RCOEM

Shri Ramdeobaba College of
Engineering and Management, Nagpur

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR – 440013

An Autonomous College affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharashtra (INDIA)

PROGRAMME SCHEME & SYLLABI
2022-23

B. TECH. (COMPUTER SCIENCE & ENGINEERING)

About the Department:

The Department of Computer Science & Engineering was established in 2002, is well-equipped with state-of-the-art infrastructure. The state of art infrastructure includes latest configuration desktops organized in four different laboratories.

The department hosts computers, laptops and lab with internet facility. The 24X7 network managed with Cyberoam UTM firewall, and CISCO router offers intranet and internet connectivity. The computer laboratories have high-end servers of IBM and WIPRO along with industry-standard software, viz., Oracle, NetSim, Wireshark, AIX, Robotics Platform, IOT Kit and MSDN. The department promotes high-end computing through Open Source technologies and hosts NVIDIA DGX DL Workstation.

The Department has a distinction of consistently achieving above 95% results in the final year. Students are encouraged to appear in GATE, CAT, GRE and other competitive examinations which have resulted in increasing number of students clearing these exams.

Students teams of CSE have emerged winners at the Grand Finale of 2018, 2019, 2020 and 2022 editions of Smart India Hackthoan and have been excelling at the world renowned prestigious International Collegiate Programming Contest, ACM ICPC Asia West Regional Contents since 2015.

Departmental Vision:

To continually improve the education environment, in order to develop graduates with strong academic and technical background needed to achieve distinction in the discipline. The excellence is expected in various domains like workforce, higher studies or lifelong learning. To strengthen links between industry through partnership and collaborative development works.

Department Mission:

To develop strong foundation of theory and practices of computer science amongst the students to enable them to develop into knowledgeable, responsible professionals, lifelong learners and implement the latest computing technologies for the betterment of the society.

Program Education Objectives:

1. To be able to comprehend, understand and analyze Computer Science Engineering problems related to real life which can be better resolved by artificial intelligence and machine learning.
2. To impart exhaustive knowledge of Computer Science Engineering, AI and Machine Learning to cater the industrial needs and excel in innovation and management fields by prediction analysis.
3. To promote collaborative learning and spirit of team work through multidisciplinary AI based projects and diverse professional ethics.
4. To inculcate a conviction to believe in self, impart professional and ethical attitude and nurture to be an effective team member, infuse leadership qualities, and build proficiency in soft skills and the abilities to relate engineering with the social, political and technical issues as per the current scenario.

Programme Outcomes (POs):

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs):

1. **Foundation of Computer System:** Ability to understand fundamental concepts of computer science & engineering, operating system, networking & data organization systems, hardware & software aspects of computing,
2. **Software development Ability:** Ability to understand the software development life cycle. Possess professional skills and knowledge of software design process. Familiarity and algorithmic competence with a broad range of programming languages and open-source platforms.
3. **Research Ability:** Ability to apply knowledge base to identify research gaps in various domains, model real world problems, solve computational tasks, to provide solution for betterment of society with innovative ideas.

Teaching Scheme for B. Tech Computer Science & Engineering

Semester - I

Sr. No.	Course Code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	CHT154	Chemistry	2	0	0	2	40	60	100	03
2.	CHP154	Chemistry Lab	0	0	2	1	25	25	50	-
3.	MAT153	Mathematics - I	3	0	0	3	40	60	100	03
4.	CST121	Digital Electronics	3	0	0	3	40	60	100	03
5.	CSP121	Digital Electronics Lab	0	0	2	1	25	25	50	
6.	CST151	Programming for Problem Solving	4	0	0	4	40	60	100	03
7.	CSP151	Programming for Problem Solving Lab	0	0	2	1	25	25	50	-
8.	CST122	Computer Workshop-1	1	0	0	1	20	30	50	1.5
9.	CSP122	Computer Workshop-1 Lab	0	0	2	1	25	25	50	-
10.	IDT151	Creativity, Innovation & Design Thinking	1	0	0	1	20	30	50	1.5
		TOTAL	**14**	**0**	**8**	**18**			**700**	

Semester - II

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	PHT154	Introduction to Quantum Computing	3	1	0	4	40	60	100	03
2.	PHP154	Introduction to Quantum Computing Lab	0	0	2	1	25	25	50	-
3.	MAT154	Mathematics - II	3	1	0	4	40	60	100	03
4.	MAP151	Computational Mathematics Lab	0	0	2	1	25	25	50	-
5.	CST123	Object Oriented Programming	4	0	0	4	40	60	100	03
6.	CSP123	Object Oriented Programming Lab	0	0	2	1	25	25	50	-
7.	CST124	Computer Workshop-2	1	0	0	1	20	30	50	1.5
8.	CSP124	Computer Workshop-2 Lab	0	0	2	1	25	25	50	-
9.	HUT151	English	2	0	0	2	40	60	100	03
10.	HUP151	English Lab	0	0	2	1	25	25	50	-
11.	HUT152	Constitution of India	2	0	0	0	-	-	-	-
12.	PEP151	Yoga/Sports	0	0	2	0	-	-	-	-
		TOTAL	**15**	**2**	**12**	**20**			**700**	

Semester - III

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	MAT252	Linear Algebra and Statistics	3	0	0	3	40	60	100	03
2.	CST221	Computer Architecture & Organization	3	0	0	3	40	60	100	03
3.	CST222	Data Structures	3	1	0	4	40	60	100	03
4.	CSP222	Data Structures Lab	0	0	2	1	25	25	50	-
5.	CST223	Theory of Computation	3	1	0	4	40	60	100	03
6.	HUT253	Business Communication	3	0	0	3	40	60	100	03
7.	CSP224	Programming Lab-I	0	0	4	2	25	25	50	-
8.	CHT252	Environmental Sciences	2	0	0	0	-	-	-	-
9.	HUT256	Indian Traditional Knowledge	2	0	0	0	-	-	-	-
		TOTAL	**19**	**2**	**6**	**20**			**600**	

Semester - IV

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	MAT281	Discrete Mathematics	3	0	0	3	40	60	100	03
2.	CST225	Software Engineering	3	0	0	3	40	60	100	03
3.	CSP225	Software Engineering Lab	0	0	2	1	25	25	50	-
4.	CST226	Operating Systems	3	1	0	4	40	60	100	03
5.	CSP226	Operating Systems Lab	0	0	2	1	25	25	50	-
6.	CST227	Design and Analysis of Algorithms	3	1	0	4	40	60	100	03
7.	CSP227	Design and Analysis of Algorithms Lab	0	0	2	1	25	25	50	-
8.	HUT257	Cyber Laws & Ethics in IT	2	0	0	2	40	60	100	03
9.	CSP228	Programming Lab-II [Cross Platform Application]	0	0	2	1	25	25	50	-
10.	CST299	Open Elective-I/MOOC	3	0	0	3	40	60	100	03
		TOTAL	**17**	**2**	**8**	**23**			**800**	

Semester - V

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	CST321	Database Management Systems	3	1	0	4	40	60	100	03
2.	CSP321	Database Management Systems Lab	0	0	2	1	25	25	50	-
3.	CST322	Artificial Intelligence	3	0	0	3	40	60	100	03
4.	CSP322	Artificial Intelligence Lab	0	0	2	1	25	25	50	-
5.	CST323	Computer Networks	3	1	0	4	40	60	100	03
6.	CSP323	Computer Networks Lab	0	0	2	1	25	25	50	-
7.	CST324	Design Pattern	3	0	0	3	40	60	100	03
8.	CST325	Program Elective-I	3	0	0	3	40	60	100	03
9.	CST398	Open Elective-II/MOOC	3	0	0	3	40	60	100	03
10.	CSP326	Mini Project	0	0	4	2	25	25	50	-
		TOTAL	18	2	10	**25**			**800**	

Semester - VI

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	CST327	Compiler Design	3	1	0	4	40	60	100	03
2.	CSP327	Compiler Design Lab	0	0	2	1	25	25	50	-
3.	CST328	Machine Learning	3	1	0	4	40	60	100	03
4.	CSP328	Machine Learning Lab	0	0	2	1	25	25	50	-
5.	CST329	Data Warehousing & Mining	3	0	0	3	40	60	100	03
6.	CSP329	Data Warehousing & Mining Lab	0	0	2	1	25	25	50	-
7.	CST330	Program Elective-II	3	0	0	3	40	60	100	03
8.	CST399	Open Elective-III/MOOC	3	0	0	3	40	60	100	03
9.	CSP331	Project-I	0	0	6	3	25	25	50	-
		TOTAL	**15**	**2**	**12**	**23**			**700**	

Semester - VII

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	CST421	Cloud Computing	3	0	0	3	40	60	100	03
2.	CSP421	Cloud Computing Lab	0	0	2	1	25	25	50	-
3.	CST422	Program Elective-III	3	0	0	3	40	60	100	03
4.	CSP422	Program Elective-III Lab	0	0	2	1	25	25	50	-
5.	CST423	Program Elective-IV	3	0	0	3	40	60	100	03
6.	CSP423	Program Elective-IV Lab	0	0	2	1	25	25	50	-
7.	CST498	Open Elective-IV/MOOC	3	0	0	3	40	60	100	03
8.	CSP424	Project-II	0	0	8	4	50	50	100	-
9.	CSP425	Industry Internship Evaluation	0	0	2	0	-	-	-	-
		TOTAL	**12**	**0**	**16**	**19**			**650**	

Semester - VIII

Sr. No.	Course code	Course Name	Hours/ week			Credits	Maximum marks			ESE Duration (Hrs)
			L	T	P		Continuous Evaluation	End Sem Exam	Total	
1.	CST426	Program Elective-V	3	0	0	3	40	60	100	03
2.	CST427	Program Elective-VI	3	0	0	3	40	60	100	03
3.	CSP428	Project-III	-	-	-	6	50	50	100	-
		TOTAL	**6**	**0**	**0**	**12**			300	
OR										
							Continuous Evaluation	**Industry Evaluation**	**Total**	
1.	CSP429	Full Semester Industry Internship	-	-	-	12	100	100	200	-
		TOTAL							**200**	

Program Electives [PE]:

Semester	Course Code	Course Name
V [PE-I]	CST325-1	Information Security and Privacy
	CST325-2	IOT Systems
	CST325-3	Computer Graphics
	CST325-4	Advanced Algorithms
VI [PE-II]	CSP330-1	Blockchain Technology
	CSP330-2	Vulnerability Assessment & Penetration Testing
	CSP330-3	System Design
	CSP330-4	Advanced OOPS
VII [PE-III]	CST422-1	Data Analytics and Visualization
	CST422-2	Customer Relationship Management
	CST422-3	Web Intelligence and Big Data
	CST422-4	Business Intelligence
VII [PE-IV]	CST423-1	Natural Language Processing
	CST423-2	Deep Learning
	CST423-3	Digital Image Processing
	CST423-4	Parallel Computing
VIII [PE-V]	CST426-1	Bioinformatics
		Quantum Algorithm
	CST426-2	Bio Inspired Intelligence Techniques
VIII [PE-VI]	CST427-1	Intrusion Detection and Prevention System
	CST427-2	Information Retrieval
	CST427-3	Augmented and Virtual Reality

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CHT154 **Course: Chemistry**

L: 2 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 2**

Course Outcomes:

After the successful completion of the course, students shall be able to

- Predict the properties and interactions of chemical substances by understanding their composition at the atomic and molecular level.
- Discuss unique properties of nano-materials to solve challenges in our life and applications in computational world.
- Discuss how spectroscopic methods are used for qualitative and quantitative analysis.
- Analyze the utilization of green computing technology for environmental issues

Syllabus:

Module 1: Atomic and Molecular Structure [6 hours]

Equations for atomic and molecular orbitals. Energy level diagrams of diatomic. HOMO, LUMO, Crystal field theory and the energy level diagrams for transition metal ions and their optical and magnetic properties. Band Theory

Module 2: Nanomaterials for Advanced Computing: [6 hours]

Basics of Nanochemistry, classification, synthesis and Industrial applications, quantum dots for quantum computers, Doping of nanocrystals, Nanowires, Nanofibers, Nanotechnological advancements for computing.

Module 3: Characterization Techniques and computational tools: [6 hrs]

Fundamentals of spectroscopy, Electronic Spectroscopy, Nuclear Magnetic Resonance Spectroscopy. Basics of Nuclear magnetic resonance quantum computer Synthesis of drugs, basic soft-wares for bio-chemical assessment of drugs.

Module 4: Green Computing and Chemistry [6 hrs]

Metal extraction from E-wastes: Constraints and opportunities, Chemical exposure (Lead, Mercury, Cadmium, Chromium etc.) and contamination, Principles of Green Chemistry and Green Computing, Role of Green Computing in Environment and Research, Green devices and Green data Server.

Suggested Text Books:

1. Shikha Agrawal , Engineering Chemistry : Fundamentals and Applications, Cambridge University Press.
2. Dr. Rajshree Khare, A Textbook of Engineering Chemistry(AICTE), S.K. Kataria & Sons.
3. S. S. Dara, A Textbook of Engineering Chemistry, S. Chand Publications.
4. A. K. Das and M. Das, An introduction to nanomaterials and nanoscience, CBS Publishers and Distributors

5. M Afshar Alam, Sapna Jain, Hena Parveen, Green Computing Approach Towards Sustainable Development, Wiley Interscience Publications

Suggested Reference Books:

1. C. J. Cramer, Essentials of Computational Chemistry: Theories and Models, Second Edition, Wiley Interscience Publications.
2. Hans-Eckhardt Schaefer, Nanoscience: The Science of the Small in Physics, Engineering, Chemistry, Biology and Medicine, Springer-Verlag Berlin Heidelberg.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CHP154 **Course: Chemistry Lab**

L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 1**

Course Outcomes:

The Chemistry laboratory course will consist of experiments illustrating the principles of chemistry relevant to the study of science and engineering.

The students will learn to:

1. Apply the fundamental principles of measurement and skills in preparation and handling of hazardous chemicals and interpret the statistical data related to measurements.
2. Estimate the rate constants of reactions and order of the reaction and/or to validate adsorption isotherms.
3. Use of various computational tools for analysis of different spectral properties and bio-activities.

List of Experiments: [Any six from the list]

1. Preparation of different Solutions: Molar solution, Normal solution and percent solution and Determination of concentration.
2. Demonstration of Handling of hazardous chemicals, MSDS (material safety data sheet), waste minimization strategies and chemical waste disposal.
3. Basic statistical analysis of results of neutralization of acid against the base and preparing acceptable graphs using software.
4. Prediction of infrared/NMR spectral and analytical data of organic molecules using Computational Software.
5. Spectroscopic/Colorimetric determine of wavelength of maximum absorption of chemical/biological compound in solution and determination of concentration using Lambert-Beer's Law.
6. To study chemical kinetics of peroxydisulphate and iodide ions reactions and to find out order of the reaction and analysis of experimental data using Computational Software.
7. Molecular docking of drugs using open computational software.
8. Determination of rate of the reaction at room temperature and analysis of experimental data using Computational Software
9. Use of open access software for the interpretation of various parameters of materials including drugs

Suggested Books/Reference Books:

1. S. S. Dara, A Textbook on Experiments and Calculations in Engineering Chemistry, S. Chand Publications.
2. J. B. Yadav, Advanced Practical Physical Chemistry, Krishna's Prakashan Media (P) Limited.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: MAT153 **Course: Mathematics-I**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objective

The objective of this course is to familiarize the prospective engineers with techniques in ordinary differential equation, statistics, probability and Matrices. It aims to equip the students to deal with advanced level of mathematics and applications that would be essential for their disciplines.

