



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)



Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B. Tech in COMPUTER SCIENCE AND ENGINEERING

Course Structure (KR21)

Applicable from 2021-22 Admitted Batch

III Year I Semester

S. No.	Course Code	Course Title	L	T	P	Credits
1	21CC501PC	Design and Analysis of Algorithms	3	1	0	4
2	21CC502PC	Software Engineering	3	0	0	3
3	21CC503PC	Web Technologies	3	1	0	4
4	21CS501PC	Computer Networks	3	0	0	3
5		Professional Elective-I	3	0	0	3
6	21CC505PC	Software Engineering Lab	0	0	3	1.5
7	21CC506PC	Web Technologies Lab	0	0	4	2
8	21CS502PC	Computer Networks Lab	0	0	3	1.5
9	21CS512HSM	Professional and Soft Skills Lab	0	0	2	1
10	*MC501AI	Artificial Intelligence	3	0	0	0
		Total Credits	18	2	12	23

Note: *Mandatory Course(*MC)- Satisfactory/Unsatisfactory

Professional Elective-I

21CE511PE	Parallel Programming
21CE512PE	Data Analytics



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)



Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B. Tech in COMPUTER SCIENCE AND ENGINEERING

Course Structure (KR21)

Applicable from 2021-22 Admitted Batch

III Year II Semester

S. No.	Course Code	Course Title	L	T	P	Credits
1	21CC601PC	Cloud Computing	3	0	0	3
2	21CS601PC	Competitive Programming	3	0	0	3
3		Professional Elective-II	3	0	0	3
4		Professional Elective-III	3	0	0	3
5		Open Elective-I	3	0	0	3
6	21CC602PC	Cloud Computing Lab	0	0	3	1.5
7	21CS602PC	Competitive Programming Lab	0	0	2	1
8		Professional Elective- III Lab	0	0	3	1.5
9	*MC209ES	Environmental Science	3	0	0	0
10	*MC601CS	Cyber Security	3	0	0	0
		Total Credits	21	0	8	19

Note: *MC-Environmental Science–Should be Registered by Lateral Entry Students only.

*Mandatory Course(*MC)- Satisfactory/Unsatisfactory

Professional Elective-II

21CE611PE	Neural Networks
21CE612PE	Design Patterns

Professional Elective-III

21CS621PE	Front End Web Development
21CE621PE	Software Testing Methodologies

Professional Elective-III LAB

21CS631PE	Front End Web Development Lab
21CE631PE	Software Testing Methodologies Lab

#Courses in PE-III and PE-III Lab must be in 1-1 correspondence.

Open Elective - I

21IT601PC	Internet of Things
21IT601OE	Introduction to Embedded Systems



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Syllabus (KR21)
DESIGN AND ANALYSIS OF ALGORITHMS (21CC501PC)
Common to CSE, IT, CSE (AI&ML) and CSE (DS)

L	T	P	C
3	1	0	4

Pre-requisites/ Co-requisites:

1. PP102ES -Programming for Problem Solving Course
2. 21CS303PC -Data Structures through C++ Course
3. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Introduce the notations for analysis of the performance of algorithms and recursion.
2. Relate major algorithmic techniques(divide-and-conquer, backtracking, greedy) and mention problems for which each technique is appropriate.
3. Introduce the applications of graphs and trees.
4. Understand dynamic programming with applications.
5. Introduce applications of strings.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Analyse the performance of algorithms and illustrate the use of divide and conquer in applications.
2. Illustrate the use of greedy method and backtracking in real world applications.
3. Apply BFS and DFS and branch and bound in different applications.
4. Develop applications using dynamic programming.
5. Design different applications of string processing and understand P, NP problems.

UNIT-I

Introduction: Algorithm, Performance Analysis-Space Complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

Recursion: Introduction, Fibonacci sequence, Climbing Stairs, Reverse String, Happy Number, Greatest Common Divisor, Strobo grammatic Number II.

Divide and Conquer: General method, Quick sort, Merge sort, Applications: Majority Element, Calculate pow(x,n).

UNIT-II

Binary Search- Introduction, Applications: Median of two sorted arrays, Find the fixed point in a given array, Find Smallest Common Element in All Rows, Longest Common Prefix, Koko Eating Bananas.

Greedy Method: General method – Applications –Minimum product subset of an array, Best Time to Buy and Sell Stock, Knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT-III

Breadth First Traversal and Depth First Traversal: BFS Introduction, Applications: Find All The Lonely Nodes, Max Area of Island, Number of Distinct Islands. DFS Introduction, Applications: The Maze, Boundary of Binary Tree.

Trees: Binary Tree Introduction, Applications: Symmetric Tree, Balanced Binary Tree, Average of Levels in Binary Tree, Find Largest Value in Each Tree Row, Binary Tree Right Side View.

Backtracking: General method, Applications: N Queens Problem, Hamiltonian Cycle, Brace Expansion, Gray Code, Path with Maximum Gold, Generalized Abbreviation, Campus Bikes II.

UNIT-IV

Dynamic Programming: Introduction, DP Techniques, Applications – Matrix Chain Multiplication, Optimal Binary Search Tree, All Pairs Shortest Paths, Traveling Salesperson Problem, Climbing Stairs, Min Cost Climbing Stairs, Maximum Sub Array, Number of Corner Rectangles, 0/1 Knapsack Problem.

Strings Problems: Introduction, Count Substrings with Only One Distinct Letter, Valid Word Abbreviation, Longest Repeating Substring, Longest Common Subsequence, Longest Increasing Subsequence.

UNIT-V

Branch and Bound: General Method, FIFO Branch and Bound, LC Branch and Bound, Applications: 0/1 knapsack Problem, Traveling Salesperson Problem.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP-Hard and NP-Complete classes, Cook's theorem.

TEXT BOOKS:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Raja sekharan, University Press.
2. Introduction to Algorithms, Thomas H Cormen, Charles E. Leiser, Ronald L. Rivest, Clifford Stein, 3rd edition, 2009.
3. Design and Analysis of Algorithms: A Contemporary Perspective, Cambridge University Press, Sandeep Sen, Amit Kumar, 2019

REFERENCE BOOKS:

1. Design and analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.
2. Algorithm design: foundations, Analysis and Internet Examples, M.T. Goodrich and Tamassia, John Wiley and Sons.
3. Algorithms Design And Analysis 2015 Edition by Harsh Bhasin, Oxford Publishers.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (KR21)

SOFTWARE ENGINEERING (21CC502PC)

Common to CSE, IT, CSE (AI&ML) and CSE (DS)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Understand the basic Software engineering methods, practices and application and a general understanding of software process models like Agile and DevOps.
2. Analyse software requirements, the SRS documents, data models, object models, context models, behavioural models and DevOps tools including Git and GitHub.
3. Learn how to use DevOps tools like Jenkins for continuous integration in Design Engineering concepts.
4. Recognise the understanding of testing strategies and continuous deployment using Dockers and Kubernetes.
5. Understand the interpretation of software risks and the idea of applying AWS cloud service for deployment on cloud.

Course Outcomes: After learning the concepts of the course, the student is able to

1. Identify and translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Illustrate appropriate process model depending on the user requirements understanding the concept of DevOps.
3. Apply appropriate software architectures and patterns to carry out the high-level design of a system, CI/CD process using Jenkins,
4. Demonstrate different testing strategies and develop a simple testing report
5. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

Software Process Structure: Software engineering- a layered technology, a process framework.

Process models: The waterfall model, incremental process models, evolutionary process models.

Introduction to Agile and DevOps: Agile versus traditional method comparison, various agile methodologies, Introduction and Benefits of working in a DevOps environment, DevOps Lifecycle, DevOps Stages, DevOps Delivery Pipeline.

UNIT - II

Understanding Requirements: Functional and non-functional requirements, user requirements, system requirements, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation, and analysis, requirements monitoring, and validation.

Git & GitHub: Version control, Branches, Merge, Revert, Fork and working with remote repository.

UNIT - III

Design Engineering: The design process and design quality, design concepts, the design model. Creating an architectural design: software architecture architectural styles and patterns, context model, a conceptual model of UML, class diagrams, sequence diagrams, use case diagrams, component diagrams.

Build Tools and Continuous Integration using Jenkins: Introduction to Maven - Installation and configuration, Jenkins Architecture, Build Pipeline project using Jenkin script.

UNIT – IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black- box and white-box testing, validation testing, system testing, the art of debugging.

Continuous Deployment: Containerization using Docker - Understand Containerization, Docker Image and Docker CLI Commands, Setup Docker Compose, Deploy a multi-container application using Docker Compose, Orchestration using Kubernetes - Understand Container Orchestration, Kubernetes Core Concept, Deploy Pods, Create Deployments to manage Pods **Continuous Monitoring:** Introduction to Continuous Monitoring using Nagios

UNIT-V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Introduction to cloud and its services: Cloud, cloud services, Understand and create cloud infrastructure using AWS. Deployment of the web application on to the cloud.

