VBScript, JavaScript.
3. Configuration of web servers: Apache Web Server, Internet Information Server (IIS).
4. Working with ActiveX Controls in web documents.
5. Experiments in Java Server Pages: Implementing MVC Architecture using Servlets, Data Access. Programming (using ADO), Session and Application objects, File System Management
6. Working with other Server Side Scripting: Active Server Pages, Java Servlets, PHP
7. Experiments in Ajax Programming
8. Developing Web Services
9. Developing any E-commerce application (Mini Project)
10. Application Development in cloud computing Environment
11. Experiment Using Open Source Tool e.g. ANEKA

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

Scheme Code - 2015CSE

6CSU09 Compiler Construction lab

MAX_MARKS(50+25)

Objectives:

At the end of the semester, the students should have clearly understood and implemented the following:

1. Develop an in depth understanding of system programming concept. Lexical analysis, syntax analysis, semantics analysis, code optimization, code generation. Language specification and processing.
2. Develop an Understanding of Scanning by using concept of Finite state automaton. Parse tree and syntax tree, Top down parsing (recursive decent parsing, LL (1) parser) Bottom up parsing (operator precedence parsing) .Managing symbol table, opcode table, literal table, pool table.
3. Develop an Understanding of Intermediate code form: Three address code, Polish notation (Postfix strings)
4. Develop an Understanding of Allocation data structure. Heaps.
5. Develop an Understanding about Language processor development tools: LEX, YACC. Language processing activities (Program generation and execution)

It is expected that each laboratory assignments to given to the students with an aim to In order to achieve the above objectives

Indicative List of exercises:

1. Write grammar for a fictitious language and create a lexical analyzer for the same.
2. Develop a lexical analyzer to recognize a few patterns in PASCAL and C (ex: identifiers, constants, comments, operators etc.)
3. Write a program to parse using Brute force technique of Top down parsing
4. Develop on LL (1) parser (Construct parse table also).
5. Develop an operator precedence parser (Construct parse table also)
6. Develop a recursive descent parser
7. Write a program for generating for various intermediate code forms
   i) Three address code ii) Polish notation
8. Write a program to simulate Heap storage allocation strategy
9. Generate Lexical analyzer using LEX
10. Generate YACC specification for a few syntactic categories
11. Given any intermediate code form implement code optimization techniques

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

Scheme Code - 2015CSE

6CSU10 Advance Java lab

MAX_MARKS(50+25)

List of Experiment

1. Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, and 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.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. Develop an applet that displays a simple message.
4. Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
5. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.
6. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
7. Write a Java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.
8. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an 63-69 6 appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially there is no message shown.
9. Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea() that prints the area of the given shape.
10. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are

separated by commas. Write a java program to display the table using Labels in Grid Layout.

11. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
12. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
13. Implement the above program with database instead of a text file.
14. Write a Java program that takes tab separated data (one record per line) from a text file and inserts them into a database.
15. Write a Java program that prints the meta-data of a given table

SYLLABUS

Semester	**VII**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2018-19**

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

Scheme Code - 2015CSE

7CSU01 CLOUD COMPUTING

MAX_MARKS(50+100)

Objectives:

The student will learn about the cloud environment, building software systems and components that scale to millions of users in modern internet, cloud concepts capabilities across the various cloud service models including Iaas,Paas,Saas, and developing cloud based software applications on top of cloud platforms.

Syllabus:

Introduction Cloud Computing: Nutshell of cloud computing, Enabling Technology, Historical development, Vision, feature Characteristics and components of Cloud Computing. Challenges, Risks and Approaches of Migration into Cloud. Ethical Issue in Cloud Computing, Evaluating the Cloud's Business Impact and economics, Future of the cloud. Networking Support for Cloud Computing. Ubiquitous Cloud and the Internet of Things

Cloud Computing Architecture: Cloud Reference Model, Layer and Types of Clouds, Services models, Data center Design and interconnection Network, Architectural design of Compute and Storage Clouds. Cloud Programming and Software: Fractures of cloud programming, Parallel and distributed programming paradigms-MapReduce, Hadoop , High level Language for Cloud. Programming of Google App engine.