Course Outcomes

On successful completion of the course, the students will learn:

1. The effective mathematical tools for the solutions of ordinary differential equations that model physical processes.
2. The essential tool of matrices in a comprehensive manner.
3. The ideas of probability and various discrete and continuous probability distributions and the basic ideas of statistics including measures of central tendency, correlation and regression.

Syllabus

Module 1: First order ordinary differential equations (7 hours)

Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Module 2: Ordinary differential equations of higher orders (8 hours)

Second order linear differential equations with constant and variable coefficients, method of variation of parameters, Cauchy-Euler equation.

Module 3: Basic Statistics: (7 hours)

Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves, correlation and regression – Rank correlation, multiple regression and correlation.

Module 4: Basic Probability: (8 hours)

Probability spaces, conditional probability, independence; Discrete random variables, Binomial distribution, Poisson distribution, Normal distribution. Relation between binomial, Poisson and Normal distributions.

Module 5: Matrices (10 hours)

Algebra of matrices, Inverse and rank of a matrix, rank-nullity theorem; System of linear equations; Symmetric, skew-symmetric and orthogonal matrices; Eigenvalues and eigenvectors; Diagonalization of matrices; Cayley-Hamilton Theorem, Orthogonal transformation and quadratic to canonical forms.

Topics for Self Learning

Application of Differential

Equations. **Textbooks /**

References

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
3. S. L. Ross, Differential Equations, 3rd Ed., Wiley India, 1984.
4. E. A. Codington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
5. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
6. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
7. Theory and Problems of probability and statistics: 2nd ed: J. R. Spiegal, Schaum series
8. A text book of Applied Mathematics Volume I & II, by P. N. Wartikar and J. N. Wartikar, Pune Vidhyarthi Griha Prakashan, Pune-411030 (India).
9. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CST121 **Course: Digital Electronics**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Outcomes

After successful completion of this course, the student will be able to,

1. Understanding of various optimization techniques used to minimize and design digital circuits.
2. Analyze and design various combinational logic circuits.
3. Analyze and design various sequential circuits.
4. Design different microprocessor based components of computer system using combinational and sequential circuits.

Course Contents

UNIT-I

Basics of Digital Electronics

Motivation for digital systems: Logic and Boolean algebra, Number Systems. Logic Gates & Truth Tables, Demorgan's law, Minimization of combinational circuits using Karnaugh maps up to five variable. Map manipulation-essential prime implicants, non-essential prime implicants.

UNIT-II

Combinational Circuit Design

Design procedure: Multiplexers, Demultiplexer, Encoders, Decoders, Code Converters, Adders, Subtractor (Half, Full), BCD Adder/ Subtractor, ripple and carry look-ahead addition.

UNIT-III

Sequential circuit Design-I

Storage elements, Flip-flops and latches: D, T, J/K, S/R flip-flops. Master Slave Conversion of one of type of F/F to another Sequential circuit. Analysis –Input equations, state table, and analysis with J-K Flip flops. Sequential circuit Design, Design procedure, designing with D & J-K Flip flop.

UNIT-IV

Sequential circuit Design-II

Counters, asynchronous and synchronous design using state and excitation tables. Registers & Shift registers.

UNIT-V

Programmable logic Design

Memory & Programmable logic Devices: RAM, Array of RAM IC's, Read only Memory, PLA,

PAL, Flash Memories

UNIT-VI

Fundamental of Microprocessor

Introduction to µp 8085, Addressing modes, Instruction set, Programming of µp 8085.

Text Books

1. Morris Mano; Digital Logic Design; Fourth edition, McGraw Hill
2. R.P.Jain; Modern Digital Electronic; Fourth edition; Tata McGraw-Hill.
3. V.J.Vibhute; 8-Bit Microprocessor & Microcontrollers; fifth edition.

Reference books

1. A. Anand Kumar; Fundamental of Digital Electronics; Second Edition, PHI
2. A.P.Godse; Digital circuit & design; Technical Publications; 2009.
3. Ramesh Gaonkar; 8 bit Microprocessor; CBS Publishers; 2011.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CSP121 **Course: Digital Electronics Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hr Per Week **Total Credits: 1**

Course Outcome

On Successful completion of course, students will be able to:

1. Use logic gates for designing digital circuits
2. Implement combinational circuits using VHDL
3. Implement sequential circuits using VHDL
4. Apply the knowledge gained for their project work based on the hardware digital circuits

Practicals based on above theory syllabus

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CST151 **Course: Programming for Problem Solving**

L: 4 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 4**

Course Outcomes

On successful completion of course student will learn:

1. To formulate simple algorithms for arithmetic and logical problems, translate the algorithms to programs (in C language), test and execute the programs and correct syntax and logical errors.
2. To implement conditional branching, iteration and recursion, to decompose a problem into functions and synthesize a complete program using divide and conquer approach.
3. To use arrays, pointers, structures and I/O operations for the formulation of algorithms and programs.
4. To apply programming to solve matrix addition, multiplication problems and searching & sorting problems.

UNIT-I: Introduction to Programming

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.)Idea of Algorithm: Steps to solve logical and numerical problems. Representation of Algorithm: Flowchart /Pseudo code with examples. Arithmetic expressions and precedence

UNIT-II: C Programming Language

Introduction to C language: Keywords, Constant, Variable, Data types, Operators, Types of Statements,

Pre-processor Directives, Decision Control Statement-if, if-else, nested if-else statement, switch case, Loops and Writing and evaluation of conditionals and consequent branching.

UNIT-III: Arrays and Basic Algorithms

Arrays: 1-D, 2-D, Character arrays and Strings. Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

UNIT-IV: Functions and Recursion

User defined and Library Functions, Parameter passing in functions, call by value, passing arrays to functions: idea of call by reference. Recursion: As a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

UNIT-V: Pointers and Structures

Structures, Defining structures, Array of Structures, Introduction to pointers, Defining pointers, Pointer arithmetic, pointer operators, Use of Pointers in self-referential structures, notion of linked list (no implementation)

UNIT-VI: File handling

Streams in C, Types of Files, File Input/ Output Operations: Modes of file opening, Reading and writing the file, Closing the files, using fflush ().

Text Books:

1. Programming in ANSI C: E. Balguruswami McGraw Hill
2. Mastering C: K. R. Venugopal and S. R. Prasad, Tata McGraw Hill

Reference Books

1. Programming with C: Byron Gottfried, Schaums Outline Series.
2. Let Us C: Yashwant Kanetkar, BPB Publication

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CSP151 **Course: Programming for Problem Solving Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits : 1**

Course Outcomes

On successful completion of course student will be able to:

1. Understand the fundamentals of C programming and choose the loops and decision making statements to solve and execute the given problem.
2. Implement different Operations on arrays also design functions to solve the given problem using C programming.
3. Understand pointers, structures, unions and apply them to develop programs.
4. Implement file Operations in C programming for a given application.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code : CST122 **Course : Computer Workshop-1**

L: 1 Hrs, T: 0 Hr, P: 0 Hr, Per Week **Total Credits: 1**

Course Objectives

1. Understand the definition and principles of UI/UX Design in order to design with intention.
2. Achieve a deep understanding of the entire life-cycle of design—the process, purpose, and tools.
3. Learn the basics of HCI (human-computer interaction) and the psychology behind user decision-making.
4. Discover the industry-standard tools and specific project deliverables in UI/UX.
5. Explain why you made design decisions, through presentations of assignments and your personal portfolio.

Unit 1:

UI/UX Overview: Intro to UI/UX, Notion & Figma Setup, Design Thinking.
User Research: How to identify stakeholders, Figma Basics, How to identify user needs.

Unit 2:

User Journeys: Mapping the user journey, Figma Grayscales, Finding solutions & constraint cards, Grayscales & User Testing: UX Principles, Figma Prototype, Understanding user testing.

Unit 3:

UI Principles: UI Principles, Color and Font. Style Guide: Components, Responsive Design.

Course Outcomes
On successful completion of the course, students will be able to:

1. Understand basics of UI/UX
2. Find solutions and constraint cards.
3. Design responsive UI.

Text Books

1. UI/UX design for designer and developers: by Nathan Clark
2. User Story mapping software for agile age [Paid subscription on yearly basis]
3. User story mapping by Jeff Patton, O'Reilly Publication

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: CSP122 **Course : Computer Workshop-1 Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hrs, Per Week **Total Credits: 1**

Course Objectives

Throughout the course, students will be expected to learn following concept:

1. Understand UI/UX basics and its use in software industry
2. Understand basic use cases of UI/UX.
3. Develop small utilities using UI/UX tools
4. Develop and integrate UI/UX with basic programs

Syllabus

Programs based on:

1. Illustration tool box
2. Storytelling and typography tools
3. UX writing and AR/VR tools
4. Voice technology tools
5. Motion Design, Animated graphics

Course Outcomes

On successful completion of the course, students will be able to:

1. Design UI/UX use cases using Illustration tool box
2. Design and use storytelling and typography for requirement specification.
3. Use UX writing, AR and VR models to develop interfaces for use cases
4. Develop small applications using voice technology, motion design, and animation.

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: IDT151 **Course: Creativity, Innovation & Design Thinking**

L: 1 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 1**

Course Outcomes

1. Be familiar with processes and methods of creative problem solving
2. Enhance their creative and innovative thinking skills
3. Practice thinking creatively and innovative design and development

Detailed Topics

UNIT I. Introduction: Making a case for creativity, Creative thinking as a skill, Valuing diversity in thinking: Thinking preferences, Creativity styles, Creativity in problem solving

UNIT 2. Pattern Breaking: Thinking differently, Lateral thinking, Mind stimulation: games, brain-twisters and puzzles, Idea-collection processes, Brainstorming/Brain writing, The SCAMPER methods, Metaphoric thinking, Outrageous thinking, Mapping thoughts, other (new approaches)

UNIT 3. Using Math and Science, Systematic logical thinking, Using math concepts, Eight-Dimensional (8D) Approach to Ideation: Uniqueness, Dimensionality, Directionality, Consolidation, Segmentation, Modification, Similarity, Experimentation

UNIT4. Systematic Inventive Thinking: Systematic inventive thinking: The TRIZ methodology, Decision and Evaluation: Focused thinking framework, six thinking hats, Ethical considerations

UNIT 5. Design for Innovation: Introduction to design for interaction, nine lessons for innovation, difference in creativity and innovation, Building blocks for innovation

UNIT 6.Intellectual Property: Introduction to intellectual property: Patents, Copyrights©, Trademarks ®, Trade Secret, Unfair Competition.

Reference Books and Text Book:

1. Creative Problem Solving for Managers - Tony Proctor - Routledge Taylor & Francis Group
2. 101 Activities for Teaching creativity and Problem Solving - By Arthur B Vangundy - Pfeiffer
3. H. S. Fogler and S.E. LeBlanc, Strategies for Creative Problem Solving, Prentice Hall
4. E. Lumsdaine and M. Lumsdaine, Creative Problem Solving, McGraw Hill,
5. J. Goldenberg and D. Mazursky, Creativity in product innovation. Cambridge University Press, 2002.

Course Assignments for internal continuous assessment of 20 Marks (NO T1 and T2)

- Brain teasers (aka Puzzle Busters, to be solved individually)
- Cartoon captions (small teams)
- TRIZ, a systematic ideation method, reading (individual)
- Book readings and discussions (small teams)
- Small teams presentations on innovation: (1) innovative individual, (2) innovative company, (3) innovative movie / game, (4) sustainable innovation, (5) innovation in business, (6) innovation in art, (7) innovation in architecture, (8) innovative nation, (9) innovation in science, and (10) innovation in engineering.
- Large group's hands-on projects
- Eight-dimensional (8D) ideation method examples Large teams videos

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: PHT154 **Course: Introduction to Quantum Computing**

L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week **Total Credits: 4**

Course Objectives

1. To introduce the fundamentals of quantum computing to students
2. The problem solving approach using finite dimensional mathematics

Course Outcomes: After successful completion of the course, the students will learn,

1. Basics of complex vector spaces
2. Quantum mechanics as applied in Quantum computing
3. Architecture and algorithms
4. Fundamentals of Quantum computations

Module 1: Complex Vector Spaces

Algebra and Geometry of Complex numbers, Real and Complex Vector Spaces, definitions, properties, basis and dimensions, Generalization to n-dimensional space

Module 2: Linear Algebra

Inner products, Hilbert Spaces, Eigenvalues and Eigenvectors, Hermitian and Unitary Matrices, Tensor Product, Applications of linear algebra in computer graphics, Geometric transforms, positioning the virtual camera

Module 3: Basic Quantum Theory

Introduction to Quantum mechanics, Schrodinger's time dependent equation, Wave nature of Particles, expectation values, variance, standard deviation, probability density, Stationary states, Infinite square well, Uncertainty principle

Module 4: Classical and Quantum Systems

Deterministic and Probabilistic Systems, Quantum Systems, Observations, Quantum measurement principles, Stochastic matrices, Probabilistic double slit experiment with photons, Entangled states, Quantum clocks

Module 5: Architecture

Bits and Qubits, Classical Gates, Reversible Gates, Quantum Gates, Toffoli and Fradkin Gates, Bloch Sphere, Deusch Gate, No-cloning theorem, Applications in Cryptography and Quantum teleportation

Module 6: Quantum algorithms

Deutsch's algorithm, The Deutsch-Jozsa algorithm, Simon's periodicity algorithm, Grover's search algorithm, Shor's factoring algorithm, Quantum Fourier Transform

Text Book

1. Quantum computing for computer scientists, Noson S. Yanofsky, Mirco A. Mannucci, Cambridge University Press 2008

2. Introduction to Quantum Mechanics, 2nd Edition, David J. Griffiths, Prentice Hall New Jersey 1995

Reference Books

1. Quantum computing explained, David McMahon, Wiley-interscience, John Wiley & Sons, Inc. Publication 2008

2. Quantum computation and quantum information, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press 2010

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: PHP154 **Course: Introduction to Quantum Computing lab**

L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 1**

Course Outcomes

The physics laboratory will consist of experiments and programming exercises illustrating the principles of physics relevant to the study of computer science and engineering. During the training in the Physics Lab, the students will be able,

1. To develop skills for experimental verification of physics laws
2. To analyze the results using the mathematical tools
3. To learn the computational techniques
4. To write the project reports

The laboratory will consist of general physics experiments and computational physics practicals.

General Physics:

1. Measuring scales and error estimation
2. Verification of Ohm's law and linear least square fitting method
3. Verification of Newton's law of cooling
4. Simple harmonic motion
5. Magnetic flux measurement using the graphical method of integration
6. Measurement, analysis and fitting of non-linear IV characteristics of PN junction diode

Computational Physics

1. Linear least square fit method for data analysis
2. Plotting of Plank's function and verification of Stefan's law
3. Finding inverse, norm and inner products, rank of a matrix
4. Introduction to quantum computing packages (GitHub repository)
5. Implementation of Deutsch-Josza algorithm using Cirq library

Project

Project on the applications of linear algebra, quantum mechanics or quantum computing to solve science and engineering problems.

Reference Books

1. Lab manual prepared by Physics Department, RCOEM, Nagpur

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: MAT154 **Course: Mathematics-II**

L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week **Total Credits: 4**

Course Objective

The objective of this course is to familiarize the prospective engineers with techniques in Calculus and multivariate analysis. It aims to equip the students with standard concepts and tools at an intermediate to advanced level that will serve them well towards tackling more advanced level of mathematics and applications that they would find useful in their disciplines.

Course Outcomes

On successful completion of the course, the students will learn:

1. The fallouts of Mean Valve Theorems that is fundamental to application of analysis to Engineering problems, to deal with functions of several variables that are essential in most branches of engineering.

2. Basics of improper integrals, Beta and Gamma functions, Curve Tracing, tool of power series and Fourier series for learning advanced Engineering Mathematics.

