



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### Course Structure (KR20)

Applicable from 20-21 Admitted Batch

### III YEAR I SEMESTER

Sr. No.	Course Code	Course Title	L	T	P	Credits
1	CS501PC	Design and Analysis of Algorithms	3	1	0	4
2	CS502PC	Software Engineering	3	0	0	3
3	CS503PC	Web Technologies	3	1	0	4
4	CS504PC	Computer Networks	3	0	0	3
5		Professional Elective-I	3	0	0	3
6	CS505PC	Software Engineering Lab	0	0	3	1.5
7	CS506PC	Web Technologies Lab	0	0	4	2
8	CS507PC	Computer Networks Lab	0	0	3	1.5
9	CS512HSM	Professional and Soft Skills Lab	0	0	2	1
		<b>Total Credits</b>	<b>15</b>	<b>2</b>	<b>12</b>	<b>23</b>

### Professional Elective-I

CS511PE	Introduction to Machine Learning
CS512PE	Human Computer Interaction
CS513PE	Data Analytics
CS514PE	Game Theory



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### Course Structure (KR20)

Applicable from 20-21 Admitted Batch

### III Year II Semester

Sr. No.	Course Code	Course Title	L	T	P	Credits
1	CS601PC	Cloud Computing	3	0	0	3
2	CS602PC	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	CS603PC	Cloud Computing Lab	0	0	3	1.5
7	CS604PC	Competitive Programming Lab	0	0	2	1
8		Professional Elective- III Lab	0	0	3	1.5
9	*MC609	Environmental Science	3	0	0	0
		<b>Total Credits</b>	<b>15</b>	<b>0</b>	<b>8</b>	<b>19</b>

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

\* Mandatory Course

**Note:** Industrial Oriented Mini Project/ Summer Internship is to be carried out during the summer vacation between 6th and 7th semesters.

### Professional Elective-II

CS611PE	Parallel Programming
CS612PE	Artificial Intelligence
CS613PE	Cyber Security
CS614PE	Design Patterns

## Professional Elective-III

CS621PE	Front-End Web Development
CS622PE	Mobile Application Development
CS623PE	Linux Programming
CS624PE	Cryptography and Network Security

## Professional Elective-III Lab

CS631PE	Front-End Web Development Lab
CS632PE	Mobile Application Development Lab
CS633PE	Linux Programming Lab
CS634PE	Cryptography and Network Security Lab

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

## Open Elective - I

CM600OE	Intelligent Database Systems
CD600OE	ANN and Deep Learning
EI600OE	Basics of Sensor Technology



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

### DESIGN AND ANALYSIS OF ALGORITHMS (CS501PC)

L	T	P	C
3	1	0	4

#### Prerequisites/ Corequisites:

1. CS203ES -Programming for Problem Solving Course
2. CS303PC -Data Structures Course
3. CS401PC- 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. Analyze 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, Strobogrammatic 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.

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

#### 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.

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

#### **UNIT-IV**

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

#### **UNIT-V**

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

**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, SatrajSahni and Rajasekharan, 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/LINKS:**

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.
4. <https://leetcode.com/problemset/algorithms/>



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

### SOFTWARE ENGINEERING (CS502PC)

L	T	P	C
3	0	0	3

**Prerequisite/ Corequisite:** NIL

**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, the capability maturity model integration (CMMI), process assessment, personal and team process models.

**Process models:** The waterfall model, incremental process models, evolutionary process models, the unified process.

**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.

**System models:** Context models, behavioral models, data models, object models.

#### UNIT - III

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

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

**Build Tools and Continuous Integration using Jenkins:** Introduction to Maven - Installation and configuration, Gradle and webpack bundler, 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

**Product metrics:** Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

#### UNIT-V

**Estimation of Software Projects:** Software project estimation, Decomposition techniques, Cost Estimation Models, and the make/buy decision

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

**Introduction to Cloud and AWS:** Understand and create cloud infrastructure using AWS, and deploy the web application.

#### TEXTBOOKS:

1. Software Engineering, A practitioner's Approach-Roger S. Pressman, 6<sup>th</sup> 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.
5. <http://khitguntur.ac.in/images/syllabus/btech/B.Tech%20R19%204-2%20sem.pdf>



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

### WEB TECHNOLOGIES (CS503PC)

L	T	P	C
3	1	0	4

#### Prerequisites/ Corequisites:

1. CS401PC–Java Programming Course
2. CS402PC – 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 Express JS.
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 Express JS.

#### 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 Express JS.
4. Analyze and Implement routing methodologies using Express JS.
5. Design and create a NoSQL (Mongo)database and then integrate it with the front end using Express JS.

#### UNIT-I

**Client-Side Scripting:** Basic Java Script, Strings and regular expressions, functions, DE structuring, arrays, sets and maps, Iterators and Generators, Promises and Asynchronous operations.

#### UNIT-II

**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-III

**Introduction to Server Side:** CGI using Express JS, life cycle and routing of Express App, deploying to Node. Handling request and response parameters, reading configuration parameters. Connecting to Database (Mongo DB)

#### UNIT-IV

**Introduction to Middleware:** Express Router methods, Route paths, Route parameters, Route handlers, Response methods

#### UNIT-V

**Introduction to No-SQL Databases:** Importing, exporting and querying – creating and manipulating documents. Advanced CRUD operations. 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 Course Syllabus (KR20)

### COMPUTER NETWORKS (CS504PC)

L	T	P	C
3	0	0	3

#### Prerequisite/ Corequisite:

1. CS403PC - 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, HDLC, Point to Point Protocols. Medium Access sub layer: Multiple Access Protocols- Random Access Protocols- ALOHA, CSMA/CD, CSMA/CA, Controlled Access Protocols- Reservation, Polling, Channelization, LAN, Ethernet IEEE 802.3, IEEE 802.11.