Virtualization Technology: Definition, Understanding and Benefits of Virtualization. Implementation Level of Virtualization, Virtualization Structure/Tools and Mechanisms , Hypervisor VMware, KVM, Xen. Virtualization: of CPU, Memory, I/O Devices, Virtual Cluster and Resources Management, Virtualization of Server , Desktop, Network, and Virtualization of data-center

Securing the Cloud : Cloud Information security fundamentals, Cloud security services, Design principles, Policy Implementation, Cloud Computing Security Challenges, Cloud Computing Security Architecture . Legal issues in cloud Computing. Data Security in Cloud: Business Continuity and Disaster Recovery , Risk Mitigation , Understanding and Identification of Threats in Cloud, SLA-Service Level Agreements, Trust Management

Cloud Platforms in Industry: Amazon web services , Google AppEngine, Microsoft Azure Design, Aneka: Cloud Application Platform -Integration of Private and Public Clouds Cloud applications: Protein structure prediction, Data Analysis, Satellite Image Processing, CRM and ERP ,Social networking . Cloud Application- Scientific Application, Business Application. Advance Topic in Cloud Computing: Federated Cloud/InterCloud, Third Party Cloud Services

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

Scheme Code - 2015CSE

TEXT BOOKS:

1. " Distributed and Cloud Computing " By Kai Hawang , Geofrey C.Fox, Jack J. Dongarra Pub: Elservier

2. Cloud Computing ,Principal and Paradigms, Edited By Rajkumar Buyya, Jemes Broberg, A. Goscinski, Pub.- Wiley

3. Kumar Saurabh, "Cloud Computing" , Wiley Pub

4. Krutz , Vines, "Cloud Security " , Wiley Pub 5. Velte, "Cloud Computing- A Practical Approach" ,TMH Pub .

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

Scheme Code - 2015CSE

7CSU02 REAL TIME SYSTEM

MAX_MARKS(50+100)

Objectives:

To introduce students to the fundamental problems, concepts, and approaches in the design and analysis of real-time systems. To study issues related to the design and analysis of systems with real-time constraints.

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. Introduction to scheduling of flexible computations –flexible applications, imprecise computation model and firm deadline model.

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 MultipleUnit Resources, Controlling Concurrent Accesses to Data Objects

TEXT BOOKS

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

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

Scheme Code - 2015CSE

7CSU03 SOFT COMPUTING

MAX_MARKS(50+100)

Objectives:

At the end of this course the student should be able to understand how to use Soft Computing techniques and programming language to design a small intelligent system for a specific application. Students can learn how to write a high-quality review, conference paper with theoretical investigation

Syllabus:

Soft Computing: Introduction, requirement, different tools and techniques, usefulness and applications.
Fuzzy Sets And Fuzzy Logic: Introduction, Fuzzy sets versus crisp sets, operations on fuzzy sets, Extension principle, Fuzzy relations and relation equations, Fuzzy numbers, Linguistic variables, Fuzzy logic, Linguistic hedges, Applications, fuzzy controllers, fuzzy pattern recognition, fuzzy image processing, fuzzy database.

Artificial Neural Network: Introduction, basic models, Hebb's learning, Adaline, Perceptron, Multilayer feed forward network, Back propagation, Different issues regarding convergence of Multilayer Perceptron, Competitive learning, Self-Organizing Feature Maps, Adaptive Resonance Theory, Associative Memories, Applications.

Evolutionary and Stochastic techniques: Genetic Algorithm (GA), different operators of GA, analysis of selection operations, Hypothesis of building blocks, Schema theorem and convergence of Genetic Algorithm, Simulated annealing and Stochastic models, Boltzmann Machine, Applications.

Rough Set: Introduction, Imprecise Categories Approximations and Rough Sets, Reduction of Knowledge, Decision Tables, and Applications.
Hybrid Systems: Neural-Network-Based Fuzzy Systems, Fuzzy Logic-Based Neural Networks, Genetic Algorithm for Neural Network Design and Learning, Fuzzy Logic and Genetic Algorithm for Optimization ,Applications.

REFRENCE BOOKS:

1. Fuzzy sets and Fuzzy logic by George Klir , Bo Y uan, PHI
2. Neural Networks, Fuzzy logic and Genetic Algorithms, Synthesis and applications by S. Rajsekharan, Vijayalaxmi Pai
3. Intelligent Hybrid Systems, D. Ruan, Kluwer Academic Publisher, 1997
4. Neural Fuzzy Systems, Chin-Teng Lin & C. S. George Lee, Prentice Hall PTR.
5. Neural Networks, S. Haykin, Pearson Education, 2ed, 2001.
6. Genetic Algorithms in Search and Optimization, and Machine Learning, D. E. Goldberg, AddisonWesley, 1989.
7. Learning and Soft Computing, V. Kecman, MIT Press, 2001.