3. Multivariable Integral Calculus and Vector Calculus and their applications to Engineering problems.

Syllabus

Module - I : Differential Calculus: (12hours)

Taylor's and Maclaurin's series expansions; radius of curvature (Cartesian form), evolutes and involutes, Limit and continuity of functions of several variables and their partial derivatives, Eulers Theorem, chain rule, total derivative, Jacobians, Maxima, minima and saddle points; Method of Lagrange multipliers.

Module - II : Integral Calculus: (6 hours)

Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Tracing of curves (Cartesian form)

Module - IV : Sequences and series: (7 hours)

Convergence of sequence and series, tests for convergence, power series, Fourier series: Half range sine and cosine series, Parseval's theorem.

Module - V : Multiple Integrals (10 hours)

Multiple Integration: Double and triple integrals (Cartesian and polar), change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications: area, mass and volume by double integration, Center of mass and Gravity (basic concepts).

Module - VI : Vector Calculus (10 hours)

Vector Differentiation, Directional derivatives, total derivative, Gradient, Curl and Divergence. Vector integration , Theorems of Green, Gauss and Stokes and their applications.

Topics for self learning

Rolle's theorem, Mean value theorems, Indeterminate forms , Maxima and minima for function of one variable, Geometrical interpretation of Partial Differentiation(Tangent plane and Normal line) , Applications of definite integrals to evaluate perimeter, area, surface areas and volumes of revolutions.

Textbooks/References

1.Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

2.Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.

3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
g Mathematics, Tata McGraw Hill New Delhi, 11th Reprint,2010.

5. P. N. Wartikar and J. N. Wartikar, A text book of Applied Mathematics Volume I & II, Pune Vidhyarthi Griha Prakashan, Pune-411030 (India).

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: MAP151 **Course: Computational Mathematics Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 1**

Course Outcomes

The Computational Mathematics Lab course will consist of experiments demonstrating the principles of mathematics relevant to the study of science and engineering. Students will show that they have learnt laboratory skills that will enable them to properly acquire and analyze the data in the lab and draw valid conclusions.

At the end of the Course the students will learn to:

1. Develop skills to impart practical knowledge in real time.
2. Understand principle, concept, working and application of areas in mathematics and compare the results obtained with theoretical calculations.
3. Understand basics of mathematics, and report the results obtained through proper programming.

The Lab turns will be utilized for performing the experiments based on the following list

1. Calculus
2. Ordinary Differential Equations
3. Statistics
4. Linear Algebra

Suggested References

1. Computational Mathematics Lab Manual written by the Teaching Faculty of Mathematics Department,RCOEM.
2. A minimum of 8 experiments to be performed based on the above list.

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: CST123 **Course: Object Oriented Programming**

L: 4 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 4**

Course Objectives

1. To make students understand Fundamental features of an object oriented language like Java: object classes and interfaces, exceptions and libraries of object collections
2. Introduce students with fundamental concepts like exception handling, generics, multithreading and streams.

Course Outcomes

On successful completion of the course, students will be able to demonstrate

1. Understand the principles of object-oriented programming; create classes, instantiate objects and invoke methods.
2. Understand concept of generics and implement collection classes. Use exception handling mechanism.
3. Efficiently work with streams, use multithreading for solving classic synchronization problems. Perform java database connectivity and execute basic SQL commands.
4. Understand characteristics and need of Design Pattern in Software Design Process.

SYLLABUS

UNIT I

Features of Object Oriented Programming languages, Abstraction, Encapsulation, Inheritance, polymorphism and late binding. Concept of a class, Access control of members of a class, instantiating a class, constructor and method overloading.

UNIT II

Concept of inheritance, methods of derivation, use of super keyword and final keyword in inheritance, run time polymorphism, abstract classes and methods, Interface, implementation of interface, creating packages, importing packages, static and non-static members, Lambda Expressions Introduction, Block, Passing Lambda expression as Argument.

UNIT III

Exceptions, types of exception, use of try catch block, handling multiple exceptions, using finally, throw and throws clause, user defined exceptions, Introduction to streams, byte streams, character streams, file handling in Java, Serialization.

UNIT IV

Generics, generic class with two type parameter, bounded generics. Collection classes: Arraylist, Linked List, Hashset, Treeset.

UNIT V

Multithreading: Java Thread models, creating thread using runnable interface and extending Thread, thread priorities, Thread Synchronization, InterThread communications.

UNIT VI

Introduction to Design Patterns, Need of Design Pattern, Classification of Design Patterns, and Role of Design Pattern in Software design, Creational Patterns, Structural Design Patterns and Behavioral Patterns.

Text Books

1. Herbert Schildt; JAVA, the Complete Reference; Ninth Edition, Tata McGraw- Hill Publishing Company Limited.
2. Design Patterns by Erich Gamma, Pearson Education.

Reference Books

1. Cay S. Horstmann and Gary Cornell; Core JAVA Volume-II Advanced Features; Eighth Edition; Prentice Hall, Sun Microsystems Press 2008.
2. Herbert Schildt and Dale Skrien; Java Fundamentals A Comprehensive Introduction; Tata McGraw- Hill Education Private Ltd 2013.

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code : CSP123 **Course : Object Oriented Programming Lab**

L: 0Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits : 1**

Course Objectives

1. To develop ability of students to implement basic concepts and techniques of object oriented programming paradigm like encapsulation, inheritance, polymorphism, exception handling.
2. Develop solution to problems using collection classes, generics, streams, multithreading.

Course Outcomes

On completion of the course the student will be able to

1. Design solution to problems using concepts of object oriented programming like classes, objects, inheritance with proper exception handling.
2. Use collection classes, generic classes to design programs and perform database connectivity.
3. Implement programs based on streams and multithreading.

SYLLABUS

Experiments based on above Syllabus.

Syllabus for Semester II, B. TECH Computer Science & Engineering

Course Code : CST124 **Course : Computer Workshop-2**

L: 1Hrs, T: 0 Hr, P: 0 Hr, Per Week **Total Credits: 1**

Course Objective

The objective of this course is to familiarize the students with an important web framework for developing user interfaces. It aims for developing high end web applications by the use of ReactJS features.

Course Outcomes

After successful completion of this course, the student will be able to,

1. Implement the fundamentals of React with Java Script and JSX
2. Understand Templating concept along with different types of components in ReactJS
3. Understand different state and Life Cycle Methods
4. Implement Router with react Router.

Course Contents

UNIT-I

Introduction to React

React JS Introduction, Advantages of React JS, Introduction to JSX, Difference between JS and JSX.

UNIT-II

Templating using JSX

Working with React, Create Element, Expressions, logical operators, specifying attributes, children and Fragments

UNIT-III

Basics in React

React Components overview, Types of components, Controlled, Split Up, Composable, Reusable, Component Declarations and Styling Components

UNIT-IV

Props and State

State and it significance, Read state and set state, Passing data to component using props, Validating props using prop Types, Supplying default values to props using default Props

UNIT-V

Lifecycle Methods- EVENT HANDLING

Lifecycle Methods, Fetching Data, ES6 Spread Operators, Conditional Rendering, Client- or Server-side Search, Error Handling.

UNIT-VI

Routing with react router

Introduction to React Router, Routing in single page applications, Browser Router and Hash Router components Configuring route with Route component.

Text Books

- Pure React- a step by step guide - Dave Ceddia
- Road to learn react - Robin Wieruch
- React in Action 1st Edition - Mark Tielens Thomas

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: CSP124 **Course : Computer Workshop-2 Lab**

L: 0Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 1**

Course Objective

The objective of this course is to familiarize the students with an important web framework for developing user interfaces. It aims for developing high end web applications by the use of ReactJS features.

- Practical based on CST124

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: HUT151 **Course: English**

L: 2 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 2**

Course Objectives

The main objective of the subject is to enhance the employability skills of engineering students as well as communication skills at work place. The sub-objectives are:

1. To develop vocabulary of students.
2. To orient students in basic writing skills.
3. To orient students in functional grammar.
4. To orient students in the process of effective writing.
5. To provide practice and improve students' oral communication skills.

Course Outcomes

1. Students will have good word power.
2. Students will acquire basic writing skills.
3. Students will understand functional grammar and its usage.
4. Students will organize and express their thoughts effectively through written communication.
5. Students will learn oral communication skills in order to handle themselves effectively in an interview and group discussion

SYLLABUS

1. Vocabulary Building

1.1. The concept of Word Formation

1.2. Root words from foreign languages and their use in English

1.3. Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives

1.4. Synonyms, Antonyms and standard abbreviations

2. Basic Writing Skills

2.1 Sentence Structures

2.2 Use of phrases and clauses in sentences

2.3 Importance of proper punctuation

2.4 Creating coherence

2.5 Organizing principles of paragraphs in documents

2.6 Techniques for writing precisely

3. Identifying Common Errors in Writing

3.1 Subject-verb agreement

3.2 Noun-pronoun agreement

3.3 Misplaced modifiers

3.4 Articles

3.5 Redundancies

3.6 Cliches

4. Nature and Style of sensible Writing

4.1 Describing

4.2 Defining

4.3 Classifying

4.4 Providing examples or evidence

5. Writing Practices

5.1 Comprehension

5.2 Precise Writing

5.3 Essay Writing

5.4 Letter Writing

5.5 Email Writing

6. Oral Communication

(This unit involves interactive practice sessions in Language Lab)

- Listening Comprehension
- Pronunciation, Intonation, Stress and Rhythm
- Common Everyday Situations: Conversations and Dialogues
- Communication at Workplace
- Interviews
- Formal Presentations

Books

1. Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011.
2. Practical English Usage. Michael Swan. OUP. 1995.
3. Remedial English Grammar. F.T. Wood. Macmillan.2007
4. On Writing Well. William Zinsser. Harper Resource Book. 2001
5. Study Writing. Liz Hamp-Lyons and Ben Heasly. Cambridge University Press. 2006.
6. Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: HUP151 **Course : English Lab**

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 1**

Course objective

1. To enhance competency of communication in English among learners.

Course outcomes

1. Students learn presentation and public speaking skills
2. Students learn to practice effective strategies for Personal Interview and Group Discussions
3. Students learn and effectively apply language skills – listening, speaking, reading and writing

List of Practical (2 hours each for each batch) based on unit 6 (oral communication).

1. Common Everyday Situations: Conversations and Dialogues
2. Pronunciation, Intonation, Stress, and Rhythm
3. Formal Presentations: Orientation
4. Formal Presentations: Practice Session
5. Interviews: Orientation
6. Interviews: Practice Session
7. Communication at Workplace: Group Discussion- Orientation
8. Communication at Workplace: Practice Session

Syllabus for Semester I, B. Tech. Computer Science & Engineering

Course Code: HUT152 **Course : Constitution of India**

L: 2 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 0**

Course outcome

1. Students will understand the role of constitution in democratic India
2. Students will be responsible students by knowing their fundamental rights and duties
3. Students will develop better understanding of democratic functions of the government of India
4. Students will form better understanding of system of governance for effective participation

Course content

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India
4. Scheme of the Fundamental Rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India
9. Union Executive: structure, functions
10. Judiciary: Structure, role with special reference to PIL, writ petitions, strengthening of democracy & social Justice
11. Amendment of the Constitutional Powers and Procedure
12. Emergency Provisions: National Emergency, President Rule, Financial Emergency
13. Local Self Government – Constitutional Scheme in India
14. Provisions of civil services: Characteristics, functions, merits and demerits
15. Democratic principles in industry

Book

1. Durga Das Basu “An Introduction to Constitution of India” 22nd Edition, LexisNexis

Syllabus for Semester II, B. Tech. Computer Science & Engineering

Course Code: PEP151 **Course : Yoga/Sports**

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 0**

Course outcome

On successful completion of the course, students will be able to:_

1. Understand fundamental skills and basic rules of games offered by the Physical Education Department of RCOEM.
2. Obtained health related physical fitness.
3. Develop body-mind co-ordination through games and yoga.
4. Changed sedentary life styles towards active living.

Brief Objectives of Sports/Yoga Practical Classes

It has long been proven that a healthy body leads to a healthy mind. With a strong belief in this, Physical Education Department at RCOEM will conduct Sports/Yoga Classes with the objective of maintaining health, fitness and wellness of students as well as create awareness about need for good health and physical fitness. The objective would also be to make the all-round development with team spirit, social values as well as to identify and develop leadership qualities in students through various sports activities. Sports activities would also be conducted with the objective to provide better interaction and recreation to the students which is an important neutralizer for stress. Additionally, the objective would be to evaluate the health related fitness of students so as to recommend and conduct specific Yoga and Sports activities. The emphasis is on participation, with healthy competition.

Programme Outline

Sports

1. Introduction to sports, offered by the department.
2. Health and safety issues related to sports; knowledge, recognition and ability to deal with injuries and illness associated with sports.
3. Practicing the fundamental skills and bringing awareness of basic rules and regulations.
4. Conduction of small recreational games and activities.
   Yoga : Includes various sitting, standing and lying Asanas, Suryanamaskars and Pranayamas.

   Physical Efficiency Tests : This includes 6 health related physical fitness tests.

Syllabus for Semester III, B. TECH (Computer Science & Engineering)

Course Code:	**CST221**	**Course:**	**Computer Architecture and Organization**
L: 3 Hrs, T: 0 Hr, P: 0 Hr, Per Week		**Total Credits:**	**03**

Course Objectives

The objective of this course is to familiarize the prospective engineers with:

1. Concepts of computer architecture and its organization by developing understanding of various functional units, components of computers and working of all the modules.
2. Design principles of modern computers including memory, bus system, input/output operation, interrupt handling mechanism and parallelization.

Syllabus

Unit I **Basic Structure of Computers:** Functional units of computer, basic operational concepts- Instruction, processor and memory, operating steps, address, Big- and Little-endian assignments, Instructions set architecture of a CPU- Instruction Formats, Instruction sequencing, addressing modes, and instruction set classification, subroutine & parameter passing, expanding opcode, RISC and CISC.

Unit II **Basic Processing Unit:** Basic Concepts- Instruction execution, Bus architecture- One bus and Multi-bus, Execution of a Complete Instruction, sequencing of control signals, Hardwired control, Micro-programmed Control.

Unit III **Computer Arithmetic:** Signed number operations, Multiplication- shift and add, booth's Algorithm, bit-pair recoding, Integer Division- restoring and non-restoring division. Floating point numbers representation, arithmetic, guard bits and rounding.

Unit IV Memory and its Design: Concept of hierarchical memory**,** Memory System-Memory organization, RAM memories- Static and Dynamic Memories, ROM, higher order memory design, multi-module memories, Memory interleaving, Cache memory, Cache size vs. block size, mapping functions, replacement algorithms, Cache read/write policy, Virtual Memory.

Unit V Input/output Organization: Accessing I/O devices, I/O mapped I/O and memories mapped I/O, interrupt and interrupt handling mechanisms, vectored interrupts, synchronous vs. asynchronous data transfer, Bus Arbitration, Direct Memory Access.

Unit VI Pipelining: Basic concepts of pipelining, hazards, throughput and speedup, Introduction of Parallel Computing: SISD, MISD, SIMD, MIMD.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Demonstrate the understanding about the functional units of a digital computer system and their organization.
2. Execute complete instruction on different types of bus architectures with control signal generation.
3. Perform mathematical operations on arithmetic and floating-point numbers.
4. Design efficient memories and pipelined system.
5. Analyze cost performance and design trade-offs in designing and constructing a computer processor, including instruction set and memory.

Text Books

1. V. C. Hamacher, Z. G. Vranesic and S. G. Zaky; “Computer Organisation”; 5th edition; Tata McGrawHill, 2002.
2. W. Stallings; “Computer Organization & Architecture”; PHI publication; 2001.
3. J. P. Hayes; “Computer Architecture & Organization”; 3rd edition; McGraw-Hill;1998.