TEXTBOOKS:

1. Software Engineering, A practitioner's Approach-Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.
4. Cloud Native DevOps with Kubernetes by John Arundel.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach-James F.Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice-Waman S Jawadekar, The McGraw-Hill Companies.
3. The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices, 1st Edition, Viktor Farcic, CreateSpace Independent Publishing Platform publications, 2016
4. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, 1st Edition, Jez Humble and David Farley, 2010.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (KR21)

WEB TECHNOLOGIES (21CC503PC)

Common to CSE, IT and CSE (AI&ML)

L T P C

3 1 0 4

Pre-requisites/ Co-requisites:

1. 21CS401PC–Java Programming Course
2. 21CS402PC – Database Management Systems Course

Course Objectives: The course will help to

1. Comprehend and apply the concepts of JavaScript.
2. Gain knowledge and to build a web page using HTML and Bootstrap.
3. Implement the server-side functionality using NodeJS and ExpressJS.
4. Comprehend and demonstrate the application of Express routing mechanism.
5. Practice and demonstrate the working of NoSQL databases and to implement the integration of front end with backend database (MongoDB) using ExpressJS.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply asynchronous programming techniques using java script.
2. Apply HTML and JavaScript effectively to create an interactive and responsive website
3. Develop a simple client server model using NodeJS and ExpressJS.
4. Analyze and Implement routing methodologies using ExpressJS.
5. Design and create a NoSQL (Mongo)database and then integrate it with the front end using ExpressJS.

Unit-1

HTML Common Tags: Lists, Tables, images, forms, CSS -using Bootstrap Framework, Bootstrap tags like Tables, images, forms, grid model. Bootstrap components like Navbar, Card, Badge, buttons and pagination.

Unit-II

Client Side Scripting: Basic Java Script, DOM Manipulation from JS , Strings and regular expressions, functions, Destructuring, arrays, sets and maps, Iterators and Generators, Promises and Asynchronous operations.

Unit-III

Introduction to Server Side: Introduction to NodeJS, Deploying to Node, NPM, Introduction to ExpressJS , life cycle and routing of Express App, deploying to Node. Handling request and response parameters, reading configuration parameters.

Unit-IV

Introduction to Middleware: Express Route Methods- GET, POST, PUT, DELETE, Route Paths- strings, string patterns, or regular expressions, Route parameters, Route , Route Handlers - as a form of a function, an array of functions, or combinations of both,
Response methods - download, end, json, redirect, render, send, sendFile, sendStatus

Unit-V

Introduction to No-SQL Databases: Introduction to MongoDB, Importing, exporting and querying (CRUD)– creating and manipulating documents. Aggregation Pipeline - group, limit, project, sort, match, addFields, count, lookup out, Integration with Express.

TEXTBOOKS:

1. Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers 1st Edition, Nicholas C. Zakas, 2016
2. Web Programming with Html, CSS, Bootstrap, Javascript, JQuery, Php, and Mysql, Larry Sanchez, 2017
3. Web Development with Node and Express, Ethan Brown, 2014 MongoDB: The Definitive Guide, 2nd Edition, Kristina Chodorow, 2013

REFERENCE BOOKS:

1. Exploring ES6, A book by Dr. Axel Rauschmayer, 2018
2. Beginning Node.js, Express & MongoDB Development, Greg Lim, 2019
3. MongoDB Basics by Peter Membrey David HowsEelcoPlugge, aPress, 2014



B. Tech. in COMPUTER SCIENCE AND ENGINEERING III Year I Semester Syllabus (KR21) COMPUTER NETWORKS (21CS501PC)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CS403PC - Operating Systems Course.

Course Objectives: The course will help to

1. Learn the basic concepts of computer networks.
2. Gain the knowledge on Framing, Error – Detection and Correction and Routing Protocols.
3. Understand the TCP, UDP protocols and DNS.
4. Understand the Socket and I/O Function.
5. Understand usage of Raw/Web sockets.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Understand the networking concepts
2. Illustrate the client-server applications using Framing, Error – Detection and Correction and Routing Protocols.
3. Apply TCP, UDP protocols and DNS.
4. Analyze and implement Socket Programming.
5. Develop and implement client-server applications using Raw and Web sockets.

UNIT – I

Data Communications: Components –Direction of Data flow –Networks –Components and Categories – Types of Connections – Topologies –Protocols and Standards – ISO / OSI model.

Physical layer: Transmission modes, Multiplexing, Transmission Media- Guided Transmission Media, Unguided Transmission Media.

Data link layer: Introduction, Framing and Error–Detection and Correction–Hamming code, CRC, Checksum.

UNIT – II

Flow and Error Control: Noiseless Channels- Simplex, Simplex-Stop and Wait Protocol, Noisy Channels- Stop and Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ. Medium Access sub layer: Multiple Access Protocols- Random Access Protocols- ALOHA, CSMA/CD, CSMA/CA, LAN, Ethernet IEEE 802.3.

UNIT – III

Network layer: Logical Addressing- IPV4- Classful and Classless Addressing, Subnetting, NAT, IPV6 Addressing.

Internetworking- Tunneling, Address mapping- ARP, RARP, Bootp, DHCP, ICMP, IGMP.

Routing –Distance Vector Routing, Link State Routing, Shortest path routing.

Transport Layer: Process to Process Delivery, UDP and TCP protocols, TCP Connection Control, Congestion Control- Open loop Congestion Control, Closed Loop Congestion Control, Congestion control in TCP.

UNIT – IV

Application Layer: Domain name space, Remote Logging, Telnet, electronic mail, SMTP, FTP, WWW, HTTP, SNMP.

Introduction to Sockets: Socket Types, API's, socket addressing, socket lifecycle, socket states and security.

Socket options: getsockopt and setsockopt functions. Socket states, Socket Programming – client socket, server socket, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.

UNIT – V

Web Sockets: Web sockets Roles: web sockets –events and actions, opening connections, handling errors, send and receive a message, closing a connection Java script and web sockets API: Server working, web socket API, communicating with server, security.

TEXT BOOKS:

1. Data Communications and Networking-Behrouz A. Forouzan, Fourth Edition TMH,2006.
2. Computer Networks- Andrew S Tanenbaum, 4th Edition. Pearson Education,PHI.
3. UNIX Network Programming-W.Richard Stevens, Bill Fenner, Andrew M. Rudoff, PearsonEducation.
4. UNIX Network Programming- – W. Richard Stevens, PHI 1st Edition.

REFERENCE BOOKS:

1. Data communications and Computer Networks- P.C Gupta,PHI.
2. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, PearsonEducation.
3. Understanding communications and Networks- W.A. Shay, CengageLearning 3rd Edition.
4. Data and Computer Communication-William Stallings, Pearson Education,6thEdition ,2000.
5. UNIX for Programmers and Users- Graham GLASS, King abls, Pearson Education 3rd Edition.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (KR21)

PARALLEL PROGRAMMING (21CE511PE)

Common to CSE, IT, CSE (AI&ML) and CSE (DS)

(Professional Elective-I)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. PP102ES – Programming for Problem Solving Course
2. 21CS303PC- Data Structures through C++ Course
3. 21CS403PC – Operating Systems Course

Course Objectives: The course will help to

1. Understand the concepts of parallel programming.
2. Familiarize with GPU concepts.
3. Understand GPU architecture.
4. Understand and become comfortable in parallel programming on MPI.
5. Understand parallel programming in CUDA on GPU.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Implement MPI applications.
2. Develop applications with CUDA on GPU.
3. Use NVIDIA board for applications
4. Apply deep learning applications on GPU. with CUDA.
5. Explain and use advanced tools cuBLAS, cuDNN for parallel programming.

UNIT-I

Introduction to Parallel Computing: Basics, fundamental laws, Categorizing parallel approaches, Parallel strategies, and Parallel speedup versus comparative speedups.

Performance limits and profiling: Application's potential performance limits, determine your hardware capabilities, characterizing your application.

Parallel algorithms and patterns: Algorithm analysis for parallel computing applications, Performance models versus algorithmic complexity, Parallel algorithms, Hash function, Spatial hashing, Prefix sum, Parallel global sum.

UNIT- II

Parallel Programming on CPU: Vectorization, SIMD overview, Hardware trends for vectorization, Vectorization methods, Programming style for better vectorization,

MPI: The basics for an MPI program, the send and receive commands for process-to-process communication, Collective communication, Data parallel examples, Advanced MPI functionality to simplify code and enable optimizations

UNIT- III

GPU Architectures and CUDA Introduction: The CPU, GPU system as an accelerated computational platform, The GPU and the thread engine, Characteristics of GPU memory spaces The PCI bus: CPU to GPU data transfer overhead, multi-GPU platforms and MPI, Potential benefits of GPU accelerated platforms. Introduction to CUDA Programming: The history of high-performance computing – Technical requirements, Hello World from CUDA, Thread hierarchy, Vector addition using CUDA, Error reporting in CUDA, Data types support in CUDA.

