



P K M Educational Trust[®]

R.R. INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belagavi | Approved by AICTE , New Delhi & Government of Karnataka)

DEPARTMENT OF COMPUTER SCIENCE & ENGG

COURSE OBJECTIVES & COURSE OUTCOMES

2010 SCHEME

Core	Electives
Programming in C and Data Structures Data Structures and Applications Computer Organization Software Engineering Design and Analysis of Algorithms Object Oriented Programming with c++ System software Operating system Data base management system Computer networks Formal languages and automata theory Unix system programming Compiler design Computer graphics and visualization Object oriented modeling and design Embedded computing systems Programming the web Advanced computer architecture Software architecture System modeling and simulation	Unix and shell programming Python programming Operation research Java and J2EE C# programming and .net Signals and system Data compression Pattern recognition Programming languages Multimedia computing Neural networks Data warehousing and data mining Digital image processing Game theory Artificial intelligence Storage area networks Fuzzy logic

Subjects	Objectives	Outcomes
Computer organization, Embedded computing system Advanced computer architecture.	Understand the architecture of a modern computer with its various processing units. Also the performance measurement of the computer system. In addition to this the memory management system of computer.	Students are able to design their own system and also have the ability to improve the system performance.

Software engineering,OOMD,Software testing,software architecture	These are the subjects at various semesters provides an insight about developing a quality software according to client requirements. SE focuses on development process,OOMD & SA explores the software design and modeling, whereas software testing highlights on 2various software testing strategies according to client requirements	Students are expected to learn various process models, software architecture and testing methodologies.
C,C++,Java,C#	Introducing the fundamental principles of language design. Introducing formal syntax and semantics. Discussing control structures and abstractions. Introducing data typing and abstractions	To be able to express computational solutions in the main programming idioms.To know and understand the principal programming abstractions. To know and understand the functional programming language
Data base management system, Data mining and data warehousing	To educate students with fundamental concepts of Data Base Management System, Data Models, Different Data Base Languages and analyze Data Base design methodology	Analyze Data Base design methodology and also deal with online transactions and control Concurrency. Understand types of Data Base failures and Recovery.
Computer graphics, Digital image processing, Digital signal processing. Pattern recognition, Data compression	Introducing the basic concepts of images ,performing operation on images to improve the quality of image and to retrieve the information from the images	To learn the basics of image, learning about various image processing steps and it will be applied on various real world applications.
Operating system, System software, FLAT, Compiler design,	Demonstrate the relationship among formal languages, formal grammars, and automat and different phases of compiler.The emphasis of the course will be placed on understanding how the various elements that underlie operating system interact and provides services for execution of application software.	Classify machines by their power to recognize languages.Master various processmanagement concepts including scheduling, synchronization, and deadlocks.

Data structure, ADA	Demonstrate familiarity with major algorithms and data structures. Analyse the performance of algorithms. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, trees and graphs	Students are able to choose the appropriate data structure and algorithm design method for a specified application.
Computer networks, Network management system,PWEB	Data Communications introduction Bandwidth utilization. Error Detection and Correction. Connecting LANs. Demonstrate all the network layers. And also demonstrate the client and server side communication by WWW and HTTP.	Students are able to analyse the services and features of the various layers of data networks. Design, calculate, and apply subnet masks and addresses to fulfil networking requirements. Analyse the features and operations of various application layer protocols.
Artificial intelligence, Neural networks, Fuzzy logic	Simulation and theoretical analysis for machine learning	Designing and implementing soft computing based solutions for real-world and engineering problems

LAB COURSES

Programming in C and Data Structures lab
 Data Structures and Applications lab
 System software & OS lab
 Data base application lab
 Programming in web lab
 Network lab
 Programming in C and Data Structures lab
 Design and analysis of algorithms lab
 Unix system programming lab
 Computer Graphics and Visualization lab
 Projects