Reference Books

1. M. Mano; “Computer System and Architecture”; PHI publication; 1993.
2. A. S. Tanenbaum; “Structured Computer Organization”; Prentice Hall of India Ltd.
3. Patterson & Hennessy, “Computer Organization and Design”, Morgan Kaufmann, 2007

Syllabus for Semester III, B. TECH. (Computer Science and Engineering)

Course Code: CST222 **Course: Data Structures**

L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week **Total Credits: 04**

Course Objectives

1. To impart to students the basic concepts of data structures and algorithms.
2. To prepare students to use linear (stacks, queues, linked lists) and non-linear (trees, graphs) data structures.
3. To familiarize students on different searching and sorting techniques.
4. To enable students to devise algorithms for solving real-world problems.

SYLLABUS

UNIT – I: Data Structures and Algorithms Basics

Introduction: basic terminologies, elementary data organizations, data structure operations; abstract data types (ADT) and their characteristics.

Algorithms: definition, characteristics, analysis of an algorithm, asymptotic notations, time and space tradeoffs.

Array ADT: definition, operations and representations – row-major and column-major.

UNIT – II: Stacks and Queues

Stack ADT: allowable operations, algorithms and their complexity analysis, applications of stacks – expression conversion and evaluation (algorithmic analysis), multiple stacks.

Queue ADT: allowable operations, algorithms and their complexity analysis for simple queue and circular queue, introduction to double-ended queues and priority queues.

UNIT – III: Linked Lists

Singly Linked Lists: representation in memory, algorithms of several operations: traversing, searching, insertion, deletion, reversal, ordering, etc.

Doubly and Circular Linked Lists: operations and algorithmic analysis. Linked

representation of stacks and queues, header node linked lists.

UNIT – IV: Trees

Trees: basic tree terminologies, binary tree and operations, binary search tree [BST] and operations with time analysis of algorithms, threaded binary trees.

Self-balancing Search Trees: tree rotations, AVL tree and operations.

UNIT – V: Graphs and Hashing

Graphs: basic terminologies, representation of graphs, traversals (DFS, BFS) with complexity analysis, path finding (Dijkstra's SSSP, Floyd's APSP), and spanning tree (Prim's method, Kruskal's method) algorithms.

Hashing: hash functions and hash tables, closed and open hashing, randomization methods (division method, mid-square method, folding), collision resolution techniques.

UNIT – VI: Sorting and Searching

Sorting: different approaches to sorting, properties of different sorting algorithms (insertion, Shell, quick, merge, heap, counting), performance analysis and comparison.

Searching: necessity of a robust search mechanism, searching linear lists (linear search, binary search) and complexity analysis of search methods.

Course Outcomes:

On completion of the course the student will be able to

1. Analyze the efficiency of algorithms through time and space complexities.
2. Demonstrate different linear data structures (viz., stack, queue, linked list).
3. Realize different non-linear data structures (trees and graphs).
4. Use efficient algorithms for searching and sorting.

Text Books:

1. Ellis Horowitz, Sartaj Sahni & Susan Anderson-Freed, Fundamentals of Data Structures in C, Second Edition, Universities Press, 2008.
2. Mark Allen Weiss; Data Structures and Algorithm Analysis in C; Second Edition; Pearson Education; 2002.
3. G.A.V. Pai; Data Structures and Algorithms: Concepts, Techniques and Application; First Edition; McGraw Hill; 2008.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein; Introduction to Algorithms; Third Edition; PHI Learning; 2009.
2. A. K. Sharma; Data Structures using C, Second Edition, Pearson Education, 2013.

Syllabus for Semester III, B. TECH. (Computer Science and Engineering)

Course Code: CSP222 **Course: Data Structures Lab**

L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 01**

Course Objectives

1. To prepare students to identify and apply linear (stacks, queues, linked lists) and non-linear (trees, graphs) data structures in solving problems.
2. To encourage students to design and execute tree-based algorithms for solving real-world problems.
3. To enable students to employ different searching and sorting methods.

Experiments based on CST222 Syllabus in C | C++.

Course Outcomes:

On completion of the course the student will be able to

1. Realize different linear data structures.
2. Demonstrate various operations on binary search trees and AVL trees.
3. Implement algorithms for graph traversal, shortest paths and spanning trees.
4. Apply specific methods of searching and sorting to solve a problem.

Reference Books:

1. K R. Venugopal and Sudeep. R Prasad; Mastering C; Second Edition; McGraw Hill; 2015.
2. Ellis Horowitz, Sartaj Sahni & Susan Anderson-Freed, Fundamentals of Data Structures in C, Second Edition, Universities Press, 2008.
3. Mark Allen Weiss; Data Structures and Algorithm Analysis in C; Second Edition; Pearson Education; 2002.

Syllabus for Semester III, B. TECH (Computer Science & Engineering)

Course Code:	**CST223**	**Course:**	**Theory of Computation**
L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week		**Total Credits:**	**04**

Course Objectives

1. To provide students an understanding of basic concepts in the theory of computation
2. To teach formal languages and various models of computation.
3. To exhibit fundamental concepts related with computability theory.

Syllabus

Unit I

Basics of Sets and Relation, Fundamentals of formal languages and grammars, Chomsky hierarchy of languages, Countability and Diagonalization, Pigeon- hole principle.

Unit II

Finite automata: Deterministic finite automata (DFA), Nondeterministic finite automata (NFA) and equivalence with DFA, Optimization of finite automata, NFA with Epsilon Transitions, Finite Automata with output.

Unit III

Regular expressions and Regular languages, Regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, Context-free grammars (CFG), Context-free language (CFL), parse trees, ambiguity in CFG, Reduction of CFGs, Chomsky and Greibach normal forms.

Unit IV

Push Down Automata: Deterministic and Non-Deterministic pushdown automata, Designing of Push down automata for language, Language acceptance by two methods: empty stack and final state, Equivalence of PDA with CFG

Unit V

Turing machines: Turing machines model (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages, Designing of Turing machine for language, unrestricted grammars and equivalence with Turing machines.

Unit VI

Decidability and Undecidability, Church-Turing thesis, variants of Turing Machines, Universal Turing machine, Undecidable problems about languages, Recursive Function Theory.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Analyze the formal relationships among machines, languages and grammars.
2. Design an optimized finite automata for given regular language.
3. Design Push Down Automata, Turing Machine for given languages.
4. Apply computability, decidability, recursive function theory for problem solving.

Text Books:

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Reference Books

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory Computation, Pearson Education Asia.
2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill

Syllabus for Semester III, B. TECH (Computer Science & Engineering)

Course Code:	**CSP224**	**Course:**	**Programming Lab-I**
L: 0 Hrs, T: 0 Hr, P: 4 Hr, Per Week		**Total Credits:**	**02**

Course Objectives:

1. The course focuses on developing the python programming skills to do a variety of programming tasks where the students are encouraged to develop application using python.

2. To cover the basic constructs of python programming, data structures and object- oriented concepts.

3. The course also targets the coverage of important modules and libraries available in python.

Syllabus

Introduction to Python: Basic Building Blocks of Python, Installation and Working with Python, Keywords, Variables and Operators.

Data Types: int , float, complex, User Input, Arithmetic Expressions ,Using Strings and Operations on Strings, List, Slicing List and Strings, Split, Tuples, Dictionary, Sets.

Flow Control: Conditional blocks: if, else, elseif, for Loops in Python: Loops with range, Strings, List and Dictionaries, While Loop, break, continue and pass statements.

Python Functions: Library Functions, User Defined Functions, Function Argument Types, Recursion, Returning Multiple Values, Lambda, Map, Filter, Reduce.

Python OOPS Basics: Classes, Object, Class Variable and Instance Variable. Files: Reading and Writing Files in Python, File Operations and Modes.

Modules and Packages: To Create and Import Module. Open-Source Python library- Pandas, Sci Py, NumPy, Matplotlib and Seaborn.

Web scrapping: Web scrapping with the help of standard libraries like Requests and Beautiful Soup.

Python Pygame (Game Development Library)

Course Outcomes:

On successful completion of the course, students will be able to:

1. Design Python Programs using Different Data and Control Structures.

2. Use Functions, Python Files, Modules and Packages to handle complex python programs.

3. Develop Mathematical Models and Scientific Applications in Python using various Libraries.

4. Write Small Python Applications using Web Scrapping and PyGame Modules.

Text Books

1. Python Programming Using Problem Solving Approach: Reema Thareja, Oxford University, Press; First edition.
2. Learning Python: Powerful object-oriented programming, Mark Lutz, O'REILLY publications 5th addition.
3. Introduction to Computing & Problem Solving with Python Jeeva Jose and P Sojan Lal Ascher.
4. Problem Solving with Algorithms and Data Structures using Python by Brad Miller and David Ranum, 2nd addition.

Reference Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers, Learning with Python, Dreamtech Press
2. David M. Baezly “Python Cookbook” O'Reilly Media; Third edition, 2013.

Syllabus for Semester IV, B. Tech (Computer Science & Engineering)

Course Code:	**CST255**	**Course:**	**Software Engineering**
L: 3 Hrs, T: 0 Hr, P: 0 Hr, Per Week		**Total Credits:**	**03**

Course Objectives

1. To make students a successful professionals in the field with solid fundamental knowledge of software engineering.
2. To prepare students with strong communication and interpersonal skills, as well as professional and ethical principles when functioning as members and leaders of multi-disciplinary teams.
3. To enable students apply their foundations in software engineering to adapt to readily changing environments using appropriate theory, principles and processes.

Syllabus

Unit I

Introduction to Software Engineering, Software engineering principles, Software Myths, Software Engineering- A Layered Technology, Software Process Framework, Requirements Engineering Tasks, Requirement Engineering Process, Eliciting Requirement.

Unit II

Software Process Models, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, The Unified Process Model, Agile Process Models: Extreme Programming (XP), Scrum.

Unit III

An overview, Requirements Analysis, Analysis Modeling Approaches, Data Modeling, Class-based Modeling, Scenario-Based Modeling, Flow-Oriented Modeling, Behavioral Model. Design Engineering Concepts, Design Model. Software reengineering, Reverse engineering.

Unit IV

Software Testing fundamentals, Testing Life Cycle, Structural Testing, Functional Technique, Static testing, Dynamic testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Debugging. Black Box Testing, White Box Testing, Web Testing, Automated Testing.

Unit V

Software Project management- Plans and Methods. Project management concept, Project Evaluation, Cost-benefit evaluation technique, Software Effort Estimation- Albrecht Function Point Analysis, COSMIC Function Point, Cost Estimation, COCOMO Model.

Unit VI

Software Quality, Achieving Software Quality, Software Measurement, Metrics for Software Quality; Quality Management - Concepts, Software Quality Assurance.

Risk management - Risk strategies, Software risks, Risk identification, Risk refinement, RMMM, Software Reviews, Formal Technical Review; Software Configuration Management, SCM Repository, SCM Process, Software reengineering, Reverse engineering.

Course Outcomes:
On successful completion of the course, students will be able to:

1. Implement software engineering practices and various models.
2. Use software engineering process models in solving real-world problems.
3. Assess impact of software testing and project management approaches.
4. Apply approaches to assessment of software quality, risk and change.

Text books

1. Roger Pressman; Software Engineering-A Practitioner's Approach; Seventh Edition, MaGraw Hill, 2010.
2. Ian Somerville; Software Engineering; Seventh Edition; Pearson Education. 2008.
3. Rajib Mall, Software Project Management, 5th Edition, McGrawHill.

Reference Books

1. David Gustafsan, Software Engineering; Schaum's Series, Tata McGraw Hill,2002.
2. Software engineering: a precise approach, Pankaj Jalote, 3rd edition, Wiley India, 2010.
3. Object-Oriented Software Engineering: Practical Software Development Using UML and Java by Timothy Lethbridge, Robert Laganiere, McGraw-Hill, 2002.
4. Ethics in Information Technology, George W. Reynalds, 4th Edition, Cengage Leraning Publication.

Syllabus for Semester IV, B. Tech (Computer Science & Engineering)

Course Code:	**CSP255**	**Course:**	**Software Engineering Lab**
L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week		**Total Credits:**	**01**

Course Objectives

1. To teach students UML modeling tool employed in the software development life cycle.
2. To make students familiar with the hundreds of hierarchical and interrelated engineering requirements necessary for large and/or complex systems.
3. To teach students software testing tools employed in the software testing.
4. To teach students prototyping tool employed in the software industry to develop software prototype.

Syllabus

Practical based on CST357 syllabus.

Course Outcomes

After successful completion of this course, the student should able to:

1. Analyze the software engineering problem(s) to provide intended solution (SRS).
2. Design different structural models for the underlying problem.
3. Design different behavioral models for the underlying problem.
4. Implement the constructed model using white box testing and black box testing.

Text books and Reference Books

1. Rajib Mall, Software Project Management, 5th Edition, McGrawHill.
2. Object-Oriented Software Engineering: Practical Software Development using UML and Java by Timothy Lethbridge, Robert Laganiere, McGraw-Hill, 2002.

Syllabus for Semester IV, B. TECH (Computer Science & Engineering)

Course Code:	**CST226**	**Course:**	**Operating Systems**
L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week		**Total Credits:**	**04**

Course Objectives

1. To learn the mechanisms of OS to handle processes and threads and their communication
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that include architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
4. To know the components and management aspects of concurrency management.

Syllabus:

Unit I:

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine, Case study on LINUX and Windows Operating System.

Unit II:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.

Threads: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SRTF, Priority, RR, Case study on Process Management in LINUX Operating System.

Unit III:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Peterson's solution, Hardware Solution, Semaphores, Monitors, Message Passing, Classical IPC Problems: Producer-Consumer Problem, Reader-Writer Problem, Dinning Philosopher Problem etc.

Unit IV:

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Unit V:

Memory Management: Basic concept, Logical and Physical address mapping, Memory allocation: Contiguous Memory allocation – Fixed and variable partition, Internal and External fragmentation and Compaction, Paging: Principle of operation
– Page allocation, Hardware support for paging, Protection and sharing, Advantages
& Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory, Hardware and control structures, Locality of reference, Page fault, Working Set, Dirty page/ Dirty bit, Demand paging; Page Replacement algorithms: First in First Out (FIFO), Least Recently used (LRU), and Optimal.

Unit VI:

File Management: Concept of File, Access methods, File types, File operations, Directory structure, File System structure, Allocation methods, Free-space management.
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, Disk reliability, Disk formatting, Boot block, Bad blocks, case study on File Systems in LINUX operating System.

Course Outcomes

1. Contrast differing structures for operating systems.
2. Analyze the role of various components (process, page, file systems etc) of operating system.
3. Design solutions for challenges in inter-process communication.
4. Implement resource (CPU, Memory, Disk) management policies.

Text Books

1. Operating System Concepts, 8th Edition by A. Silberschatz, P.Galvin, G. Gagne, Wiley India.

2. Modern Operating Systems, 2nd Edition by Andrew Tanenbaum, PHI.

Reference Books:

1. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

2. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly

Syllabus for Semester IV, B. TECH (Computer Science & Engineering)

Course Code: CSP226	**Course: Operating System Lab**
L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week	**Total Credits: 1**

Course Objectives

Using C language in Linux environment

1. To develop ability of students to design and implement concepts of operating systems such as system calls, CPU scheduling, process/thread management.
2. To develop the components and management aspects of concurrency management, memory management, and File management.

SYLLABUS

Experiments based on CST226 Syllabus.

Course Outcomes:

On completion of the course the student will be able to demonstrate

1. Implement system commands by making use of LINUX system calls.
2. Implement processes and process schedulers.
3. Design solutions to process synchronization and deadlock handling.
4. Implement Memory management and File management solutions.