#### UNIT – III

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

**Internetworking-** IPV4 and IPV6 delivery mechanisms, Tunneling, Address mapping- ARP, RARP, Bootp, DHCP, ICMP, IGMP.

**Routing** – Distance Vector Routing, Link State Routing, Path Vector Routing, Protocols- RIP, OSPF, BGP. Multicast Routing Protocols- Multicast Distance Vector Routing, Multicast linkstate routing, Core Based Tree. **Transport Layer:** Process to Process Delivery, UDP and TCP protocols, TCP Connection Control, Data Traffic. Congestion Control- Open loop Congestion Control, Closed Loop Congestion Control, Congestion control in TCP. QoS- Techniques to improve QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

## UNIT – IV

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

**Socket options:** getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.

**Advanced I/O Functions-**Introduction, Socket Timeouts, recv and send Functions, readv and writev Functions, recvmsg and sendmsg Functions, Ancillary Data, How Much Data Is Queued?, Sockets and Standard I/O, T/TCP: TCP for Transactions.

## UNIT – V

**Raw Sockets:** Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Trace route Program, An ICMP Message Daemon, Data link Access- Introduction, BPF: BSD Packet Filter, DLPI: Data Link Provider Interface.

**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.

**Introduction to Security:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security.

## TEXT BOOKS:

1. Data Communications and Networking-Behrouz A. Forouzan, Fourth Edition TMH,2006.
2. Computer Networks- Andrew S Tanenbaum, 4<sup>th</sup> 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 1<sup>st</sup> Edition.

## REFERENCE BOOKS:

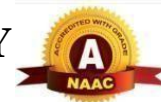
1. Data communications and Computer Networks- P.C Gupta, PHI.
2. An Engineering Approach to Computer Networks-S. Keshav, 2<sup>nd</sup> Edition, Pearson Education.
3. Understanding communications and Networks- W.A. Shay, Cengage Learning 3<sup>rd</sup> Edition.
4. Computer Networking: A Top-Down Approach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



**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTE)

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



## **B. Tech. in COMPUTER SCIENCE AND ENGINEERING**

### **III Year I Semester Course Syllabus (KR20)**

### **INTRODUCTION TO MACHINE LEARNING (CS511PE)**

#### **(Professional Elective – I)**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Prerequisites/ Corequisites:**

1. MA405BS- Probability Theory and Statistics Course
2. CS306PC – Python Programming Lab Course

#### **Course Objectives: The course will help to**

1. Introduce basic concepts of Probability and Machine Learning.
2. Introduce Descriptive Statistics and data analysis along with visualization
3. Gain knowledge on Regression analysis.
4. Learn about Classification Techniques.
5. Develop knowledge of non-Parametric machine learning algorithms and SVMs

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

1. Outline the basic concepts of Probability and Machine Learning
2. Develop the Statistics and data analysis along with visualization
3. Implement the different types of regression models.
4. Implement the classification model for categorical data.
5. Analyze and develop the non-Parametric models.

#### **UNIT – I**

**Introduction:** What is Machine Learning, Use Machine Learning, and Types of Machine Learning Systems: supervised, unsupervised, semi-supervised, Reinforcement Learning, Batch and Online Learning, Main Challenges of Machine Learning.

#### **UNIT – II**

**Descriptive Statistics:** Data representation, types of data- nominal, ordinal, interval and continuous, central tendency- calculating mean mode median, mean vs median, variability, variance, standard deviation, Mean Absolute Deviation using sample dataset, finding the percentile, inter-quartile range, Box Plot, Outlier, whisker, calculating correlation, covariance, causation. Exploratory data analysis, Data preparation and preprocessing, Data visualization.

#### **UNIT – III**

**Regression:** Introduction to Regression analysis, measure of linear relationship, Regression with stats models, determining coefficient, meaning and significance of coefficients, coefficient calculation with least square method, Types of regression, Simple Linear Regression, Using Multiple features, Polynomial Regression, Metrics for Regression: MSE, RMSE, MAE.

#### **UNIT – IV**

**Classification:** Classification problem, Probability based approach, Logistic Regression- log-odd, sigmoid transformation, Metrics: Confusion Matrix, Accuracy, Error Rate, Precision, Recall, ROC curve, F1 score, and introduction to gradient descent.

## **UNIT – V**

**Non-Parametric& SVM classification:** About Non parametric classification, Decision Trees: Entropy, gain ratio, Information Gain, Splitting criteria,

**Ensemble Method:** Introduction to Random Forest, Accuracy measure & performance

**Instance based learning-** Introduction, KNN algorithm, Distance measures, model building, locally weighted regression, radial basis functions, SVM classifier, hyper-plane, slack variables, geometric transformation kernel trick, kernel transformation.

### **TEXT BOOKS:**

1. Hands-On Machine Learning with Scikit-Learn and Tensor Flow -Aurélien Géron,O'Reilly Media, 2017.
2. Practical Python Data Visualization: A Fast Track Approach to Learning Data Visualization with Python, Ashwin Pajankar, A Press.
3. Python: End-to-end Data Analysis - Phuong Vo.T.H, Martin Czygan, Ivan Idris, Magnus Vilhelm Persson, Luiz Felipe Martins, Packet Pub.

### **REFERENCE BOOKS:**

1. Machine Learning in Action, Peter Harrington, Manning Publications.
2. Python for Data Analysis–wes McKinny, 2nd Edition, O'REILLY Publications.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

### HUMAN COMPUTER INTERACTION (CS512PE)

#### (Professional Elective – I)

L	T	P	C
3	0	0	3

**Prerequisites/ Corequisites:** Nil

**Course Objectives:** The course will help to

1. Gain an overview of Human-Computer Interaction (HCI), with an understanding of user interface design in general, and alternatives to traditional "keyboard and mouse" computing.
2. Become familiar with the vocabulary associated with sensory and cognitive systems as relevant to task performance by humans; be able to apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks and recognize the limits of human performance as they apply to computer operation.
3. Recognise the significance of a design and evaluation methodology that begins with and maintains a focus on the user; be familiar with a variety of both conventional and non-traditional user interface paradigms, the latter including virtual and augmented reality, mobile and wearable computing, and ubiquitous computing.
4. Recognise the social implications of technology and their ethical responsibilities as engineers in the design of technological systems.
5. Understand the working in small groups on a product design from start to finish will provide you with invaluable team-work experience.

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

1. Outline and apply HCI and principles to interaction design.
2. Select and design certain tools for blind or PH people.
3. Operate with windows navigation and devices.
4. Practice HCI in software process.
5. Synthesize to design connection between menus and windows.