UNIT- IV

Parallel Programming on CUDA: CUDA Memory Management: NVIDIA Visual Profiler, Global memory/device memory, Shared memory, Read, only data/cache, Registers in GPU, pinned memory, unified memory, GPU memory evolution CUDA Thread Programming: CUDA threads, blocks, and the GPU, CUDA occupancy, understanding parallel reduction, Identifying the application's performance limiter.

UNIT- V

Kernel Execution Model and Optimization Strategies, Kernel execution with CUDA streams, Pipelining the GPU execution, The CUDA callback function, CUDA streams with priority, Kernel execution time estimation using CUDA events, CUDA dynamic parallelism, Grid, level cooperative groups, Multi, Process Service

Advanced Concepts in CUDA: Scalable Multi-GPU Programming: Solving a linear equation using Gaussian elimination, GPU Direct peer to peer, GPU Direct RDMA, CUDA streams Parallel Programming Patterns in CUDA: Matrix multiplication optimization, Convolution, Prefix sum (scan), Compact and split, Quicksort in CUDA using dynamic parallelism, Radix sort.

Overview of Libraries: cuBLAS, cuRAND, cuFFT, NPP, cuDNN

TEXT BOOKS:

1. Parallel and High-Performance Computing, Robert (Bob) Robey and Yuliana (Yulie) Zamora, Manning Publications, 2021
2. Learn CUDA Programming: A beginner's guide to GPU programming and parallel computing with CUDA 10.x and C/C++, Jaegeun Han, Bharatkumar Sharma, 2019

REFERENCE BOOKS:

1. Programming Massively Parallel Processors: A Hands, on Approach, Third Edition, David B. Kirk, Wen, mei W. Hwu, 2016.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (KR21)

DATA ANALYTICS (21CE512PE)

Common to CSE, IT, CSE (AI&ML) and CSE (DS)

(Professional Elective – I)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21MA406BS – Statistical Methods for Computer Science Course
2. 21CS402PC – Database Management Systems Course

Course Objectives: The course will help to

1. Explore the fundamental concepts of data analytics.
2. Learn the principles and methods of statistical analysis.
3. Discover interesting patterns, analyse supervised and unsupervised models.
4. Estimate the accuracy of the algorithms.
5. Understand the various search methods and visualization techniques.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline the impact of data analytics for business decisions and strategy.
2. Illustrate data analysis/statistical analysis.
3. Demonstrate standard data visualization and formal inference procedures.
4. Design Data Architecture.
5. Summarize various Data Sources.

UNIT - I

Data Management: Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Data Quality (noise, outliers, missing values, duplicate data) and Data Processing & Processing.

UNIT - II

Data Analytics: Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and variables, Data Modeling Techniques, Missing Imputations etc. Need for Business Modeling.

UNIT - III

Regression: Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.

Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT - IV

Object Segmentation: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc. Time Series Methods: Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction

UNIT - V

Data Visualization: Pixel-Oriented Visualization Techniques, Geometric Projection Visualization Techniques, Icon-Based Visualization Techniques, Hierarchical Visualization Techniques, Visualizing Complex Data and Relations.

TEXT BOOKS:

1. Student's Handbook for Associate Analytics – II, III.
2. Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers, 2010.

REFERENCE BOOKS:

1. Introduction to Data Mining, Tan, Steinbach and Kumar, Addison Wesley, 2006.
2. Data Mining Analysis and Concepts, M. Zaki and W. Meira, 2014
3. Mining of Massive Datasets, Jure Leskovec Stanford Univ. Anand Rajaraman Millway Labs, Jeffrey D Ullman Stanford Univ, 2014.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Syllabus (KR21)
SOFTWARE ENGINEERING LAB (21CC505PC)
Common to CSE, IT, CSE(AI&ML) and CSE(DS)

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. PP102ES – Programming for Problem Solving Course
2. 21CS403PC – Operating Systems Course
3. 21CC502PC – Software Engineering Course
4. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Develop the process of problem statement.
2. Understand the process of development of Software Requirement Specifications.
3. Have hands on experience in Design, develop and testing various modules in a project.
4. Understand the usage of GitHub and Jenkins.
5. Devise the deployment of project in AWS cloud using Docker and Kubernetes.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline to translate end-user requirements into system and software requirements.
2. Illustrate a high-level design of the system from the software requirements.
3. Apply usecase tools in design phase.
4. Use Jenkins to build project.
5. Devise a project in AWS cloud using Dockers and Kubernetes.

Software to be used: The students must use JDK 11 Version, STAR UML, GIT Bash, Jenkins, Dockers Desktop, Mini KUBE, Eclipse, Tomcat, Visual Studio Editor.

List of Experiments:

Do the following exercises for any one project given in the list of sample projects or any other projects?

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Study and usage of any Design phase CASE tool
4. Creating the project and committing using Git and GitHub
5. Creating Maven Java and Maven Web project using Eclipse.
6. Building the project in Jenkins
7. Deploying the project in AWS cloud using Docker and Kubernetes and monitoring using Nagios Tool

Sample Projects:

1. Book Bank
2. Online course reservation system
3. E-ticketing
4. Recruitment system
5. Hospital Management system
6. Online Banking System

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition, 2015.
2. Software Engineering- Sommerville, 7th edition, Pearson Education, 2017.
3. The unified modeling language user guide Grady Brooch, James Rumbaugh, Ivar Jacobson, Pearson Education, 2016.
4. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2015.

REFERENCE BOOKS:

1. Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale, 2018.
2. Cloud Native DevOps with Kubernetes by John Arundel, 2016.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (KR21)

WEB TECHNOLOGIES LAB (21CC506PC)

Common to CSE, IT and CSE(AI&ML)

L	T	P	C
0	0	4	2

Pre-requisites/ Co-requisites:

1. 21CS401PC – Java Programming Course
2. 21CS402PC- Database Management Systems Course
3. 21CC503PC - Web Technologies Course

Course Objectives: The course will help to

1. Comprehend and apply the concepts of JavaScript
2. Gain knowledge to build a web page using HTML and Bootstrap.
3. Implement the server-side functionality using NodeJS and ExpressJS.
4. Comprehend and demonstrate the application of Express routing mechanism.
5. Practice and demonstrate the working of NoSQL databases and to implement the integration of front end with backend database (MongoDB) using ExpressJS.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply asynchronous programming techniques using java script.
2. Apply HTML and JavaScript effectively to create an interactive and responsive website.
3. Develop a simple client server model using NodeJS and ExpressJS.
4. Analyze and Implement routing methodologies using ExpressJS.
5. Design and create a NoSQL (Mongo)database and then integrate it with the front end using ExpressJS, this, by better understanding client-server communication.

Software to be used: The students must use VS Code, Node Js, Mongo DB

List of Programs:

1. Write a JavaScript program --
 - a) which accepts a string as input and swap the case of each character. For example, if you input 'The Quick Brown Fox' the output should be 'tHEqUICKbROWNfOX'.
 - b) to find the most frequent item of an array
 - c) to remove duplicate items from an array (ignore case sensitivity)
 - d) to perform a binary search
 - e) to list the properties of a JavaScript object.
 - f) to check whether an object contains given property.
 - g) to sort a list of elements using Quick sort.
 - h) to implement Bubble Sort
 - i) to read from a JSON object and display the data in a table (HTML page)
 - j) that takes name, rollno, marks from user from a form and on form submission displays them in a tabular format, with GPA (like a marks sheet)
2. Write JS code in an HTML page such that based on location selected by user an AJAX request is made and weather details for that location are fetched and displayed
3. Write a Node JS program that accepts a port from the user and runs a node server at that port
4. Write a NodeJS program to read from a file and display the content on screen

5. Write a NodeJS program to accept a file name from user, text from user, if file exists append the text to the file. If not create a new file and add the text to it.
6. Create a student database in Mongo DB with all the details of students of a class.
7. Create a form such that, based on student roll number provided by user, the student details should be fetched (usingExpressJS)

Additional Programs-

8. Create a form such that CRUD operations can be performed on the student DB using ExpressJS
9. Create a simple website for the CRUD operations on student DB and apply Express Routing.

TEXTBOOKS:

1. Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers 1st Edition, Nicholas C. Zakas, 2016.
2. Web Programming with Html, CSS, Bootstrap, Javascript, JQuery, Php, and Mysql, Larry Sanchez, 2017.
3. Web Development with Node and Express, Ethan Brown, 2014 MongoDB: The Definitive Guide, 2nd Edition, Kristina Chodorow, 2013.

REFERENCE BOOKS:

1. Exploring ES6, A book by Dr. Axel Rauschmayer, 2018.
2. Beginning Node.js, Express & MongoDB Development, Greg Lim, 2019.
3. MongoDB Basics by Peter Membrey David HowsEelcoPlugge, aPress, 2014.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Syllabus (KR21)
COMPUTER NETWORKS LAB (21CS502PC)

L	T	P	C
0	0	3	1.5

Pre-requisite/ Co-requisites:

1. PP102ES - Programming for problem solving Course.
2. 21CS401PC - Java Programming Course.
3. 21CS501PC – Computer Networks Course.

