INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KOTA

REVISED CURRICULUM

for

B.TECH. (COMPUTER SCIENCE AND ENGINEERING)

&

B.TECH. (CSE) with Specialization in “AI and MACHINE LEARNING”

w.e.f. 2021-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.Tech CSE 1st Year

1st Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST101	Computer Systems and Programming	3	0	0	3
2	ECT101	Digital Design	3	1	0	4
3	ECT103	Circuit Theory	3	1	0	4
4	HST101	Communication Skills	2	0	0	2
5	MAT101	Mathematics – I	3	1	0	4
		Labs				
6	CSP101	Computer Systems and Programming Lab	0	1	2	2
7	CSP111	IT Workshop – I	0	1	2	2
8	ECP101	Digital Design Lab	0	0	2	1
9	ECP111	System Simulation Techniques Lab	0	0	2	1
10	HSP101	Communication Skills Lab	0	0	2	1
11	OTP101	Upnayan-The Induction Programme	0	1	2	2
Total			**14**	**6**	**12**	**26**

2nd Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST102	Data Structures and Algorithms	3	0	0	3
2	ECT102	Electronic Devices and Circuits	3	1	0	4
3	ECT104	Fundamentals of Electrical Engineering	3	1	0	4
4	HST102	Technical Writing and Presentation Skills	2	0	0	2
5	MAT102	Mathematics – II	3	1	0	4
6	OTT102	Health, Safety and Environment	2	0	0	2
		Labs				
7	CSP102	Data Structures and Algorithms Lab	0	1	2	2
8	CSP112	IT Workshop – II	0	1	2	2
9	ECP102	Electronic Devices and Circuits Lab	0	0	2	1
10	ECP112	Circuit Design and Printing Lab	0	0	2	1
11	HSP102	Technical Writing and Presentation Skills Lab	0	0	2	1
Total			**16**	**4**	**11**	**26**

B.Tech CSE 2nd Year

3rd Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST201	Computer Architecture and Organization	3	0	0	3
2	CST203	Database Management Systems	3	0	0	3
3	CST205	Software Engineering	3	0	0	3
4	ECT211	Microprocessors and Microcontrollers	3	0	0	3
5	MAT201	Discrete Mathematical Structures	3	1	0	4
6	HST201	Engineering for Social Empowerment	3	0	0	3
		Labs				
7	CSP201	Computer Architecture and Organization Lab	0	0	2	1
8	CSP203	Database Management Systems Lab	0	0	2	1
9	ECP211	Microprocessors and Microcontrollers Lab	0	0	2	1
10	CSP211	IT Workshop – III	0	1	2	2
11	OTP201	Engineering Creativity, Innovation and Design	0	1	2	2
Total			**18**	**3**	**10**	**26**

4th Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST202	Object Oriented System Design	3	0	0	3
2	CST204	Design and Analysis of Algorithms	3	0	0	3
3	CST206	Operating Systems	3	0	0	3
4	CST208	Artificial Intelligence	3	0	0	3
5	ECT212	Communication Systems	3	0	0	3
6	MAT202	Probability and Statistics	3	0	0	3
		Labs				
7	CSP202	Object Oriented System Design Lab	0	0	2	1
8	CSP204	Design and Analysis of Algorithms Lab	0	1	2	2
9	CSP206	Operating Systems Lab	0	0	2	1
10	CSP210	Foundation of Data Science Lab	0	0	2	1
11	OTP202	Entrepreneurship and Business Incubation	0	1	2	2
Total			**18**	**1**	**11**	**25**

B.Tech CSE 3rd Year

5th Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST301	Theory of Computation	3	1	0	4
2	CST303	Computer Networks	3	0	0	3
3	CST305	Cryptography and Cyber Security	3	0	0	3
4	CST307	Machine Learning	3	0	0	3
5	CST309	Elective - 5.1	3	0	0	3
6	CST311	Elective - 5.2	3	0	0	3
7	CST3XX	Specialization Core Course (Optional)	3	0	0	3
		Labs				
8	CSP303	Computer Networks Lab	0	0	2	1
9	CSP307	Machine Learning Lab	0	0	2	1
10	CSP309	Elective - 5.1 Lab	0	0	2	1
11	CSD301	Survey of Technical Articles	0	2	2	3
Total			**18**	**3**	**8**	**25**

6th Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST302	Compiler Design	3	1	0	4
2	CST304	Elective - 6.1	3	0	0	3
3	CST306	Elective - 6.2	3	0	0	3
4	HST302	Professional Development	2	0	0	2
5	CST3XX	Specialization Core Course (Optional)	3	0	0	3
6	CST3XX	Specialization Core Course (Optional)	3	0	0	3
		Labs				
7	CSD302	Project – I	0	2	6	5
8	CSP302	Compiler Design Lab	0	0	2	1
9	CSP304	Elective - 6.1 Lab	0	0	2	1
10	HSP302	Professional Development Lab	0	0	2	1
11	CSP3XX	Specialization Core Course Lab (Optional)	0	0	2	2
Total			**11**	**3**	**12**	**20**

B.Tech CSE 4th Year

7th Semester On Campus with 8th On/Off Campus / 8th Semester On Campus with 7th Off Campus						
			Scheme			
S. No	**Subject Code**	**Subject**	**L**	**T**	**P**	**Credits**
1	CST401 / CST402	Elective - 7.1 / Elective - 8.1	3	0	0	3
2	CST403 / CST404	Elective - 7.2 / Elective - 8.2	3	0	0	3
3	OTT401 / OTT402	Open Elective – 7.1 / Open Elective – 8.1	3	0	0	3
4	HST401 / HST402	Living Ethics	2	0	0	2
5	CST4XX / CST4XX	Specialization Elective Course (Optional)	3	0	0	3
6	CST4XX / CST4XX	Specialization Elective Course (Optional)	3	0	0	3
		Labs				
4	CSP401 / CSP402	Elective - 7.1 Lab / Elective - 8.1 Lab	0	0	2	1
5	CSD401 / CSD402	Project - II	0	2	6	5
Total			**11**	**0**	**8**	**17**

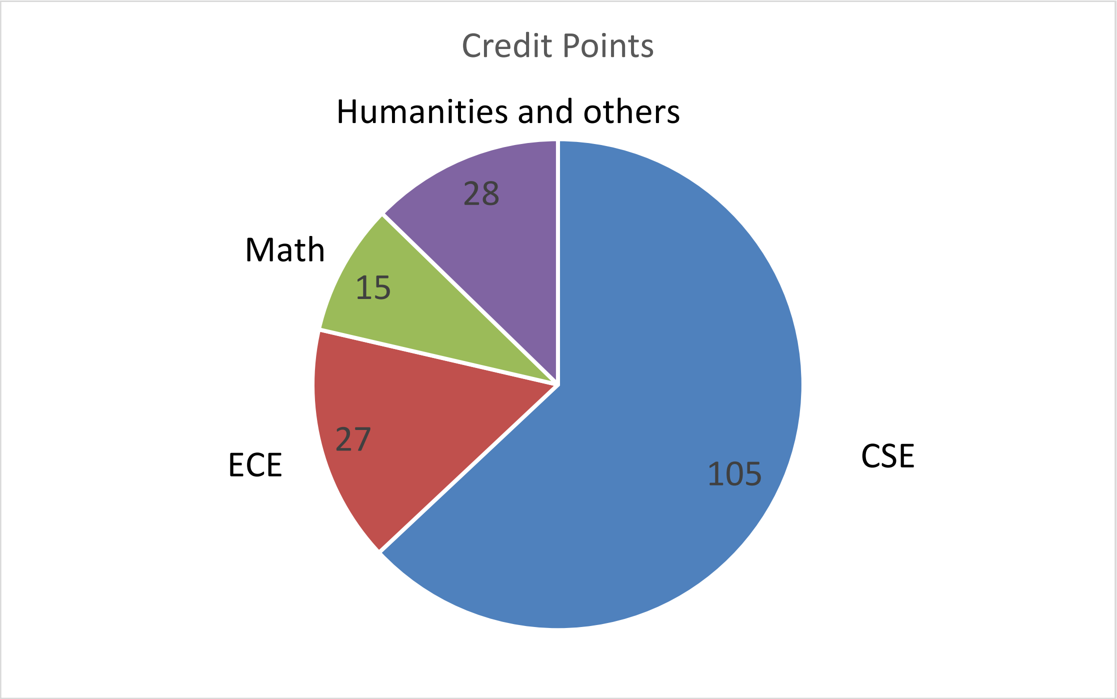
7th Semester Off Campus / 8th Semester Off Campus						
			Scheme			
S. No	**Subject Code**	**Subject**	**L**	**T**	**P**	**Credits**
1	OTD401	Internship	0	0	20	10
Total			**0**	**0**	**20**	**10**

8th Semester On Campus with 7th On Campus						
			Scheme			
S. No	**Subject Code**	**Subject**	**L**	**T**	**P**	**Credits**
1	CST402	Elective - 8.1	3	0	0	3
2	CST404	Elective - 8.2	3	0	0	3
3	OTT402	Open Elective - 8.1	3	0	0	3
		Labs				
5	CSP402	Elective - 8.1 Lab	0	0	2	1
Total			**9**	**0**	**2**	**10**

Note 1: Internship is only allowed either in 7th or 8th Sem.
Note 2: Student is eligible for Diploma in Computer Science and Engineering after successful completion of 3rd Year.

Total Credits: 175

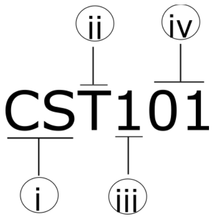
CSE Discipline = 105 (60%)
Other Discipline = 70 (40%)



Specialization in "AI and Machine Learning" Courses (Additional 15 Credits)

- Students with no backlog and CGPA greater than 7 at the end of 4th / 5th / 6th / 7th Sem are eligible to register Specialization courses in 5th / 6th / 7th / 8th Sem.
- B. Tech level Three Core Courses (9 Credits) and Two Elective Courses (6 Credits) of the Specialization shall be offered from 5th Sem onwards for B. Tech with Specialization.
- Students can reconvert from B. Tech with Specialization to B. Tech. In this case, B. Tech with Specialization Course Credits earned will be accounted in terms of Audit Courses.