Name of the lab	Objectives	Outcomes
Programming in C and Data Structures lab	This course is designed to provide a comprehensive study of the C programming language. It stresses the strengths of C, which provide students with the means of writing efficient, maintainable, and portable code.	Understand the basic terminology used in computer programming. Write, compile and debug programs in C language. Use different data types in a computer program. Design programs involving decision structures, loops and functions.
Data Structures and Applications lab, Design and analysis of algorithms lab	Determine which algorithm or data structure to use in different scenarios. Implementation of stack, linked list and queue using array. Implementing different sorting techniques.	Students are able to choose the appropriate data structure and algorithm design method for a specified application
System software & OS lab, Unix system programming lab	To provide an understanding of the language translation. Simulation of CPU Scheduling Algorithms. Simulation of Banker's Algorithm for Deadlock Avoidance, Prevention.	Will be able to create a complete translator for a mini language and describe and use the UNIX operating system.
Data Base Application lab	To educate students with fundamental concepts of Data Base Design, Data Models, Different Data Base Languages (SQL/Oracle).To analyze Data Base design methodology and DB connectivity.	Able to handle with different Data Base languages. Introduction to different Database packages(Oracle/MySQL/ DB2/ etc.

Programming in web lab	Design dynamic and interactive web pages by embedding, CSS, Java Script, php, pearl code in HTML. Understand the various steps in designing a creative and dynamic website.	Able to create a fully functional website. To understand evolution of web services and their architecture. To be able to describe, discover & develop web services.
Computer Graphics and Visualization lab	This course is designed to provide the knowledge about Geometric transformations, geometric algorithms, software systems (OpenGL, shades), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis.	Able to create interactive graphics applications in C++ using one or more graphics application programming interfaces and write programs that demonstrate computer graphics animation.
Network lab	To provide the understanding of different network topology, Bandwidth utilization, Error Detection and Correction, Connecting LANs.	Understand the importance of data communications, role of protocols in networking and Analyze the features and operations of various Networks.

2015-CBCS SCHEME

Core	Electives
Engineering Mathematics - III Analog and Digital Electronics Data Structures and Applications Computer Organization Unix and Shell Programming Discrete Mathematical structures Analog and Digital Electronics Laboratory Data Structures Laboratory Engineering Mathematics - IV Software Engineering Design and Analysis of Algorithms Microprocessors and microcontrollers Object Oriented Programming with JAVA Data communications	Data Mining and Data Warehousing Software Architecture and Design Patterns Operation research Distributed Computing system

Design and Analysis of Algorithm Laboratory Microprocessors Laboratory Management and Entrepreneurship for IT Industry Computer Networks Database Management System Automata theory and Computability Computer Network Laboratory DBMS Laboratory with mini project	
--	--

Subjects	Objectives	Outcomes
Analog and Digital Electronics	Recall and Recognize construction and characteristics of JFETs and MOSFETs and differentiate with BJT. Demonstrate and Analyze Operational Amplifier circuits and their applications. Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques. Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops. Describe, Design and Analyze Synchronous and Asynchronous Sequential	Acquire knowledge of JFETs and MOSFETs , Operational Amplifier circuits and their applications. Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique. Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors. Working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters. Analyze the performance of JFETs and MOSFETs , Operational Amplifier circuits. Simplification Techniques using Karnaugh Maps, Quine McClusky Technique. Synchronous and Asynchronous Sequential Circuits. Apply the knowledge gained in the design of Counters, Registers and A/D & D/A converters
Data Structures and Applications	Explain fundamentals of data structures and their applications essential for programming/problem solving. Analyze Linear Data Structures: Stack, Queues, Lists. Analyze Non-Linear Data Structures: Trees, Graphs. Analyze and Evaluate the sorting & searching algorithms. Assess appropriate data structure during	Acquire knowledge of Various types of data structures, operations and algorithms. Sorting and searching operations. File structures. Analyse the performance of-Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques. Implement all the applications of Data structures in a high-level language. Design and apply appropriate data structures for