Course Objectives: The course will help to

1. Introduce CRC Mechanism.
2. Understand the concepts of Data link layer.
3. Gain the knowledge on network layer.
4. Understand the concepts of TCP and UDP Protocols.
5. Learn the concepts of sockets and DNS.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Compute CRC Mechanisms.
2. Demonstrate and implement the Go-Back-N mechanism.
3. Demonstrate and Apply routing algorithms.
4. Illustrate and implement TCP and UDP Client and server Applications.
5. Develop DNS and Ping service.

Software to be used: The students can use any OS with Java.

List of Programs: Using C/Java programming

1. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
2. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
3. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
4. Implement distance vector routing algorithm for obtaining routing tables at each node.
5. Design the following
 - a. TCP iterative Client and server application to reverse the given input sentence.
 - b. TCP client and server application to transfer file.
 - c. TCP concurrent server to convert a given text into upper case using multiplexing system call “select”.
 - d. TCP concurrent server to echo given set of sentences using poll functions.
6. Design the following
 - a. UDP Client and server application to reverse the given input sentence. 2018-2019 173.
 - b. UDP Client server to transfer a file.
7. Programs to demonstrate the usage of Advanced socket system calls like getsockopt(), setsockopt(), getpeername(), getsockname(), readv() and writev().
8. Implementation of concurrent chat server that allows current logged in users to communicate one with other.
9. Implementation of DNS.
10. Implementation of Ping service.

TEXT BOOKS:

1. Data Communications and Networking-Behrouz A. Forouzan, Fourth Edition TMH,2006.
2. Computer Networks- Andrew S Tanenbaum, 4th Edition. Pearson Education,PHI.
3. UNIX Network Programming-W.Richard Stevens, Bill Fenner, Andrew M. Rudoff, Pearson Education.
4. UNIX Network Programming- – W. Richard Stevens, PHI 1st Edition.

REFERENCE BOOKS:

1. Data communications and Computer Networks- P.C Gupta,PHI.
2. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education.
3. Understanding communications and Networks- W.A. Shay, CengageLearning 3rd Edition.
4. Computer Networking: A Top-Down Approach Featuring the Internet-James F.Kurose & KeithW. Ross, 3rdEdition, Pearson Education.
5. Data and Computer Communication-William Stallings, Pearson Education,6thEdition ,2000.
6. UNIX for Programmers and Users- Graham GLASS, King abls, Pearson Education 3rd Edition.
7. Advanced UNIX Programming- M. J. ROCHKIND, Pearson Education, 2nd Edition.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Syllabus (KR21)
PROFESSIONAL AND SOFT SKILLS LAB (21CS512HSM)
COMPUTER SCIENCE AND ENGINEERING

L	T	P	C
0	0	2	1

Pre-requisites/ Co-requisites:

1. EN106HS – English Language and Communication Skills Lab Course

Course Objectives: The course will help to

1. Improve the student's fluency in English through a well-developed vocabulary.
2. Listen and respond appropriately in formal context.
3. Enhance Reading skills by adequate exposure to different techniques of reading.
4. Communicate ideas relevantly and coherently.
5. Instil writing skills to respond effectively.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Express effectively by using sound vocabulary.
2. Apply various techniques of reading to comprehend and analyze the given matter.
3. Develop flair in writing and felicity in written expression
4. Demonstrate expertise by using structured presentation methodology
5. Modify the requisite skills for enhanced job prospects and better placements.

Software to be used: The software consisting of the prescribed topics elaborated above should be procured and used.

- Oxford Advanced Learner's Compass, 7th Edition
- DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.
- Lingua TOEFL CBT Insider, by Dream tech
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)

INTRODUCTION:

The introduction of the Advanced Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalized context.

- The proposed course is a laboratory course to enable students to use 'good' English and perform the following:
- Gathering ideas and information to organize ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

The following course content to conduct the activities is prescribed for the subject

1. Activities on Fundamentals of Inter-personal Communication and Building Vocabulary

Starting a conversation – responding appropriately and relevantly – using the right body language – Role Play in different situations & Discourse Skills- using visuals - Synonyms and antonyms, word roots, one- word substitutes, prefixes and suffixes, study of word origin, business vocabulary, analogy, idioms and phrases, collocations & usage of vocabulary.

SWOT Analysis.

2. **Activities on Reading Comprehension** –General Vs Local comprehension, reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, critical reading& effective googling. Time and Stress Management.

3. **Activities on Writing Skills** – Structure and presentation of different types of writing – letter writing/Resume writing/ e-correspondence/Technical report writing/ – planning for writing –improving one’s writing. Writing statement of purpose. Writing Portfolio. Emotional Intelligence.

4. **Activities on Presentation Skills** – Oral presentations (individual and group) through JAM sessions/seminars/PPTs and written presentations through posters/projects/reports/ emails/assignments etc. Positive thinking and Decision-making Skills.

5. **Activities on Group Discussion and Interview Skills** – Dynamics of group discussion,intervention, summarizing, modulation of voice, body language, relevance, fluency and organizationof ideas and rubrics for evaluation- Concept and process, pre-interview planning, openingstrategies, answering strategies, interview through tele-conference & video-conference and MockInterviews. Case study interviews. Problem Solving, Leadership Skills, Team Building skills.

MINIMUM REQUIREMENT:

The Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab:

Spacious room with appropriate acoustics.

- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- P – IV Processor, Hard Disk – 80 GB, RAM–512 MB Minimum, Speed – 2.8 GHZ
- T. V, a digital stereo & Camcorder
- Headphones of High quality

TEXT BOOKS:

1. Effective Technical Communication - M Asharaf Rizvi, 2nd Edition, McGraw Hill Education (India) Pvt. Ltd.
2. Academic Writing: A Handbook for International Students - Stephen Bailey, Routledge, 5th Edition.

REFERENCES BOOKS:

1. Learn Correct English – A Book of Grammar, Usage and Composition - Shiv K. Kumar and Hemalatha Nagarajan. Pearson 2007.
2. Professional Communication -ArunaKoneru, McGraw Hill Education (India) Pvt. Ltd, 2016.
3. Technical Communication - Meenakshi Raman & Sangeeta Sharma, Oxford University Press2009.
4. Technical Communication - Paul V. Anderson, Cengage Learning pvt. Ltd. New Delhi,2007.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Syllabus (KR21)
ARTIFICIAL INTELLIGENCE (*MC501AI)
Common to CSE, IT, CSE (AI&ML) and CSE (DS)

L	T	P	C
3	0	0	0

Pre-requisites/ Co-requisites:

1. PP102ES–Programming for Problem Solving Course
2. 21CS303PC – Data Structures through C++ Course
3. 21CS501PC - Design and Analysis of Algorithms Course
4. 21MA406BS- Statistical Methods for Computer Science Course

Course Objectives: The course will help to

1. Learn the distinction between optimal reasoning Vs. human like reasoning.
2. Understand the concepts of state space representation, exhaustive search.
3. Understand the heuristic search together with the time and space complexities.
4. Learn different knowledge representation techniques.
5. Understand the applications of AI, namely game playing, theorem proving, and machine learning.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline an efficient problem space for a problem expressed in natural language.
2. Illustrate a search algorithm for a problem and estimate its time and space complexities.
3. Demonstrate the skill for representing knowledge using the appropriate technique for a given problem.
4. Design algorithms for planning and acting in real world
5. Apply AI techniques to solve problems of game playing, and machine learning.

UNIT - I

Introduction: AI problems, Agents and Environments, Structure of Agents, Problem Solving Agents

Basic Search Strategies: Problem Spaces, Uninformed Search (Breadth-First, Depth-First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A*), Constraint Satisfaction (Backtracking, Local Search)

UNIT - II

Advanced Search: Constructing Search Trees, Stochastic Search, A* Search Implementation, Minimax Search, Alpha-Beta Pruning

Basic Knowledge Representation and Reasoning: Propositional Logic, First-Order Logic, Forward Chaining and Backward Chaining, Introduction to Probabilistic Reasoning, Bayes Theorem

UNIT - III

Advanced Knowledge Representation and Reasoning: Knowledge Representation Issues, Non monotonic Reasoning, Other Knowledge Representation Schemes

Reasoning Under Uncertainty: Basic probability, Acting Under Uncertainty, Bayes' Rule, Representing Knowledge in an Uncertain Domain, Bayesian Networks

UNIT - IV

Learning: What Is Learning? Rote Learning, Learning by Taking Advice, Learning in Problem Solving, Learning from Examples, Winston's Learning Program, Decision Trees.

UNIT - V

Expert Systems: Representing and Using Domain Knowledge, Shell, Explanation, Knowledge Acquisition.

TEXT BOOK:

1. Russell, S. and Norvig, P, Artificial Intelligence: A Modern Approach, Third Edition, Prentice-Hall, 2010.

REFERENCE BOOKS:

1. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivasankar B. Nair, The McGraw Hill publications, Third Edition, 2009.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 6th ed., 2009.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year II Semester Syllabus (KR21)

CLOUD COMPUTING (21CC601PC)

Common to CSE and IT

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CS403PC – Operating Systems Course.
2. 21CS402PC– Database Management Systems Course.