Note: Course Code nomenclature



i. First and Second characters together represents the specific department offering the course: CS for Computer Science and Engineering (CSE), EC for Electronics and Communication Engineering (ECE), MA for Mathematics (MAT), HS for Humanities and Social Sciences (HSS), OT for Other than those mentioned (OTH)

ii. Third character represents the mode of conduction of course: T for Theory Course, P for Practical Course, D for Project / Study / Training and Internship Course

iii. Fourth character represents the level of the course: 1 for 1st Year, 2 for 2nd Year, 3 for 3rd Year, 4 for 4th Year

iv. Fifth and Sixth characters together represent semester and nature of the course:

   a. Numbers are odd for Odd Semesters (1st / 3rd / 5th / 7th) and even for Even Semesters (2nd / 4th / 6th / 8th)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KOTA

REVISED SYLLABI

for

B.TECH. (COMPUTER SCIENCE AND ENGINEERING)

&

B.TECH. (CSE) with Specialization in “AI and MACHINE LEARNING”

w.e.f. 2021-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.Tech CSE 1st Year

1st Semester						
S. No	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	CST101	Computer Systems and Programming	3	0	0	3
2	ECT101	Digital Design	3	1	0	4
3	ECT103	Circuit Theory	3	1	0	4
4	HST101	Communication Skills	2	0	0	2
5	MAT101	Mathematics – I	3	1	0	4
		Labs				
6	CSP101	Computer Systems and Programming Lab	0	1	2	2
7	CSP111	IT Workshop – I	0	1	2	2
8	ECP101	Digital Design Lab	0	0	2	1
9	ECP111	System Simulation Techniques Lab	0	0	2	1
10	HSP101	Communication Skills Lab	0	0	2	1
11	OTP101	Upnayan-The Induction Programme	0	1	2	2
Total			**14**	**6**	**12**	**26**

Course Code : CST101	**Course Credit : 3**
Course Name : Computer Systems and Programming	**L-T-P : 3-0-0**

Course Prerequisite: Nil

Course Syllabus:

Basics: C language introduction, C language Standards, System Software, Application Software. Compiler - Compilation process - Compiler and interpreter. **[6 Lectures]**

Data Types and Storage Classes: Different data types, Storage Classes – auto, static, extern, register. Macro & Preprocessor in C. Operator Precedence and Associativity. Control Statements: If-else condition, If-else if Ladder, Switch case, Loop – for, while, do while. Nested loop, break, continue, exit, goto and problem with goto. **[10 Lectures]**

Functions: Passing arguments in main() function, Call by value, Call by reference. Array & Strings: Introduction to Array, Number type array, Character type array (String), Multi-dimensional array, Operations on strings (User defined functions for strlen, strcpy, strcmp, strrev, etc.), gets(), puts(), getc(), getch(), getchar(), putc(), putch(), putchar() functions. **[10 Lectures]**

Pointers: Introduction to pointer, Double pointer. Pointer to int, Pointer to char, Pointer to function, Function to pointer, Pointer to array, Pointer to structure, Array of pointers.Static & Dynamic Memory Allocation: malloc(), calloc(), realloc() and free() functions. **[8 Lectures]**

Structure and Union: Structure in C, Union in C, Enum operator. File Handling: Basics of working with text files, File read, write, append and other similar operations, EOF and feof() functions, File pointer, fopen(), fgetc() and fgets() functions, fputc and fprintf() functions. **[6 Lectures]**

Text/Reference Books:

1. The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Latest Edition, Prentice Hall.
2. Programming in ANSI C, E. Balagurusamy, Latest Edition, McGraw Hill
3. Let us C, Yashavant Kanetkar, Latest Edition, BPB Publication

Course Outcome (CO):

CO1: To be able to understand and operate Linux the operating system.

CO2: Basic understanding of the compiler, interpreter, assembler, and library functions.

CO3: Develop the ability to implement the fundamental knowledge of mathematics and science in computer programming.

CO4: Design the flowchart of the solution and develop the computer program to solve real-life problems.

CO5: Develop the ability to analyze the problem, develop an algorithm and finally implement using the C programming language.

Course Code : ECT101	**Course Credit : 4**
Course Name : Digital Design	**L-T-P : 3-1-0**

Course Prerequisite: Nil

Course Syllabus:

Number base conversion (binary, octal, decimal, hexadecimal), Binary codes (weighted, unweighted, self-complementary), Signed and unsigned binary numbers, complements (1's, 2's, 9's, 10's), Binary arithmetic (addition, subtraction, multiplication, division), Binary logic (positive and negative logic) **[8 Lectures]**

Boolean algebra (basic theorems and properties, truth tables, DeMorgan's theorem, duality, operator precedence), Boolean function (canonical and standard forms), Digital logic gates, Boolean function simplification (2 to 4 variable Karnaugh maps, don't care conditions, Quine-McCluskey method), NAND and NOR implementation. **[9 Lectures]**

Analysis and design of combinational logic circuits (code conversion, error detector, binary adder and subtractor, look-ahead carry and BCD adders, binary magnitude comparator, decoder, encoder, priority encoder, multiplexer, demultiplexer), Programmable logic devices (design using read only memory, and programmable logic arrays). **[9 Lectures]**

Level and edge-triggered flip-flops (RS flip-flop, D flip-flop, JK flip-flop, T flip-flop, timing specifications of flip-flops, characteristic table and equation of flip-flops, excitation table of flip-flops). **[7 Lectures]**

Analysis of clocked sequential circuits (state table, state diagram, state reduction and assignment), Design of synchronous and asynchronous counters, Shift registers and its timing considerations. **[7 Lectures]**

Text/Reference Books:

1. M. Morris Mano, Michael D. Ciletti, "Digital Design", Prentice Hall, 4th Edition
2. R.P. Jain, "Modern Digital Electronics", Tata McGraw Hill, 3rd Edition
3. Albert Paul Malvino, Donald P. Leach, "Digital Principles and Applications", Tata McGraw Hill, 6th Edition
4. John F. Wakerly, "Digital Design: Principles and Practices", Pearson Education, 4th Edition

Course Outcome (CO):

CO1: Represent and convert decimal numbers in various other number systems.

CO2: Use Boolean algebra to construct, minimize and implement real time problems in digital system design.

CO3: Implement, analyze, optimize and debug design based on various logic gates.

CO4: Design and analyze circuits for digital arithmetic. To describe the operation and timing constraints for latches, Flip-flops and registers etc.

Course Code : ECT103	**Course Credit : 4**
Course Name : Circuit Theory	**L-T-P : 3-1-0**

Course Prerequisite: Nil

Course Syllabus:

Introduction to electric circuits and networks, Circuit variables (charge, current, voltage, electric field, power), Circuit elements (independent and dependent voltage and current sources, resistors, capacitors, inductors), Kirchhoff's laws (KCL and KVL), Ohm's law, Series and parallel circuits, Linear circuit analysis techniques (nodal analysis, mesh analysis). **[8 Lectures]**

Network topology and graphs, Network theorems (Superposition, Thevenin's, Norton's, Maximum Power Transfer, Millman's, Tellegen's, Reciprocity, Compensation), Circuit transformation (Wye - Delta), Source transformations, Duality principle. **[10 Lectures]**

Time-domain transient analysis (natural and forced) of first-order and second-order circuits. **[8 Lectures]**

Phasor-domain or frequency-domain steady-state analysis, AC power, Polyphase circuits, Three-phase loads, Frequency response, Basic filters, Resonance, Quality factor and bandwidth. **[8 Lectures]**

Two Port Parameters (Impedance, Admittance, Transmission, Hybrid), Relationships between parameters, Interconnection of two port networks, Symmetrical two port network. **[6 Lectures]**

Text/Reference Books:

1. Charles K. Alexander, Mathew N.O. Sadiku, "Fundamentals of Electric Circuits", McGraw Hill, 3rd Edition
2. William H. Hayt, Jack Kemmerley and Steven M. Durbin, "Engineering Circuit Analysis", 8th Ed., Tata McGraw-Hill, 2012
3. M. E. Van Valkenburg, "Network Analysis", 3rd Ed., Prentice Hall of India, 2003
4. Leonard S. Bobrow, Navneet Gupta, "Foundations of Electrical Engineering", Oxford University Press, Asian Edition
5. Allan R. Hambley, "Electrical Engineering Principles and Applications", Prentice Hall, 5th Edition

Course Outcome (CO):

CO1: Learn the essentials of electrical circuits and networks.
CO2: Learn the linear circuit analysis techniques and network theorems.
CO3: Learn the time-domain response for RLC circuits.
CO4: Learn the frequency-domain response of RLC circuits.
CO5: Learn the two port networks and its parameters.

Course Code : HST101	**Course Credit : 2**
Course Name : Communication Skills	**L-T-P : 2-0-0**

Course Prerequisite: Nil

Course Syllabus:

Basic Grammar: Sentence Construction and Types; Simple, Complex and Compound sentences; Tenses; Agreement of Subject and Verb; Conditional Sentences; Direct and Indirect Narration; Active and Passive Voice; Error Spotting; Question tags and short responses. **[9 Lectures]**

Vocabulary and Usage: Word Formation (by adding suffixes and prefixes), Confusing Word Pairs; Homophones, and Homonyms; One Word Substitution; Phrasal Verbs; Punctuation. **[4 Lectures]**

Writing Skills: Precis writing; Note-making; Expressing ideas within a restricted word limit; Email writing; Reading Comprehension. **[5 Lectures]**

Texts for Appreciation and Analysis:
Animal Farm (1945) by George Orwell. Penguin India, 2011.(ISBN: 9781502492791) and
Selected chapters from the prescribed textbook: Insights: A Course in English Literature
and Language (2009) by K. Elango, Orient Blackswan Publishers:
‘The Diary of a Young Girl’
‘Wings of Fire’
‘Our Urgent Need for Self-esteem’ **[9 Lectures]**

Text/Reference Books:

1. Murphy, Raymond. English Grammar in Use, Cambridge UP. 2012.
2. Stuart Redman, English Vocabulary in Use: Pre-Intermediate and Intermediate, Cambridge UP, 2012.
3. Barker, Alan. Improve Your Communication Skills: How to Build Trust, Be Heard and Communicate with Confidence. Kogan Page, 2019.
4. Swan, Michael. Practical English Usage, Oxford UP, 2017.
5. Barnet, Sylvan, &amp; William E. Cain. A Short Guide to Writing about Literature. Longman, 2005.
6. O&#39;Brien, Terry. Modern Writing Skills, Rupa, 2011.

Course Outcome (CO):

CO1: Understand the essential rules of syntax in the English language.
CO2: Learn the techniques to expand the knowledge of vocabulary.
CO3: Learn to use appropriate idiomatic expressions in speech and writing.
CO4: Learn the techniques for effective written communication.
CO5: Learn to develop the skills of comprehending and analyzing a written work.

Course Code : MAT101	**Course Credit : 4**
Course Name : Mathematics - I	**L-T-P : 3-1-0**

Course Prerequisite: Nil

Course Syllabus:

Differential Calculus: Asymptotes, curve tracing (Cartesian, parametric and five polar curves-Folium of Descartes, Limacon, Cardioids, Lemniscuses of Bernoulli and Equiangular spiral and other simple polar curves). Partial differentiation, Euler's theorem on homogeneous functions, total differentiation, approximate calculation. **[8 Lectures]**

Integral Calculus – Improper integrals, Area and length of curves, Surface area and volume of solid of revolution. Multiple integrals, Change of order of integration. **[6 Lectures]**

Differential Equations – Differential equations of first order and first degree - linear form, reducible to linear form, exact form, reducible to exact form. Linear differential equations of higher order with constant coefficients. Second order ordinary differential equations with variables coefficients –Homogeneous, exact form, reducible to exact form, change of dependent variable (normal form), change of independent variable, method of variation of parameters. **[9 Lectures]**

Matrices – Rank and inverse of matrix by elementary transformations, Consistency of linear system of equations and their solution. Eigenvalues and eigenvectors. Cayley-Hamilton theorem (statement only) & its applications. **[8 Lectures]**

Numerical Analysis- Finite differences, interpolations and numerical differentiations – Forward, Backward, Central differences and relations between them, Newton's forward, backward interpolation formulas and Stirling's central difference interpolation formulas. Lagrange's interpolation formula, Numerical differentiations using Newton's forward, backward, Stirling's central difference interpolation formulas. Numerical integrations - Trapezoidal rule, Simpson's one-third rule, Simpson's 3/8 rule. **[9 Lectures]**

Text/Reference Books:

1. R.K. Jain, S.R.K. Iyengar, "Advanced Engineering Mathematics", Narosa
2. Srimanta Pal and Subodh C. Bhunia, "Engineering Mathematics", Oxford
3. Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley India
4. D. W. Jordan, P. Smith, "Mathematical Techniques", Oxford
5. Peter V. O'Neil, "Advanced Engineering Mathematics", Cengage Learning, New Delhi
6. B.V. Ramana, "Higher Engineering Mathematics", McGraw–Hill

Course Outcome (CO):

CO1: Understand Differential equations and its applicability in different engineering fields.

CO2: Incorporate the knowledge of calculus to support their concurrent and subsequent engineering studies.