	program development/Problem Solving	solving computing problems
Computer Organization	<p>Understand the basics of computer organization: structure and operation of computers and their peripherals. Understand the concepts of programs as sequences or machine instructions. Expose different ways of communicating with I/O devices and standard I/O interfaces. Describe hierarchical memory systems including cache memories and virtual memory.</p> <p>Describe arithmetic and logical operations with integer and floating-point operands. Understand basic processing unit and organization of simple processor, concept of pipelining</p>	<p>Acquire knowledge of-The basic structure of computers & machine instructions and programs, Addressing Modes, Assembly Language, Stacks, Queues and Subroutines.Input/output Organization such as accessing I/O Devices, Interrupts. Memory system basic Concepts, Semiconductor RAM Memories, Static memories, Asynchronous DRAMS, Read Only Memories, Cache Memories and Virtual Memories. Some Fundamental Concepts of Basic Processing Unit, Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control and Micro programmed Control. Pipelining, embedded and large computing system architecture. Analyse and design arithmetic and logical units. Apply the knowledge gained in the design of Computer. Design and evaluate performance of memory systems. Understand the importance of life-long learning</p>
Unix and Shell Programming	<p>Understand the UNIX Architecture, File systems and use of basic Commands. Use of editors and Networking commands.</p> <p>Understand Shell Programming and to write shell scripts. Understand and analyze UNIX System calls, Process Creation, Control & Relationship.</p>	<p>Explain multi user OS UNIX and its basic features. Interpret UNIX Commands, Shell basics, and shell environments. Design and develop shell programming, communication , System calls and terminology. Design and develop UNIX File I/O and UNIX Process</p>
Discrete Mathematical structures	<p>Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science. Understand and apply logic, relations, functions, basic set theory, countability and counting</p>	<p>Verify the correctness of an argument using propositional and predicate logic and truth tables. Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability.Solve</p>

	arguments, proof techniques, Understand and apply mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory. Understand and apply graph theory and mathematical proof tech	problems involving recurrence relations and generating functions. Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction. Explain and differentiate graphs and tree
Analog and Digital Electronics Laboratory	This laboratory course enable students to get practical experience in design, assembly and Evaluation / testing of-Analog components and circuits including Operational Amplifier, Timer, etc. Combinational logic circuits. Flip - Flops and their operations. Counters and Registers using Flip-flops. Synchronous and Asynchronous Sequential Circuits.	Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit. Design and demonstrate various combinational logic circuits. Design and demonstrate various types of counters and Registers using Flip-flops. Use simulation package to design circuits. Understand the working and implementation of ALU.
Data Structures Laboratory	This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of- Asymptotic performance of algorithms. Linear data structures and their applications such as Stacks, Queues and Lists. Non Linear Data Structures and their Applications such as Trees and Graphs	Analyze and Compare various linear and non-linear data structures .Code, debug and demonstrate the working nature of different types of data structures and their applications. Implement, analyze and evaluate the searching and sorting algorithms. Choose the appropriate data structure for solving real world problems
Management and Entrepreneurship for IT Industry	Explain the principles of management, organization and entrepreneur. Discuss on planning, staffing, ERP and their importance .Infer the importance of intellectual property rights and relate the institutional support.	Define management, organization, entrepreneur, planning, staffing, ERP and outline their importance in entrepreneurship .Utilize the resources available effectively through ERP. Make use of IPRs and institutional support in entrepreneurship.
Computer Networks	Demonstration of application layer protocols. Discuss transport layer services and understand UDP and TCP protocols. Explain routers, IP	Explain principles of application layer protocols. Recognize transport layer services and infer UDP and TCP protocols. Classify routers, IP

	and Routing Algorithms in network layer. Disseminate the Wireless and Mobile Networks covering IEEE 802.11 Standard. Illustrate concepts of Multimedia Networking, Security and Network Management	and Routing Algorithms in network layer. Understand the Wireless and Mobile Networks covering IEEE 802.11 Standard. Describe Multimedia Networking and Network Management.
Database Management System	Provide a strong foundation in database concepts, Technology, and practice. Practice SQL programming through a variety of database problems. Demonstrate the use of concurrency and transactions in database. Design and build database applications for real world problems.	Identify, analyze and define database objects, enforce integrity constraints on a database using RDBMS. Use Structured Query Language (SQL) for database Manipulation. Design and build simple database systems. Develop application to interact with databases
Automata theory and Computability	Introduce core concepts in Automata and Theory of Computation .Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different formal languages. Prove or disprove theorems in automata theory using their properties. Determine the decidability and intractability of Computational problems	Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models). Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers. Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness. Classify a problem with respect to different models of Computation.
Computer Network Laboratory	Demonstrate operation of network and its management Commands. Simulate and demonstrate the performance of GSM and CDMA. Implement data link layer and transport layer	Analyze and Compare various networking protocols. Demonstrate the working of different concepts of networking.
DBMS Laboratory with mini project	Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers. Strong practice in SQL	Create, Update and query on the database. Demonstrate the working of different concepts of DBMS. Implement, analyze and evaluate the project developed for