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

Scheme Code - 2015CSE

7CSU04 DIGITAL IMAGE PROCESSING

MAX_MARKS(50+100)

Objectives:

To learn and understand the fundamentals of digital image processing, and various image Transforms, Image Enhancement Techniques, Image restoration Techniques and methods, image compression and Segmentation used in digital image processing.

Syllabus:

Introduction to Image Processing: Digital Image representation, Sampling & Quantization, Steps in image Processing, Image acquisition, color image representation

Image Transformation & Filtering: Intensity transform functions, histogram processing, Spatial filtering, Fourier transforms and its properties, frequency domain filters, colour models, Pseudo colouring, colour transforms, Basics of Wavelet Transforms

Image Restoration: Image degradation and restoration process, Noise Models, Noise Filters, degradation function, Inverse Filtering, Homomorphism Filtering

Image Compression: Coding redundancy, Interpixel redundancy, Psychovisual redundancy, Huffman Coding, Arithmetic coding, Lossy compression techniques, JPEG Compression

Image Segmentation & Representation: Point, Line and Edge Detection, Thresholding, Edge and Boundary linking, Hough transforms, Region Based Segmentation, Boundary representation, Boundary Descriptors, Regional

TEXT BOOKS:

Pakhera Malay K: Digital Image Processing and Pattern Recogination, PHI.

REFRENCE BOOKS:

1. Gonzalez and Woods: Digital Image Processing ISDN 0-201-600- 781, Addison Wesley 1992. Boyle and Thomas: Computer Vision - A First Gurse 2nd Edition, ISBN 0-632-028-67X, Blackwell Science 1995.
2. Gonzalez and Woods: Digital Image Processing ISDN 0-201-600- 781, Addison Wesley 1992

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

Scheme Code - 2015CSE

7CSU05 DISTRIBUTED SYSTEMS

MAX_MARKS(50+100)

Objective : The course aims to provide an understanding of the principles on which the Internet and other distributed systems are based; their architecture, algorithms and how they meet the demands of contemporary distributed applications.

Syllabus:

Distributed Systems: Features of distributed systems, nodes of a distributed system, Distributed computation paradigms, Model of distributed systems, Types of Operating systems: Centralized 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).[1] 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.[1] Distributed File Systems: Transparencies and Characteristics of DFS, DFS Design and implementation, Transaction Service and Concurrency Control, Data and File Replication.[1,2] Case studies: Sun network file systems, General Parallel file System and Window's file systems. Andrew and Coda File Systems [2,3]

Distributed Shared Memory: Non-Uniform Memory Access Architectures, Memory Consistency Models, Multiprocessor Cache Systems, Distributed Shared Memory, Implementation of DSM systems.[1] Models of Distributed Computation: Preliminaries, Causality, Distributed Snapshots, Modeling a Distributed Computation, Failures in a Distributed System, Distributed Mutual Exclusion, Election, Distributed Deadlock handling, Distributed termination detection.

Distributed Agreement: Concept of Faults, failure and recovery, Byzantine Faults, Adversaries, Byzantine Agreement, Impossibility of Consensus and Randomized Distributed Agreement.[1] Replicated Data Management: concepts and issues, Database Techniques, Atomic Multicast, and Update Propagation.[1] CORBA case study: Introduction, Architecture, CORBA RMI, CORBA Services.

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

Scheme Code - 2015CSE

TEXT 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

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

Scheme Code - 2015CSE

7CSU06.1 ROBOTICS

MAX_MARKS(50+100)

Objective:

At the end of this course the student should be able to understand

1. The basics of robot
2. End effectors and robot controls
3. Robot Transformations and Sensors
4. Robot cell design and applications
5. Micro/Nano robotic systems

Syllabus:

INTRODUCTION: Robot anatomy-Definition, law of robotics, History and Terminology of Robotics-Accuracy and repeatability of Robotics-Simple problems Specifications of Robot-Speed of Robot-Robot joints and links-Robot classifications-Architecture of robotic systems-Robot Drive systems Hydraulic, Pneumatic and Electric system.

END EFFECTORS AND ROBOT CONTROLS : Mechanical grippers-Slider crank mechanism, Screw type, Rotary actuators, cam type-Magnetic grippers-Vacuum grippers-Air operated grippers-Gripper force analysis-Gripper design-Simple problems-Robot controls-Point to point control, Continuous path control, Intelligent robot-Control system for robot joint-Control actions-Feedback devices-Encoder, Resolver, LVDT-Motion Interpolations-Adaptive control.