CO3: Have the idea of matrices, its physical interpretation and applications in real life examples.

CO4: To develop mathematical skills so that students are able to apply mathematical methods & principles in solving problems from engineering fields.

Course Code : CSP101	**Course Credit : 2**
Course Name : Computer Systems and Programming Lab	**L-T-P : 0-1-2**

Course Prerequisite: Nil

Course Syllabus:
First program in C, Variable Declaration and Initialization, Scope of a variable, Use of Constant, Use of Escape sequences, Use of printf() and scanf() functions, Different data types, Use of static, extern, Use of Macro, Use of Logical and Relational operators, Operator Precedence and Associativity, Evaluation order, Post-increment and Pre-increment, sizeof operator, If-else condition, If-else if Ladder, Switch case, Loop – for, while, do while. Nested loop, break, continue, exit. **[3 Labs]**
User defined functions, Function prototype, Argument passing, return type, Passing arguments in main() function, Evaluation order of arguments, Return multiple values from a function, Number type array, Character type array (String), Multi-dimensional array. **[3 Labs]**
Operations on strings (User defined functions for strlen, strcpy, strcmp, strrev, etc.), gets(), puts(), getc(), getch(), getchar(), putc(), putch(), putchar() functions, Call by value, Call by reference. **[2 Labs]**
Null, void pointers, Double pointer, Pointer to int, Pointer to char, Pointer to function, Function to pointer, Pointer to array, Pointer to structure, Array of pointers. **[2 Labs]**
Structure in C, Different operations on struct variables, Enum operator, malloc(), calloc(), realloc() and free() functions, Basics of working with text files, File read, write, append and other similar operations, EOF and feof() functions, File pointer, fopen(), fgetc() and fgets() functions, fputc and fprintf() functions. **[2 Labs]**

Text/Reference Books:
1. The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Latest Edition, Prentice Hall.
2. Programming in ANSI C, E. Balagurusamy, Latest Edition, McGraw Hill
3. Let us C, YashavantKanetkar, Latest Edition, BPB Publication

Course Outcome (CO):
CO1: To understand and use variables, data types and functions to implement various algorithms.
CO2: To handle loop execution, if-else conditions, array and preprocessing directives.
CO3: To understand the use and implementation of arrays, structures and unions as user defined datatypes.
CO4: To handle pointer variables, static and dynamic memory allocation, array of pointers and other uses of pointers.

Course Code : CSP111	**Course Credit : 2**
Course Name : IT Workshop - I	**L-T-P : 0-1-2**

Course Prerequisite: Nil

Course Syllabus:

<u>Part 1: COMPUTER ANIMATION:</u> [6 Labs]

Software Used: Blender

Animation Basics – Timeline, Frame rate, Shots and Scenes, Keyframes, In-between frames, Timing, Spacing, etc
Animation Principles – Squash and Stretch, Anticipation, Overlapping actions, etc
Type of Animation – Difference among 3D animation, Stop Motion Animation, Character Animation.
Animation Process – Story and Script, Concept Art, Animatic, Creating Assets, Animation, Texturing and Rendering, Editing, Add Music and Sound.
Project: Making an animated movie / video game / any other project

<u>Part 2: ROBOTICS:</u> [6 Labs]

Introduction to Robots and Autonomous Systems
Getting started with Arduino Uno
Installation of Arduino IDE for coding and sample socket programming
Write the first LED on-off program and transfer it to the Arduino Uno
Connecting Ultrasonic sensor and other sensors
Practice with L298N motor controller and DC motors
Embedding and data transfer using Bluetooth module
Lecture and small demonstration on advanced applications of IoT

Text/Reference Books:

1. Oliver Villar, “Learning Blender: A Hands-On Guide to Creating 3D Animated Characters”, Addison-Wesley
2. Gordon Fisher, “Blender 3D Basics”, Packt Publishing Limited
3. Mark Geddes, Arduino Project Handbook, No Starch Press
4. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill Education, Latest Edition
5. Andy Beane, “3D Animation Essentials”, John Wiley
6. Isaac Kerlow, “The Art of 3D Computer Animation and Effects”, Latest Edition, Wiley
7. Simon Monk, 30 Arduino Projects for Evil Genius, McGraw-Hill Education, Latest Edition

Course Outcome (CO):

CO1: To understand the importance and industrial scope of computer animation and gaming
CO2: To understand the principles of animation and its vocabulary
CO3: To design and develop an animated film step-by-step
CO4: To understand the microcontrollers, analog and digital sensors and motors.
CO5: To connect sensors and actuators with the microcontroller and power supply system.
CO6: Mathematical simple dynamic modelling and programming to make a working robotic prototype to develop a simple robot control system, environment perception, planning and action.

Course Code : ECP101	**Course Credit : 1**
Course Name : Digital Design Lab	**L-T-P : 0-0-2**

Course Prerequisite: Nil

Course Syllabus:

Verification of truth table for various logic gates using TTL ICs and implementation of basic gates universal NAND and NOR gates.
Design of four bit Binary to Gray and Gray to Binary code Converter.
Design of Half and Full Adder and Subtractor circuits.
Design of Two-bit multiplier.
Design of One- and Two-bit Comparators.
Design of Even and Odd parity generator and checker.
Design of 2:1 and 4:1 MUX using basic gates, and design of 4:1 MUX using 2:1 MUX.
Design a binary to decimal and octal to decimal decoder.
Design and verification truth table of flip-flops (SR latch with NOR and NAND Gates, SR flip-flop with control input using NOR and NAND Gates).
Design and verification truth table of flip-flops (D, JK and T).
Design and implement binary ripple and synchronous up/down counters using flip-flops.
Design and implement shift registers using flip-flops.

Text/Reference Books:

1. M. Morris Mano, Michael D. Ciletti, “Digital Design”, Prentice Hall, 4th Edition
2. R.P. Jain, “Modern Digital Electronics”, Tata McGraw Hill, 3rd Edition

Course Outcome (CO):

CO1: Design and verify the truth table of various logic gates.
CO2: Design and analyze the universal gates using basic gates.
CO3: Design and analyses of different combinational circuits.
CO4: Design and analyses of different sequential circuits.

Course Code : ECP111	**Course Credit : 1**
Course Name : System Simulation Techniques Lab	**L-T-P : 0-0-2**

Course Prerequisite: Nil

Course Syllabus: [12 Labs]

Brief Introduction Installation of MATLAB History Use of MATLAB Key features.

Introduction to MATLAB Software MATLAB window Command window Workspace Command history Setting directory Working with the MATLAB user interface Basic commands Assigning variables Operations with variables.

Data files and Data types-Character and string Arrays and vectors Column vectors Row vectors.

Basic Mathematics-Operations on matrix, Mathematical and logical operators, Creating rows and columns Matrix Matrix operations Finding transpose, determinant and inverse Solving matrix.

M files- Working with script tools Writing Script file Executing script files The MATLAB Editor Saving m files.

Plots- 2D plots, Basic Plotting Functions Creating a Plot Plotting Multiple Data Sets in One Graph Specifying Line Styles and Colors Graphing Imaginary and Complex Data Figure Windows Displaying Multiple Plots in One Figure Controlling the Axes.

3D plots- Creating Mesh and Surface About Mesh and Surface Visualizing Subplots.

MATLAB Simulink- Introduction Of Simulink Simulink Environment & Interface Study of Library Circuit Oriented Design Equation Oriented Design Model Subsystem Design Connect Call back to subsystem Application.

MATLAB Programming -Loops and Conditional Statements, Control Flow Conditional Control — if, else, switch Loop Control — for, while, continue, break Program Termination — return.

Functions- Writing user defined functions Built in Function Function calling Return Value Types of Functions Global Variables.

Text/Reference Books:

1. S.N. Alam and S.S. Alam, Understanding MATLAB: A Textbook for Beginners, I K International Publishing House Pvt. Ltd, Latest Edition.
2. Stormy Attaway, MATLAB: A Practical Introduction to Programming and Problem Solving, Butterworth-Heinemann, Latest Edition
3. Holly Moore, MATLAB for Engineers, Pearson, Latest Edition
4. MATLAB Primer by MathWorks, Latest Edition

Course Outcome (CO):

CO1: To install and use MATLAB functions to analyze and visualize data.

CO2: To apply basic computer numeric techniques and simulations to solve engineering problems.

CO3: To write programs to generate 2D and 3D plots and user interface graphics in MATLAB.

CO4: To learn MATLAB programming

Course Code : HSP101	**Course Credit : 1**
Course Name : Communication Skills Lab	**L-T-P : 0-0-2**

Course Prerequisite: Nil

Course Syllabus: [12 Labs]
Active Listening, Interactive Vocabulary building, Grammar Practice
Extempore Speaking, Group discussions, Interaction on Topics Of Social & General awareness,Turncoat Debates, Grammar practice
Story Telling, Screening Select episodes/Clips from Movies/Series, Viewing Skills (Writing Activities using silent videos), Grammar practice
Jigsaw reading, Drills & training on the combined skills of Vocal, Written, Visual, & Non-verbal Communication, Grammar practice

Text/Reference Books:
1. Daniel Jones, “Cambridge English Pronouncing Dictionary”, Cambridge, ELBS Cambridge
2. J. Sethi, P.V. Dhamija, “A Course in Phonetics and Spoken English”, PHI Learning
3. Matthew McKay, Martha Davis, Patrick Fanning, “Messages: The Communication Skills Book”, New Harbinger Publications, 3rd Edition
4. Barun K. Mitra, “Personality Development and Soft Skills”, Oxford University Press

Course Outcome (CO):
CO1: Effective Communication as a Must-have skill
CO2: Receiving Information Successfully.
CO3: Transmitting Information Successfully, Effectively, & Constructively.

Course Code : OTP101	**Course Credit : 2**
Course Name: Upnayan-The Induction Programme	**L-T-P : 0-1-2**
Course Prerequisite: Nil	

Course Syllabus: Session 90 min each

Day 1:

Inaugural Session: Welcome Note, Introduction to the institute, Introduction to Leadership
Session : Role of Effective studentship for a better life ahead
Introduction to the curriculum, evaluation metrics, time table and annual calendar
Ice Breaking: Knowing Each Other

Day 2:

Recap of Day 1
Understanding and Managing Change and Transition -1
Understanding and Managing Change and Transition -2
Introduction to the functioning of institution: Committees, Clubs, Events, Activities, Student Support Services
Ragging, Regulations (class rules, discipline, ragging etc
Campus tour

Day 3:

Recap
Expand your learning styles
Study Skills -1(Introduction, Self-Evaluation, Attention Management)
Study Skills -2 (Reading, Note Making, Comprehension)
Study Skills-3 (Memory, Time Management, Test Taking Skills)

Day 4:

Recap of Day 3
Enhancing 21st Century Skills-1(Introduction and Relevance)
Enhancing 21st Century Skills-2 : Imagination and Creativity
Enhancing 21st Century Skills-3: Digital Literacy Skills
Enhancing 21st Century Skills-4: Leadership and Team Culture

Day 5:

Getting ready for career -1
Getting ready for career -2
Evaluation and Feedback: Directions for improvement
Wrap up session

- Sharing of Experience
- An inspirational connect with an influencer
- Vote of Thanks

B.Tech CSE 1st Year

2nd Semester						
S. No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST102	Data Structures and Algorithms	3	0	0	3
2	ECT102	Electronic Devices and Circuits	3	1	0	4
3	ECT104	Fundamentals of Electrical Engineering	3	1	0	4
4	HST102	Technical Writing and Presentation Skills	2	0	0	2
5	MAT102	Mathematics – II	3	1	0	4
6	OTT102	Health, Safety and Environment	2	0	0	2
		Labs				
7	CSP102	Data Structures and Algorithms Lab	0	1	2	2
8	CSP112	IT Workshop – II	0	1	2	2
9	ECP102	Electronic Devices and Circuits Lab	0	0	2	1
10	ECP112	Circuit Design and Printing Lab	0	0	2	1
11	HSP102	Technical Writing and Presentation Skills Lab	0	0	2	1
Total			**16**	**4**	**11**	**26**