#### UNIT - I

**Introduction: Importance of user Interface** – definition, importance of good design. Benefits of good design. A brief history of Screen design. The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

#### UNIT - II

**Design process** – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business junctions. Screen Designing: Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

#### UNIT- III

**Windows** – New and Navigation schemes selection of window, selection of devices based and screen-based controls. Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

#### UNIT- IV

**HCI in the software process, the software life cycle Usability engineering Iterative design and prototyping Design Focus-** Prototyping in practice Design rationale Design rules Principles to support usability Standards Golden rules and heuristics HCI patterns Evaluation techniques, Goals of evaluation, Evaluation through expert analysis, Evaluation through user participation, Choosing an evaluation method. Universal design, Universal design principles Multi-modal interaction

## **UNIT- V**

**Cognitive models Goal and task hierarchies Design Focus-** GOMS saves money Linguistic models The challenge of display-based systems Physical and device models Cognitive architectures Ubiquitous computing and augmented realities Ubiquitous computing applications research Design Focus: Ambient Wood – augmenting the physical Virtual and augmented reality Design Focus: Shared experience Design Focus: Applications of augmented reality Information and data visualization Design Focus: Getting the size right.

### **TEXT BOOKS:**

1. The essential guide to user interface design, Wilbert O Galitz, Wiley Dream Tech, 2007.
2. Human – Computer Interaction. Alan Dix, Janet Fincay, GreGoryd, Abowd, Russell Bealg, Pearson Education, 2004

### **REFERENCE BOOKS:**

1. Designing the user interface. 3rd Edition Ben Shneidermann, Pearson Education Asia, 1998.
2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech, 2005.
3. User Interface Design, Soren Lauesen , Pearson Education, 2003.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

#### DATA ANALYTICS (CS513PE)

#### (Professional Elective – I)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. MA405BS – Probability Theory and Statistics Course
2. CS402PC – 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, analyze 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 Course Syllabus (KR20)**  
**GAME THEORY (CS514PE)**  
**(Professional Elective – I)**

L	T	P	C
3	0	0	3

**Prerequisites/ Corequisites:**

1. MA101BS – Mathematics- I Course

**Course Objectives: The course will help to**

1. Understand the concept of games.
2. Understand cooperative games and algorithmic issues.
3. Apprehend extensive games and their form.
4. Recognize Nash and correlated equilibria in games.
5. Specify utility functions.

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

1. Analyze games based on complete and incomplete information about the players.
2. Analyze games where players cooperate.
3. Compute Nash equilibrium.
4. Apply game theory to model network traffic.
5. Analyze auctions using game theory.

**UNIT-I**

Games, Old and New; Games, Strategies, Costs, and Payoffs; Basic Solution Concepts Finding Equilibria and Learning in Games, Refinement of Nash: Games with turns and Subgame Perfect Equilibrium; Nash Equilibrium without Full Information: Bayesian Games;

**UNIT-II**

Cooperative Games, Markets and Their Algorithmic Issues, Is the NASH-Equilibrium Problem NP-Complete? The Lemke-Howson Algorithm. The Class PPAD. Succinct Representations of Games; The Reduction, Correlated Equilibria, Bit matrix Games and Best Response Condition; Equilibria Via Labeled Polytopes, The Lemke-Howson Algorithm; Integer Pivoting and Degenerate Games;

**UNIT-III**

Extensive Games and Their Strategic Form; Sub game Perfect Equilibria, Computing Equilibria with Sequence Form. Model and Preliminaries; External Regret Minimization, Regret minimization and Game Theory, Generic Reduction from External to Swap Regret, On the Convergence of Regret- Minimizing Strategies to Nash Equilibrium in Routing Games.

**UNIT-IV**

Fisher's Linear Case and the Eisenberg –Gale Convex Program, checking if Given Prices are Equilibrium Prices, Two Crucial Ingredients of the Algorithm, The Primal-Dual Schema in the Enhanced Setting; Tight Sets and the Invariants; Balanced Flows, The Main Algorithm and Running Time, The Linear-Case of Arrow-Debreu Model.

**UNIT-V**

Algorithm for Single-Source Multiple-Sink Markets; Fisher Model with Homogeneous Consumers; Exchange Economics Satisfying WGS; Specific Utility Functions; Computing Nash Equilibria in Tree Graphical Games; Graphical Games and Correlated Equilibria; Graphical Exchange Economies.

**TEXT BOOKS:**

1. Noam Nisan, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, Algorithmic Game Theory, Cambridge University Press, 2007.
2. Ronald Cohn Jesse Russell, Algorithmic Game Theory, VSD Publishers, 2012.

**REFERENCE BOOKS:**

1. Game Theory, An Introduction, Steven Tadelis, Princeton University Press, 2016
2. Game Theory, Michael Maschler, Eilon Solan, Shmuel Zamir, Cambridge University Press, 2013



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20) SOFTWARE ENGINEERING LAB (CS505PC)

L	T	P	C
0	0	3	1.5

#### Prerequisites/ Corequisites:

1. CS203ES – Programming for Problem Solving Course
2. CS403PC – Operating Systems Course
3. CS502PC – Software Engineering 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 use case tools in design phase.
4. Use Jenkins to build project.
5. Devise a project in AWS cloud using Dockers and Kubernetes.

#### List of Experiments:

Execute the following exercises for any one project given in the list of sample 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 static pages of the project and committing using Git and GitHub
- 5) Building the project in Jenkins
- 6) Deploying the project in AWS cloud using Docker and Kubernetes
- 7) Develop test cases for unit testing and integration testing

#### 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.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year I Semester Course Syllabus (KR20)

### WEB TECHNOLOGIES LAB (CS506PC)

L	T	P	C
0	0	4	2

#### Prerequisites/ Corequisites:

1. CS401PC – Java Programming Course
2. CS402PC- Database Management Systems Course
3. CS503PC- 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 Express JS.
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 Express JS.

#### 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 Express JS.
4. Analyze and Implement routing methodologies using Express JS.
5. Design and create a NoSQL (Mongo)database and then integrate it with the front end using Express JS, this, by better understanding client-server communication.

#### 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 (using ExpressJS)

**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 Course Syllabus (KR20)**  
**COMPUTER NETWORKS LAB (CS507PC)**

L	T	P	C
0	0	3	1.5

**Prerequisite/ Corequisites:**

1. CS203ES - Programming for problem solving Course.
2. CS401PC - Java Programming 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.