	programming through a variety of database problem	an application.
OBJECT ORIENTED MODELING AND DESIGN	Describe the concepts involved in Object-Oriented modeling and their benefits. Demonstrate concept of use-case model, sequence model and state chart model for a given problem. Explain the facets of the unified process approach to design and build a Software system. Translate the requirements into implementation for Object Oriented design. Choose an appropriate design pattern to facilitate development procedure.	Describe the concepts of object-oriented and basic class modeling. Draw class diagrams, sequence diagrams and interaction diagrams to solve problems. Choose and apply a befitting design pattern for the given problem
INTRODUCTION TO SOFTWARE TESTING	Differentiate the various testing techniques. Analyze the problem and derive suitable test cases. Apply suitable technique for designing of flow graph. Explain the need for planning and monitoring a process.	Derive test cases for any given problem Compare the different testing techniques Classify the problem into suitable testing model Apply the appropriate technique for the design of flow graph. Create appropriate document for the software artifact.
ADVANCED JAVA AND J2EE	Identify the need for advanced Java concepts like Enumerations and Collections Construct client-server applications using Java socket API. Make use of JDBC to access database through Java Programs. Adapt servlets to build server side programs. Demonstrate the use of JavaBeans to develop component based Java software	Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs. Build client-server applications and TCP/IP socket Programs. Illustrate database access and details for managing information using the JDBC API. Describe how servlets fit into Java-based web application architecture. Develop reusable software components using Java Beans
ADVANCED ALGORITHMS	Explain principles of algorithms analysis approaches. Compare and contrast a number theoretic based strategies. Describe complex signals and data flow in networks. Apply the computational geometry criteria	Explain the principles of algorithms analysis approaches. Apply different theoretic based strategies to solve Problems. Illustrate the complex signals and data flow in networks with usage of tools. Describe the computational geometry criteria.

SOFTWARE ENGINEERING	Outline software engineering principles and activities involved in building large software programs. • Identify ethical and professional issues and explain why they are of concern to software engineers. • Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation. • Differentiate system models, use UML diagrams and apply design patterns. • Discuss the distinctions between validation testing and defect testing. • Recognize the importance of software maintenance and describe the intricacies involved in software evolution. • Apply estimation techniques, schedule project activities and compute pricing. • Identify software quality parameters and quantify software using measurements and metrics. • List software quality standards and outline the practices involved. • Recognize the need for agile software development, describe agile methods, apply agile practices and plan for agility	Design a software system, component, or process to meet desired needs within realistic constraints. • Assess professional and ethical responsibility • Function on multi-disciplinary teams • Use the techniques, skills, and modern engineering tools necessary for engineering practice • Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems.
DESIGN AND ANALYSIS OF ALGORITHMS	Explain various computational problem solving techniques. • Apply appropriate method to solve a given problem. • Describe various methods of algorithm analysis.	Describe computational solution to well known problems like searching, sorting etc. • Estimate the computational complexity of different algorithms. • Devise an algorithm using appropriate design strategies for problem solving.
MICROPROCESSORS AND MICROCONTROLLERS	Make familiar with importance and applications of microprocessors and microcontrollers • Expose architecture of 8086 microprocessor and ARM processor • Familiarize instruction set of ARM processor	Differentiate between microprocessors and microcontrollers • Design and develop assembly language code to solve problems • Gain the knowledge for interfacing various devices to x86 family and ARM processor • Demonstrate design of interrupt routines for interfacing.
OBJECT ORIENTED CONCEPTS	Learn fundamental features of object oriented language and JAVA • Set up Java JDK environment to create, debug and run simple Java programs.	Explain the object-oriented concepts and JAVA. • Develop computer programs to solve real world problems in Java. • Develop simple GUI