Course Code: CST102	**Course Credit: 3**
Course Name: Data Structures and Algorithms	**L-T-P: 3-0-0**

Course Prerequisite: Basic programming in C language

Course Syllabus:
Introduction: Concept of Data Structures, Algorithms and ADT (Abstract Data Type), Program v/s algorithms, Execution time and storage space, Complexity - time and space, Asymptotic notations: $O(n)$, $\Omega(n)$ $Q(n)$. [6 lectures]
Array: Array as storage element, computing address in n-dimensional array. Insertion and Deletion, Searching (Sequential and binary), Sorting (Bubble sort, Insertion, Selection, Merge sort, Quick sort, radix sort), Representation of polynomial and its applications, Representation of Sparse matrix and its applications. Linked lists: Single and double linked lists, Insertion/deletion/searching in linked lists, Comparison of arrays and linked lists, Implementation of circular lists. [9 lectures]
Stack and Queue: Stack, Queue, Circular queue, Concept of overflow and underflow, Concept of precedence and associativity in expressions, Resolving precedence of operators and association of operands, Evaluation of Expression: Infix, Prefix & Postfix notations, conversion of expression from one form to other form, Recursion: concepts, use and implementation. Strings, Hash tables (open and close), Dictionary, Sets. [10 lectures]
Trees: Concept of Trees, Binary and Multiway tree, Representing multiway tree as Binary tree, Tree Traversal, constructing Binary tree from Traversal, BST (Binary Search Tree), threaded and unthreaded BST as data structure, Insertion/Deletion/Search in BST, Heap Tree and Heap sort, Introduction to height balanced tree. [9 lectures]
Graphs: Introduction to graphs (directed and undirected), representation of graphs using adjacency matrix and list, Graph Traversals: DFS and BFS, Topological sorting. [6 lectures]

Text/Reference Books:
1. Ellis Horowitz, SartajSahni, Fundamentals of Data Structures, Computer Science Press, Latest Ed.
2. Robert Kruse, et al. Data Structures and Program Design in C, Pearson, Latest Edition.
3. Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, Data Structures and Algorithms, Addison Wesley, Latest Edition.
4. Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein, Data Structures Using C, PHI.

Course Outcome (CO):
CO1: To understand the basic data structures and analyze them to use in different problems.
CO2: To understand the linear and nonlinear search data structures and their implementation.
CO3: Select the appropriate data structures and analyze time and space complexities.
CO4: To derive the mathematical details to compute the complexity asymptotically.
CO5: Identify different parameters to analyze and implement various types of data structures and design algorithms for solving real world problems.

Course Code: ECT102	**Course Credit: 4**
Course Name: Electronic Devices and Circuits	**L-T-P:** 3-1-0

Course Prerequisite: NIL

Course Syllabus:
Types of materials, Characteristics of intrinsic and extrinsic semiconductors, Junction diode and its characteristics, Ideal diode and its applications (half-wave and full-wave rectifiers in voltage regulators, positive and negative clippers, positive and negative clampers), Non-ideal diode models, Zener diodes and its applications (clipper, voltage regulator), Diode capacitance and switching times, Types of diodes (LED, Varactor diode, Schottky diode, Photodiode) [8 lectures]
Bipolar Junction Transistor (BJT types, operation, configurations, characteristics), Cutoff and saturation operations, BJT switching times. [8 lectures]
Field Effect Transistor (FET types, operation, configurations, characteristics), Metal-Oxide Semiconductor FET (MOSFET types, operation, configurations, characteristics), Complimentary MOSFET (CMOS). [8 lectures]
BJT biasing and small-signal analysis of BJT amplifiers, FET biasing and small-signal analysis of FET amplifiers, Frequency response (low-frequency and high-frequency responses of amplifiers), Large-signal power amplifiers (class A, class B, class AB). [8 lectures]
Feedback (concept of negative and positive feedback, characteristics of negative feedback amplifiers, negative feedback amplifiers topologies, sinusoidal oscillators). [8 lectures]

Text/Reference Books:
1. Robert Boylestad, Louis Nashelsky, “Electronic Devices and Circuit Theory”, Prentice Hall, 7th Edition
2. Jacob Millman, Christos C. Halkias, “Integrated Electronics: Analog and Digital Circuits and Systems”, Tata McGraw Hill
3. Adel S. Sedra, Kenneth C. Smith, “Microelectronic Circuits”, Oxford University Press, 5th Edition
4. Leonard S. Bobrow, Navneet Gupta, “Foundations of Electrical Engineering”, Oxford University Press, Asian Edition
5. Donald A. Neamen, “Microelectronics: Circuit Analysis and Design”, McGraw Hill, 4th Edition

Course Outcome (CO):
CO1: Learn the essentials of semiconductor materials and devices.
CO2: Learn the operation of BJT and FET.
CO3: Learn the design and frequency-domain analysis of amplifiers using BJT and FET.
CO4: Learn the basic power amplifiers.
CO5: Learn the concept of feedback and their circuit applications.

Course Code: ECT104	**Course Credit: 4**
Course Name: Fundamental of Electrical Engineering	**L-T-P:** 3-1-0
Course Prerequisite: NIL	

Course Syllabus:

Brief history of electrical engineering, Sources of Electrical Energy, Generation and Transmission of Electrical Energy. [6 lectures]

Measurement of electrical quantities (current, voltage, power, energy, resistance, inductance, capacitance, frequency). [8 lectures]

Magnetic circuits, Ideal transformer, Non-ideal transformer parameters determination [8 lectures]

DC machines (DC generator and motor), AC machines (synchronous and induction generators and motors). [10 lectures]

Realization of Digital Logic Gates (AND, OR, NOT, NAND, NOR, Latch, Flip-flops) using Semiconductor Devices (Diode / BJT / FET), Operation of Digital Logic Families (RTL, DTL, TTL, ECL, MOS, CMOS), Parameters of Digital Logic Families (Fan-out, Power dissipation, Propagation delay, Noise margin). [10 lectures]

Text/Reference Books:

1. Charles A. Gross, Thaddeus A. Roppel, “Fundamentals of Electrical Engineering”, CRC Press
2. A.K. Sawhney, “Electrical & Electronics Measurement and Instrumentation”, DhanpatRai& Co
3. M. Morris Mano, Michael D. Ciletti, “Digital Design”, Prentice Hall, 4th Edition
4. Edward Hughes, “Electrical & Electronic Technology”, Pearson Education, 10th Edition
5. Vincent Del Toro, “Electrical Engineering Fundamentals”, Prentice Hall India, 2nd Edition

Course Outcome (CO):

CO1: Learn the essentials of electrical energy.
CO2: Learn the measurement of electrical quantities.
CO3: Learn the principle of operation of transformer.
CO4: Learn the principle of operation of DC and AC motors and generators.
CO5: Learn the realization of digital logic gates and families.

Course Code: HST102	**Course Credit: 2**
Course Name: Technical Writing and Presentation Skills	**L-T-P:** 2-0-0
Course Prerequisite: NIL	
Course Syllabus: Communication Strategy, Data Visualization and Delivery, Communication Across Cultures [6 lectures] Communication to Build Brands/Values/Promise (Slogan Writing, Demos, Sales Pitch etc.), Communication in Crisis (Negotiation, Brainstorming for Deadlines, Precision in Extreme Situations) [5 lectures] Communication in different conversations (Emails, Meetings, Interviews, Presentations, Networking), Style, Tone & Voice [5 lectures] Types of Presentations (Formal, Informal, Speeches, Demos, etc.), Preparation, Writing, Method, and delivery, Tailoring Information to Suit the Audience) [6 lectures] Writing to Create Quality Documents: 7Cs of Communication; Structured Writing: Paragraph Expansion, Essay, Presentation; Style, Coherence, Emphasis. [6 lectures]	
Text/Reference Books: 1.Handbook of Technical Writing: Charles T Brusaw, Gerald J Alred& Walter E Oliu, St. Martin's Press, New York. 2.Technical Writing 101: Alan S Pringle & Sarah S O'Keefe, Scriptorium Publishing Services Inc 3. Every Page is Page One: Mark Baker XLM Press 4. How to Talk to Anyone: Leil Lowndes, McGraw Hill 5. Talk Like Ted: Carmine Gallo, Pan Macmillan	
Course Outcome (CO): CO1: Confidence Building. CO2: Effective Participation CO3: Developing Skills for Digital Communication CO4: Developing Critical, Independent, and Creative Thinking	

Course Code: MAT102	**Course Credit: 4**
Course Name: Mathematics – II	**L-T-P:** 3-1-0

Course Prerequisite: NIL

Course Syllabus:

Vector Calculus – Differentiation and integration of vector functions of scalar variables, Scalar and vector fields, Gradient, Directional derivative, Divergence, Curl. Line integral, Surface integral and Volume integral. Green's, Gauss's and Stokes's theorems (statement only) and their simple applications. [8 lectures]

Fourier series- full range and half range series, change of intervals, Harmonic analysis. [6 lectures]

Partial Differential Equation – Formulation and classification of PDE; Linear partial differential equation of the first order (Lagrange's method) Non-linear PDE of the first order. Four standard forms, Charpit's method. [8 lectures]

Integral Transforms – Laplace Transform and Convergence, Properties of Laplace Transform, Inverse Laplace Transform, Fourier Transform, Inverse Fourier Transform, Laplace Transform and Fourier Transform. [6 lectures]

Complex Variable – Limit, Continuity and Differentiability of complex function, Analytic functions, Cauchy-Riemann Equations, Necessary and Sufficient condition for analyticity, Properties of Analytic functions and their Engineering Applications. Complex Integration: Line Integral (contour integral) and its properties, Cauchy's integral theorem, Cauchy Integral Formula, Taylor's series and Laurent's series, Applications of Contour Integration –Residue theorem, calculation of residues, Evaluation of various types of definite real integrals using contour. [12 lectures]

Text/Reference Books:

1. R.K. Jain, S.R.K. Iyengar, "Advanced Engineering Mathematics", Narosa
2. Srimanta Pal and Subodh C. Bhunia, "Engineering Mathematics", Oxford
3. Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley India
4. R.V. Hogg, J.W. McKean, A. Craig, "Introduction to Mathematical Statistics", Pearson Education India, 6th Edition
5. N.P. Bali, Manish Goyal, "A textBook of Engineering Mathematics", Laxmi Publications
6. J. Ravichandran, "Probability and Statistics for Engineers", Wiley India, 2010

Course Outcome (CO):

CO1: Understand Integral transforms and its applicability in different engineering fields.

CO2: Incorporate the knowledge of calculus to support their concurrent and subsequent engineering studies.

CO3: To develop mathematical skills so that students are able to apply mathematical methods & principles in solving problems from Engineering fields.

CO4: To make aware students about the importance and symbiosis between Mathematics and Engineering.