Course Objectives: The course will help to

1. Gain a thorough understanding of cloud computing, including its basics and management challenges.
2. Recognize security issues and future research trends.
3. Know how to use and investigate cloud computing platforms.
4. Understand on-demand computing as an Internet-based service for a shared pool of resources, such as servers, storage, networking, software, databases, and applications.
5. Recognize a paradigm for providing ubiquitous, on-demand access to a shared pool of configurable computing resources that can be deployed and released quickly and with little administration work.

Course outcomes: After learning the concepts of this course, the student is able to

1. Leverage essential AWS to compute and store various use cases.
2. Demonstrate and comprehend the underlying systems that the cloud is built on.
3. Recognize the differences between NoSQL and relational databases.
4. Gain experience with core Amazon Web Capabilities (AWS) services.
5. Assess and select acceptable cloud technologies, methods, and methodologies for installation.

UNIT-I

Computing Paradigms: High performance computing, parallel computing, Distributed computing, cluster computing, Grid computing, cloud computing, Bio computing, Mobile computing, Quantum computing, optical computing, Nano computing.

UNIT-II

Cloud computing fundamentals: Motivation for cloud computing with AWS as a case study, 5 essential characteristics: on demand services like Amazon LEX, Amazon Lambda, VPC Service, Elastic resource pooling using Amazon Elastic Compute Cloud (EC2) as example, Rapid elasticity using Amazon EBS, Amazon EFS, Amazon S3, overview of Docker CLI commands cloud deployment using Docker.

UNIT-III

Cloud computing Architecture and Management: Managing the cloud and managing the cloud infrastructure using AWS cloud Front, AWS BillingService. Managing Identity and Access (IAM), Migrating Application to cloud, Phases of cloud migration, Approaches for Cloud Migration.

UNIT-IV

Cloud Service Models: Infrastructure as service, characteristics of IaaS, Suitability of IaaS, pros and cons of IaaS, summary of IaaS providers, Platform as a Service with examples of Amazon DynamoDB, characteristics of PaaS, Suitability of PaaS, pros and cons of PaaS, summary of PaaS providers, software as a service, characteristics of SaaS, Suitability of SaaS, summary of SaaS providers.

UNIT-V

Governance, Applications with (Azure and GCP) and ML using AWS cloud: Organizational readiness and change management in the cloud age, Data Security in the cloud, legal issues in cloud computing.

Introduction to Azure, GCP services and its Architecture with its features.

ML Application and Integration: Train and Deploy models using Amazon sage maker.

TEXT BOOKS:

1. Cloud Computing, Theory and Practice 2nd Edition, Dan C Marinescu, MK Elsevier publisher, 2019
2. Cloud Computing, A Practical Approach, 1st Edition, Anthony T Velte, Toby J Velte, Robert Elsenpeter, TMH, 2017.
3. Learn AWS Serverless Computing: A beginner's guide to using AWS Lambda, Amazon API Gateway, and services from Amazon Web Services by Scott Patterson.
4. Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud– by Mark Wilkins, Kindle E Textbook, 1st edition, 2019.
5. Microsoft Azure for Dummies by Timothy L. Warner, Wiley publications, 1st Edition, 2021.

REFERENCE BOOKS:

1. Cloud computing principles and paradigms by Rajkumar Buyya, Wiley Publication, 1st Edition, 2013.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
COMPETITIVE PROGRAMMING (21CS601PC)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. PP102ES – Programming for Problem Solving Course
2. 21CS303PC – Data Structures through C++ Course

Course Objectives: The course will help to

1. Understand the sliding window and two pointer approach problems.
2. Understand bit manipulation operations
3. Recognize the various forms of trees and their interconnected components.
4. Understand graph applications and topological sort
5. Recognize disjoint set union and its applications.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline and solve different array applications.
2. Implement bit manipulation for compression, encryption, optimization
3. Illustrate different types of trees and tries applications.
4. Demonstrate graph applications and use topological sort in applications.
5. Construct different applications of disjoint sets.

UNIT-I

Sliding Window – Introduction- Applications – Naive Approach, Diet Plan Performance, Distinct Numbers in Each Subarray, Kth Smallest Subarray Sum, Maximum of all subarrays of size k.

Two Pointer Approach -Introduction –Palindrome Linked List, Find the Closest pair from two sorted arrays, Valid Word Abbreviation.

UNIT-II

Bit Manipulation: Introduction, Applications: Counting Bits, Palindrome Permutation, Remove All Ones with Row and Column Flips, Encode Number

UNIT-III

Tree Algorithms: Fenwick Tree, Segment Tree – Applications- Range Sum Queries, Treap – Applications – Kth Largest Element in an Array.

Trie: Introduction, Suffix Tree, Applications: Index Pairs of a String, Longest word with all prefixes, Top k frequent words.

UNIT-IV

Graphs Algorithms: Connected Components in a graph, Finding Bridges in a Graph and Finding Articulation Point in a Graph, Maximum Flow Algorithms, Lowest Common Ancestor.

Topological Sort: Introduction, Applications: Parallel Courses, Course Schedule.

UNIT-V

Disjoint Set Union: Disjoint set and its operations, Union Find Algorithm, Applications: Lexicographically Smallest Equivalent String, Number of Distinct Islands, Number of Connected Components in an Undirected Graph.

TEXT BOOKS:

1. Guide to Competitive Programming, Antti Laaksonen, 2017.
2. Competitive Programming 3, Steven Halim, Felix Halim, 2013.
3. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein, MIT Press, 2009.
4. The Algorithm Design Manual, Steven S, Skiena, Springer Verlag London Limited, 3rd Edition 2020.

REFERENCE BOOKS:

1. Algorithms, by Robert Sedgewick and Kevin Wayne, Addison Wesley, 4th Edition
2. An Introduction to the Analysis of Algorithms, Robert Sedgewick and Philippe Flajolet, Addison Wesley, 2nd edition



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
NEURAL NETWORKS (21CE611PE)
Common to CSE and IT
(Professional Elective-II)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CS305PC - Introduction to Machine Learning
2. 21MA406BS - Statistical Methods for Computer Science
3. PP204ES - Python Programming

Course Objectives: The course will help to

1. To make the participant comfortable with Artificial Neural Networks and Deep Learning concepts.
2. To implement Neural network algorithms with Tensorflow.
3. To implement Neural network algorithms with Keras.
4. To understand Image processing and analysis with CNN.
5. To understand sequence learning with RNN.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Develop ANN without using Machine Learning/Deep learning libraries.
2. Develop ANN model for tabular data.
3. Develop Neural network algorithms with Keras.
4. Develop image classification model using CNN.
5. Develop model for sequence learning using RNN.

Unit-I

Introduction to Artificial Intelligence: Artificial intelligence, Machine Learning, Neural Networks in Deep Learning, Shallow Neural Networks, Deep Learning, Characteristics, Applications, Deep Learning Framework.

Regression: Neuron Model, Optimization Method, Linear Model in Action.

Classification: Hand Written Digital Picture Dataset, Build a model, Error Calculation, Non-Linear model, model complexity, Optimization Method, Hands-On Hand-Written Digital Image Recognition.

Unit – II

TensorFlow: Data Types, Numerical Precision, Tensors to Be Optimized, Create Tensors, Typical Applications of Tensors, Indexing and Slicing, Dimensional Transformation, Broadcasting, Mathematical Operations, Merge and Split, Common Statistics, Tensor Comparison, Fill and Copy, Data Limiting, Advanced Operations, Load Classic Datasets.

Neural Networks: Perceptron, Fully Connected Layer, Neural Network, Activation function, Design of Output Layer, Error Calculation, Types of Neural Networks, Hands-On of Automobile Fuel Consumption Prediction.

UNIT-III

Backward Propagation Algorithm: Derivatives and Gradients, Common Properties of Derivatives, Derivative of Activation Function, Gradient of Loss Function, Gradient of Fully Connected Layer, Chain Rule, Back Propagation Algorithm, Hands-On Optimization of Himmelblau, Hands-On Back Propagation Algorithm, Hands-On Handwritten Digital Image Recognition.

Keras Advanced API: Common Functional Modules, Model Configuration, Training, and Testing, Model Configuration, Model Saving and Loading, Custom Network, Model Zoo, Metrics, Hands-On Accuracy Metric, Visualization.

Overfitting: Model Capacity, Overfitting and Underfitting, Dataset Division, Model Design, Regularization, Dropout, Data Augmentation, Hands-On Overfitting.

Unit-IV

Convolutional Neural Networks: Problems with Fully Connected, Convolutional Neural Network, Convolutional Layer Implementation, Hands-On LeNet-5, Representation Learning, Gradient Propagation, Pooling Layer, BatchNorm Layer, Classical Convolutional Network, Hands-On CIFAR10 and VGG, Convolutional Layer Variants, Deep Residual Network, DenseNet, Hands-On CIFAR10 and ResNet18.