ROBOT TRANSFORMATIONS AND SENSORS: Robot kinematics-Types- 2D, 3D Transformation-Scaling, Rotation, Translation- Homogeneous coordinates, multiple transformation-Simple problems. Sensors in robot – Touch sensors-Tactile sensor – Proximity and range sensors – Robotic vision sensor-Force sensor-Light sensors, Pressure sensors.

ROBOT CELL DESIGN AND APPLICATIONS : Robot work cell design and control-Sequence control, Operator interface, Safety monitoring devices in Robot-Mobile robot working principle, actuation using MATLAB, NXT Software Introductions-Robot applications Material handling, Machine loading and unloading, assembly, Inspection, Welding, Spray painting and undersea robot.

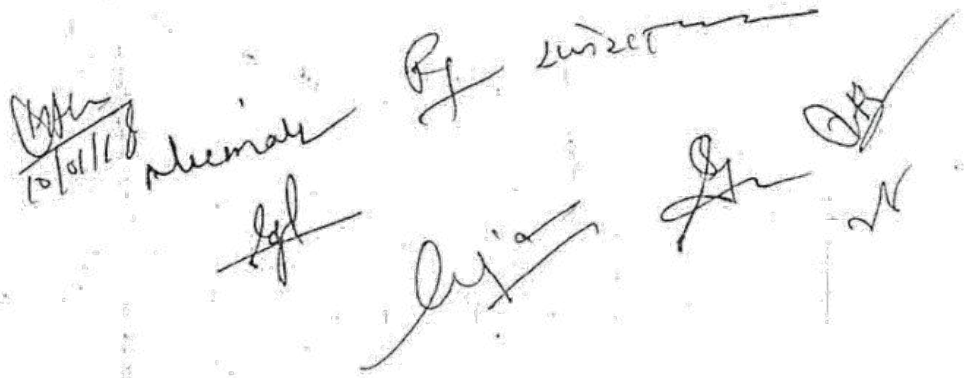
MICRO/NANO ROBOTICS SYSTEM : Micro/Nanorobotics system overview-Scaling effect-Top down and bottom up approach- Actuators of Micro/Nano robotics system-Nanorobot communication techniques-Fabrication of micro/nano grippers-Wall climbing micro robot working principles-Biomimetic robot-Swarm robot-Nanorobot in targeted drug delivery system.

Textbook:

- S.R. Deb, Robotics Technology and flexible automation, Tata McGraw-Hill Education., 2009
- Mikell P Groover& Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, Industrial Robotics, Technology programming and Applications, McGraw Hill, 2012
- Richard D. Klafter, Thomas .A, ChriElewski, Michael Negin, Robotics Engineering an Integrated Approach, Phi Learning., 2009.
- Francis N. Nagy, AndrasSiegler, Engineering foundation of Robotics, Prentice Hall Inc., 1987.

Reference Books:

- P.A. Janaki Raman, Robotics and Image Processing an Introduction, Tata McGraw Hill Publishing company Ltd., 1995.
- Carl D. Crane and Joseph Duffy, Kinematic Analysis of Robot manipulators, Cambridge University press, 2008.
- Fu. K. S., Gonzalez. R. C. & Lee C.S.G., "Robotics control, sensing, vision and intelligence", McGraw Hill Book co, 1987
- Craig. J. J. "Introduction to Robotics mechanics and control", Addison- Wesley, 1999. 9. Ray Asfahl. C., "Robots and Manufacturing Automation", John Wiley & Sons Inc.,1985.
- Bharat Bhushan., "Springer Handbook of Nanotechnology", Springer, 2004. 11. Julian W. Gardner., "Micro sensor MEMS and Smart Devices", John Wiley & Sons, 2001



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

Scheme Code - 2015CSE

7CSU06.2 CYBER SECURITY

MAX_MARKS(50+100)

Objectives:

The objective of this course is to gain a fundamental knowledge of what Cyber Security is and how it applies to your daily work. Gain a fundamental understanding of what an attack is, and how to identify and prevent them from occurring.