Course Code: OTT102	**Course Credit: 2**
Course Name: Health, Safety and Environment	**L-T-P:** 2-0-0

Course Prerequisite: NIL

Course Syllabus:

Health and Safety:

Health and Safety Foundations, Key Elements of Health and Safety Policy, History of Occupational Safety and Health, Organizing and Promoting a Positive Health and Safety Culture, Risk Assessment and Principles of Control, Monitoring as well as Review and Audit, Hazards Control and Personal Protective Equipment at work places (Hazards related to Food, Body Posture, Fire, Electrical, Office, Construction, Chemical, Mining, Oil and Gas), International and National Aspects of Health and Safety, Mental Health and Physical Wellbeing. [14 lectures]

Environment: Introduction to Water, Air and Land Pollution, Environmental Legislation and Standards in India [4 lectures]

Parameters of Water Pollution, Biochemical Oxygen Demand, Five-day BOD Test, Modeling BOD (First order reaction), Effect of Oxygen demanding wastes on rivers, Oxygen sag curve, Streeter-Phelps Equation, Overview of air pollution, Global warming and climate change. [6 lectures]

Introduction to Solid Waste Management, Municipal Solid Waste (MSW) Management Practices, Introduction to Hazardous Waste and its Management, E-waste, Waste Electrical & Electronic Equipment (WEEE) Directives, Water Conservation, Role of IT in Environment Protection & Human Health [3 lectures]

Text/Reference Books:

1. Phil Hughes, Ed Ferrett, “Introduction to Health and Safety at Work”, Elsevier, 2009
2. S. Z. Mansdorf, “Handbook of Occupational Safety and Health”, Wiley, 2019
3. G. M. Masters and W. P. Ela, Introduction to Environmental Engineering and Science, 3rd edition, Pearson
4. Benny Joseph, Environmental Studies, 2nd edition, Tata McGraw Hill
5. S. K. Dhameja, Environmental Engineering & Management, 2nd edition, S. K. Kataria & Sons

Course Outcome (CO):

CO1: Understanding the fundamentals of Health and Safety.

CO2: Understanding the common Hazards and methods to control them.

CO3: Understanding the issues and challenges related to environment and pollution.

CO4: Understanding the methods to estimate and quantify environmental parameters.

Course Code: CSP102	**Course Credit: 2**
Course Name: Data Structures and Algorithms Lab	**L-T-P:** 0-1-2

Course Prerequisite: Basic programming in C language

Course Syllabus:
Concepts revision of C Programming Language, Data Types Revisited, Variable and Constant, Static and Dynamic Memory Allocation, Array, Pointer, Structure, Strings. [2 labs]
Sorting (Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort), Searching (Linear search and binary search). [2 labs]
Linked List (Creation, Insertion, Deletion and Search operations in Singly Linked List, Circular Linked List, Doubly Linked List and Circular Doubly Linked List). [2 labs]
Stack, Queue, Circular Queue, Priority Queue, Double Ended Queue, Infix, Prefix and Postfix expression conversion. [3 labs]
Tree (Creation of Binary and Multiway tree, Insertion, Deletion and Search in Binary Tree, Creation, Insertion, Deletion in Binary Search Tree, Inorder, Preorder and Postorder Traversal, Creation of Heap Tree, Heap sort), Graph (Creation of Directed and Undirected Graph, Depth First Traversal and Breadth First Traversal). [3 labs]

Text/Reference Books:
1.Ellis Horowitz, SartajSahni, Fundamentals of Data Structures, Computer Science Press, Latest Edition.
2.Robert Kruse, et al. Data Structures and Program Design in C, Pearson, Latest Edition.
3.Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, Data Structures and Algorithms, Addison Wesley, Latest Edition.
4.Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein, Data Structures Using C, PHI.

Course Outcome (CO):
CO1: To implement all basic data structures in C programming language.
CO2: To implement dynamic array, stack, queue, linked list and priority queue.
CO3: To implement various sorting and searching algorithms using linear data structures.
CO4: To analyze and implement binary search trees, graphs, heaps, B-Tree, B+-Tree and other non-linear data structures to solve various computing problems.
CO5: Design algorithms and implement using the combination of linear and nonlinear data structures.

Course Code: CSP112	**Course Credit: 2**
Course Name: IT Workshop – 2	**L-T-P:** 0-1-2
Course Prerequisite: NIL	

Course Syllabus:

Web Development, Beginning HTML and CSS, Creating HTML Content. [1 lab]
Photoshop Basics, CSS, Customizing Colors and Fonts, Styling Web Pages and Navigation. [2 labs]
Java Script, Adding Pages to a Website, Responsive design, testing [1 lab]
Debugging HTML/CSS. [1 lab]
PHP Basics and Functions, Error Handling, SQL Basics [1 lab]
Integrating PHP with Database. [1 lab]
Building Dynamic Website with PHP. [1 lab]
Build website with Django, Angular. [1 lab]
Integrate React in dynamic website. [1 lab]
Use AJAX in dynamic website. [1 lab]
Project: Building Dynamic Website with PHP, CSS, JavaScript, Ajax, SQL. [1 lab]

Text/Reference Books:

1. Craig Knuckles, David Yuen, “Web Applications Technologies Concepts and Real-World Design”, John Wiley, 1st Edition
2. Robert W. Sebesta, “Programming with World Wide Web”, Pearson, 6th Edition
3. Internet & Intranet Engineering, - Daniel Minoli, TMH.
4. W. Jason Gilmore, “Beginning PHP and MySQL: From Novice to Professional”, Apress, 2008

Course Outcome (CO):

CO1: To provide the students an exposure to develop dynamic and static websites using state-of-the-art technologies.
CO2: Hands-on the latest web development practices in Industry and provide practical exposure to develop effective and efficient websites using the latest open-source technologies.

Course Code: ECP102	**Course Credit: 1**
Course Name: Electronic Devices and Circuits Lab	**L-T-P:** 0-0-2

Course Prerequisite: NIL

Course Syllabus: [12 labs]

List of Experiments:

1. To study following: Basic circuit elements (resistor, capacitor, diode, transistor) and Basic measurements using lab equipment's (DMM, DSO, function generator, power supply).
2. To study I-V characteristics of pn junction and Zener diodes.
3. To study Positive and negative level clippers using diode.
4. To study Positive and negative clamper circuits using diode.
5. To study Voltage regulator using diode.
6. To study BJT input and output characteristics in CB configurations.
7. To study BJT input and output characteristics in CE configurations.
8. To study FET input and output characteristics.
9. To study FET transfer characteristics.
10. To study frequency response of BJT amplifier in CE configurations.
11. To study frequency response of FET amplifier.

To design an oscillator circuit.

Text/Reference Books:

1. Robert Boylestad, Louis Nashelsky, "Electronic Devices and Circuit Theory", Prentice Hall, 7th Edition
2. Jacob Millman, Christos C. Halkias, "Integrated Electronics: Analog and Digital Circuits and Systems", Tata McGraw Hill
3. Adel S. Sedra, Kenneth C. Smith, "Microelectronic Circuits", Oxford University Press, 5th Edition
4. Donald A. Neamen, "Microelectronics: Circuit Analysis and Design", McGraw Hill, 4th Edition
5. Leonard S. Bobrow, Navneet Gupta, "Foundations of Electrical Engineering", Oxford University Press, Asian Edition

Course Outcome (CO):

CO1: Understand the fundamental concepts of various electronic equipment's/components.

CO2: Utilize the various concepts of diodes for various diode circuits such as rectifiers, clippers, clampers, voltage regulators etc.

CO3: Learn and implement the various concepts of transistors.

CO4: To design oscillators.

Course Code: ECP112	**Course Credit: 1**
Course Name: Circuit Design and Printing Lab	**L-T-P:** 0-0-2
Course Prerequisite: NIL	

Course Syllabus:

Circuit Design and Printing fundamentals:

1. PCB Design overview, Prototyping and the PCB Design Flow.
2. Customizing the board outline and handling board outline errors.
3. Configure the Board Outline, Layer selection, Placing the parts at desired location.
4. Selection and application of routing method, optimal routing methods.
5. Finalizing a design for fabrication and exporting Gerbers.

Mini Project on circuit design and printing.

Introduction to 2D and 3D design using EDA tools.

Text/Reference Books:

1. https://www.ni.com/tutorial/12242/en/
2. Printed Circuit Board by RS Khandpur, Tata McGraw Hill Education Pvt Ltd., New Delhi
3. Electronic Product Design Volume-I by S D Mehta, S Chand Publications
4. Open-source EDA Tool KiCad Tutorial: http://kicad-pcb.org/help/tutorials/
5. Bernd S. Palm, Introduction to AutoCAD 2020, CRC Press, 2020.
6. https://www.tinkercad.com/dashboard.

Course Outcome (CO):

CO1. Understand the need for PCB Design and steps involved in PCB Design and Fabrication process. CO2. Familiarize Schematic and layout design flow using Electronic Design Automation (EDA) Tools
CO3. Understand the steps involved in schematic, layout, fabrication, and assembly process of PCB design.
CO4. Design PCB for analog and digital circuits.
CO5: Learn the 2D and 3D design fundamentals.

Course Code: HSP102	**Course Credit: 1**
Course Name: Technical Writing and Presentation Skills Lab	**L-T-P:** 0-0-2

Course Prerequisite: NIL

Course Syllabus:
Exercise to Lower Anxiety, Build Confidence. [4 labs]
Exercise in Clarity of Messaging. [4 labs]
Exercise in Effective Speaking. [5 labs]
Exercise in Structured Writing. [5 labs]
Exercise in Organizing, & delivering a Memorable Presentation. [6 labs]

Text/Reference Books:
1.Handbook of Technical Writing: Charles T Brusaw, Gerald J Alred& Walter E Oliu, St. Martin’s Press, New York.
2.Technical Writing 101: Alan S Pringle & Sarah S O’Keefe, Scriptorium Publishing Services Inc
3. Every Page is Page One: Mark Baker XLM Press
4. How to Talk to Anyone: Leil Lowndes, McGraw Hill
5. Talk Like Ted: Carmine Gallo, Pan Macmillan

Course Outcome (CO):
CO1: Clear Communication
CO2: Communicating Complex Ideas & Projects effectively.
CO3: Practical application of the Lecture Content.
CO4: Building Confidence and effectiveness.

B.Tech CSE 2nd Year

3rd Semester						
			Scheme			
S.No	Subject Code	Subject	L	T	P	Credits
1	CST201	Computer Architecture and Organization	3	0	0	3
2	CST203	Database Management Systems	3	0	0	3
3	CST205	Software Engineering	3	0	0	3
4	ECT211	Microprocessors and Microcontrollers	3	0	0	3
5	MAT201	Discrete Mathematical Structures	3	1	0	4
6	HST201	Engineering for Social Empowerment	3	0	0	3
		Labs				
7	CSP201	Computer Architecture and Organization Lab	0	0	2	1
8	CSP203	Database Management Systems Lab	0	0	2	1
9	ECP211	Microprocessors and Microcontrollers Lab	0	0	2	1
10	CSP211	IT Workshop – III	0	1	2	2
11	OTP201	Engineering Creativity, Innovation and Design	0	1	2	2
Total			**18**	**3**	**10**	**26**

Course Code : CST201	**Course Credit : 3**
Course Name : Computer Architecture and Organization	**L-T-P : 3-0-0**

Course Prerequisite: Digital Logic Design, Microprocessors

Course Syllabus:

Introduction: Architecture vs Organization, Abstraction level, Y-chart, Views, Design Matrices, Instruction Set Architecture, Von Neumann and Harvard Architecture; RISC versus CISC; Flynn's Classification, Synthesis and design Metrics, Fundamentals of HDL (System Verilog/ VHDL)
[8 Lectures]

Basic CPU Organization: General purpose Registers Organization; Stack Organization; Bit-sliced CPU; Accumulator-based CPU Data Representation - Basic Data-type formats; Storage order: Big-endian and Little-endian Instruction Formats - RISC and CISC type; Instruction Types; Instruction Cycle and Machine Cycle. Addressing Modes **[6 Lectures]**

Multiplication: Shift & Add Multiplier, Two's Complement Multiplier, Array Multiplier, Booth Multiplier; Division - Restoring & Non-Restoring Division, Floating Point Arithmetic - Addition, Subtraction and Multiplication for IEEE 754 standard, Arithmetic-Logic Units - Combinational ALUs and Sequential ALUs (basic concepts). **[8 Lectures]**

Processor Design: Logic Design, Conventions, Hardwired Control versus microprogrammed control, Single cycle implementation, Multi-cycle implementation, Performance enhancement using pipelining, Pipelining - Instruction & Arithmetic Pipeline, Concept, Structure and Space-time diagram.
[8 Lectures]

Memory: Memory Characteristics, Memory Access modes, Design and Interfacing; Multi-level Hierarchy; Cache Memories, Organization, R/W Operations, Address Mapping, Performance issues and Replacement Policies **[6 Lectures]**

Interconnects: Intra system versus Inter system; Buses, Interconnection Structures; Bus Control - Features and Data Transfers, I/O Control Methods - Programmed I/O, Interrupt Driven I/O and I/O Processors with an example. **[4 Lectures]**

Text/Reference Book(s):

1. D.A. Patterson, J.L. Hennessy, “Computer Organization and Design”, Elsevier, 5th Edition
2. John P. Hayes, “Computer Architecture and Organization”, McGraw Hill, 5th Edition
3. William Stalling, “Computer Organization and Architecture”, Prentice Hall India
4. C. Hamacher, Z.Vranesic, S. Zaky, “Computer Organization”, McGraw Hill, 5th Edition

Course Outcome (CO): At the end of the course, students will be able to

CO1: Describe the interconnection between various functional units of a computer system and be able to assess the performance of a computer.