Syllabus for Semester IV, B. TECH (Computer Science and Engineering)

Course Code: CST227	**Course: Design and Analysis of Algorithms**
L: 3 Hrs, T: 1 Hr, P: 0 Hr, Per Week	**Total Credits: 04**

Course Objectives

1. Students should learn techniques for effective problem solving in computing.
2. Students should analyze different paradigms of problem solving to solve a given problem in efficient way.
3. Students should analyze the complexity of the problem for efficient problem solving.

Syllabus

UNIT I

Mathematical foundations for arithmetic and geometric series, Recurrence relations and their solutions, Principles of designing algorithms and complexity calculation, Asymptotic notations for analysis of algorithms, worst case and average case analysis, amortized analysis and its applications.

UNIT II

Divide and Conquer- basic strategy, Binary Search, Quick sort, Maximum sub-array problem, Closest pair of points problem, Convex Hull: Quick Hull, Graham's scan.

UNIT III

Greedy method – basic strategy, fractional knapsack problem, Minimum cost spanning trees, Huffman Coding, activity selection problem, find maximum sum possible equal to sum of three stacks.

UNIT IV

Dynamic Programming -basic strategy, Bellmen ford algorithm, all pairs shortest path, multistage graphs, optimal binary search trees, traveling salesman problem, String Editing, Longest Common Subsequence problem and its variations.

UNIT V

Basic Traversal and Search Techniques, breadth first search and depth first search, connected components. Backtracking basic strategy, 8-Queen's problem, graph colouring, Hamiltonian cycles, sum of subset problem, Introduction to Approximation algorithm.

UNIT VI

NP-hard and NP-complete problems, basic concepts, non-deterministic algorithms, NP-hard and NP complete, decision and optimization problems, polynomial reduction, graph based problems on NP Principle , vertex cover problem, clique cover problem

Course Outcomes

On successful completion of the course, students will be able to:

1. Analyze efficiency of algorithms using mathematical formulation, and recurrence relations methodologies.
2. Design Greedy and Divide andConquer algorithms and their usage
3. Design Dynamic programming and Backtracking Paradigms to solve the real-life problems.
4. Solve NP class problems using standard approaches.

Text Books

1. Thomas H. Cormen et.al; “Introduction to Algorithms”; 3 Edition; Prentice Hall, 2009.
2. Horowitz, Sahani and Rajasekaram; “Computer Algorithms”, Silicon Press, 2008.
3. Sridhar S.; “” Design and Analysis of Algorithms, Oxford University Press.
4. Brassard and Bratley; “Fundamentals of Algorithms”, 1 Edition; Prentice Hall, 1995. 4. Richard Johnsonbaugh, “Algorithms”, Pearson Publication, 2003.

Reference Books

1. Parag Himanshu Dave, Balchandra Dave, “Design and Analysis of Algorithms” Pearson Education, O’relly publication
2. Richard Johnsonbaugh, “Algorithms”, Pearson Publication, 2003.

Syllabus for Semester IV, B. TECH (Computer Science and Engineering)

Course Code:	**CSP227**	**Course:**	**Design and Analysis of Algorithm Lab**
L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week		**Total Credits:**	**01**

Course Objectives

1. Analyze the performance of algorithms.
2. Demonstrate a familiarity with major algorithms and data structures.
3. Apply important algorithmic design paradigms and methods of analysis.

SYLLABUS

Experiment based on syllabus of Design and Analysis Algorithms (CST352).

Course Outcomes:

After successful completion of this course, the student should able to:

1. Implement greedy algorithms to solve real world problems.
2. Implement divide-and-conquer algorithms to solve real world problems.
3. Implement algorithms using Dynamic Programming Approach.
4. Apply backtracking paradigm to realize real world problems.

Text Books

1. Thomas H. Cormen et.al. Introduction to Algorithms, Prentice Hall of India.
2. Horowitz, Sahani, Rajsekharam, Computer Algorithms, Galgotia Publications Pvt. Ltd.

Reference Books

1. Brassard, Bratley, Fundamentals of Algorithms, Prentice Hall
2. Algorithms - A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Syllabus for Semester IV, B. TECH (Computer Science and Engineering)

Course Code:	**HUT257**	**Course:**	**Cyber Law and Ethics in IT**
L: 2 Hrs, T: 0 Hr, P: 0 Hr, Per Week		**Total Credits:**	**02**

Course Objectives

1. Describe laws governing cyberspace and analyze the role of Internet Governance in framing policies for Internet security
2. Identify intellectual property right issues in the cyberspace and design strategies to protect your intellectual property
3. Understand the importance of freedom of expression, defamation and hate speech in cyber world.
4. Recognize the importance of digital divide, contingent workers and whistle blowing situations.

SYLLABUS

Unit-I

Cyber laws and rights in today's digital age; IT Act, Intellectual Property Issues connected with use and management of Digital Data, Emergence of Cyberspace, Cyber Jurisprudence.

Unit-II

Cyber Crimes against Individuals, Institution and State, Hacking, Digital Forgery, Cyber Stalking/Harassment, Cyber terrorism, Cyber Defamation, Different offences under IT Act, 2000, Cyber Torts**.**

Unit-III

Ethics in business world, Ethics in IT, Ethics for IT professionals and IT users, IT professional malpractices, communications eavesdropping, computer break-ins, denial-of-service, Cyber extortion. Types of Exploits and Perpetrators.

Unit-IV

Intellectual Property: Copy rights, Patents, Trade Secret Laws, Key Intellectual property issues, Plagiarism, Competitive Intelligence, Cybersquatting.

Unit-V

Privacy: The right of Privacy, Protection, Key Privacy and Anonymity issues, Identity Theft, Consumer Profiling, Freedom of Expression, Defamation and Hate Speech.

Unit-VI

Ethics of IT Organization: Contingent Workers H- IB Workers, Whistle- blowing, Protection for Whistle- Blowers, Handling Whistle- blowing situation, Digital divide. **Course Outcome:**

1. Analyze statutory, regulatory, constitutional, and organizational laws that affect the software professional.
2. Evaluate relationship between ethics and cyber laws with respect to legal dilemmas in the Information Technology field.
3. Demonstrate Privacy and Intellectual property rights related ethics issues that are in practices.
4. Distinguish between Business ethics roles applicable to IT users, IT professional Malpractice, IT organization workers

Text Books:

1. George Reynolds, "Ethic s in information Technology",5th edition Cengage Learning
2. Hon C Graff, Cryptography and E-Commerce - A Wiley Tech Brief, Wiley Computer Publisher, 2001
3. Introduction to Open-source Intelligence techniques.by Michael bazzel 6th Edition.

Reference Books:

1. Michael Cross, Norris L Johnson, Tony Piltzecker, Security, Shroff Publishers and Distributors Ltd.
2. Debora Johnson," Computer Ethic s",3/e Pearson Education.
3. Sara Baase, "A Gift of Fire: Social, Legal and Ethical Issues, for Computing and the Internet ," PHI Publications.
4. Chris Reed & John Angel, Computer Law, OUP, New York, (2007)
5. Cyber Crime Law and Practice by CS Mamta Binani, THE INSTITUTE OF company secretaries of india.

Syllabus for Semester III, B. TECH (Computer Science & Engineering)

Course Code:	**CSP228**	**Course:**	**Programming Lab-II**
L: 0 Hrs, T: 0 Hr, P: 2 Hr, Per Week		**Total Credits:**	**01**

Course Objectives

1. The objective of this course is to develop the ability of students to design android applications.
2. Use various features of android like broadcast receivers, content providers etc.
3. Effectively use files and database to store the data.
4. Use location-based services to develop navigation-based applications.

Syllabus

- Layout Manager: Linear, Relative, Table, Frame, Constraint Layout
- UI Widgets: Basic Views, Picker Views, List Views
- Activity, Intent and Android Menu
- Data Storage: Shared Preference, Internal Storage and External Storage
- Android Notification and Dialog
- SMS and Broadcast Receiver
- SQLite, Content Provider
- Location Services

Course Outcomes

On successful completion of the course, students will be able to:

1. Design basic android applications using UI resources: Activity, View group, menus
2. Develop android applications to enhance user interactivity by using toast, notification, dialogs etc.
3. Apply android persistence with shared preferences and files, store and retrieve data with databases and content providers.
4. Implement Android's communication APIs for SMS, location-based services.

Text Books

1. Beginning Android Programming with Android Studio, 4Ed by J. F. DiMarzio, Wrox publication.

2. Professional Android 4 Application Programming by Reto Meier, Wiley Publication

Reference Book

1. Android Programming for Beginners - Second Edition by John Horton, Packt Publishing Pvt. Ltd.

Syllabus for Semester IV, B. TECH (Computer Science & Engineering)

Course Code:	**CST299**	**Course:**	**Object Oriented Programming OPEN ELECTIVE**
L: 3 Hrs, T: 0 Hr, P: 0 Hr, Per Week		**Total Credits:**	**03**

Course Objectives

1. To make students understand Fundamental features of an object-oriented language like Java: object classes and interfaces, exceptions and libraries of object collections
2. Introduce students with concepts like exception handling, generics, multithreading, streams, and data persistence.

Syllabus

Unit I

Introduction to Programming Languages and its types, Features of Object- Oriented Programming Language, Features of Java, Java Runtime Environment, JDK, Java Variables and Data Types, Expressions and Operators, Decision Constructs and switch, Object Oriented Programming Paradigms, Class & Objects, String class, Math class, Packages.

Unit II

Using Object as a data member, Object as an argument, object as a return type. Introduction to Arrays, Input & Loop Constructs, Multi-dimensional arrays, Jagged arrays. Access control of members of a class, instantiating a class, method and constructor overloading.

Unit III

Concept of inheritance, methods of derivation, use of this keyword, super keyword and final keyword in inheritance. Static and non-static members. Abstract classes and methods, Interface, Implementation of interface, Runtime polymorphism, Dynamic method dispatch.

Unit IV

Exceptions, types of exception, use of try catch block, handling multiple exceptions, using finally, throw and throws clause, user defined exceptions.

Introduction to streams, byte streams, character streams, file handling in Java, Serialization.

Unit V

Generics: Generic class with one and two type parameters, bounded generics. Generic methods. Collection classes: Array list, Tree set.

Multithreading: Java Thread models, creating thread using runnable interface and extending Thread, thread priorities, Thread Synchronization, Interthread communications.

Unit VI

Basic SQL commands, DDL and DML commands, Java Database Connectivity, Working with Connection, Statement and result set, Data navigation and Data Manipulation using JDBC.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply object-oriented programming paradigms for creating stand-alone applications.
2. Develop robust programs to handle any type of exceptions.
3. Implement scalable code with the help of concepts like generics and multithreading.
4. Utilize resources such as files and databases for data storage in applications.

Text Books

1. Herbert Schildt; JAVA The Complete Reference; Ninth Edition, Tata McGraw- Hill Publishing Company Limited.
2. C Xavier; Java Programming: A Practical Approach; Tata McGraw - Hill Education Private Ltd 2011.

Reference Books

1. Cay S. Horstmann and Gary Cornell; Core JAVA Volume-II Advanced Features; Eighth Edition; Prentice Hall, Sun Microsystems Press 2008.
2. Herbert Schildt and Dale Skrien; Java Fundamentals A Comprehensive Introduction; Tata McGraw-Hill Education Private Ltd 2013.

Course Code: CST421 **Course: Cloud Computing**
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Pre-requisites
CST255 - Operating Systems, CST353 - Computer Networks

Course Objectives

The objective of this course is to impart necessary and practical knowledge of components of Cloud computing and develop skills required to design real-life cloud-based projects by:
1. Learning basics of cloud and challenges in implementation.
2. Identifying areas where cloud computing can be applied.
3. Understanding the cloud environment and its security issues.
4. Understanding the various cloud programming and software environments.

Syllabus

Unit I

Introduction: Traditional server concept: pros and cons. Need of Virtualized Technology, Benefits of Virtualization. Hypervisors, Types of Hypervisors, Full Virtualization, Para Virtualization, Hardware Assisted Virtualization, Types of virtualizations: Various forms of virtualization: Desktop, Application, Server, Hardware, Storage, Memory and I/O virtualization. Virtualization at different levels. Virtualization Vs Containerization, Concept of Containers, Container Orchestration.

Unit II

Virtualization Technologies: Software Virtualization, Hardware Virtualization, Application Virtualization, Storage Virtualization, OS Virtualization. Accomplishing Virtualization. High availability of Data Centers, Planned and unplanned maintenance activities in a datacenter. Migration: Migration consideration, Things to do before migration of servers- Discovery, Assessment and Migration. Concepts of Networking: CIDR Blocks, Subnet Mask, Designing a network.

Unit III

Cloud Computing Fundamental: Cloud deployment models, Cloud service models. Benefits and challenges of cloud computing. Regions, Availability zones, Edge locations. Fundamentals of pricing: Total cost of ownership (TCO) and monthly calculator.

Unit IV

Amazon Web services (AWS): AWS core services: Storage: Simple Storage service (S3), Elastic Block Store (EBS), Elastic File Store (EFS), Content Delivery Network (CDN), Cloudfront, Snowball, Snowmobile, Route53.

Unit V
Introduction to Microsoft Azure, Subscription, Resource Group, Resource Deployment Models, Different methods of creating resources: Azure portal, Cloud shell, ARM templates, Virtual Networking in cloud.

Unit VI

Compute Services in AWS and Azure: Elastic Compute Cloud (EC2) in AWS and VMs in Azure, Virtual Machine creation in Windows and Linux platform, Remote Desktop Protocol (RDP), Secure Shell (SSH) protocol, Security Groups, Load balancer, Auto scaling.

Course Outcomes:
On successful completion of the course, students will be able to:

1. Exhibit the understanding of virtualization technologies.
2. Apply the existing virtualization technologies to solve real-world problems.
3. Analyze architecture, services, and challenges in Cloud computing.
4. Know the services of the AWS platform and their usage.
5. Know services on the Microsoft Azure platform and their usage.

Text Books

1. Distributed and Cloud Computing: Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, 1st Edition Elsevier
2. Cloud Computing Basics: A Non-Technical Introduction, Anders Lisdorf, Apress Publication.
3. Cloud Computing Bible: Barrie sosinsky, Wiley- India Edition
4. Cloud Computing: A Practical approach for learning and Implementation, A. Srinivasan, J. Suresh, 1stEdition, Pearson Publication

Reference Books

1. The Complete Cornerstone Guide to Virtualization Best Practices: Ivanka Menken, Paperback, 2ndEdition, Emereo Pty Ltd.
2. Cloud Computing Explained: Implementation Handbook for Enterprise, 2013 Edition, Recursive Press Publication.
3. Enterprise Cloud Computing: Technology, Architecture, Applications, Gautam Shroff, 1st Edition, Cambridge University Press
4. Cloud Computing: Dr. Kumar Saurabh, 2nd Edition, Wiley- India Edition

Syllabus for Semester VII, B. Tech. Computer Science & Engineering

Course Code: CSP421 **Course: Cloud Computing Lab**
L: 3 Hrs, T: 0 Hr, P: 2 Hr, Per Week **Total Credits: 1**

Course Objectives

The objective of this course is to impart necessary and practical knowledge of components of Cloud computing and develop skills required to build real-life cloud-based projects by:

1. Studying various cloud environments.
2. Implementing various cloud programming concepts.
3. Designing and developing processes involved in creation of a cloud based application.

Practicals based on Cloud Computing Theory syllabus.

Course Outcomes

On completion of this course, the students will be able to:

1. Identify and use the appropriate cloud service for a given application.
2. Apply Virtualization technology.
3. Demonstrate the need for high availability in the Cloud environment.
4. Design and create a Virtual Network in the Cloud.
5. Deploy storage and application services on the Cloud platform.