Syllabus:

Systems Vulnerability Scanning :Overview of vulnerability scanning, Open Port / Service Identification, Banner / Version Check, Traffic Probe, Vulnerability Probe, Vulnerability Examples, OpenVAS, Metasploit. Networks Vulnerability Scanning - Netcat, Socat, understanding Port and Services tools - Datapipe, Fpipe, WinRelay, Network Reconnaissance – Nmap, THC-Amap and System tools. Network Sniffers and Injection tools – Tcpdump and Windump, Wireshark, Ettercap, Hping Kismet

Network Defense tools: Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding, the basic of Virtual Private Networks, Linux Firewall, Windows Firewall, Snort: Introduction Detection System

Web Application Tools: Scanning for web vulnerabilities tools: Nikto, W3af, HTTP utilities - Curl, OpenSSL and Stunnel, Application Inspection tools – Zed Attack Proxy, Sqlmap. DVWA, Webgoat, Password Cracking and Brute-Force Tools – John the Ripper, L0htcrack, Pwdump, HTC-Hydra

Introduction to Cyber Crime and law: Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behavior, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world, A Brief History of the Internet, Recognizing and Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT 2000.

Introduction to Cyber Crime Investigation: Firewalls and Packet Filters, password Cracking, Keyloggers and Spyware, Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks

REFERENCE BOOKS:

1. Anti-Hacker Tool Kit (Indian Edition) by Mike Shema, Publication Mc Graw Hill.
2. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and Sunit Belpure, Publication Wiley

7CSU06.3 INTERNET OF THINGS

Objectives: MAX_MARKS(50+100)

The objective of the course is to:

1. Vision and Introduction to IoT .
2. Understand IoT Market perspective.
3. Data and Knowledge Management and use of Devices in IoT T echnology .
4. Understand State of the Art – IoT Architecture.
5. Real World IoT Design Constraints, Industrial Automation and Commercial Building Automation in IoT

Syllabus:

M2M to IoT-The Vision-Introduction, From M2M to IoT , M2M towards IoT -the global context, A use case example, Differing Characteristics.

M2M to IoT – A Market Perspective– Introduction, Some Definitions, M2M V alue Chains, IoT V alue Chains, An emerging industrial structure for IoT , The international driven global value chain and global information monopolies.

M2M toIoT –An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

M2M and IoT T echnology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT , Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management

IoT Architecture-State of the Art – Introduction, State of the art,

Architecture Reference Model- Introduction, Reference Model and architecture, IoT reference Model

Textbook:

- Jan Holler , Vlasios T siatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence",1st Edition, Academic Press, 2014.

Reference Books:

- Vijay Madisetti and ArshdeepBahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT , 2014.
- Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything",1st Edition, A press Publications, 2013.

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

Scheme Code - 2015CSE

7CSU07 DIP LAB

Maximum Marks (50+25)

LIST OF LAB EXPERIMENTS:

1. Color image segmentation algorithm development
2. Wavelet/vector quantization compression
3. Deformable templates applied to skin tumor border finding
4. Helicopter image enhancement
5. High-speed film image enhancement
6. Computer vision for skin tumor image evaluation
7. New Border Images

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

Scheme Code - 2015CSE

7CSU08 ANDROID LAB

Maximum Marks (50+25)

LIST OF LAB EXPERIMENTS:

1. Develop an application that uses GUI components, Font and Colors.
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi threading.
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Write a mobile application that creates alarm clock.

SYLLABUS

Semester	**VIII**
Branch	**CSE**
Admission Year	**2015-16**
Academic Year	**2018-19**

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

Scheme Code - 2015CSE

8CSU01 MACHINE LEARNING

MAX_MARKS(50+100)

Objectives: This course aims to cover the potential applications of machine learning in practice, learning different machine learning models, evaluation of machine learning models based on mathematical analysis.

Syllabus:

Introduction: Types of learning, application, Supervised learning: Linear Regression Model, Naive Bayes classifier Decision Tree, K nearest neighbor, Logistic Regression, 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, Backpropagation, Introduction to Deep learning.

(For the students admitted in session 2015-16 and 2016-17)

03/11/19

03/11/19

3/11/19

Director
(Rajesh Dadhich)

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

Scheme Code - 2015CSE

8CSU02BIG DATA USING HADOOP

MAX_MARKS(50+100)

Objectives: The objective of this course is tooptimize business decisions and create competitive advantage with Big Data analytics, introduce Java concepts required for developing map reduce programs, derive business benefit from unstructured data, impart the architectural concepts of Hadoop and introducing map reduce paradigm and to introduce programming tools 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.

(Applicable for the students admitted in session 2015-16 and 2016-17)

03/01/19

03/01/19

3/1/19

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

Scheme Code - 2015CSE

8CSU03.1 COMPUTER VISION

MAX_MARKS(50+100)

Objective: This course offers an introduction to computer vision concepts, including: image formation, edge detection, segmentation, perceptual grouping and object/activity recognition. Students will be able toapply the knowledge earned to develop real time projects related to the field.