CO2: Describe the importance of the design matrices like power and thermal.

CO3: Describe the pipelines-based performance issues and performance enhancement.

CO4: Describe various parts of a system memory hierarchy and caching techniques.

CO5: Evaluate the performance of CPU, memory and I/O operations.

Course Code : CST203	**Course Credit : 3**
Course Name : Database Management Systems	**L-T-P : 3-0-0**

Course Prerequisite: Basic understanding of computers and logics

Course Syllabus:

Introduction: An overview of database management system, database system Vs file system, Database system concept and architecture, data model schema and instances, data independence and database language and interfaces, data definitions language, DML, Overall Database Structure. ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationship of higher degree. **[10 Lectures]**

Relational data Model and Language: Relational data model concepts, integrity constraints, entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus, Characteristics of SQL, advantage of SQL. SQL data type and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and subqueries. Aggregate functions. Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL **[6 Lectures]**

Database Design & Normalization: Functional dependencies, normal forms- 1NF, 2NF, 3NF, BCNF, inclusion dependence, lossless join decompositions **[8 Lectures]**

Transaction Processing Concept: Transaction system, Testing of serializability, 8 serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling. Distributed Database: distributed data storage,directory system. **[10 Lectures]**

Concurrency Control Techniques: Concurrency control, Locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction **[6 Lectures]**

Text/Reference Book(s):

1. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill
2. Date C J, " An Introduction to Database Systems", Addision Wesley
3. Elmasri, Navathe, " Fundamentals of Database Systems", Addision Wesley
4. O'Neil, Databases, Elsevier Pub.

Course Outcome (CO):

CO1: Students understand the basic concepts of DBMS and various databases used in real applications

CO2: Students will be able to design relational database using E-R model and normalization

CO3: Students will be able to demonstrate structured query languages for various database applications

CO4: Students will be able to explain transaction management, recovery management, and concurrency control for real application

Course Code : CST205	**Course Credit : 3**
Course Name : Software Engineering	**L-T-P : 3-0-0**

Course Prerequisite: Basic knowledge of C programming language

Course Syllabus:

Introduction to Software Engineering: Reasons for software project failure – Similarities and differences between software and other engineering products. Software Components, Software Characteristics, Software Crisis, Software Development Life Cycle (SDLC)– Phases Overview, Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Choosing a social relevant problem-Summary Team Report. **[8 Lectures]**

Problem partitioning (subdivision): Power of Abstraction, Concept of functional decomposition , process modeling : DFDs, Concept of data modeling : ER diagrams, Class and component level design : UML diagrams. **[4 Lectures]**

Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Assessment: Impact of Requirement Engineering. Decision Tables, SRS Document, IEEE Standards for SRS, Architectural design, component level design, user interface design, WebApp Design, SRS Documentation for Team Project. **[8 Lectures]**

Coding and Testing: Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing, Software Testing Strategies - Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Introduction to secure programming. Types of testing – Specification of test cases – Code review process. **[8 Lectures]**

Software Quality Assurance (SQA): Quality concepts, Review techniques, Verification and Validation, SQA Plans, Software Quality Frameworks. Assessment: Framing SQA Plan. ISO 9000 Models, SEI-CMM Model and their relevance to project Management-other emerging models like People CMM. **[6 Lectures]**

Software Configuration Management (SCM): versioning, Reusable components, Mathematical methods of risk assessment and management, Methods of software licensing and introduction to free software. Software Maintenance: Maintenance Characteristics, Maintainability, Maintenance tasks and side effects. Risk Management, Maintenance and Reengineering. Risk Assessment: Preparation of Risk mitigation plan. **[6 Lectures]**

Text/Reference Book(s):

1. Pressman R.S: Software Engineering: A Practitioner approach, McGraw Hill.
2. Sommerville I: Software Engineering, Addison Wesley
3. Ghezzi C. Jazayeri M and Mandrioli: Fundamentals of Software Engg. , PHI
4. PankajJalote, Software Engineering: A Precise Approach (1/e), Wiley India, 2010.

Course Outcome (CO):

CO1: Learn basic SW engineering methods and practices, and understanding of software models.

CO2: Learn software requirements and the SRS documentation, software design process and principles.

CO3: Learn implementation issues i.e., modularity, coding standards and software testing approaches.

CO4: Learn planning, scheduling, risk management, ethical and professional issues of software engineers.

Course Code : ECT211	**Course Credit : 3**
Course Name : Microprocessors and Microcontrollers	**L-T-P : 3-0-0**

Course Prerequisite: Digital Logic Design

Course Syllabus:

Introduction: Motivation, history, growth curves and issues that microprocessors face till now. Basic system stack – hardware and software. ARM- motivation and objectives. Microprocessor trends and development, Processor design trade-offs, Components of the Microprocessor, Motivation to design CISC and RISC, Level of abstractions and View (Y-Chart), Design Matrices (Performance, power, thermal and reliability). ISA design concepts and needs. **[4 Lectures]**

ARM Architecture and Assembly Language Programming: The Acorn RISC Machine, Architectural inheritance, The ARM programmer's model, ARM development tools. Data processing instructions, Data transfer instructions, Control flow instructions. **[4 Lectures]**

ARM Organization and Implementation: Comparison Case study of ARM 7, 9, microarchitectures with 8085. ARM organization- 3-stage and 5-stage pipeline. ARM instruction execution, ARM implementation, The ARM coprocessor interface. ARM buses. **[8 Lectures]**

ARM Instruction Set: General Purpose, special purpose architecture, control word, status registers and operating modes, Introduction, Exceptions, Conditional execution, Branch and Branch with Link (B, BL) and eXchange (BX, BLX), Software Interrupt (SWI), Data processing instructions, Multiply instructions, Count leading zeros (CLZ - architecture v5T only), Single word, Half-word and signed byte data transfer instructions, Multiple register transfer instructions, Swap memory and register instructions (SWP), Status register to general register transfer instructions and vice-versa, Coprocessor- instructions, data operations, data transfers, register transfers, Breakpoint instruction (BRK - architecture v5T only), Unused instruction space, Memory faults, ARM architecture variants.
[9 Lectures]

ARM Operation Modes: The Thumb bit in the CPSR, The Thumb programmer's model, Thumb branch instructions, Thumb software interrupt instruction, Thumb data processing instructions, Thumb single register data transfer instructions, Thumb multiple register data transfer instructions, Thumb breakpoint instruction, Thumb implementation, Thumb applications Example and exercises, The ARM memory interface, The Advanced Microcontroller Bus Architecture (AMBA), The ARM reference peripheral specification **[6 Lectures]**

Architectural Support for System Development: Prototyping tools, JTAG boundary scan test architecture, The ARM debug architecture, Embedded Trace Signal processing support Example and exercises. Architectural Support for High-Level Languages: Abstraction in software design, Data types, Floating-point data types, The ARM floating-point architecture, Expressions Conditional statements, Loops, Functions and procedures
[6 Lectures]

Text/Reference Book(s):

1. Steve Furber, "ARM system-on-chip architecture", Addison-Wesley, 2000,2nd ed.
2. Andrew Sloss, Dominic Symes, Chris Wright, "ARM System Developer's Guide: Designing and Optimizing System Software", Morgan Kaufmann, Year: 2004

Course Outcome (CO):

CO1: Understand the concepts and methodologies employed in designing microprocessor core.

CO2: Understand the principles of microprocessor design using ARM.

Course Code : MAT201	**Course Credit : 4**
Course Name : Discrete Mathematical Structures	**L-T-P : 3-1-0**
Course Prerequisite: NIL	

Course Syllabus:

Set theory: Definition of Sets, Venn Diagrams, complements, Cartesian products, power sets, counting principle, cardinality and countability, proofs of some general identities on sets. **[4 Lectures]**

Relation and Functions: Definition, types of relation, composition of relations, domain and range of a relation, pictorial representation of relation, properties of relation, equivalence relation partial ordering relation. Function: Definition and types of function, composition of functions), pigeonhole principle. **[5 Lectures]**

Propositional logic: Proposition logic, basic logic, logical connectives, truth tables, tautologies, contradiction, converse, inverse, contrapositive, negation, and contradiction. Deduction, Resolution, Predicates and Quantifiers, Mathematical Proofs. **[10 Lectures]**

Combinatorics: Mathematical induction, recursive mathematical definitions, basics of counting, permutations, combinations, inclusion-exclusion, recurrence relations (nth order recurrence relation with constant coefficients, Homogeneous recurrence relations, and Inhomogeneous recurrence relation), and generating function (closed form expression, properties of G.F., solution of recurrence relation using G.F, solution of combinatorial problem using G.F.) **[8 Lectures]**

Algebraic Structure: Binary composition and its properties definition of algebraic structure; Semi group, Monoid Groups, Abelian Group, properties of groups, Homomorphism, isomorphism, Permutation Groups, Sub Group, Cyclic Group, Rings and Fields (definition and standard results). **[6 Lectures]**

Graph Theory: Graph terminology, types of graph connected graphs, components of graph, Euler graph, Hamiltonian path and circuits, Graph coloring, Chromatic number. Tree: Definition, types of tree (rooted, binary), properties of trees, binary search tree, tree traversing (preorder, inorder, postorder). **[7 Lectures]**

Text/Reference Book(s):

1. J.P. Tremblay and R. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, McGraw Hill.
2. Kenneth H. Rosen, “Discrete Mathematics and its Applications’’, Mc.Graw Hill, 2002.
3. Grimaldi, R.P. “Discrete and Combinatorial Mathematics”, Pearson Education, 2002
4. C.L.Liu, Elements of Discrete Mathematics, McGraw-Hill Book

Course Outcome (CO):

CO1: To enable the students to think logically and mathematically.
CO2: To apply mathematical reasoning in which mathematical problems could be solved.
CO3: To see the practical aspects of mathematical reasoning, combinatorial analysis, discrete structures, and mathematical modeling.
CO4: To observe the real life problems where the concepts of logic, set theory, counting, probability theory, graph theory, trees, Boolean algebra, and modeling computation can be applied.