Unit-V

Recurrent Neural Network: Sequence Representation Method, Recurrent Neural Network, Gradient Propagation, How to Use RNN Layers, Hands-On RNN Sentiment Classification, Gradient Vanishing and Gradient Exploding, RNN Short-Term Memory, LSTM Principle, How to Use the LSTM Layer, GRU, Hands-On LSTM/GRU Sentiment Classification.

Autoencoder: Principle of Autoencoder, Hands-On Fashion MNIST Image Reconstruction, Autoencoder Variants, Variational Autoencoder.

TEXT BOOKS:

1. Beginning Deep Learning with TensorFlow: Work with Keras, MNIST DataSets, and Advanced Neural Networks by Liangqu Long, Xiangming Zeng, A Press.
2. Deep Learning from the Basics, Koki Saitoh, Packt Publishing.

REFERENCE BOOKS:

1. Deep Learning Methods and Applications by Li Deng, Dong Yu, Now Publishers Inc, 2014.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
DESIGN PATTERNS (21CE612PE)
Common to CSE and IT
(Professional Elective-II)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CC502PC – Software Engineering Course
2. 21CS401PC – Java Programming Course

Course Objectives: The course will help to

1. Understand the concept of design patterns
2. Comprehend the use of design patterns in designing an interface
3. Recognize various creational patterns
4. Interpret various structural patterns
5. Elucidate different behavioural patterns

Course Outcomes: After learning the concepts of this course, the student is able to

1. Identify software designs that are scalable and easily maintainable
2. Examine and apply creational design patterns to class instantiation in software.
3. Use structural design patterns to improve the composition of classes and objects.
4. Use behavioural patterns to improve object organization and communication.
5. For effective code packaging, use refactoring to assemble the functions.

UNIT - I

Introduction: What is a design pattern? design patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

UNIT - II

Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look and Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary

UNIT - III

Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

UNIT - IV

Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

UNIT - V

Behavioural Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.

TEXT BOOK:

1. Design Patterns, Erich Gamma, Pearson Education, 1995

REFERENCE BOOKS:

1. Pattern's in Java, Vol-I, Mark Grand, Wiley Dream Tech, 2002.
2. Patterns in Java, Vol-II, Mark Grand, Wiley Dream Tech, 1999.
3. Java Enterprise Design Patterns Vol-III, Mark Grand, Wiley Dream Tech, 2002.
4. Head First Design Patterns, Eric Freeman, O'reilly publications, 2004.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
FRONT END WEB DEVELOPMENT (21CS621PE)
(Professional Elective-III)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CC503PC – Web Technologies Course

Course Objectives: The course will help to

1. Comprehend and apply the concepts of HTML, CSS and Bootstrap
2. Gain knowledge and implement the component-based architecture
3. Demonstrates the application and implementation of event and form handling mechanisms.
4. Practice and demonstrate the functionality of React Component life cycle and React-hooks
5. Analyse and integrate the UI with react routing and AJAX

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply HTML, CSS and Bootstrap effectively to create interactive and responsive website
2. Develop a component-based UI using ReactJS.
3. Apply event handling mechanism for the component-based UI
4. Comprehend, design and build a web application using React Hooks
5. Develop a weather application using routing and AJAX methodology.

UNIT-I

HTML5: Basics, Lists, Tables, working with links, HTML forms for user Inputs.

Cascading Style sheets: Introduction, Selectors, Fonts and Text Effect, Border and Box Effects, inline, internal and external styling.

Bootstrap: Introduction, Bootstrap components- Helper Classes & Responsive Utilities, Working with Buttons, List, Tables, Forms, Typography, Page Header, Button Groups, navs, modals, grid system.

UNIT-II

React-JS Introduction: Introduction, Hello World app in React, React App Project Directory Structure React Component Basic, Understanding JSX, Limitations of JSX, Original DOM vs Virtual DOM, React Components with JSX, Class Components, Functional Components.

Props and State: Understanding and using Props and State, Set State in Depth, Destructuring Props and State, Functional (Stateless) VS Class (Stateful) Components, Parent – Child Communication.

UNIT-III

Event Handling: Basics of Event Handling, Binding Event Handlers, Methods as Props, Conditional Rendering, Form **handling:** Creating a Custom Dynamic Input Component, Setting Up a JS Config for the Form, Dynamically Create Inputs based on JS Config, Handling User Input, Handling Form Submission

UNIT-IV

React Hooks: Introduction to Hooks, important hooks-useState, useEffect, useContext, useRef, useMemo and custom hooks.

React Component life cycle: Updating life cycle hooks, Pure Components, React's DOM Updating, Fragments, React Higher Order Components

UNIT-V

React Routing: Routing and SPAs, Setting Up the Router Package, react-router vs react-router-dom, switching between Pages, passing and extracting route/query parameters, Using Switch to Load a Single Route

HTTP Requests/Ajax Calls: HTTP Requests in React, Introduction of Axios package, HTTP GET Request, fetching data.

TEXTBOOKS:

1. Web Programming with Html, CSS, Bootstrap, Javascript, JQuery, Php, and Mysql, Larry Sanchez, 2017
2. The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React.Js.
3. Beginning React with Hooks, Greg Lim

REFERENCE BOOKS:

1. HTML CSS Bootstrap JavaScript JQuery for absolute beginners, Anne Pryor, 2021.
2. React.js, Learning React Javascript Library from Scratch, Greg Sidelnikov, Learning Curve, 2015.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
SOFTWARE TESTING METHODOLOGIES (21CE621PE)
Common to CSE and CSE (AI&ML)
(Professional Elective-III)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. 21CC502PC – Software Engineering course.

Course Objectives: The course will help to

1. To provide knowledge of the concepts in software testing
2. To provide knowledge on testing process, criteria, strategies, and methodologies.
3. To discuss various software testing issues and solutions in software unit test, integration and system testing.
4. To expose the advanced software testing topics, such as object-oriented software testing methods.
5. To develop skills in software test automation and management using latest tools.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline the necessity of testing, debugging using program control flow.
2. Apply transaction flow, data flow testing to unit and integration testing. Able to test a domain or an application and identify the nice and ugly domains.
3. Apply reduction procedures to a control flowgraph and simplify it into a single path expression and the use of decision tables in testcase design.
4. Understand the testing of state graphs.
5. Analyze graph matrices for optimizing the code and use of testing tools like WinRunner, JMeter.

UNIT – I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.

Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: transaction flows, transaction flow testing techniques.

Dataflow testing: Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III

Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.



B.Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Course Syllabus (KR21)
INTERNET OF THINGS (21IT601PC)
OPEN ELECTIVE-I

L	T	P	C
3	0	0	3

Prerequisite/Corequisite:

1. PP204ES –PythonProgrammingLabCourse.

Course Objectives: The course will help to

1. Understand the concepts of Internet of Things and able to build IoT Applications.
2. Understand the need of Sensors & Actuators in IoT.
3. Learn the programming and use of Arduino boards.
4. Learn Raspberry Pi python programming and use of Raspberry Pi boards.
5. Learn interfacing I/O devices with ESP32.

Course outcomes: After learning the concepts of this course, the student is able to

1. Identify basic concepts of Internet of Things in real world applications.
2. Identify the need and working of Sensors & Actuators in IoT
3. Practice Programs and configure Arduino boards for various designs.
4. Practice Raspberry Pi Python programming and interfacing I/O devices with Raspberry Pi.
5. Design IoT Applications in different domains using Raspberry Pi.

UNIT-I

Introduction to IOT: Definition & Characteristics of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Communication models & APIs, IoT Enabling Technologies, Domain Specific Applications of IOT: Home automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health and lifestyle.

UNIT –II

IoT Sensing & Actuation: Introduction to sensors, sensor characteristics, sensorial deviations, sensing types, Temperature & Humidity Sensor, PIR Sensor, Ultrasonic Sensor, LDR, sensing considerations, Actuators, Actuator types, LED, Switch, Picamera, Actuator considerations

UNIT–III

Arduino Programming: This is Arduino, getting started, Arduino Programming: variables, Conditional execution, loops, functions, arrays and strings, input and output, the serial monitor, Integration of Sensors and Actuators with Arduino: Controlling LED, integration of an LED and Switch, integration of a LED and light sensor (LDR), Interfacing LCD.

UNIT IV

IoT Physical Devices & Endpoints: what is an IoT Device: Basic building blocks of an IoT device, Introduction to Raspberry: about the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces (serial, SPI, I2C), Programming Raspberry Pi with Python: Controlling LED, integration of LED and Switch, interfacing an Ultrasonic Sensor, PIR sensor, DHT22 sensor, Picamera.