	<ul style="list-style-type: none"> • Create multi-threaded programs and event handling mechanisms. • Introduce event driven Graphical User Interface (GUI) programming using applets and swings. Module 1 T 	<p>interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles using Applets and swings.</p>
DATA COMMUNICATION	<p>Comprehend the transmission technique of digital data between two or more computers and a computer network that allows computers to exchange data.</p> <ul style="list-style-type: none"> • Explain with the basics of data communication and various types of computer networks; • Illustrate TCP/IP protocol suite and switching criteria. • Demonstrate Medium Access Control protocols for reliable and noisy channels. 	<p>Illustrate basic computer network technology.</p> <ul style="list-style-type: none"> • Identify the different types of network topologies and protocols. • Enumerate the layers of the OSI model and TCP/IP functions of each layer. • Make out the different types of network devices and their functions within a network • Demonstrate the skills of subnetting and routing mechanisms.
DESIGN AND ANALYSIS OF ALGORITHM LABORATORY	<p>Design and implement various algorithms in JAVA</p> <ul style="list-style-type: none"> • Employ various design strategies for problem solving. • Measure and compare the performance of different algorithms. 	<p>Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)</p> <ul style="list-style-type: none"> • Implement a variety of algorithms such as sorting, graph related, combinatorial, etc., in a high level language. • Analyze and compare the performance of algorithms using language features. • Apply and implement learned algorithm design techniques and data structures to solve real world problems
MICROPROCESSOR AND MICROCONTROLLER LABORATORY	<p>To provide practical exposure to the students on microprocessors, design and coding knowledge on 80x86 family/ARM. To give the knowledge and practical exposure on connectivity and execute of interfacing devices with 8086/ARM kit like LED displays, Keyboards, DAC/ADC, and various other devices.</p>	<p>Learn 80x86 instruction sets and gain the knowledge of how assembly language works.</p> <ul style="list-style-type: none"> • Design and implement programs written in 80x86 assembly language • Know functioning of hardware devices and interfacing them to x86 family • Choose processors for various kinds of applications.
CRYPTOGRAPHY, NETWORK SECURITY AND CYBER LAW	<p>Explain the concepts of Cyber security</p> <ul style="list-style-type: none"> · Illustrate key management issues and solutions. · Familiarize with Cryptography and very essential algorithms · Introduce cyber Law and ethics to be followed. 	<p>Design and implement algorithms for 2D graphics primitives and attributes.</p> <ul style="list-style-type: none"> · Illustrate Geometric transformations on both 2D and 3D objects. · Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models. · Decide suitable hardware and software for developing graphics packages using OpenGL

SYSTEM SOFTWARE AND DESIGN	SOFTWARE COMPILER Define System Software such as Assemblers, Loaders, Linkers and Macroprocessors · Familiarize with source file, object file and executable file structures and libraries · Describe the front-end and back-end phases of compiler and their importance to students	Explain system software such as assemblers, loaders, linkers and macroprocessors · Design and develop lexical analyzers, parsers and code generators · Utilize lex and yacc tools for implementing different concepts of system software
OPERATING SYSTEMS	Introduce concepts and terminology used in OS · Explain threading and multithreaded systems · Illustrate process synchronization and concept of Deadlock · Introduce Memory and Virtual memory management, File system and storage	Demonstrate need for OS and different types of OS · Apply suitable techniques for management of different resources · Use processor, memory, storage and file system commands · Realize the different concepts of OS in platform of usage through case studies
DATA MINING AND DATA WAREHOUSING	Define multi-dimensional data models. · Explain rules related to association, classification and clustering analysis. · Compare and contrast between different classification and clustering algorithms	Identify data mining problems and implement the data warehouse · Write association rules for a given data pattern. · Choose between classification and clustering solution.
SOFTWARE ARCHITECTURE AND DESIGN PATTERNS	To Learn How to add functionality to designs while minimizing complexity. · What code qualities are required to maintain to keep code flexible? · To Understand the common design patterns. · To explore the appropriate patterns for design problems	Design and implement codes with higher performance and lower complexity · Be aware of code qualities needed to keep code flexible · Experience core design principles and be able to assess the quality of a design with respect to these principles. · Capable of applying these principles in the design of object oriented systems. · Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary. · Be able to select and apply suitable patterns in specific contexts
OPERATIONS RESEARCH	Formulate optimization problem as a linear programming problem. · Solve optimization problems using simplex method. · Formulate and solve transportation and assignment problems.	Select and apply optimization techniques for various problems. · Model the given problem as transportation and assignment problem and solve. · Apply game theory for decision support system.
DISTRIBUTED COMPUTING SYSTEM	Explain distributed system, their characteristics, challenges and system models. · Describe IPC mechanisms to	Explain the characteristics of a distributed system along with its and design challenges · Illustrate the