Course Code : HST201	**Course Credit : 3**
Course Name : Engineering for Social Empowerment	**L-T-P : 3-0-0**

Course Prerequisite: An interest in applying Engineering for Social Empowerment

Course Syllabus:

Introduction to Engineering for Social Empowerment: Engineers as the Problem Solvers- How can they bring change in the society, Elements of engineering for social empowerment, Complex Systems, Wicked Problems, Dancing with the system, Ethics, principles and practice in Community **[8 Lectures]**

Planning for Social Empowerment: How can engineers plan and bring positive social change through their work, Impact pathways for social empowerment, Understanding the system that you want to influence, Theory of Change, Gender, Case study **[8 Lectures]**

Engaging and Partnering: Who are the stakeholders anyway & why should engineers work with them? Developing Strategic partnerships, Principles of stakeholder engagement, Stakeholder engagement methodologies, Stakeholder engagement and power relations, challenges and opportunities when engaging with diverse groups of stakeholders. **[10 Lectures]**

Communicating for Impact: Moving beyond dissemination to bi-directional communication, what is strategic communication and why it is important for engineers to communicate their work to diverse stakeholders of the community? Planning for effective and impactful communication, The elevator pitch, Case Study **[7 Lectures]**

Building Capacities: Why do we need capacity development of all stakeholders, Different types and aspects of capacity development, dealing with challenges, whose capacities need to be developed, Case Study India - Developing Stakeholder Capacity through games and 3D models **[7 Lectures]**

Text/Reference Book(s):

1. Meadows, D. (2001). Dancing with systems. *Whole Earth*, *106*, 58-63.

2. Carden, F. (2009). *Knowledge to policy: Making the most of development research*. IDRC.

3. Reed, M. S. (2016). *The research impact handbook*. Fast Track Impact.

4. McLaughlin, M. W., & Mitra, D. (2001). Theory-based change and change-based theory: Going deeper, going broader. *Journal of educational change*, *2*(4), 301-323.

5. Bourne, L., & Weaver, P. (2010). Mapping stakeholders. *Construction stakeholder management*, 99-120.

6. Bryson, J. M. (2004). What to do when stakeholders matter: stakeholder identification and analysis techniques. *Public management review*, *6*(1), 21-53.

7. Chen, P. G., Diaz, N., Lucas, G., & Rosenthal, M. S. (2010). Dissemination of results in community-based participatory research. *American journal of preventive medicine*, *39*(4), 372-378.

Course Outcome (CO):

CO1: Develop an ability to identify, formulate, and solve engineering problems with a realistic world view.

CO2: Understand the need and develop the sensibility to work with community

CO3: Develop an ability to work with multi-disciplinary teams and various stakeholders

CO4: Understand professional and ethical responsibility towards community

CO5: Develop an ability to communicate effectively to all stakeholders

CO6: Develop a recognition of the need for, and an ability to engage in life-long learning and partnership with stakeholders

Course Code : CSP201	**Course Credit : 1**
Course Name : Computer Architecture and Organization Lab	**L-T-P : 0-0-2**
Course Prerequisite: Digital logics	

Course Syllabus:

Introduction to HDL and Xilinx toolchain for synthesis and FPGA Kit Concept: Implementation of logic gates, and combinational circuits using Dataflow, structured and behavioral coding. **[2 Labs]**

Single-cycle and multicycle microarchitecture design, Pipeline implementation, Concept of switching activity, delay, area and power calculation, Timing Analysis-XDC/SDC, SDF, SAF for FPGA and ASIC technology. **[3 Labs]**

Understanding of Architecture Level Simulators: SimpleScalar, McPAT, HotSpot, and Reliability tools. **[3 Labs]**

Cache proliferation, design, and protocol implementation for design metrics (area, power, and performance).
Power and thermal aware design implementation at the architectural level. **[3 Labs]**

Introduction to RISC V - RTL and simulator. **[1 Labs]**

Text/Reference Book(s):

1. SimpleScalar Download and Help available at http://www.simplescalar.com/
2. McPAT Download and Help available at https://www.hpl.hp.com/research/mcpat/
3. HotSpot Download and Help available at http://lava.cs.virginia.edu/HotSpot/
4. RISC-V Helps available at https://riscv.org/software-tools/riscvemu/
5. SystemVerilog for Design Second Edition: A Guide to Using SystemVerilog for Hardware Design and Modeling by Stuart Sutherland.
6. Timing Analysis and Optimization of Sequential Circuits Author(s): Naresh Maheshwari, Sachin S. Sapatnekar (auth.) Publisher: Springer US
7. Synopsys, Cadence, and Xilinx manuals.
8. Static Timing Analysis For Nanometer Designs: A Practical Approach by J. Bhasker

Course Outcome (CO):

CO1: Describe the exact choice of implementation of a system choosing the correct HDL language.
CO2: Design the performance and power aware microarchitecture of hardware components.
CO3: Use industry specific and research-based tools and design.
CO4: Describe the memory design and implement the protocols on the simulator to view the impact of these protocols and techniques on the performance and power of the system.

Course Code : CSP203	**Course Credit : 1**
Course Name : Database Management Systems Lab	**L-T-P : 0-0-2**

Course Prerequisite: Basic understanding of computers and logics

Course Syllabus:

Practice My SQL queries for Data Manipulation (Insert, Update, Delete, Select) and Data Definition (Create, Drop, Truncate, Rename, etc.) Language **[2 Labs]**

Practice SQL queries using logical operations and operators (Arithmetic, Comparison, Logical, etc.) **[1 Lab]**

SQL queries using group by and order by functions **[1 Lab]**

SQL queries for group functions(Avg, Count, Max, Min ,Sum) **[1 Lab]**

Practice Subqueries / Nested Queries **[1 Lab]**

SQL queries to implement joins **[1 Lab]**

SQL Queries for extracting data from more than one table **[1 Lab]**

Implement a mini database project with all the sql query concepts learnt above **[4 Labs]**

Text/Reference Book(s):

1. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill
2. Date C J, " An Introduction to Database Systems", Addision Wesley
3. Elmasri, Navathe, " Fundamentals of Database Systems", Addision Wesley
4. O'Neil, Databases, Elsevier Pub.
5. Leon & Leon, "Database Management Systems", Vikas Publishing House
6. Bipin C. Desai, "An Introduction to Database Systems", Galgotia Publications
7. Majumdar & Bhattacharya, "Database Management System", TMH

Course Outcome (CO): At the end of the course the students are able to:

CO1: Apply the basic concepts of Database Systems and Applications.
CO2: Use the basics of SQL and construct queries using SQL in database creation and interaction.
CO3: Students will be able to combine the theoretical knowledge and practical skills learn in the course to build a mini database project. stakeholders

Course Code : ECP211	**Course Credit : 1**
Course Name : Microprocessors and Microcontrollers Lab	**L-T-P : 0-0-2**
Course Prerequisite: NIL	

Course Syllabus:

Introduction to ARM Board, cables, connection procedure, and Keil software- How to write a Embedded C program on it, Build, and load the program on board **[2 Labs]**

Interface and control a DC motor, ADC **[1 Lab]**

Interface a stepper motor and rotate it clock and anti-clockwise and different application **[2 Labs]**

LED interface and executing delay and pattern **[1 Lab]**

LCD controller and its interfacing to board **[2 Labs]**

Keyboard Interfacing **[1 Lab]**

How PWM can be used and application programs **[2 Labs]**

Project Evaluation: Build any automation systems using software and hardware **[2 Labs]**

Text/Reference Book(s):

1. Steve Furber, “ARM system-on-chip architecture”, Addison-Wesley, 2000,2nd ed.
2. Andrew Sloss, Dominic Symes, Chris Wright, “ARM System Developer's Guide: Designing and Optimizing System Software”, Morgan Kaufmann, Year: 2004

Course Outcome (CO):

CO1: Understand the architecture of ARM- based assembly programming.

CO2: Learn to design, construct, program, verify, analyze and troubleshoot ARM assembly and C language programs and supporting hardware.
CO3: Learn basic knowledge of sensor and board interfacing

Course Code : CSP211	**Course Credit : 2**
Course Name : IT Workshop-III	**L-T-P : 0-1-2**

Course Prerequisite: Basic C programming and computer knowledge

Course Syllabus:

Installation of Python Tool, Introduction to Python programming **[1 Lab]**

Data types, Input/Output and library imports **[1 Lab]**

Python strings operations, Doc strings **[1 Lab]**

Objects - List, Tuples and Dictionaries **[1 Lab]**

Control flow, functions working and some advanced functions **[1 Lab]**

File handling and third party library integration **[1 Lab]**

Usage of image processing library **[1 Lab]**

Data exchange mechanism - JSON, Understanding web services - REST APIs **[1 Lab]**

Advanced part of python functions, Database interaction **[1 Lab]**

Regular expressions and their uses in searching **[1 Lab]**

Numpy, Matlabplot, pandas utility functions **[1 Lab]**

JSON format for NoSQL database **[1 Lab]**

Text/Reference Book(s):

1. John Zelle and Michael Smith, Python Programming: An Introduction to Computer Science, Franklin, Beedle& Associates Inc
2. Allen Downey, Jeff Elkner and Chris Meyers, Learning with Python: How to Think Like a Computer Scientist, Latest Edition
3. David Beazley and Brian K. Jones, Python Cookbook: Recipes for Mastering Python 3, O'Reilly Media

Course Outcome (CO): After completing the course, a student will be able:

CO1: To acquire programming skills in core Python.
CO2: To implement basic principles of python programming language and implement object-oriented concepts.
CO3: To use backend database services and make graphical user interface applications.
CO4: To handle large dataset in real-time engineering problems and develop real-time, fast and flexible solutions.

Course Code : OTP201	**Course Credit : 2**
Course Name : Engineering Creativity, Innovation and Design	**L-T-P : 0-1-2**
Course Prerequisite: None	

Course Syllabus:

Creativity as a human need, its utility and creative expression, *Discourse, philosophy of Human Creativity, Psychological basis of Human Creativity* **[2 Lab]**

Creative Expression in Art Design and Engineering: *Art v/s Design, Craft and Creative process, Engineering and Design* **[2 Lab]**

Fundamentals of Structuring creative process: *Design Methodologies , Design Thinking, Design Criticism and Subjectivity.* Creative ideation and thinking visually: *Doodling as Visual Thinking, Design Sketching basics, Diagramming as structured design communication* **[3 Lab]**

Engineering concepts for design and creativity: *Simple machines, Common mechanisms,* Innovation as a deliberate creative pursuit**,** Designing and Engineering of Everyday objects **[2 Lab]**

Hands-on: Workshop involving ideation and brainstorming on selected themes (For example, Agriculture, Education, Energy, Environment Conservation, Health, Infrastructure, Safety, Transportation, Waste Management, Water Conservation, etc), Mind mapping exercises and design assignment, Theme based concept evaluation and allocation of project. **[3 Lab]**

Text/Reference Book(s):

1. Berger, J. (2008). Ways of seeing. Penguin UK.
2. Deutsch, R. (2020). Think Like An Architect: How to develop critical, creative and collaborative problem-solving skills. RIBA Publishing. and 2. Berger, J. (2008).
3. Ways of Seeing. Penguin Books Ltd;
4. Norman, D. (2013). The design of everyday things: Revised and expanded edition. Basic books.
5. Lawson, B. (2006). How designers think: The design process demystified. Routledge.
6. Cross, N., & Roy, R. (1978). Design Methods Manual: Prepared for the {Open University, Man-Made Futures; Design and Technology} Course Team. Open UP.
7. Hanington, B., & Martin, B. (2012). Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions. Rockport Publishers.