Description
1. Configure Virtual Machines in Cloud
2. Configure High Availability of servers in Cloud
3. Configure Storage in Cloud
4. Configure Load Balancing and Auto Scaling for servers in Cloud
5. Design and Configure Network in Cloud
6. Establish Communication among Cloud Networks.

Course Code: CST422-1 Course: Program Elective-III: Data Analytics and Visualization
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To understand data analytics life cycle for solving challenging business problems.
2. To adopt appropriate statistical procedures for analysis based on goals and nature of data.
3. To employ best practices in data visualization to develop charts, maps tables and other visual representations of data.

Syllabus

UNIT - I:
Introduction and Overview
Importance of analytics and visualization, data preprocessing, Basic Analysis Techniques, Data Analytics Lifecycle and Different Phases

UNIT - II:
Association Rules and Regression
Association Rules: Overview, Apriori Algorithm, Evaluation of Candidate Rules, Frequent Itemsets and Rule Generation, Validation and Testing, Diagnostics.
Regression: Linear Regression, Logistic Regression, Choice of a Model.

UNIT - III:
Classification and Clustering
Overview, k-Means, k-Modes, Partitioning around Medoids (PAM), Hierarchical Agglomerative and Density Clustering Methods. **Classification**: Decision Trees – Overview, Detecting Significant Split, Algorithms and Evaluation; Naïve-Bayes – Bayes' Theorem, Naïve Bayes
Classifier, Smoothing; Diagnostics of Classifiers.

UNIT - IV:
Time Series Analysis
Time Series Analysis: Box-Jenkins Methodology, ARIMA (Auto Regressive Integrated Moving Average) Model, Choice of a Model, Overview of ARMAX, Spectral Analysis and GARCH.

UNIT - V :
Data Visualization Understanding
Understanding Data Visualization Principles, Mapping Data onto Aesthetics, Visualizing - Distributions, Proportions, Time Series, Trends and Uncertainty; Commonly used File Formats and Software.

UNIT - VI :

Creating Stories with Data

Why Planning?, Creating Interesting Stories with Data – Reader-driven Narratives, Author-driven Narratives; Perceptions and Presentation Methods, Best Practices in Visualization, Interactive Visualization, Event Listeners and Layouts, Case Studies for Visualization.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply data preprocessing and basic data analysis techniques.
2. Conduct data analytics using scientific methods.
3. Analyze time series data.
4. Create presentations and visualizations.

Text Books

1. David Dietrich, Barry Heller and Beibel Yang, - Data Science and Big Data Analytics – Discovering, Analyzing, Visualizing, and Presenting Data, John Wiley and Sons [EMC Education Services], 2015.
2. Claus O. Wilke, - Fundamentals of Data Visualization – A Primer on Making Informative and Compelling Figures, O'Reilly, 2019.
3. Python: Data Analytics and Visualization, Packt Publishing, 2017.

Reference Books

1. Jiawei Han, Micheline Kamber and Jian Pei, - Data Mining Concepts and Techniques, 3rd edition; Morgan Kaufmann Publishers, 2011.

Course Code: CSP422 Course: Program Elective-III Lab: Data Analytics and Visualization
L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Course Objectives

1. To apply statistical methods for data analytics to provide business solutions.
2. To develop insights based on analytical results to facilitate better understanding of consumer attitude, perceptions and behavior.
3. To create data visualizations for effective communication to user.

Syllabus

Experiments based on Data Visualization and Analytics Theory Syllabus preferably using R, Python, JavaScript.

Course Outcomes:
On successful completion of the course, students will be able to:

1. Apply different data preparation techniques.
2. Apply various data analysis techniques.
3. Apply analytics on time series data.
4. Design effective presentations and visualizations.

Reference Books

1. David Dietrich, Barry Heller and Beibel Yang, - Data Science and Big Data Analytics – Discovering, Analyzing, Visualizing, and Presenting Data, John Wiley and Sons [EMC Education Services], 2015.
2. Kyran Dale, - Data Visualization with Python and JavaScript – Scrape, Clean and transform Your Data, O'Reilly, 2016.

Course Code: CST422-2 Course: Program Elective-III: Customer Relationship Management
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To make the students understand the organizational need, benefits and process of creating long-term value for individual customers.
2. To disseminate knowledge regarding the concept of Salesforce and Salesforce technologies.
3. To enable the students understand the technological and human issues relating to implementation of Customer Relationship Management in the organizations.

Syllabus

UNIT I :

Introduction to CRM and Salesforce : Definition and importance of CRM, Key CRM concepts, Benefits of CRM for businesses, Introduction to Salesforce, Salesforce's role in business processes, Salesforce Cloud offerings , Overview of Salesforce architecture, Multi-tenant cloud architecture, Salesforce Data Model (Objects, Records, Fields), Understanding Tabs, Apps, and Objects

UNIT II :

Salesforce Administration Basics : Understanding Salesforce Setup menu, Creating and managing users, Profiles, Roles, and Permission Sets, Organizing security settings (Organization-Wide Defaults, Sharing Rules), Data Validation Rules, Creating and customizing Objects, Object Relationships, Creating and managing Fields, Workflow Rules, Process Builder, and Flow.

UNIT III:

Introduction to Apex Programming : Apex basics (Syntax, Variables, Methods), Apex classes and triggers, Working with SOQL and SOSL (Salesforce Object Query Language, Handling exceptions in Apex, Apex Triggers, Writing Apex triggers to handle database events, Trigger context variables and best practices, Governor limits and optimization techniques.

Unit IV:

Advanced Salesforce Development – Lightning Web Components (LWC) : Overview of Lightning Web Components (LWC), LWC architecture and lifecycle, Creating and deploying LWC components, Handling events in LWC, Working with Apex from LWC, LWC Integration with Salesforce Data, Displaying Salesforce data in LWC, Handling record pages and lightning layouts, Best practices for LWC development.

UNIT V :

Salesforce Integration and Deployment : Overview of integration in Salesforce, Integration tools: REST API, SOAP API, and Bulk API, Salesforce Connect, Introduction to Mulesoft for Salesforce integration, Introduction to Salesforce DX, Source-driven development and version control, Continuous Integration and Continuous Delivery (CI/CD) in Salesforce.

Unit VI:

Reports and Dashboards Reports : Introduction, types of reports, report builder, formatting reports, formula, dashboard introduction, dashboard generation, charts in dashboards, limitations of Salesforce

reports.

Course Outcomes:

On successful completion of the course, students will be able to:

6. Understand the fundamentals of Salesforce and its role in CRM.
7. Gain practical skills in configuring and managing Salesforce environments.
8. Apply the basics of Salesforce development using Apex to customize the platform.
9. Develop modern web applications on the Salesforce platform using Lightning Web Components.
10. Integrate Salesforce with external systems and deploy applications in a production environment.

Text Books

1. Jason Ouellette; Development with the Force.com Platform, Second Edn, Addison Wesley, 2011. 2. Mohith Shrivastava; Salesforce Lightning Application Development, 2018.
2. Mohith Shrivastava; Salesforce Lightning Application Development, 2018
3. Judith W .Kincaid , Customer Relationship Management Getting it Right, Pearson Education
4. Customer Centricity –Focus on right customer for strategic advantage, by Peter Fader, Wharton Digital Press, 2012

Reference Books

1. Learning Salesforce Development with Apex – Paul Battisson
2. Salesforce for Beginners – Sharif Shaalan

Course Code: CSP422-2 Course: Program Elective-III Lab: Customer Relationship Management Lab

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Course Objectives

4. To make the students understand the organizational need, benefits and process of creating long-term value for individual customers.
5. To disseminate knowledge regarding the concept of Salesforce and Salesforce technologies.
6. To enable the students understand the technological and human issues relating to implementation of Customer Relationship Management in the organizations.

Syllabus

The syllabus will be based on Salesforce environment management and use of Apex and Visualforce for implementing CRM using Salesforce

11. Creating and navigating a Salesforce Developer account and exploring Salesforce Classic and Lightning Experience interfaces
12. Basic Salesforce configurations and customizations
13. Creating custom objects and fields and setting up user roles and profiles
14. Implementing workflow rules and process automation
15. Writing basic Apex classes and triggers and Developing Visualforce pages to customize user interface
16. Building triggers for automation
17. Building basic LWC components and creating interactive user interfaces using LWC

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand the fundamentals of Salesforce and its role in CRM.
2. Gain practical skills in configuring and managing Salesforce environments.
3. Apply the basics of Salesforce development using Apex to customize the platform.
4. Develop modern web applications on the Salesforce platform using Lightning Web Components.

Text Books

5. Jason Ouellette; Development with the Force.com Platform, Second Edn, Addison Wesley, 2011. 2. Mohith Shrivastava; Salesforce Lightning Application Development, 2018.
6. Mohith Shrivastava; Salesforce Lightning Application Development, 2018
7. Judith W .Kincaid , Customer Relationship Management Getting it Right, Pearson Education
8. Customer Centricity –Focus on right customer for strategic advantage, by Peter Fader, Wharton Digital Press, 2012

Reference Books

3. Learning Salesforce Development with Apex – Paul Battisson
4. Salesforce for Beginners – Sharif Shaalan

Course Code: CST422-3 Course: Program Elective-III: Web Intelligence and Big Data
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To understand the principles, tools and methods of web intelligence.
2. To understand the basic concepts of big data analytics and the Big Data landscape.
3. Learn about various tools to address big data to extract insight from it and analyze the big data using intelligent techniques.

Syllabus

UNIT – I

Introduction to Web Intelligence, Architecture of a Web search engine, Web Indexing, Inverted indexes - Construction, Query processing; Link Analysis - Page Rank, Modified page rank, Search Engine Optimization using page rank; Finding Similar Items: Applications of Near-Neighbor Search, Shingling of Documents, Locality Sensitive hashing, Distance Measures.

UNIT - II

Information and Language, Web Data Mining, Mining Data Streams, Web Scraping for information extraction, Importance of Words- TF, IDF, TF-IDF, Analyzing Sentiment and Intent Load, Mutual information, Naïve bayes classifier and Bayesian networks.

UNIT - III

Recommender system - long tail problem, content-based recommendations, Collaborative filtering-Item based and user based, Frequent Itemsets, Graph based clustering, mining social network graphs.

UNIT – IV

Introduction to Big Data- Characteristics, Challenges and applications; Introduction to Big Data Stack, Introduction to Big Data Platforms, Hadoop: Features, advantages, Hadoop 1.0 - Hadoop2.0, overview of hadoop ecosystems, Hadoop Distributed File System (HDFS), HBase Architecture, Introduction to Map Reduce, Internal working of Map - Reduce, Map-reduce way of designing solutions with examples.

UNIT – V

Consistency Availability Partition Tolerance (CAP), Eventual Consistency, Consistency Trade-O-s, Basically Available Soft State Eventual Consistency (BASE),
Introduction to Cassandra: Architecture, Data Replication in Cassandra, Data model: cluster, keyspace, column family, Cassandra Keyspace Operations, CURD Operations, Cassandra CQL: data types, collections.

UNIT - VI

Introduction to Big Data Applications (Machine Learning), Overview of Big Data Machine Learning, Mahout Introduction, Introduction to Machine Learning with MLlib, Linear and logistic regression, Course Outcomes

Course Outcomes:

On Successful completion of this course, students will able to:

1. Apply Web analytics and artificial intelligence on web-based systems to improve their performance.

2. Demonstrate information retrieval, working of recommender systems, and extraction of data from the web.
3. Develop Big Data Solutions using Map-Reduce method and Hadoop Eco System.
4. Implement various functionalities of the NoSQL database and Machine Learning Techniques to address analyzing large data sets.

Text Books

1. Leskovec, Rajaraman, Ullman, Mining of Massive Datasets, Cambridge University Press.
2. Tom White; Hadoop: The Definitive Guide, 4th Edition, O'Reilly, 2015.
3. Hien Luu; Beginning Apache Spark 2: With Resilient Distributed Datasets, Spark SQL, Structured Streaming And Spark Machine Learning Library, Apress, 2018
4. Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Wiley, 2014
5. Michael Minelli and Michele Chambers; Big Data, Big Analytics: Emerging Business Intelligence and Analytic trends for Today's Business, John Wiley & Sons, 2013.
6. Judith Hurwitz and Alan Nugent; Big Data for Dummies; John Wiley & Sons, 2013.

Reference Books

1. Eric Siegel, Thomas H. Davenport; Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die, John Wiley & Sons, 2013.
2. Dirk Deroos et al., Hadoop for Dummies, Dreamtech Press, 2014.
3. I.H. Witten and E. Frank, Data Mining: Practical Machine learning tools and techniques.
4. P. Simon, Too Big to Ignore: The Business Case for Big Data; Wiley, 2015.

Course Code: CSP422-3 Course: Program Elective-III Lab: Web Intelligence and Big Data Lab
L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Course Objectives:

1. To develop understanding about working of search engines and perform optimization for improving the search results.
2. To apply various techniques to perform tasks related to web intelligence.
3. Understand the big data platforms, analytic techniques and applications.
4. Perform analysis of big data using suitable technique and use of different tools.

Experiment based on syllabus of Web Intelligence and Big Data Theory.

Course Outcomes:

On Successful completion of this course, students will able to:

1. To implement intelligent web-based systems for solving a well-defined web intelligence problem using tools and techniques, by analyzing the data to extract insight from it.
2. To demonstrate the working of the recommender systems.
3. To perform big data analytics on structured and unstructured data, using various big data analytic techniques.
4. To apply appropriate Machine Learning algorithms and big data techniques to address real-world problems.

References:

1. Tom White; Hadoop: The Definitive Guide, 4th Edition, O'Reilly, 2015.
2. Hien Luu; Beginning Apache Spark With Resilient Distributed Datasets, Spark SQL, Structured Streaming and Spark Machine Learning Library, Apress, 2018.
3. Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Wiley, 2014
4. H. Witten and E. Frank, Data Mining: Practical Machine learning tools and techniques.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering

Course Code: CST422-4 **Course: Program Elective-III: Business Intelligence**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objectives

1. To learn about Business Intelligence
2. To learn about how to take effective and timely decisions in an organization.
3. To learn how to extract knowledge from data and information.
4. To learn how to draw conclusions, predictions and take futuristic actions.
5. To learn the architecture of BI system.

Syllabus

UNIT – I

Business intelligence: Effective and timely decisions, Data, information and knowledge, The role of mathematical models, Business intelligence architectures, Ethics and business intelligence Decision support systems: Definition of system, Representation of the decision-making process, Evolution of information systems, Definition of decision support system, Development of a decision support system

UNIT –II

Mathematical models for decision making: Structure of mathematical models, Development of a model, Classes of models Data mining: Definition of data mining, Representation of input data , Data mining process, Analysis methodologies Data preparation: Data validation, Data transformation, Data reduction

UNIT –III

Business intelligence applications: Marketing models: Relational marketing, Sales force management, Logistic and production models: Supply chain optimization, Optimization models for logistics planning, Revenue management systems. Data envelopment analysis: Efficiency measures, Efficient frontier, The CCR model, Identification of good operating practices

UNIT –IV

Decision Support Systems modeling, Structure of mathematical models for decision support, Certainty, Uncertainty, and Risk, Decision modeling with spreadsheets, Mathematical programming optimization, Decision Analysis with Decision Tables and Decision Trees, Multi-Criteria Decision Making With Pairwise Comparisons.