UNIT- V

Implementation of IoT with Raspberry Pi: Implementation of data sharing between server & client, Fire base implementation, Traffic light control system, IoT applications: implementation of Home automation, Healthcare, Smart device

TEXT BOOKS:

1. Internet of Things-A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015.
2. Introduction to IoT- Sudip Misra, Anandarup Mukherjee, Arijit Roy, Cambridge University Press, 2021
3. Introduction to Internet of Things- Sudip Misra, IIT Kharagpur https://onlinecourses.nptel.ac.in/noc22_cs53/unit?unit=113&lesson=114
4. Programming Arduino Next Steps: Going Further with Sketches- Simon Monk, 2nd Edition, 2019.
5. Getting Started with Raspberry Pi- Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014.
6. <https://projects.raspberrypi.org>

REFERENCE BOOKS:

1. Beginning Sensor networks with Arduino and Raspberry Pi- Charles Bell, Apress, 2013.
2. The Internet of Things: Enabling Technologies, Platforms, and Use Cases- Pethuru Raj and Anupama C. Raman (CRC Press)



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



B.Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Course Syllabus (KR21)
INTRODUCTION TO EMBEDDED SYSTEMS (21IT601OE)
OPEN ELECTIVE-I

L	T	P	C
3	0	0	3

Prerequisite/ Corequisites:

1. 21CS304ES-AnalogandDigitalElectronicsCourse.
2. 21CS302PC -Computer Organization and Architecture Course.

Course Objectives: The course will help to

1. Provide an over view of principles of Embedded System.
2. Gain knowledge on Typical Embedded System.
3. Provide a clear understanding Embedded Firmware Design and Development.
4. Gain knowledge on operating systems.
5. Provide the knowledge on role offirm ware,operating systems in correlation with hardware systems.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Understand the selection procedure of processors in the embedded domain.
2. Design procedure of embedded firmware.
3. Develop Firmware using Embedded C programming.
4. Analyze the role of real time operating systems in embedded systems.
5. Evaluate the correlation between task synchronization and latency issues

UNIT – I

Introduction to Embedded Systems: Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification of Embedded Systems, Major application areas, Purpose of E bedded Systems, Characteristics and Quality attributes of Embedded Systems.

UNIT – II

The Typical Embedded System: Core of the Embedded System, Memory, Sensors and Actuators, Communication Interface, Embedded Firmware, Other System components.

UNIT – III

Embedded Firmware Design and Development: Embedded Firmware Design, Embedded Firmware Development Languages, Programming in Embedded C.

UNIT - IV

RTOS Based Embedded System Design: Operating System basics, Types of Operating Systems, Tasks, Process, Threads, Multiprocessing and Multi-tasking, Task Scheduling, Threads-Processes-Scheduling putting them together, Task Communication, Task Synchronization, Device Drivers, How to choose an RTOS

UNIT – V

Integration and Testing of Embedded Hardware and Firmware: Integration of Hardware and Firmware, Boards Bring up The Embedded System Development Environment: The Integrated Development Environment (IDE), Types of files generated on Cross-Compilation, Disassembler/Decompiler, Simulators, Emulators and Debugging, Target Hardware Debugging, Boundary Scan.

TEXT BOOKS:

1. Shibu K V, “Introduction to Embedded Systems”, Second Edition, Mc Graw Hill

REFERENCE BOOKS:

1. Rajkamal, Embedded Systems Architecture, Programming and Design, TATA McGraw-Hill
2. Frank Vahid and Tony Givargis, “Embedded Systems Design” - A Unified Hardware/Software Introduction, John Wiley.
3. Lyla, “Embedded Systems” –Pearson
4. David E.Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year II Semester Syllabus (KR21)

CLOUD COMPUTING LAB (21CC602PC)

Common to CSE and IT

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. CS402PC – Database Management Systems Course
2. CS403PC – Operating Systems Course
3. CS306PC – Python Programming Lab Course

Course Objectives: The course will help to

1. Understand the working definition of the AWS Cloud
2. Become familiar with Amazon's storage offerings
3. Understand machine learning models
4. Understand event driven architectures
5. Use tools, gain a better understanding of Continuous Integration and Continuous Delivery Pipeline.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Identify the main concepts, key technologies, strengths and limitations of cloud computing.
2. Illustrate the key and enabling technologies that help in the development of the cloud.
3. Develop the ability to understand and use the architecture of compute and storage cloud, service and delivery models.
4. Analyze the core issues of cloud computing such as resource management and security.
5. Develop PAAS and CICD applications

Software to be used: The students can use any OS with AWS account free 12 months services.

List of Experiments:

1. Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.
2. Create an AWS Free Tier account and launch your first virtual servers
3. (Amazon EC2 instances) on the AWS Cloud
4. Create your First AWS S3 Bucket and Upload Content to Bucket and Manage their Access and Create Static Website using AWS S3
5. Create and configure storage services and upload files and objects using
6. Amazon EBS, Amazon EFS and Amazon S3
7. Create, Train and Deploy a simple Machine Learning Model (using python) using AWS Amazon Sage maker (use P type instance)
8. Create visual search & image recognition capability application using AWS Amazon Recognition.
9. Implement serverless computing and Docker containers on AWS using AWS Lambda and Amazon ECS access the services with amazon API gateway
10. Create a Platform as a Service (PaaS) application on AWS Elastic Beanstalk
11. Create serverless event-driven architectures on Lambda to run backend ML application integrated with frontend React component.
12. Launch a relational database on Amazon RDS and a NoSQL database using Amazon DynamoDB
13. Create loosely coupled services with Amazon SQS and Amazon SNS to process data received from the applications.

14. Build a continuous integration and continuous delivery pipeline using DevOps tools on AWS to automate a (CI/CD) pipeline
15. Create 3D scenes with AWS Sumerian Artificial humans that can serve as help desk assistants integrating Amazon LEX and embed into web pages.

TEXT BOOKS:

1. Learn AWS Serverless Computing: A beginner's guide to using AWS Lambda, Amazon API Gateway, and services from Amazon Web Services - by Scott Patterson, 2019.
2. Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud - by Mark Wilkins, 2019
3. Microsoft Azure For Dummies 1st Edition - by Timothy L. Warner, 2020

REFERENCE BOOKS:

1. Essentials of Cloud Computing- k. Chandrasekharan, 2014.
2. Cloud computing principles and paradigms by Rajkumar Buyya, 2013.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
COMPETITIVE PROGRAMMING LAB (21CS602PC)

L	T	P	C
0	0	2	1

Pre-requisites/ Co-requisites:

1. 21CS405PC–Java Programming Lab Course
2. 21CS601PC – Competitive Programming Course

Course Objectives: The course will help to

1. Understand Sliding Window applications.
2. Understand Bit Manipulation problems.
3. Understand different types of trees and applications.
4. Know how to work with graph algorithms.
5. Solve problems with disjoint sets.

Course Outcomes: After learning the concepts of this course, the student will be able to

1. Design and implement solutions for arrays.
2. Design and implement solutions for different trees.
3. Design solutions for graph applications.
4. Design solutions for compression techniques.
5. Design solutions for disjoint set applications.

Software to be used: The students can use any OS with Java latest version.

List of Programs:

1. Write a java program
 - a. to find Subarrays with K Different Integers
 - b. to find shortest sub array with sum at least K
 - c. to implement Fenwick Tree
 - d. to implement a segment tree with its operations
 - e. to implement treap with its operations
 - f. to find a permutation of the vertices (**topological order**) which corresponds to the order defined by all edges of the graph.
 - g. to find all the articulation points of a graph.
 - h. to check whether the permutation of a string forms a palindrome
 - i. to return all index pairs [i,j] given a text string and words (a list of strings), so that the substring text[i]..text[j] is in the list of words.
 - j. to find the lowest common ancestor of a binary tree.
 - k. to find the Longest Increasing Path in a Matrix.
2. Develop a java program to find the Lexicographically smallest equivalent string

TEXTBOOKS:

1. Guide to Competitive Programming, Antti Laaksonen, 2017.
2. Competitive Programming 3, Steven Halim, Felix Halim, 2013.
3. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein, MIT Press, 2009.
4. The Algorithm Design Manual, Steven S, Skiena, Springer Verlag London Limited, 3rd Edition 2020.

REFERENCE BOOKS:

1. Algorithms, by Robert Sedgewick and Kevin Wayne, Addison Wesley, 4th Edition
2. An Introduction to the Analysis of Algorithms, Robert Sedgewick and Philippe Flajolet, Addison Wesley, 2nd edition



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
FRONT END WEB DEVELOPMENT LAB (21CS631PE)
(Professional Elective-III Lab)

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. 21CS621PE – Front End Web Development Course

Course Objectives: The course will help to

1. Comprehend and apply the concepts of HTML, CSS and Bootstrap
2. Gain knowledge and implement the component-based architecture
3. Demonstrates the application and implementation of event and form handling mechanisms.
4. Practice and demonstrate the functionality of React Component life cycle and React-hooks
5. Analyze and integrate the UI with react routing and AJAX

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply HTML, CSS and Bootstrap effectively to create interactive and responsive portfolio website
2. Develop a component-based web UI for site using ReactJS.
3. Apply event handling mechanism for the component-based UI
4. Comprehend, design and build a web application using React Hooks
5. Develop a web application using routing and AJAX methodology.