**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, 4<sup>th</sup> 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 1<sup>st</sup> Edition.

**REFERENCE BOOKS:**

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



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING III Year I Semester Course Syllabus (KR20) PROFESSIONAL AND SOFT SKILLS LAB (CS512HSM)

L	T	P	C
0	0	2	1

### Prerequisite/ Corequisite:

1. EN107HS – 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.

### 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

**SUGGESTED SOFTWARE:**

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)

**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, 5<sup>th</sup> 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 - Aruna Koneru, McGraw Hill Education (India) Pvt. Ltd, 2016.
3. Technical Communication - Meenakshi Raman & Sangeeta Sharma, Oxford University Press 2009.
4. Technical Communication - Paul V. Anderson, Cengage Learning pvt. Ltd. New Delhi, 2007.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

#### CLOUD COMPUTING (CS601PC)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. CS403PC – Operating Systems Course.
2. CS402PC– 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 and use.

#### 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, Amazon Sumerian, 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, Managing the cloud application, 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 with example of Amazon RDS Amazon DynamoDB, characteristics of PaaS, Suitability of PaaS, pros and cons of PaaS, summary of PaaS Providers, software as service, characteristics of SaaS, Suitability of SaaS, pros and cons of SaaS, summary of SaaS Providers.

## **UNIT-V**

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

**Introduction to Azure and GCP services and Architecture:** Create and deploy a static web app, execute a google compute engine.

**ML Application and Integration:** Train and Deploy models using Amazon sage maker and Amazon Rekognition using server less API

## **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, AmazonAPI 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, 1<sup>st</sup> Edition, 2013.



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### COMPETITIVE PROGRAMMING (CS602PC)

L	T	P	C
3	0	0	3

#### Pre-requisites/ Corequisites:

1. CS203ES – Programming for Problem Solving Course
2. CS303PC – Data Structures 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, 3<sup>rd</sup> Edition 2020.

### **REFERENCE BOOKS/ LINKS:**

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, 2<sup>nd</sup> edition
3. <https://cp-algorithms.com>





# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### PARALLEL PROGRAMMING (CS611PE)

#### (Professional Elective – II)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. CS203ES – Programming for Problem Solving Course
2. CS302PC- Object Oriented Programming Using C++ Course
3. CS403PC – 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.

**Planning for parallelization:** Approaching a new project: The preparation, Profiling, Planning, Implementation, and Commit.

**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, Compiler flags relevant for vectorization for various compilers, OpenMP SIMD directives for better portability

**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 , Hybrid MPI plus OpenMP for extreme scalability

#### 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 type 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 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 , Kernel execution overhead comparison

#### **UNIT- V**

**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, N-body, Histogram calculation, Quicksort in CUDA using dynamic parallelism, Radix sort.

**Programming with Libraries and Other Languages:** cuBLAS, cuRAND , cuFFT, NPP, Writing Python code that works with CUDA – NVBLAS Deep Learning Acceleration with CUDA : Fully connected layer acceleration with cuBLAS , Activation layer with cuDN, Softmax and loss functions in cuDNN, Convolutional neural networks with cuDNN , Recurrent neural network optimization

#### **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 II Semester Course Syllabus (KR20)**  
**ARTIFICIAL INTELLIGENCE (CS612PE)**  
**(Professional Elective – II)**

L	T	P	C
3	0	0	3

**Prerequisites/ Corequisites:**

1. CS203ES–Programming for Problem Solving Course
2. CS303PC – Data Structures Course
3. CS501PC - Design and Analysis of Algorithms
4. MA405BS - Probability Theory and Statistics

**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**

**Problem Solving by Search-I:** Introduction to AI, Intelligent Agents

**Problem Solving by Search –II:** Problem, Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth, first search, Uniform cost search, Depth-first search, Iterative deepening Depth, first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best, first search, A\* search, Heuristic Functions, Beyond Classical Search: Hill, climbing search, Simulated annealing search, Local Search in Continuous Spaces, Searching with Non, Deterministic Actions, Searching with Partial Observations, Online Search Agents and Unknown Environment .

**UNIT - II**

**Problem Solving by Search-II and Propositional Logic Adversarial Search:** Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real, Time Decisions.**Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

**Propositional Logic:** Knowledge, Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

### **UNIT - III**

**Logic and Knowledge Representation First, Order Logic:** Representation, Syntax and Semantics of First, Order Logic, Using First, Order Logic, Knowledge Engineering in First, Order Logic.

**Inference in First, Order Logic:** Propositional vs. First, Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

**Knowledge Representation:** Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

### **UNIT - IV**

#### **Planning**

**Classical Planning:** Definition of Classical Planning, Algorithms for Planning with State, Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

**Planning and Acting in the Real World:** Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent Planning.

### **UNIT - V**

#### **Uncertain knowledge and Learning**

**Uncertainty:** Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes Rule and Its Use,

**Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First, Order Probability, Other Approaches to Uncertain Reasoning; Dempster, Shafer theory.

**Learning:** Forms of Learning, Supervised Learning, Learning Decision Trees. Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation, Based Learning, Learning Using Relevance Information, Inductive Logic Programming.

### **TEXT BOOKS:**

1. Artificial Intelligence A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education, 2015.

### **REFERENCE BOOKS:**

1. Artificial Intelligence, E. Rich and K. Knight (TMH), 3rd Edition, 2017.
2. Artificial Intelligence, Patrick Henny Winston, Pearson Education, 3rd Edition, 2002.
3. Artificial Intelligence, Shivani Goel, Pearson Education, 2013.
4. Artificial Intelligence and Expert systems, Patterson, Pearson Education, 2015.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### CYBER SECURITY (CS613PE)

### (Professional Elective – II)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. CS504PC -Computer Networks Course
2. CS601PC- Cloud Computing 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 Cybercrime:** Introduction, Cybercrime and Information Security, who are Cybercriminals, Classifications of Cybercrimes, Cybercrime: The legal Perspectives and Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes.

#### UNIT - II

**Cyber Offenses: How Criminals Plan Them:** Introduction, How Criminals plan the Attacks, Social Engineering, Cyber stalking, Cyber Cafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector, Cloud Computing.