	communicate between distributed objects · Illustrate the operating system support and File Service architecture in a distributed system · Analyze the fundamental concepts, algorithms related to synchronization	mechanism of IPC between distributed objects · Describe the distributed file service architecture and the important characteristics of SUN NFS. · Discuss concurrency control algorithms applied in distributed transactions
SYSTEM SOFTWARE AND OPERATING SYSTEM LABORATORY	To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX & YACC tools and/or C/C++/Java · To enable students to learn different types of CPU scheduling algorithms used in operating system. · To make students able to implement memory management -	Implement and demonstrate Lexer's and Parser's · Evaluate different algorithms required for management, scheduling, allocation and communication used in operating system.
COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT	Demonstrate simple algorithms using OpenGL Graphics Primitives and attributes. · Implementation of line drawing and clipping algorithms using OpenGL functions · Design and implementation of algorithms Geometric transformations on both 2D and 3D objects.	Apply the concepts of computer graphics · Implement computer graphics applications using OpenGL
WEB TECHNOLOGY AND ITS APPLICATIONS	Illustrate the Semantic Structure of HTML and CSS • Compose forms and tables using HTML and CSS • Design Client-Side programs using JavaScript and Server-Side programs using PHP • Infer Object Oriented Programming capabilities of PHP • Examine JavaScript frameworks such as jQuery and Backbone	Construct and visually format tables and forms using HTML and CSS • Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically. • Appraise the principles of object oriented development using PHP • Inspect JavaScript frameworks like jQuery and Backbone which facilitates developer to focus on core features.
ADVANCED COMPUTER ARCHITECTURES	• Describe computer architecture. • Measure the performance of architectures in terms of right parameters. • Summarize parallel architecture and the software used for them.	• Explain the concepts of parallel computing and hardware technologies • Compare and contrast the parallel architectures • Illustrate parallel programming concepts
MACHINE LEARNING	Define machine learning and problems relevant to machine learning. • Differentiate supervised, unsupervised and reinforcement learning • Apply neural networks, Bayes classifier and k nearest neighbor, for problems appear	Explain theory of probability and statistics related to machine learning • Investigate concept learning, ANN, Bayes classifier, k nearest neighbor, Q,

	in machine learning. • Perform statistical analysis of machine learning techniques.	
NATURAL LANGUAGE PROCESSING	Learn the techniques in natural language processing. • Be familiar with the natural language generation. • Be exposed to Text Mining. • Understand the information retrieval techniques	Analyze the natural language text. • Generate the natural language. • Do Text mining. • Apply information retrieval techniques.
CLOUD COMPUTING AND ITS APPLICATIONS	Explain the fundamentals of cloud computing • Illustrate the cloud application programming and aneka platform	Explain cloud computing, virtualization and classify services of cloud computing • Illustrate architecture
INFORMATION AND NETWORK SECURITY	Analyze the cryptographic processes. • Summarize the digital security process. • Indicate the location of a security process in the given system	Analyze the Digital security lapses • Illustrate the need of key management
UNIX SYSTEM PROGRAMMING	Explain the fundamental design of the unix operating system • Familiarize with the systems calls provided in the unix environment • Design and build an application/service over the unix operating system	Ability to understand and reason out the working of Unix Systems • Build an application/service over a Unix system.
SOFT AND EVOLUTIONARY COMPUTING	Familiarize with the basic concept of soft computing and intelligent systems • Compare with various intelligent systems • Analyze the various soft computing techniques	Understand soft computing techniques • Apply the learned techniques to solve realistic problems • Differentiate soft computing with hard computing techniques
COMPUTER VISION AND ROBOTICS	Review image processing techniques for computer vision • Explain shape and region analysis • Illustrate Hough Transform and its applications to detect lines, circles, ellipses • Contrast three-dimensional image analysis techniques, motion analysis and applications of computer vision algorithms	Implement fundamental image processing techniques required for computer vision • Perform shape analysis • Implement boundary tracking techniques • Apply chain codes and other region descriptors • Apply Hough Transform for line, circle, and ellipse detections. • Apply 3D vision techniques. • Implement motion related techniques. • Develop applications using computer vision techniques.
MACHINE LEARNING LABORATORY	Make use of Data sets in implementing the machine learning algorithms 2. Implement the machine learning concepts and algorithms in any suitable language	1.Understand the implementation procedures for the machine learning algorithms. 2. Design Java/Python programs for various Learning algorithms. 3. Apply appropriate data sets to the Machine Learning algorithms. 4. Identify and apply