Syllabus:

Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, HOG, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

Pattern Analysis: Clustering: K-Means, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA.

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

(Applicable for the students admitted in session 2015-16 and 2016-17)

03/01/19

03/01/19

3/1/19

(Rajesh Doctor)

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

Scheme Code - 2015CSE

8CSU03.2ADVANCE OPERATING SYSTEMS

MAX_MARKS(50+100)

Objectives: This course aims at defining the basic architecture of operating systems, architecture of distributed operating systems, different types of clock synchronization methods and their limitations, mutual exclusion algorithms and deadlock models, understanding various file system managementand challenges in an operating system,different load sharing and load distribution algorithms.

Syllabus:

Operating system introduction and structure, processes, threads, interprocess communication.cpu scheduling: Scheduling algorithm, multiprocess and realtime process scheduling, algorithm evaluation. Process synchronizations: semaphores, critical regions and monitors.

Distributed OS: architecture of distributed Systems, issues in DOS, client-server computing, message-passing, remote procedure call (RPC), limitations of DS, absence of shared memory and global clock, lamport's Logical clocks, vector clocks, causal ordering of messages.

Distributed mutual exclusion and deadlock: mutual exclusion algorithms, token-based and non-token-based algorithms, deadlock models and algorithms, deadlock detection and prevention. Distributed file systems and shared memory: architecture of distributed file systems, design issues, replication algorithms, cache coherence.

Distributed scheduling: motivation and issues, load distribution, balancing and sharing algorithms, load distribution algorithms, load scheduler, task migration.

Failure recovery and fault tolerance: introduction and basic concepts, classification of failures, backward and forward recovery, check pointing and recovery, issues in fault tolerance, commit and voting protocols.

Real-time OS: characteristics of real time OS, hard versus soft real-time systems, real-time communications, real-time scheduling, case study: windows CE, palm OS.

Applicable for the students admitted in session 2015-16 and 2016-17)

03/01/19.

03/01/19

3/1/19

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

Scheme Code - 2015CSE

8CSU03.3 WIRELESS SENSOR NETWORKS

MAX_MARKS(50+100)

Objectives:

This course aims to cover the basic principles behind a Wireless Sensor Network. It provides a broad coverage of challenges and latest issues related to the design and management of WSN and aspects like protocols and their applications.

Syllabus:

Overview of Wireless Sensor Networks: Characteristics, Applications, Design objectives, challenges. Technological Background – MEMS Technology, Hardware and Software Platforms, Wireless Sensor Network Standards. Sensor network architectures and protocol stack, Current and future research on WSNs.

MAC protocols: Fundamentals, low duty cycle protocols and wakeup concepts, contention based protocols, Schedule-based protocols - SMAC, BMAC,TRAMA, Link Layer protocols.
Sensor Deployment Mechanisms, Issues of coverage, Node discovery protocols, Localization schemes, Time Synchronization, Network clustering, Query Models, In-network data aggregation, Robust route setup, Coping with energy constraints, QoS Management, Security.

Routing protocols: Issues in designing a routing protocol, classification of routing, Requirements, Taxonomy - Data-centric routing – SPIN, Directed Diffusion, Flooding and gossiping, Energy aware routing, Gradient-based routing – COUGAR, Hierarchical Routing – LEACH, TEEN,APTEEN, Location Based Routing, Data aggregation – Various aggregation techniques, Localization and positioning.

(Applicable for the students admitted in session 2017-18 and 2016-17)

03/11/19

03/11/19

3/11/19

21/12/19

(Rajesh Dadhwich)

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

Scheme Code - 2015CSE

8CSU07 MACHINE LEARNING LAB

MAX_MARKS(50+25)

Objectives: The objective of this course is to provide knowledge about identifying applications of machine learning in practice, Implementing and applying machine learning algorithms to solve problems.

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.

(Applicable to the students admitted in session 2015-16 and 2016-17

03/01/19

03/01/19

3/1/19

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

Scheme Code - 2015CSE

8CSU08 BIG DATA USING HADOOP LAB

MAX_MARKS(50+25)

Objectives: This course aims at introducing Java concepts required for developing map reduce programs, impart the architectural concepts of Hadoop and introducing map reduce paradigm and introduce programming tools PIG & HIVE in Hadoop echo system.

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, andindexes.

9. Solve some real life big data problems.

(Applicable for the students admitted in session 2017-18 and 2018-19)

03/01/19 03/01/19 3/1/19