#### 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

**Tools and Methods Used in Cybercrime:** Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spywares, Virus and Worms, Trojan Horse and Backdoors, Steganography, DoS and DDoS attacks, SQL Injection, Buffer Overflow.

**UNIT - V**

**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.

**TEXT BOOKS:**

1. Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Nina Godbole and Sunil Belapure, Wiley INDIA, 2020.

**REFERENCE BOOKS:**

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press, 2010.
2. Introduction to Cyber Security, Chwan,Hwa (john) Wu, J. David Irwin. CRC Press T&F Group, 2013



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

#### DESIGN PATTERNS (CS614PE)

#### (Professional Elective – II)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. CS502PC – Software Engineering Course
2. CS401PC – 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 behavioral 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 behavioral 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

**Behavioral 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 Course Syllabus (KR20)

### FRONT END WEB DEVELOPMENT (CS621PE)

### (Professional Elective – III)

L	T	P	C
3	0	0	3

#### Prerequisite/ Corequisite:

1. CS503PC – 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. 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 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



# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### MOBILE APPLICATION DEVELOPMENT (CS622PE)

(Professional Elective-III)

L	T	P	C
3	0	0	3

#### Prerequisite/ Corequisites:

- 1.CS403PC - Operating Systems Course.
- 2.CS302PC - Object Oriented Programming Using C++ Course.

#### Course Objectives: The course will help to

1. Understand the fundamentals of Android operating systems.
2. Develop software with reasonable complexity on mobile platform.
3. Demonstrate the ability to deploy software to mobile devices.
4. Understand persistent data management and directory Shared Preferences.
5. Explore API for SMS, Email.

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

1. Understanding android operating system and its components
2. Understanding and design of User Interface with the support of persistent data management
3. Creating and displaying notifications
4. Analyze the directory Shared Preferences
5. Develop the API for SMS, Email.

#### UNIT-I

**Introduction to Android Operating System:** Android OS design and Features – Android development framework, SDK features, Installing and running applications on Eclipse platform, Creating AVDs, Types of Android applications, best practices in Android programming, Android tools.

**Android application components:** Android Manifest file, Externalizing resources like values, Themes, Layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes.

**Android Application Lifecycle:** Activities, Activity lifecycle, Activity states, monitoring state changes.

#### UNIT-II

**Android User Interface: Measurements** - Device and pixel density independent measuring units, Layouts - Linear, Relative, Grid and Table Layouts.

**User Interface (UI) Components:** Editable and non-editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers.

**Event Handling:** Handling clicks or changes of various UI components.

**Fragments:** Creating fragments, Lifecycle of fragments, Fragment states, adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities.

### **UNIT-III**

**Intents and Broadcasts:** Intent - Using intents to launch Activities, explicitly starting new Activity, Implicit Intents, passing data to Intents, getting results from Activities, Native Actions, using Intent to dial a number or to send SMS.

**Broadcast Receivers:** Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity.

**Notifications:** Creating and Displaying notifications, Displaying Toasts

### **UNIT-IV**

**Persistent Storage:** Files - Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences - Creating shared preferences, saving and retrieving data using Shared Preference.

### **UNIT-V**

**Database:** Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update).

**Alarms** - Creating and using alarms.

**Using Internet Resources:** Connecting to internet resource, using download manager Location Based Services - Finding Current Location and showing location on the Map, updating location.

### **TEXT BOOKS:**

1. Professional Android 4 Application Development- Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers-James C Sheusi ,Cengage Learning, 2013.

### **REFERENCE BOOK:**

1. Beginning Android 4 Application Development-Wiley India (Wrox), Wei-Meng Lee, 2013ser.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### LINUX PROGRAMMING (CS623PE)

### (Professional Elective – III)

L	T	P	C
3	0	0	3

#### Pre-requisite/ Corequisite:

1. CS403PC – Operating Systems Course

#### Course Objectives: The course will help to

1. Understand and make effective use of Linux utilities and Shell scripting language (bash) to solve Problems.
2. Recognize the files and directories in Linux environment.
3. Understand the concepts process and signal management
4. Comprehend different inter process communication mechanisms.
5. Understand shared memory concept and sockets.

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

1. Outline the Linux utilities and use shell in Linux environment.
2. Illustrate different file and directory operations.
3. Implement inter process communication between processes
4. Implement message queues and semaphores
5. Implement sockets for client server model

#### UNIT- I

**Linux Utilities** - File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities. Sed-Scripts, Operation, Addresses, Commands, Applications, awk-Execution, Fields and Records, Scripts, Operation, Patterns, Actions, Associative Arrays, String and Mathematical functions, System commands in awk, Applications.

**Shell programming with Bourne again shell (bash)** - Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

#### UNIT- II

**Files and Directories** - File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, create, read, write, close, lseek, dup2, file status information-stat family, file and record locking- fcntl function, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink. **Directories** - Creating, removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir functions.

#### UNIT- III

**Process** – Process concept, Layout of a C program image in main memory, Process environment-environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system

call interface for process management-fork, vfork, exit, wait, waitpid, exec family, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

**Signals** – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise, alarm, pause, abort, sleep functions.

#### UNIT- IV

**Interprocess Communication** - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions. **Message Queues** - Kernel support for messages, APIs for message queues, client/server example. **Semaphores** - Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

#### UNIT- V

**Shared Memory** - Kernel support for shared memory, APIs for shared memory, shared memory example. **Sockets** - Introduction to Berkeley Sockets, IPC over a network, Client- Server model, Socket address structures (Unix domain and Internet domain), Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous clients, Socket options setsockopt and fcntl system calls, Comparison of IPC mechanisms.

#### TEXT BOOKS:

1. Unix System Programming using C++, T. Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
3. Unix Network Programming, W. R. Stevens, PHI.

#### REFERENCE BOOKS:

1. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
2. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
3. System Programming with C and Unix, A. Hoover, Pearson.
4. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.



**B. Tech. in COMPUTER SCIENCE AND ENGINEERING**  
**III Year II Semester Course Syllabus (KR20)**  
**CRYPTOGRAPHY AND NETWORK SECURITY (CS624PE)**  
**(Professional Elective – III)**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Prerequisites/ Corequisites**

1. CS504PC– Computer Networks Course.

**Course Objectives: The course will help to**

1. Recognize the objectives of information security.
2. Explain the importance and application of each of confidentiality, integrity, authentication and availability.
3. Understand various cryptographic algorithms.
4. Comprehend the basic categories of threats to computers and networks.
5. Understand Intrusions and intrusion detection.