		Machine Learning algorithms to solve real world problems.
WEB TECHNOLOGY LABORATORY WITH MINI PROJECT	1.Design and develop static and dynamic web pages. 2. Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages. 3. Learn Database Connectivity to web applications.	Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's. • Have a good understanding of Web Application Terminologies, Internet Tools other web services. • Learn how to link and publish web sites
INTERNET OF THINGS TECHNOLOGY	Assess the genesis and impact of IoT applications, architectures in real world. • Illustrate diverse methods of deploying smart objects and connect them to network. • Compare different Application protocols for IoT. • Infer the role of Data Analytics and Security in IoT. • Identify sensor technologies for sensing real world entities and understand the role	Interpret the impact and challenges posed by IoT networks leading to new architectural models. • Compare and contrast the deployment of smart objects and the technologies to connect them to network. • Appraise the role of IoT protocols for efficient network communication. • Elaborate the need for Data Analytics and Security in IoT. • Illustrate different sensor technologies for sensing real world entities.
BIG DATA ANALYTICS	Understand Hadoop Distributed File system and examine MapReduce Programming • Explore Hadoop tools and manage Hadoop with Ambari • Appraise the role of Business intelligence and its applications across industries • Assess core data mining techniques for data analytics	Master the concepts of HDFS and MapReduce framework • Investigate Hadoop related tools for Big Data Analytics and perform basic Hadoop Administration • Recognize the role of Business Intelligence, Data warehousing and Visualization in decision making • Infer the importance of core data mining techniques for data analytics • Compare and contrast different Text Mining Techniques
HIGH PERFORMANCE COMPUTING	Introduce students the design, analysis, and implementation, of high performance computational science and engineering applications. • Illustrate on advanced computer architectures, parallel algorithms, parallel languages, and performance-oriented computing.	• Illustrate the key factors affecting performance of CSE applications, and • Make mapping of applications to high-performance computing systems • Apply hardware/software co-design for achieving performance on real-world
USER INTERFACE DESIGN	To study the concept of menus, windows, interfaces • To study about business functions • To study the characteristics and components of	Design the user interface, design, menu creation and windows creation and connection between menu and windows

	windows and the various controls for the windows. • To study about various problems in windows design with color, text, graphics. • To study the testing methods	

[As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2017 -2018)

Subjects	Outcomes
ANALOG AND DIGITAL ELECTRONICS	Explain the operation of JFETs and MOSFETs , Operational Amplifier circuits and their application • Explain Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique. • Demonstrate Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors, working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters • Design of Counters, Registers and A/D & D/A converters
DATA STRUCTURES AND APPLICATIONS	Explain different types of data structures, operations and algorithms • Apply searching and sorting operations on files • Make use of stack, Queue, Lists, Trees and Graphs in problem solving. • Develop all data structures in a high-level language for problem solving.
COMPUTER ORGANIZATION	Explain the basic organization of a computer system. • Demonstrate functioning of different sub systems, such as processor, Input/output, and memory. • Illustrate hardwired control and micro programmed control. pipelining, embedded and other computing systems. • Build simple arithmetic and logical units.
UNIX AND SHELL PROGRAMMING	After studying this course, students will be able to: • Explain UNIX system and use different commands. • Compile Shell scripts for certain functions on different subsystems. • Demonstrate use of editors and Perl script writing
DISCRETE MATHEMATICAL STRUCTURES	Make use of propositional and predicate logic in knowledge representation and truth verification. • Demonstrate the application of discrete structures in different fields of computer science. • Solve problems using recurrence relations and generating functions. • Apply different mathematical proofs, techniques in proving theorems. • Compare graphs, trees and their applications.

ANALOG AND DIGITAL ELECTRONICS LABORATORY	On the completion of this laboratory course, the students will be able to: • Demonstrate various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit. • Design and demonstrate various combinational logic circuits. • Design and demonstrate various types of counters and Registers using Flip-flops • Make use of simulation package to design circuits. • Infer the working and implementation of ALU.
DATA STRUCTURES LABORATORY	On the completion of this laboratory course, the students will be able to: • Analyze and Compare various linear and non-linear data structures • Demonstrate the working nature of different types of data structures and their applications • Develop, analyze and evaluate the searching and sorting algorithms • Choose the appropriate data structure for solving real world problems