Software to be used: The students can use Visual Studio Editor, Internet.

List of Experiments:

1. Create your portfolio page using HTML tags
2. Update the portfolio page by adding styles to it. Add a contact Us form as well.
3. Update the same page using external style sheets or move the added styles to a CSS file and add it to the HTML page
4. Update the portfolio page to use Bootstrap.
5. Create Home, About Us, Products, Partners, Contact Us pages for a static e-commerce website using Bootstrap. Add a gallery page using Bootstrap grid model. Add Navbar, header and footer to all pages. Add a Login link (Modal).
6. Add a weather component to the home page of the above website, using ReactJS.
7. Update the 'contact us' page to a ReactJS component. An email should be sent each time contact us form is submitted
8. Add a Login component using ReactJS. User should be able to login, using pre-defined logins. Session should be maintained using react hooks
9. Each page in the site should be a React JS component and the elements in each page should be child components.
10. React Routing should be used to take care of routing from browser location bar and also from Navbar

TEXTBOOKS:

1. Web Programming with Html, CSS, Bootstrap, Javascript, JQuery, Php, and Mysql, Larry Sanchez, 2017
2. The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React.Js.
3. Beginning React with Hooks, Greg Lim

REFERENCE BOOKS:

1. HTML CSS Bootstrap JavaScript JQuery for absolute beginners, Anne Pryor, 2021
2. React.js, Learning React Javascript Library from Scratch, Greg Sidelnikov, Learning Curve, 2015



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
SOFTWARE TESTING METHODOLOGIES LAB (21CE631PE)
Common to CSE and CSE (AI&ML)
(Professional Elective-III Lab)

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. A basic knowledge of programming.

Course Objectives: The course will help to

1. To provide knowledge of Software Testing Methods.
2. To develop skills in software test automation and management using latest tools.
3. To expose the advanced software testing topics, such as object-oriented software testing methods.
4. To develop skills in software test automation and management using latest tools.
5. To discuss various software testing issues and solutions in software unit test, integration and system testing.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Design and develop the best test strategies in accordance to the development model.
2. Outline the necessity of testing, debugging using program control flow.
3. Apply transaction flow, data flow testing to unit and integration testing. Able to test a domain or an application and identify the nice and ugly domains.
4. Understand the testing of state graphs.
5. Analyze graph matrices for optimizing the code and use of testing tools like WinRunner, JMeter.

Software to be used: The students must use Selenium.

List of Experiments:

1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
 - a) Bitmap checkpoint for object/window
 - b) Bitmap checkpoint for screen area
5. Database checkpoint for Default check
6. Database checkpoint for custom check
7. Database checkpoint for runtime record check
 - a) Data driven test for dynamic test data submission
 - b) Data driven test through flat files
 - c) Data driven test through front grids
 - d) Data driven test through excel test
 - e) Batch testing without parameter passing
 - f) Batch testing with parameter passing
8. Data driven batch
9. Silent mode test execution without any interruption
10. Test case for calculator in windows application

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
ENVIRONMENTAL SCIENCE (*MC209ES)

L	T	P	C
3	0	0	0

Pre-requisite/Co-requisite: NIL

Course Objectives: The course will help to

1. Gain a Better Understanding of the Ecosystem.
2. Understanding of Natural and Mineral Resources.
3. Recognizing Biodiversity and Biotic Resources effects.
4. Environmental Pollution and Control Technologies.
5. Becoming acquainted with environmental policies and regulations.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Recognize the ecological system.
2. Explain the difference between natural and mineral resources.
3. Demonstrate the understanding of biodiversity and biotic resources.
4. Analyze the effects of pollution on the environment and the technologies that can be used to mitigate it.
5. Develop environmental policies and laws to be implemented.

UNIT-I

Ecosystems: Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Bio-magnification, ecosystem value, services and carrying capacity, Field visits.

UNIT-II

Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy resources: growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT-III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT-IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **Soil Pollution:** Sources and types, Impacts of modern agriculture, degradation of soil. **Noise Pollution:** Sources and Health hazards, standards, Solid waste: Municipal Solid Waste management, composition and characteristics of e- Waste and its management. **Pollution control technologies:** Wastewater Treatment methods: Primary, secondary and Tertiary. Overview of air pollution control technologies, Concepts of bioremediation. **Global Environmental Issues and Global Efforts:** Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

UNIT-V

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan (EMP). Towards Sustainable Future: Concept of Sustainable Development Goals, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

TEXT BOOKS:

1. Textbook of Environmental Studies for Undergraduate Courses, ErachBharucha for University Grants Commission.
2. Environmental Studies. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future, Richard T. Wright. PHL Learning Private Ltd. New Delhi, 2008.
2. Environmental Engineering and science, Gilbert M. Masters and Wendell P. Ela. PHI Learning Pvt. Ltd, 2008.
3. Environmental Science, Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies, Anubha Kaushik, 4th Edition, New age international publishers, 2010.
5. Text book of Environmental Science and Technology, Dr. M. Anji Reddy, BS Publications, 2007.



B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year II Semester Syllabus (KR21)
CYBER SECURITY (*MC601CS)
Common to CSE, IT, CSE (AI&ML) and CSE (DS)

L	T	P	C
3	0	0	0

Pre-requisites/ Co-requisites:

1. 21CS501PC - Computer Networks Course

Course Objectives: The course will help to

1. Understand cybercrime and Information Security
2. Understand cyber offences and their planning
3. Understand cybercrime in mobile and wireless devices
4. Understand the tools of cyber crime
5. Understand the implications of cyber crime

Course Outcomes: After learning the concepts of this course, the student is able to

1. Outline cybercrimes and legal perspectives
2. Identify cyber stalking, fuel of cyber crime
3. Illustrate cybercrime in mobile devices, wireless devices
4. Implement the methods used in cyber crime
5. Identify the cyber threats for organizations

UNIT - I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Spectrum of attacks, Taxonomy of various attacks, IP spoofing, Methods of defense, Security Models, risk management, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy. Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics, Special Techniques for Forensics Auditing.

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops.

UNIT- IV

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations.

Cybercrime and Cyber terrorism: Introduction, intellectual property in the cyberspace, the ethical dimension of cybercrimes the psychology, mindset and skills of hackers and other cyber criminals.

UNIT - V

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc.

Cybercrime: Examples and Mini-Cases

Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances.

Mini-Cases: The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

TEXT BOOKS:

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley
2. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335, 2018.

REFERENCE BOOKS:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu, J. David Irwin, CRC Press T&F Group.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year II Semester Syllabus (KR21)

ROBOTIC PROCESS AUTOMATION (21CS601OE)

Common to CSE, IT, CSE (AI&ML) and CSE (DS)

(Open Elective – I)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites:

1. *MC501AI–Artificial Intelligence Course

Course Objectives: The course will help to

1. To introduce the concepts of Robotic system, its components and instrumentation and control related to robotics.
2. Understand RPA and identify the fundamental capabilities, benefits, and risks of RPA
3. Understand Intelligent Automation and its impact on the transformation of business.
4. Apply the technologies and best practices used to enable process automation.
5. Describe how to expand the benefits of RPA by combining it with artificial intelligence and other emerging technologies.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Describe RPA, where it can be applied and how it's implemented.
2. Identify and understand Web Control Room and Client Introduction
3. Understand how to handle various devices and the workload
4. Understand Bot creators, Web recorders and task editors
5. Identify areas where Intelligent Automation is applicable and formulate its value (quantify and qualify).

UNIT – I

Introduction to Robotic Process Automation & Bot Creation: Introduction to RPA and Use cases – Automation Anywhere Enterprise Platform – Advanced features and capabilities – Ways to create Bots

UNIT - II

Web Control Room and Client Introduction: Features Panel - Dashboard (Home, Bots, Devices, Audit, Workload, Insights) - Features Panel – Activity (View Tasks in Progress and Scheduled Tasks) - Bots (View Bots Uploaded and Credentials)

UNIT - III

Devices (View Development and Runtime Clients and Device Pools) : Workload (Queues and SLA Calculator) - Audit Log (View Activities Logged which are associated with Web CR) - Administration (Configure Settings, Users, Roles, License and Migration) - Demo of Exposed API's – Conclusion – Client introduction and Conclusion.

UNIT - IV

Bot Creator Introduction: Recorders – Smart Recorders – Web Recorders – Screen Recorders - Task Editor – Variables - Command Library – Loop Command – Excel Command – Database Command - String Operation Command - XML Command

UNIT - V

Terminal Emulator Command: PDF Integration Command - FTP Command - PGP Command - Object Cloning Command - Error Handling Command - Manage Windows Control Command - Workflow Designer - Report Designer

TEXT BOOKS:

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath: Create Software robots. with the leading RPA tool – UiPath Kindle Edition.

REFERENCE BOOKS:

1. Robotic Process Automation A Complete Guide - 2020 Edition Kindle Edition.