UNIT –V

Knowledge Management: Introduction to Knowledge Management, Organizational Learning and Transformation, Knowledge Management Activities, Approaches to Knowledge Management, Information Technology (IT) In Knowledge Management, Knowledge Management Systems Implementation, Roles of People in Knowledge Management Artificial Intelligence and Expert Systems: Concepts and Definitions of Artificial Intelligence, Artificial Intelligence Versus Natural Intelligence, Basic Concepts of Expert Systems, Applications of Expert Systems, Structure of Expert Systems, Knowledge Engineering, Development of Expert Systems

Course Outcomes:

On successful completion of the course, students will be able to:

1. Able to analyze Business Intelligence, Analytics and Decision Support
2. Explain the technologies for decision making
3. Apply predictive modelling techniques
4. Understand the use of BI in real world applications
5. Understand the expert systems.

Text Books

1. Business Intelligence: Data Mining and Optimization for Decision Making, Carlo Vercellis, Wiley
2. Decision support and Business Intelligence Systems, Efraim Turban, Ramesh Sharda, Dursun Delen, Pearson
3. Fundamental of Business Intelligence, Grossmann W, Rinderle-Ma, Springer

Reference Books

1. Ramesh Sharda, Dursun Delen, EfraimTurban, J.E.Aronson,Ting-Peng Liang, David King, "Business Intelligence and Analytics: System for Decision Support", 10th Edition, Pearson Global Edition, 2013

Course Code: CST423-1 Course: Program Elective-IV: Natural Language Processing
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To familiarize the concepts and techniques of natural language processing.
2. To learn computational techniques that enable machines to process, understand, and generate human language efficiently.
3. To apply the statistical learning methods and cutting-edge research models to solve natural language processing problems.
4. To integrate natural language processing into real-world systems to develop, evaluate, and enhance applications.

Syllabus:

UNIT I

Introduction to NLP, Definition and Scope, A Brief History, Importance, Challenges, Tasks, Significance, NLP Pipeline and Applications, Morphological Analysis and Generation using Finite State Automata, Finite State Transducer, Hidden Markov model, Viterbi Algorithm, Applications of Tagging.

UNIT II

Lexical Analysis, Part-of-Speech (POS) Tagging, Approaches for POS Tagging, Rule-Based, Stochastic, Hybrid Approach, Taggers Evaluations, Tokenization with NLTK.

UNIT III

Syntax and Parsing, Types of Constituents in Sentences, Context-Free Grammar (CFG), CFG Parsing, Top-Down Parser, Bottom-Up Parser, Shallow Parsing and Chunking, Thematic Roles, Conditional Random Fields, Maximum Likelihood Estimation, Lexical and Probabilistic Parsing, Probabilistic Context Free Grammars, The Probability of a String, Inside-Outside Algorithm, CKY Parsing.

UNIT IV

Semantic Analysis, Lexical Vs Compositional Semantic Analysis, Word Senses and Relations, Types of Lexical Semantics, Word Sense Disambiguation, WordNet and Online Thesauri, Word Similarity and Thesaurus Methods, Text Representation, Word Embedding, TF-IDF, Bag of Words, Word2Vec, Skip-gram. **Pragmatic Analysis and Discourse,** Discourse Phenomena, Coherence and Coreference, Importance of Coreference Relations, Discourse Segmentation, Algorithms for Coreference Resolution.

UNIT V

N-Gram Language Model, Language Modeling and Chain Rule, Markov Chain in N-Gram Model, Shannon's Method in N-Gram Model, Smoothing Techniques, Extrinsic Evaluation Scheme, Zero Counts Problems, Smoothing Techniques, Laplace (Add-One) Smoothing, Add-k Smoothing, Backoff and Interpolation Smoothing, Good Turing Smoothing, The Transformer, Large Language Models, Language Model Evaluation, Entropy, Perplexity, ROUGE, BLEU.

UNIT VI

Major NLP Applications, Information Retrieval Systems, Social Network Analysis, Sentiment Analysis, Information Extraction, Named Entity Recognition, Text Classification, Text Summarization Systems, Machine Translation, Word Alignment, Content Recommendation System, Answering Questions, Applications in Finance, E-Commerce, Travel and Hospitality, Healthcare, Supply Chain,

Telecommunication, Education and Research.

Course Outcomes

On Successful completion of course, students will be able to:

1. Understand core NLP concepts and techniques.
2. Apply various POS tagging approaches and parsing techniques to analyze sentence structure and utilize probabilistic models for syntactic analysis.
3. Analyze various semantic and pragmatic analysis techniques and discourse phenomena to enhance text representation and understanding.
4. Implement N-Gram language models and Transformer-based models for effective language modeling and text generation.
5. Design and develop innovative NLP solutions to address real-world challenges across industries like finance, healthcare, e-commerce, education and research.

Textbooks

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition, 2025, https://web.stanford.edu/~jurafsky/slp3.
2. Raymond ST. Lee, Natural Language Processing: A Textbook with Python Implementation, Springer Nature Singapore Pte Ltd. 2024, ISBN: 978-9819919987.
3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana, Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media, Inc., USA, First edition, 2020, ISBN: 978-1492054054.
4. Dipanjan Sarkar, Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition, Apress Media, LLC, California, 2019, ISBN: 978-1484243534.

Reference Books

1. Natural Language Processing with Python: From Basics to Advanced Projects, Second Edition, 2024, Cuantum Technologies LLC. Plano, ISBN: 979-8894968483.
2. Jyotika Singh, Natural Language Processing in the Real World: Text Processing, Analytics, and Classification, First edition, 2023, CRC Press is an imprint of Taylor & Francis Group, LLC, ISBN: 978-1003264774.
3. Gerhard Paaß and Sven Giesselbach, Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer Nature Switzerland Pte Ltd. 2022, ISBN: 978-3031231896.
4. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media, Inc., USA, Revised First edition, May 2022, ISBN: 978-1098136796.

Course Code: CSP423-1 Course: Program Elective-III Lab: Natural Language Processing Lab
L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Syllabus:

Experiments based on the above syllabus.

1. Few lab sessions shall be conducted using virtual lab platforms to enhance learning experiences and accessibility.
2. Utilise Hugging Face, Stanford, Kaggle, MIT OpenCourseWare, OpenAI Learning, Fast.ai, AllenNLP like repositories and platforms.

Course Code: CST423-2 **Course: Program Elective-IV: Deep Learning-I**
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objectives

1. To introduce basic deep learning algorithms.
2. To understand real-world problems which can be solved by deep learning methods.
3. To identify deep learning techniques suitable for a real-world problem.

Syllabus:

UNIT I: Basics of Deep Learning
History of Deep Learning, McCulloch Pitts Neuron, Thresholding Logic, Perceptrons,, Perceptron Learning Algorithm and Convergence, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons.

UNIT II: Training of Feedforward Neural Networks
Feedforward Neural Networks, Representation Power of Feedforward Neural Networks, Training of Feedforward Neural Networks, Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam

UNIT III: Optimization Algorithm
Activation Function and Initialization Methods: Sigmoid, Tanh, ReLU, Xavier and He Initialization, Regularization: Bias and Variance, Overfitting, Hyperparameters Tuning, L1 and L2 Regularization, Data Augmentation and Early Stopping, Parameter Sharing and Tying.

UNIT IV: Convolutional Neural Network (CNN)
Convolutional Neural Networks, 1D and 2D Convolution, Visualizing Convolutional Neural Networks, Guided Backpropagation.

UNIT V: Recurrent Neural Network (RNN)
Recurrent Neural Networks, Backpropagation Through Time (BPTT), Vanishing and Exploding Gradients, Long Short-Term Memory (LSTM) Cells, Gated Recurrent Units (GRUs).

UNIT VI: Variants of CNN and RNN

Encoder-Decoder Models, Attention Mechanism, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet.

Course Outcomes:
On successful completion of the course, students will be able to:

1. **Apply** fully connected deep neural networks to real-world problem-solving scenarios.
2. **Evaluate** the performance of various deep learning models in terms of optimization, bias-variance trade-off, overfitting, and underfitting.

3. **Analyze** the role of convolutional and recurrent neural networks in addressing different real-world problems.
4. **Create** advanced deep learning models by designing variants of CNNs and RNNs tailored to specific applications.

Text Books

Sandro Skansi, *Introduction to Deep Learning*, Springer

1. Charu C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, Springer, 2019
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016
3. Dr. S. Lovelyn Rose, Dr. L. Ashok Kumar, Dr. D. Karthika Renuka, *Deep Learning using Python*, Wiley Publication

Reference Books:

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006
2. Yegnanarayana, B., *Artificial Neural Networks*, PHI Learning Pvt. Ltd., 2009
   A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering Optimization: Methods and Applications*, John Wiley & Sons, Inc., 2016

Syllabus for Semester VII, B. Tech. Computer Science & Engineering

Course Code: CSP423-2 Course: Program Elective-IV: Deep Learning-I Lab

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Course Objectives:

1. To solve problems in linear algebra, probability, optimization using artificial neurons.
2. To understand the usage of publically available datasets.
3. To use various python packages and tools for deep learning.

Course Syllabus:

Experiments based on:

Implementation of Linear Algebra, Probability etc.
CST423-2 Syllabus.

Technology: Python, Tensorflow/Pytorch

Course Outcomes:

On completion of the course the student will be able to

1. Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
2. Implement deep learning models in Python using the PyTorch/Tensorflow library and train them with real-world datasets.
3. Analyze and evaluate deep learning model's.

Text Books:

1. Sandro Skansi, Introduction to Deep Learning, Springer
2. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Reference Books:

1. Francois Chollet, Deep Learning with Python, Manning Publications Co.
2. Golub, G., H., and Van Loan, C., F., Matrix Computations, JHU Press, 2013.

Course Code: CST423-3 Course: Program Elective-IV: Digital Image Processing
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To introduce the students to digital image processing fundamentals.
2. To enable students to analyze spatial and frequency domain filtering.
3. To aid students in using image compression and morphological algorithms.
4. To facilitate use of image segmentation and feature extraction.

SYLLABUS

UNIT – I: Digital Image Fundamentals

Digital Image Processing – Introduction, Applications of Image Processing; Fundamental Steps in Digital Image Processing; Components of Digital Image Processing.

Elements of Visual Perception; Image Sensing and Acquisition; Image Sampling and Quantization; Basic Relationships Between Pixels – Neighborhood, Adjacency, Connectivity, Distance Measures and Mathematical Operations.

UNIT – II: Intensity Transformations and Special Filtering

Basic Intensity Transformation Functions; Histogram Processing; Fundamentals of Spatial Filtering; Spatial Correlation and Convolution; Smoothing Spatial Filters; Sharpening Spatial Filters; High-Pass, Band-Reject, and Band-Pass Filters; Combining Spatial Enhancement Methods.

UNIT – III: Filtering in the Frequency Domain

Preliminary Concepts; Sampling and the Fourier Transform of Sampled Functions; Discrete Fourier Transform of One Variable; Extensions to Functions of Two Variables; Some Properties of the 2-D DFT and IDFT;

Basics of Filtering in the Frequency Domain; Image Smoothing Using Low-Pass Frequency Domain; Filters; Image Sharpening Using High-Pass Filters; Selective Filtering; The Fast Fourier Transform.

UNIT – IV: Image Compression and Color Image Processing

Fundamentals; Huffman Coding; Golomb Coding; Arithmetic Coding; LZW Coding; Run-length Coding; Symbol-based Coding; Bit-plane Coding; Block Transform Coding; Predictive Coding; Wavelet Coding.

Color Fundamentals; Color Models; Pseudo-color Image Processing; Basics of Full-Color Image Processing; Color Transformations.

UNIT – V: Morphological Image Processing

Introduction; Erosion and Dilation; Opening and Closing; The Hit or Miss Transform; Basic Morphological Algorithms – Boundary Extraction, Hole Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening, Skeletons and Pruning; Basic Grayscale Morphological Algorithms and Morphological Reconstruction.

UNIT – VI: Image Segmentation and Feature Extraction

Foundation; Point, Line, and Edge Detection; Thresholding; Segmentation by Region Growing and by Region Splitting and Merging; Region Segmentation Using Clustering and Superpixels; Segmentation Using Morphological Watersheds; The Use of Motion in Segmentation.

Boundary Preprocessing; Boundary Feature Descriptors; Region Feature Descriptors; Principal Components as Feature Descriptors; Whole-Image Features; Scale-Invariant Feature Transform (SIFT).

Course Outcomes:

On completion of the course the student will be able to

1. Review fundamental concepts in digital image processing.
2. Perform image enhancement in spatial domain and frequency domain.
3. Demonstrate different algorithms for image compression.
4. Use different morphological techniques to process images.
5. Apply image segmentation and feature extraction for object detection.

Text Books:

1. Rafael C. Gonzalez and Richard E, Woods; Digital Image Processing; Fourth Edition; Pearson Education; 2018.
2. S. Jayaraman, S. Esakkirajan and T. Veerakumar; Digital Image Processing; Tata McGraw Hill Education; 2009.

References:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle; Image Processing, Analysis, and Machine Vision; Third Edition; Cengage Learning; 2017.
2. Anil K. Jain; Fundamentals of Digital Image Processing; Second Edition; Prentice Hall of India; 2004.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering

Course Code: CSP423-3 Course: Program Elective-IV: Digital Image Processing Lab

L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week Total Credits: 1

Course Objectives

1. To introduce students to wide array of filters (smoothing and sharpening) for image enhancement.
2. To understand the prominent color spaces and transformations.
3. To visualize the impact of morphological processing and supporting methods.
4. To use image segmentation for improved image understanding.

Experiments based on CST423-3 Syllabus using MATLAB/Python.

Syllabus

Experiments designed to simulate the following tasks:

- Mathematical Operations on Images
- Image Enhancement with Spatial Filters
- Image Enhancement with Frequency Filters
- Morphological Operations on Image
- Edge Detection and Linking
- Image Segmentation
- Working with Color Models

Course Outcomes:

On completion of the course the student will be able to

1. Realize basic geometric transformations on images.
2. Perform image enhancement in spatial domain and frequency domain.
3. Demonstrate different algorithms for color space transformations.
4. Use different morphological techniques to process images.
5. Apply image segmentation and feature extraction for object detection.

Reference Books:

1. Rafael C. Gonzalez, Richard Eugene Woods and Steven L. Eddins; Digital Image Processing Using MATLAB; Pearson Education; 2004.
2. Sandipan Dey; Image Processing Masterclass with Python; BPB Publication; 2021.

Course Code: CST423-4 Course: Program Elective-IV: Parallel Computing
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week Total Credits: 3

Course Objectives

1. To introduce the fundamental principles of parallel computing, including Flynn's taxonomy, types of parallelism, and performance evaluation metrics.
2. To enable students to understand and apply parallel algorithm design techniques using shared-memory and many-core architectures.
3. To develop the ability to analyze and optimize the performance of parallel programs using OpenMP and CUDA programming models.

Syllabus

UNIT – I:
Motivation and need for parallelism, Flynn's Taxonomy (SISD, SIMD, MISD, MIMD), Types of parallelism: data, task, pipeline, Performance metrics: speedup, efficiency, scalability, Amdahl's Law, Gustafson's Law, Introduction to parallel architectures (shared vs distributed memory)

UNIT – II:
Shared memory model, memory hierarchy, Thread-level parallelism and synchronization, OpenMP programming model, Concepts: critical sections, barriers, atomic operations, Case studies: Parallel Merge Sort, Parallel Prefix Sum

UNIT – III:
Message Passing Interface (MPI) basics, Point-to-point and collective communication, Latency, bandwidth, communication cost modeling, MPI send/receive, broadcast, scatter/gather, reduce, Sample applications: distributed sorting, matrix multiplication

UNIT – IV:
GPU architecture and SIMT model, CUDA programming: kernels, memory types, thread hierarchy, Optimizing memory access: shared memory and coalescing, Example problems: 2D stencil computation, matrix operations, Introduction to OpenCL, SYCL, and OpenACC

UNIT – V:
Parallel reduction, pointer jumping, list ranking, Graph algorithms: parallel BFS, DFS, SSSP, Parallel dynamic programming basics, Scalability analysis and performance tuning, Debugging and profiling parallel code.