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

1. Understand basic cryptographic algorithms, message and web authentication and security issues.
2. Illustrate information system requirements for both of them such as client and server.
3. Understand the current legal issues towards information security.
4. Implement fundamental ideas of cryptography.
5. Use PGP package to develop applications.

**UNIT - I**

**Security Concepts:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security

**Cryptography Concepts and Techniques:** Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

**UNIT - II**

**Symmetric key Ciphers:** Block Cipher principles, DES, AES, Blowfish, RC5, IDEA, Block cipher operation, Stream ciphers, RC4.

**Asymmetric key Ciphers:** Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm.

**UNIT - III**

**Cryptographic Hash Functions:** Message Authentication, Secure Hash Algorithm (SHA-512),

**Message authentication codes:** Authentication requirements, HMAC, CMAC, Digital signatures, Elgamal Digital Signature Scheme.

**Key Management and Distribution:** Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

#### **UNIT - IV**

**Transport-level Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH)

**Wireless Network Security:** Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security

#### **UNIT - V**

**E-Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange.

**Case Studies on Cryptography and security:** Secure Multiparty Calculation, Virtual Elections, Single sign On, Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability.

#### **TEXT BOOKS:**

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition, 2017.
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition, 2013.

#### **REFERENCE BOOKS:**

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition, 2011.
3. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition, 2015.
4. Information Security, Principles, and Practice: Mark Stamp, Wiley India, 2005.
5. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH, 2016.





**B. Tech. in COMPUTER SCIENCE AND ENGINEERING**  
**III Year II Semester Course Syllabus (KR20)**  
**INTELLIGENT DATABASE SYSTEMS (CM600OE)**  
**(OPEN ELECTIVE – I)**

L	T	P	C
3	0	0	3

**Prerequisites/ Corequisites:**

1. CS402PC – Database Management Systems Course
2. CS612PE – Artificial Intelligence Course

**Course Objectives: The course will help to**

1. Know the fundamentals of knowledge and expert systems.
2. Recognize the importance of multidimensional indexing and information retrieval.
3. Recognize the components of database data mining and knowledge discovery.
4. Understand knowledge database systems
5. Understand about the different applications of IDBS

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

1. Define the term intelligent database systems
2. Illustrate Database installation then create the database with user and apply SQL.
3. Demonstrate knowledge-based system concepts and apply them to AI.
4. Create modest knowledge base systems by designing and building them.
5. Analyze and implement for various real-time applications in Intelligent Database System

**UNIT-I**

**INTRODUCTION TO IDBS:** Informal definition of the domain, General characteristics of IDBSs, Data models and the relational data model, A taxonomy of intelligent database systems, Guidelines for using intelligent database systems.

**UNIT-II**

**SEMANTIC DATA MODELS:** Nested and semantic data models, Introduction, The nested relational model, Semantic models, Hyper, semantic data models, Object, oriented approaches to semantic data modeling – Object oriented database systems, Basic concepts of a core object, oriented data model, Comparison with other data models, Query languages and query processing, Operational aspects, Systems, The ODMG standard, The object, relational data model, Java and databases, Conclusions, Active database systems, Basic concepts, Issues, Architectures, Research relational prototypes, the Starburst Rule System, Commercial relational approaches.

**UNIT-III**

**KNOWLEDGE-BASED SYSTEMS- AI CONTEXT:** Characteristics and classification of the knowledge based systems – Introduction, The resolution principle, Inference by inheritance – Conclusion, Deductive database systems, Basic concepts, DATALOG language – Deductive database systems and logic programming systems, differences, Architectural approaches, Research prototypes, Updates in deductive databases, Integration of deductive database and object database technologies, Constraint databases, Conclusions.

#### **UNIT-IV**

**ADVANCED KNOWLEDGE-BASED SYSTEMS:** Introduction, Architectural solutions, The 'general bridge' solution, Extending a KBS with components proper to a DBMS, The 'tight coupling' approach – Conclusion, Advanced solutions: Introduction, A 'knowledge level' approach to the interaction with an IAS, TELOS, a language for implementing very large 'integral approach' systems, The CYC project, Other projects based on a 'conceptual representation' approach, Lexical approaches to the construction of large KBs.

#### **UNIT-V**

**APPLICATIONS IN IDBS:** Introduction, Temporal databases, Basic concepts, Temporal data models, Temporal query languages, Ontologies, Ontology theoretical foundations, Environments for building ontologies, Structured, semi-structured and unstructured data, Multimedia database, semi-structured data, Mediators, Motivation, Architecture, Application of mediators to heterogeneous systems – Proposals, Multi-Agents systems, Main issues in designing a multi-agent system, Open problems. Internet indexing and retrieval, Basic indexing methods, Search engines or meta-searchers, Internet spiders, Data mining, Data mining tasks, Data mining tools, Medical and legal information systems, medical information systems, Legal information systems – Conclusions.

#### **TEXT BOOK:**

1. Intelligent Database Systems Collection, Elisa Bertino, Barbara Catania, Gian Piero Zarri, ACM Press, 2000.

#### **REFERENCE BOOKS/ LINKS:**

1. Advances in Intelligent Information and Database Systems, Ngoc Thanh Nguyen, Radoslaw Katarzyniak, and Shyi-Ming Chen (Eds.), Springer, 2010.
2. <https://www.eyrolles.com/Informatique/Livre/intelligent-database-systems-9780201877366/>



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

#### ANN & DEEP LEARNING (CD600OE)

#### (OPEN ELECTIVE – I)

L	T	P	C
3	0	0	3

#### Prerequisites/ Co-requisites:

1. CS511PE – Introduction to Machine Learning Course
2. MA101BS – Mathematics - I Course
3. MA405BS – Probability Theory and Statistics Course
4. CS306PC – Python Programming Lab Course

#### Course Objectives: The course will help to

1. Understand Artificial Neural Networks and Deep Learning concepts.
2. Implement ANN and DL algorithms with Tensorflow and Keras.
3. Understand Image processing and analysis with CNN.
4. Understand Sequence learning with RNN.
5. Devise advanced concepts of computer vision.