UNIT – VI:
MapReduce programming paradigm, Apache Spark: RDDs, transformations, actions, Fault tolerance and lineage in Spark, Case studies: Triangle counting, Connected Components, SSSP in Spark, Big Data applications in distributed environments

Course Outcomes:
On successful completion of the course, students will be able to:

1. Explain key concepts in parallel computing such as speedup, scalability, efficiency, and hardware classifications based on Flynn's taxonomy.

2. Develop and execute parallel programs using OpenMP and CUDA for computational problems like sorting, matrix operations, and prefix computations.
3. Analyze the performance of parallel algorithms using theoretical models and identify bottlenecks and inefficiencies.
4. Evaluate and optimize parallel solutions for problem-specific requirements using appropriate techniques and memory management strategies.

Text Books

1. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). *Introduction to Parallel Computing* (2nd ed.). Pearson Education. ISBN: 978-0201648652
2. McCool, M., Reinders, J., & Robison, A. (2012). *Structured Parallel Programming: Patterns for Efficient Computation*. Morgan Kaufmann. ISBN: 978-0124159938
3. Kirk, D. B., & Hwu, W. W. (2016). *Programming Massively Parallel Processors: A Hands-on Approach* (3rd ed.). Morgan Kaufmann. ISBN: 978-0128119860

Reference Books

1. Quinn, M. J. (2003). *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill. ISBN: 978-0072822564
2. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., & McDonald, J. (2001). *Parallel Programming in OpenMP*. Morgan Kaufmann. ISBN: 978-1558606715
3. Hwu, W. W. (Ed.). (2011). *GPU Computing Gems: Emerald Edition*. Morgan Kaufmann. ISBN: 978-0123849885

Course Code: CSP423-4 **Course: Program Elective-IV: Parallel Computing Lab**
L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 1**

Course Objectives

1. To provide practical experience in writing and debugging parallel programs using OpenMP and CUDA.
2. To strengthen understanding of thread-level parallelism, synchronization, memory management, and performance tuning in multicore and GPU environments.
3. To enable students to apply parallel computing concepts to implement efficient solutions for real-world problems.

Syllabus

Practicals based on topics covered in the theory syllabus, including implementation and performance evaluation of parallel and distributed algorithms using OpenMP, MPI, CUDA, and Spark.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Write and execute parallel programs using OpenMP and CUDA for operations such as prefix sum, matrix multiplication, and sorting.
2. Measure and analyze the performance of parallel programs concerning execution time, scalability, and memory usage.
3. Design optimized solutions using shared memory and GPU techniques for computationally intensive tasks
4. Evaluate the efficiency, correctness, and code quality of parallel solutions developed in lab exercises.

Course Code: CSP424 **Course: Project-II**
L: 0 Hrs, T: 0 Hr, P: 8Hr, Per Week **Total Credits: 4**

Course Objectives

The objective of project is to let the students apply theoretical knowledge and practical skills acquired in the previous semesters to solve a real-world problem by developing an innovative, and efficient solution while enhancing problem-solving, teamwork, and research capabilities.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Identify and finalize the problem statement by investigating various domains and society needs.
2. Perform requirement analysis and design methodology for solving the identified problem.
3. Apply programming techniques and modern tools for the development of the solution.
4. Apply ethical principles, project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.

Syllabus for Semester VII, B. Tech. Computer Science & Engineering

Course Code: CSP425 **Course: Industry Internship Evaluation**
L: 0 Hrs, T: 0 Hr, P: 2Hr, Per Week **Total Credits: 0**

Course Objectives

1. To enable students to explore opportunities for alternative career development.
2. To assess interests and abilities in the respective field of study and to integrate theory and practice.
3. To learn workplace habits and develop attitudes and skills necessary for job success.

Scope

Students are expected to complete the internship before the start of the VII semester. The industry internship evaluation will be carried out during the VII Semester.

Mode of Conduction

Each student will be evaluated through Seminar-cum-Presentations on the following parameters

- Technology
- Domain Understanding
- Outcomes.

Course Outcomes

On completion of the course, the student will be able to

1. Apply fundamental science and engineering principles to identify and evaluate real-world challenges.
2. Use modern tools and demonstrate a continuous learning approach.
3. Communicate technical information employing written reports and presentations.

Course Code: CST426-1 **Course: Program Elective-V: Bio-Informatics**
L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objectives

1. Provide an introduction to the field of Bioinformatics.
2. Describe how bioinformatics data is stored and organized.
3. Provide an approach to build search query and sequence alignment.
4. Provide methods for genome analysis.

Syllabus

UNIT-I

Introduction to Bioinformatics: Genome Sequences ORFs, Genes, Introns, Exons, Splice Variants DNA/RNA Secondary Structure, Retrieval methods for DNA sequence, protein sequence and protein structure information.

UNIT-II

Databases – Format and Annotation: Conventions for database indexing and specification of search terms, Common sequence file formats. Annotated sequence databases - primary sequence databases, protein sequence and structure databases; Organism specific databases, Data retrieval tools – Entrez, DBGET and SRS, Submission of (new and revised) data.

UNIT-III

Sequence Similarity Searches: Local versus global, Distance metrics, Similarity and homology, scoring matrices, PAM, BLOSUM, PSSM, Dot Plot.

UNIT-IV

Dynamic programming algorithms: Needleman-Wunsch and Smith-waterman, Heuristic Methods of sequence alignment, FASTA, BLAST and PSI BLAST.
Multiple Sequence Alignment and software tools for pair wise and multiple sequence alignment, ClustalW algorithm - Feng Doolittle algorithm.

UNIT-V

Phylogenetic Analysis: Methods of phylogenetic analysis, UPGMA, WPGMA, neighbour joining method, Fitch/Margoliash method, Character Based Methods. Whole genome analysis, existing software tools, Genome Annotation and Gene Prediction, Comparative genomics, orthologs, paralogs.

UNIT-VI

Genome Analysis: Whole genome analysis, existing software tools, Genome Annotation and Gene Prediction, Comparative genomics, orthologs, paralogs. Bioinformatics in the Pharmaceutical Industry- Drug discovery**.**

Course Outcomes:

On successful completion of the course, students will be able to:

1. Demonstrate the fundamentals of biological processes and data acquisition.
2. Apply sequence alignment algorithms to search biological databases.
3. Implement phylogenetic tree construction algorithms.

4. Analyze genes and their sequences for applications such as drug discovery.

Text Books

1. Bioinformatics: Databases and Systems, by Stanley I. Letovsky
2. Bioinformatics Databases: Design, Implementation, and Usage (Chapman & Hall/ CRC Mathematical Biology & Medicine), by Sorin Draghici
3. Data base annotation in molecular biology, principles and practices, Arthur M.Lesk
4. Current topics in computational molecular biology, Tao, Jiang, Ying Xu, Michael Q.Zang

Reference Books

1. D. Baxevanis and F. Oulette, (2002) "Bioinformatics : A practical guide to the analysis of genes and proteins", Wiley Indian Edition.
2. Cynthia Gibas and Per Jambeck (2001), "Developing Bioinformatics Computer Skills". O'Reilly press, Shorff Publishers and Distributors Pvt. Ltd., Mumbai.
3. Bryan Bergeron MD (2003), "Bioinformatics Computing". Prentice Hall India(Economy Edition)

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering

Course Code: CST426-3 **Course: Program Elective-V: Bio-Informatics**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objectives

1. Learn the basics of bioinspired intelligence and its applications.
2. Implement Genetic Algorithms, Differential Evolution, ACO, and PSO for optimization.
3. Explore Artificial Immune Systems and other bioinspired techniques in real-world problems.

Syllabus

UNIT-I:

Introduction to Bioinspired Intelligence: Fundamentals of Bioinspired Computing, Biological and Natural Computation Principles, Differences Between Conventional AI and Bioinspired AI, Applications in Engineering, Optimization, and AI, Key Characteristics of Bioinspired Algorithms such as Adaptability, Self-organization, Parallelism, Optimization & Robustness

UNIT-II

Introduction to Genetic Algorithms: Fundamental Components of GA, Population, Chromosome Encoding, Fitness Function, Selection Mechanisms, Crossover Operators, Mutation Operators, GA Workflow & Process, Applications of Genetic Algorithms

UNIT-III

Introduction to Differential Evolution (DE): Definition and Concept of DE, Biological and Evolutionary Inspiration, Comparison of DE with Genetic Algorithms (GA) and other Evolutionary Algorithms, Fundamental Concepts of DE, Differential Evolution Algorithm, applications of DE

UNIT-IV

Ant Colony Optimization (ACO): Biological Inspiration – Ant behaviour and pheromone trails, ACO Algorithm Mechanics – Pheromone updating, exploration vs. exploitation, Variants of ACO – Max-Min ACO, Rank-Based ACO

UNIT-V

Particle Swarm Optimization (PSO): Biological Inspiration – Swarming behaviour in nature, PSO Algorithm Mechanics – Particle velocity, position updating, and social learning, Applications.

UNIT-VI

Artificial Immune Systems & Other Bioinspired Techniques: Artificial Immune Systems (AIS) – Self-learning, anomaly detection, cybersecurity, Firefly Algorithm & Bat Algorithm – Bioinspired metaheuristic optimization, Cuckoo Search & Grey Wolf Optimizer – Nature-inspired search strategies, Applications.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Describe the principles of bio-inspired intelligence and its applications in AI and optimization.
2. Apply evolutionary algorithms, such as Genetic Algorithms (GA) and Differential Evolution (DE), for problem-solving.

3. Implement swarm intelligence techniques such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).
4. Explore Artificial Immune Systems (AIS) and other bioinspired metaheuristic algorithms for their applications in real-world scenarios.

Text Books

1. "Nature-Inspired Optimization Algorithms" by Xin-She Yang
2. "Bio-Inspired Computation in Unmanned Aerial Vehicles" by Dawei Han, Wei Gao, and Zhiqiang Wei

Reference Books

1. "Evolutionary Computation: A Unified Approach" by Kenneth A. De Jong
2. “Artificial Immune Systems: A New Computational Intelligence Approach" by Leandro Nunes de Castro and Fernando J. Von Zuben
3. "Swarm Intelligence: Principles, Advances, and Applications" by Marco Dorigo, Mauro Birattari, and Thomas Stützle

Course Code: CST427-3 **Course: Program Elective-VI: Bio-Informatics**

L: 3 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 3**

Course Objectives

1. Analyse fundamental methods for designing and deploying Virtual and Augmented Reality applications.
2. Explain the working principles of Virtual and Augmented Reality systems.
3. Comprehend virtual world spaces and various input-output modalities used in Virtual Reality interfaces.
4. Develop animations and physical simulations within virtual environments.
5. Select, create, and critically evaluate designs for Augmented Reality experiences, providing rationale for design choices.

Syllabus

UNIT-I:

Introduction to Virtual Reality

Fundamental Concept and Components of Virtual Reality, Primary Features and Present Development on Virtual Reality, Computer graphics, Real time computer graphics, Flight Simulation, requirements of virtual environment, benefits of virtual reality, Hardware technologies for 3D user interface: Visual Displays Auditory Displays, Haptic Displays, Choosing Output Devices for 3D User Interfaces

UNIT-II:

3D Computer Graphics

The Virtual world space, the perspective projection, human vision, 3D clipping, Colour theory, Simple 3D modeling, Illumination models, Reflection models, Shading algorithms, Hidden Surface Removal

Multiple Modals of Input and Output Interface in Virtual Reality:

Input - Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3D Menus & 3D Scanner Output - Visual / Auditory / Haptic Devices

UNIT-III:

Environment Modeling

Geometric Modeling, behavior simulation, from 2D to 3D, 3D spaces curves, 3D boundary representation Geometrical Transformations: Frames of reference, Modeling transformations, Instances, Picking, Flying, Scaling the VE, Collision detection Generic VR system: Introduction, Virtual environment, Computer environment, VR technology, Model of interaction, VR Systems.

UNIT-IV:

Animating the Virtual Environment

Introduction, The dynamics of numbers, Linear and Nonlinear interpolation, the animation of objects, linear and non-linear translation, free from deformation, particle system

Physical Simulation

Introduction, Objects falling in a gravitational field, Rotating wheels, Elastic collisions, projectiles, simple pendulum, springs, Flight dynamics of an aircraft

UNIT-V: : Augmented Reality

Introduction to Augmented reality technology, technology and features of augmented reality Augmented reality in everyday world, Types of augmented reality, The similarities and differences between AR and VR, Multimodal displays, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality

UNIT-VI:

Augmented Reality Applications

Wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, evaluating AR systems.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand the fundamental principles, components, and technologies of Virtual and Augmented Reality.

2. Apply 3D modeling, shading algorithms, input-output interfaces, and geometric transformations to develop immersive and interactive virtual environments.

3. Analyze animation techniques such as interpolation, particle systems, and physical simulations to evaluate their influence on realism and interactivity in VR.

4. Develop AR applications using marker-less tracking, mobile projection interfaces, and multimodal displays, assessing their practical effectiveness.

Text Books

1. John Vince, -Virtual Reality Systems||, Pearson Education Asia, 2007.
2. Steven M. LaValle - Virtual Reality , Cambridge University Press, 2016
3. Adams, -Visualizations of Virtual Reality||, Tata McGraw Hill, 2000
4. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.
5. Alan B Craig, William R Sherman and Jeffrey D Will, - Developing Virtual Reality Applications: Foundations of Effective Design||, Morgan Kaufmann, 2009.

Reference Books

1. Anand R., - Augmented and Virtual Reality, Khanna Publishing House, Delhi.
2. Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, - 3D User Interfaces, Theory and Practice, Addison Wesley, USA, 2005.
3. Burdea, Grigore C and Philippe Coiffet, - Virtual Reality Technology, Wiley Interscience, India, 2003.
4. Adams, - Visualizations of Virtual Reality, Tata McGraw Hill, 2000
5. Tony Parisi - Learning Virtual Reality , O'Reilly Media, Inc., 2015, ISBN- 9781491922835
6. Howard Rheingold, - Virtual Reality: The Revolutionary Technology and how it Promises to Transform Society, Simon and Schuster, 1991.

Course Code: CSP428 **Course: Project-III**
L: 0 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 6**

Course Objectives

The objective of project is to let the students apply theoretical knowledge and practical skills acquired in the previous semesters to solve a real-world problem by developing an innovative, and efficient solution while enhancing problem-solving, teamwork, and research capabilities.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Identify and finalize the problem statement by investigating various domains and society needs.
2. Perform requirement analysis and design methodology for solving the identified problem.
3. Apply programming techniques and modern tools for the development of the solution.
4. Apply ethical principles, project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.

Syllabus for Semester VIII, B. Tech. Computer Science & Engineering

Course Code: **Course: Full Semester Industry Internship**

L: 0 Hrs, T: 0 Hr, P: 0Hr, Per Week **Total Credits: 12**

Course Objectives

1. To enable students to explore opportunities for alternative career development.
2. To assess interests and abilities in the respective field of study and to integrate theory and practice.
3. To learn workplace habits and develop attitudes and skills necessary for job success.

Course Outcomes:

On successful completion of the course, students will be able to:

1. Apply fundamental science and engineering principles to identify and evaluate real-world challenges.
2. Design and implement effective solutions while assessing their impact on projects and industry practices.
3. Use modern tools and demonstrate a continuous learning approach.
4. Apply ethical principles and project management skills and demonstrate the ability to work in teams for project development within the confines of a deadline.
5. Communicate technical information employing written reports and presentations.