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

1. Develop Artificial Neural Networks and Deep Learning concepts.
2. Implement ANN and DL algorithms with Tensorflow and Keras.
3. Understand Image processing and analysis with CNN.
4. Implement Sequence learning with RNN.
5. Understand advanced concepts of computer vision.

#### UNIT-I

**Introduction to Artificial Intelligence:** Artificial Intelligence in Action, The History of Neural Networks, Deep Learning Characteristics, Deep Learning Applications, Deep Learning Framework, Development Environment Installation.

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

**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.

#### UNIT-II

**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.

**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.

### **UNIT-III**

**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.

**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 VGG13, Convolutional Layer Variants, Deep Residual Network – DenseNet, Hands, On CIFAR10 and ResNet18.

### **UNIT-V**

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

**Generative Adversarial Networks:** Examples of Game Learning, GAN Principle, Hands, On DCGAN, GAN Variants, GAN Training Difficulty, WGAN Principle, Hands, On WGAN, GP

### **TEXT BOOKS:**

1. Beginning Deep Learning with TensorFlow: Work with Keras, MNISTDataSets, and Advanced Neural Networks by Liangqu Long, Xiangming Zeng, A Press, 2022
2. Deep Learning from the Basics, KokiSaitoh, Packt Publishing, 2021

### **REFERENCE BOOK:**

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 Course Syllabus (KR20)

### BASICS OF SENSOR TECHNOLOGY (EI600OE)

#### (OPEN ELECTIVE – I)

L	T	P	C
3	0	0	3

#### Prerequisites/ Corequisites:

1. AP202BS – Applied Physics Course
2. MA101BS – Mathematics - I Course

#### Course Objectives: The course will help to

1. Provide basic knowledge in transduction principles, sensors and transducer technology and measurement systems.
2. Provide better familiarity with the Theoretical and Practical concepts of Transducers.
3. Provide familiarity with different sensors and their applications in real life.
4. Provide the knowledge of various measurement methods of physical
5. Provide the knowledge of electrical parameters

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

1. Identify suitable sensors for real time applications.
2. Identify suitable transducers for real time applications
3. Translate theoretical concepts into working models.
4. Design the experimental applications to engineering modules and practices.
5. Design engineering solution to the Industry/Society needs and develop products.

#### UNIT-I

**Introduction to measurement systems:** General concepts and terminology, measurement systems, sensor classifications: Analog Input and Output, Digital Input and Output, general input, output configuration, methods of correction. Passive Sensors, Resistive Sensors: Potentiometers, Strain Gages, Resistive Temperature Detectors (RTDs), Thermistors, Light, dependent Resistors (LDRs), Resistive Hygrometers.

**Capacitive Sensors:** Variable capacitor and Differential capacitor.

**Inductive Sensors:** Reluctance variation sensors, Eddy current sensors, Linear variable differential transformers (LVDTs), Magneto elastic sensors, Electromagnetic sensors, Sensors based on Faraday's law of Electromagnetic induction, Touch Sensors: Capacitive, Resistive, Proximity Sensors.

#### UNIT-II

**Self-generating Sensors or active sensors, Thermoelectric Sensors:** Thermocouples, Thermoelectric effects, Common thermo couples, Practical thermo couple laws, Cold compensation in thermocouples circuits.

**Piezoelectric Sensors:** Piezoelectric effect, piezo electric materials, applications.

### **UNIT-III**

**Velocity And Acceleration Measurements:** Relative velocity–Translational and Rotational velocity measurements, Revolution counters and Timers, Magnetic and Photoelectric pulse counting stroboscopic methods. Accelerometers-different types, Gyroscopes, applications. Density measurements – Strain Gauge load cell method – Buoyancy method, Air pressure balance method, Gamma ray method, Vibrating probe method.

### **UNIT-IV**

**Density, Viscosity and Other Measurements:** Units of Viscosity, specific gravity scales used in Petroleum Industries, Different Methods of measuring consistency and Viscosity –Two float viscorator –Industrial consistency meter. Sound, Level Meters, Microphones, Humidity Measurement.

### **UNIT-V**

**Calibration and Interfacing:** Calibration using Master Sensors, Interfacing of Force, Pressure, Velocity, Acceleration, Flow, Density and Viscosity Sensors, Variable Frequency Drive

### **TEXTBOOKS:**

1. Measurement Systems–Applications and Design–by Doebelin E.O., 4/e, McGraw Hill International, 1990.
2. Principles of Industrial Instrumentation –Patranabis D.TMH. End edition 1997

### **REFERENCE BOOKS:**

1. Sensors and Transducers: D.Patranabis, TMH 2003, Wiley & Sons Ltd. 2006.
2. Sensor Technology HandBook– Jon Wilson, Newne 2004.
3. Instrument Transducers–An Introduction to their Performance and design–by Herman K. P. Neubrat, Oxford University Press, 2005.
4. Measurement System: Applications and Design–by E.O. Doebelin, McGraw Hill Publications, 2007.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

#### CLOUD COMPUTING LAB (CS603PC)

L	T	P	C
0	0	3	1.5

#### Prerequisites/ Corequisites:

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

#### 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 (Amazon EC2 instances) on the AWS Cloud
3. Create your First AWS S3 Bucket and Upload Content to Bucket and Manage their Access and Create Static Website using AWS S3
4. Create and configure storage services and upload files and objects using Amazon EBS, Amazon EFS and Amazon S3
5. Create, Train and Deploy a simple Machine Learning Model (using python) using AWS Amazon Sage maker (use P type instance)
6. Create visual search & image recognition capability application using AWS Amazon Recognition.
7. Implement serverless computing and Docker containers on AWS using AWS Lambda and Amazon ECS access the services with amazon API gateway

8. Create a Platform as a Service (PaaS) application on AWS Elastic Beanstalk
9. Create serverless event-driven architectures on Lambda to run backend ML application integrated with frontend React component.
10. Launch a relational database on Amazon RDS and a NoSQL database using Amazon DynamoDB
11. Create loosely coupled services with Amazon SQS and Amazon SNS to process data received from the applications.
12. Build a continuous integration and continuous delivery pipeline using DevOps tools on AWS to automate a (CI/CD) pipeline
13. 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.





# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### COMPETITIVE PROGRAMMING LAB (CS604PC)

L	T	P	C
0	0	2	1

#### Prerequisites/ Corequisites:

1. CS406PC–Java Programming Lab Course
2. CS604PC – 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.

#### 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, 3<sup>rd</sup> Edition 2020.

**REFERENCE BOOKS/ LINKS:**

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, 2<sup>nd</sup> edition
3. <https://cp-algorithms.com>



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

#### FRONT END WEB DEVELOPMENT LAB (CS631PE)

#### (Professional Elective – III)

L	T	P	C
0	0	3	1.5

#### Prerequisite/ Corequisite:

1. CS621PE – 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.

#### 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, AboutUs, 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 Course Syllabus (KR20)**  
**MOBILE APPLICATION DEVELOPMENT LAB (CS632PE)**  
**(Professional Elective-III)**

**Prerequisites:** Nil

L	T	P	C
0	0	3	1.5

**Course Objectives:** The course will help to

1. Learn how to develop Applications in android environment.
2. Understand the layout Management and Multi layout definition techniques
3. Learn how to develop user interface applications.
4. Demonstrate the push notifications for incoming messages
5. Analyse and integrate the applications to the Android marketplace

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

1. Apply OOPC to develop Mobile Applications.
2. Apply Layout Management and Multi layout definition techniques to create adaptable User Interface
3. Develop user interface for mobile Application using widgets with event handling.
4. Design push notifications for incoming messages
5. Deploy applications to the Android marketplace for distribution.

**List of Experiments:**

1. Create an Android application that shows
  - a) Hello + name of the user and run it on an emulator.
  - b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, and Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use
  - (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button Use Fragment transactions and Rotation event listener.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.
12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.
13. Create an application that shows the given URL (from a text field) in a browser.

#### **TEXT BOOKS:**

1. Professional Android 4 Application Development, . Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013.

#### **REFERENCES BOOK:**

1. Beginning Android 4 Application Development, Wiley India (Wrox), Wei-Meng Lee, 2013ser.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### LINUX PROGRAMMING LAB (CS633PE)

### (Professional Elective – III)

L	T	P	C
0	0	3	1.5

#### Prerequisites/ Corequisites:

1. CS403PC– Operating Systems Course
2. CS623PE – Linux Programming Course

#### Course Objectives: The course will help to

1. Write shell scripts to solve problems.
2. Implement some standard Linux utilities such as ls, cp etc using system calls.
3. Implement file and directory commands of linux.
4. Develop inter process communication applications.
5. Develop network-based applications using C.

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

1. Illustrate different file and directory applications with shell script.
2. Demonstrate the file management and multiple tasks using shell scripts in Linux environment.
3. Illustrate different applications of inter process communication.
4. Design different applications using message queues and signals.
5. Develop client server applications using sockets.

#### List of Experiments:

##### Note: Use Bash for Shell scripts.

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a C program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Linux commands using System calls a) cat b) mv
12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls -l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that redirects standard output to a file.Ex: ls > f1.

16. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
17. Write a C program to create a Zombie process.
18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with command pipe. Ex: - ls -l | sort
20. Write C programs that illustrate communication between two unrelated processes using named pipe (FIFO File).
21. Write a C program in which a parent writes a message to a pipe and the child reads the message.
22. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
23. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.
24. Write a C program that illustrates suspending and resuming processes using signals.
25. Write Client and Server programs in C for connection-oriented communication between Server and Client processes using Unix Domain sockets to perform the following: Client process sends a message to the Server Process. The Server receives the message, reverses it and sends it back to the Client. The Client will then display the message to the standard output device.
26. Write Client and Server programs in C for connection-oriented communication between Server and Client processes using Internet Domain sockets to perform the following: Client process sends a message to the Server Process. The Server receives the message, reverses it and sends it back to the Client. The Client will then display the message to the standard output device.
27. Write C programs to perform the following: One process creates a shared memory segment and writes a message (“Hello”) into it. Another process opens the shared memory segment and reads the message (i.e. “Hello”). It will then display the message (“Hello”) to standard output device.

#### **TEXT BOOKS:**

1. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
2. Advanced Unix Programming, N. B. Venkateswarulu, BS Publications.
3. Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education.
4. Unix Shells by Example, 4th Edition, Ellie Quigley, Pearson Education.
5. Sed and Awk, O. Dougherty & A. Robbins, 2nd edition, SPD.

#### **REFERENCE BOOKS:**

1. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
2. Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education. Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
3. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. Rago, Pearson Education.





**B. Tech. in COMPUTER SCIENCE AND ENGINEERING**  
**III Year II Semester Course Syllabus (KR20)**  
**CRYPTOGRAPHY AND NETWORK SECURITY LAB (CS634PE)**  
**(Professional Elective – III)**

L	T	P	C
0	0	3	1.5

**Prerequisites/ Corequisites:**

1. CS401PC – Java Programming Course
2. CS203ES – Programming for Problem Solving Course
3. CS624PE – Cryptography and Network Security Course

**Course Objectives: The course will help to**

1. Understand basics of Cryptography and Network Security.
2. Secure a message over insecure channel by various means.
3. Learn about how to maintain the Confidentiality, Integrity and Availability of a data.
4. Understand various protocols for network security to protect against the threats in the networks.
5. Understand various cryptographic algorithms

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

1. Implement basic string applications.
2. Implement algorithms of key ciphers.
3. Implement various cryptographic hash functions.
4. Implement advanced symmetric encryption algorithms.
5. Implement advanced asymmetric encryption algorithms.

**List of Experiments:**

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
  - a. Ceaser cipher
  - b. Substitution cipher
  - c. Hill Cipher
4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement RSA algorithm.
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.

10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

**TEXT BOOKS:**

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition, 2017
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition, 2013.

**REFERENCE BOOKS:**

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition, 2011.
2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition, 2015.
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India, 2005.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH, 2016.



## B. Tech. in COMPUTER SCIENCE AND ENGINEERING

### III Year II Semester Course Syllabus (KR20)

### ENVIRONMENTAL SCIENCE (\*MC609)

L	T	P	C
3	0	0	0

#### Prerequisite/Corequisite:

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.