Course Outcome (CO):

CO1: Understand the fundamentals of human creative expression and its need

CO2: Understanding how to appreciate criticize and empathise with creative expression

CO3: Learn engineering design principles, basics of design thinking, frugal innovation

CO4: Learning and applying principles of design and engineering innovation in practice, through projects

CO5: Learn technology and design through analyzing everyday objects and common mechanisms

4th Semester						
S.No	Subject Code	Subject	Scheme			
			L	T	P	Credits
1	CST202	Object Oriented System Design	3	0	0	3
2	CST204	Design and Analysis of Algorithms	3	0	0	3
3	CST206	Operating Systems	3	0	0	3
4	CST208	Artificial Intelligence	3	0	0	3
5	ECT212	Communication Systems	3	0	0	3
6	MAT202	Probability and Statistics	3	0	0	3
		Labs				
7	CSP202	Object Oriented System Design Lab	0	0	2	1
8	CSP204	Design and Analysis of Algorithms Lab	0	1	2	2
9	CSP206	Operating Systems Lab	0	0	2	1
10	CSP210	Foundation of Data Science Lab	0	0	2	1
11	OTP202	Entrepreneurship and Business Incubation	0	1	2	2
Total			**18**	**1**	**11**	**25**

Course Code : CST202	**Course Credit : 3**
Course Name : Object Oriented System Design	**L-T-P : 3-0-0**
Course Prerequisite: Elementary knowledge of programing language C	

Course Syllabus:

Introduction: Principles of OOD, programming Paradigms, benefits and applications, Comparison of Java with C and C++, Java Evolution and History, Features of Java Language. Difference between JVM, JDK JRE and JIT, Installing Java in WINDOWS. **[2 Lectures]**

Decision Making & Looping: Data types, UNICODE, Reference type. Static variables, Instance variable, Local variables, final variable, Static block and Non-static block, final, abstract, Decision Making, looping and operators in Java. **[4 Lectures]**

Classes and Objects: constructors, parameterized constructors, overloaded constructors, constructors with default arguments, Access Control, Modifiers, methods Nested, Inner Class, Anonymous Classes, Abstract Class & Interfaces, Argument Passing Mechanisms, Method Overloading, Dealing with Static Members and class. Finalize () Method. Use of "this" reference. **[5 Lectures]**

Arrays & Strings: Single, Double Dimensional Array , Arrays class, Methods in Arrays class , String–what and why Operation on String Immutable, String comparison and concatenation, Method of String class StringBuffer class and its methods. StringBuilder class, Creating Immutable class like String. Tokenizing a String. **[6 Lectures]**

Inheritance & Polymorphism: types of inheritance, constructors in derived and base class, abstract classes, Compile and run time polymorphism, Role of Constructors in inheritance, Overriding Super Class Methods. Use of "super" keyword. Implementing interfaces. Dynamic method dispatching by down-casting and up-casting. **[7 Lectures]**

Packages & Exception Handling : Organizing Classes and Interfaces in Packages. Sub-Package CLASSPATH Setting for Packages. Making JAR Files for Library Packages Import and Static Import Creating .EXE and jar executable file. Exceptions, Errors, Checked and UnChecked Exceptions, Control Flow in Exceptions, try and catch block, Multiple catch block, Nested try, finally block, throw keyword.**[8 Lectures]**

Input/ Output operations in Java: Understanding Streams File class and its methods, Creating file and folder using java code. File Input/Output Stream, File Writer & File Reader. Input from keyboard by Input Stream Reader. Print Stream class Print Writer class, Buffered Reader and Buffered Writer class. Scanner class. AWT class. **[8 Lectures]**

Text/Reference Book(s):

1. Ali Bahrami, Object Oriented Systems Development, McGraw Hill International Edition.

2. Java: A Beginner's Guide, Eighth Edition by Schildt, Herbert

3. Head First Java by Kathy Sierra & Bert Bates, O'REILLY

Course Outcome (CO):

CO1: Ability to analyze and model software specifications.
CO2: Ability to abstract object-based views for generic software systems.
CO3: Ability to deliver robust software components.

Course Code : CST204	**Course Credit : 3**
Course Name : Design and Analysis of Algorithms	**L-T-P : 3-0-0**

Course Prerequisite: Data structures and programming

Course Syllabus:

Asymptotic analysis, Worst average and best cases, Asymptotic notation, Little-o and little-omega notations, Lower and upper boundaries, Tractable and Intractable problems, Algorithms analysis using loops and trees, Solving recurrences, Amortized analysis **[8 Lectures]**

Divide and conquer: General method, Binary search, Quick sort, randomized quick sort, Merge sort, Strassen’s matrix multiplication, Recurrence equation for divide-and-conquer, Topological Sort. Graph Algorithms: Depth first search, Breadth first search, Applications of depth first search, Detecting cycle in a graph **[8 Lectures]**

Basics of greedy approach, Knapsack problem, Kruskal and Prim’s minimum spanning tree, Huffman coding, Efficient Huffman coding for sorted input, Dijkstra’s shortest path algorithm, Basics of dynamic programming, Overlapping subproblems property, Optimal substructure property, Matrix chain multiplication, 0-1 Knapsack problem, Bellman Ford algorithm, Floyd-Warshall all pair shortest path algorithm. **[10 Lectures]**

String Matching Algorithms: Naïve method, KMP algorithm, Robin-Karp algorithm, Boyer Moore algorithm, Suffix array. Backtracking: N-queen problem, Subset sum, Graph m-coloring problem.
[6 Lectures]

Polynomial time complexity and intractability, Decision Problems, Non-deterministic polynomial algorithms, Satisfiability and verification, NP-completeness, NP-hard, Cook’s theorem, 2-SAT and 3-SAT problems, Problem reduction, Vertex cover problem, Graph coloring problem, Independent Set, Travelling Salesman Problem, Introduction to approximation algorithms **[8 Lectures]**

Text/Reference Book(s):

1. Thomas H. Cormen, et al. Introduction to Algorithms, Latest Edition, MIT Press

2. NarasimhaKarumanchi, Data Structures and Algorithms Made Easy, Latest Edition, CareerMonk Publication

3. Jon Kleinberg and Eva Tardos, Algorithm Design, Latest Edition, Pearson

4. Robert Sedgewick and Kevin Wayne, Algorithms, Latest Edition, Addison Wesley

Course Outcome (CO): At the end of this course, each student should be able:

CO1: To analyze the complexity of algorithms in the form of recurrence relation.
CO2: To implement various algorithmic paradigms such as divide-and-conquer, greedy and dynamic programming.
CO3: To analyze the solutions derived from randomized and approximation algorithms in real-time problems.
CO4: Analyze the performance of various graph algorithms to find the shortest route and other operations.

CO5: Understand and analyze the NP-Hard problems and their approximate solutions.

Course Code : CST206	**Course Credit : 3**
Course Name : Operating Systems	**L-T-P : 3-0-0**

Course Prerequisite: Basic knowledge of computer systems

Course Syllabus:

Introduction and Process Management: Need of operating system, types of OS, operating system as resource manager, OS services, kernel, system calls, firmware, bootloader, process model, creation, termination, states and transitions, context switching, process control block, system calls in Linux and Windows, processes versus threads, kernel and user level threads and multi-threading. **[6 Lectures]**

Process Scheduling: Process scheduling - concepts, CPU and I/O bound, CPU scheduler - short, medium, long-term dispatcher. Scheduling - preemptive and non-preemptive, Priority, Scheduling algorithms - FCFS, SJFS, Shortest Remaining Time, round robin, priority scheduling, multilevel queue scheduling, multilevel feedback queue scheduling, fair share scheduling. **[8 Lectures]**

Inter-Process Communication: Message passing, race condition, critical section problem, mutual exclusion with busy waiting, Peterson's solution, Semaphore, Classical IPC problems, Deadlock problem, detection, prevention, avoidance, recovery from deadlock. **[10 Lectures]**

Memory Management: Memory management - concepts, logical and physical address space, address binding, degree of multiprogramming, swapping. Memory allocation schemes, Free space management, memory protection and sharing, relocation and address translation, Virtual Memory- concept, paging, segmentation, segmentation with paging, demand paging, thrashing. Page replacement algorithms - optimal, MRU, FIFO, LRU, Belady's anomaly, design issues for paging system. Page size, TLB. Inverted page table. Basic idea of MM in Linux. **[10 Lectures]**

File System and Storage: File System - concepts, operations, types. File organization and access (Sequential, Direct, Index and Sequential) methods. Memory mapped files, directory structures, file system mounting, file sharing. Overview of file system in Linux, Input/output subsystems- concepts, input/output devices, disk structure, disk storage capacity. Disk scheduling algorithm - FCFS, SSTF, Scan scheduling, C-scan schedule, Look and C-Look schedule. **[6 Lectures]**

Text/Reference Book(s):

1. Silberschatz and Galvin: Operating System Concepts, Wiley India Pvt. Ltd.
2. Tanenbaum: Modern Operating System, Prentice Hall.
3. OS – Three Easy Step by Remzi (available free online)

Course Outcome (CO):

CO1: Students will be able to describe the general architecture of computers
CO2: Students will be able to describe, contrast and compare differing structures of operating systems
CO3: Students will understand and analyse theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files

Course Code : CST208	**Course Credit : 3**
Course Name : Artificial Intelligence	**L-T-P : 3-0-0**
Course Prerequisite: Basic knowledge of data structures	
Course Syllabus: Introduction; Intelligent Agents; Problem Formulation; Problem Solving by Searching – Uninformed Search (BFS, Uniform cost, DFS, Depth limited, Iterative deepening, Bidirectional), Constraint Satisfaction Search, Heuristic Functions, Informed Search (Greedy search, A*, IDA*, SMA*), Hill Climbing, Simulated Annealing **[10 Lectures]** Knowledge-based Agents; Reasoning and Logic; Propositional Logic – Syntax, Semantics, Validity, Inference; First-order Logic; Inference in First-order Logic – Inference Rules, Forward and Backward Chaining, Completeness, unification, Resolution, Rule value approach **[8 Lectures]** Planning Agent; Problem Solving to Planning; Planning with State Space Search; Partial-Order Planning; Planning graphs; Planning and Acting in Real World – Conditional Planning, Replanning, Planning and Execution **[6 Lectures]** Uncertainty; Rational Decisions; Probability Review and Reasoning; Bayes' Rule – Simple case and Normalization; Bayesian Networks; Inference in Bayesian Networks; Dynamic Bayesian Networks; Statistical Validation Techniques; Hidden Markov Models (HMM); Expectation-maximization algorithm **[10 Lectures]** Introduction to expert systems; Features of an expert System; Knowledge based overview; Explanation based learning; Learning by Induction; Uncertainty in expert systems; Expert Systems Tools: MYCIN, EMYCIN **[6 Lectures]**	
Text/Reference Books: 1. S. Russell and P. Norvig, "Artificial Intelligence – A Modern Approach", Prentice-Hall, 2009. 2. RajendraAkerkar, "Introduction to Artificial Intelligence", PHI, 2005. 3. G. F. Luger, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", Fifth Edition, Addition Wesley, 2005. 4. N. J. Nilsson, "Artificial Intelligence: A New Synthesis", Morgan Kaufmann Publishers, 1998. 5. Kevin L. Priddy and Paul E. Keller, "Artificial Neural Networks – An Introduction", SPIE Press, 2005.	
Course Outcome (CO): CO1: To learn the basic concepts of search based searching. CO2: To learn the probabilistic learning techniques and their applications. CO3: Well known artificial intelligence techniques implementation CO4: Understanding and creating an expert system	

Course Code : ECT212	**Course Credit : 3**
Course Name : Communication Systems	**L-T-P : 3-0-0**

Course Prerequisite: Basic electronics

Course Syllabus:

Basic history of electronic communication systems, Types of communication systems, Analog vs. digital communication, Issues and design aspects of communication systems. Emerging communication technologies. **[5 Lectures]**

Classification of signals and useful signal operations. concepts of signal-to-noise ratio, Frequency domain representation of signals using Fourier transform, Important properties of Fourier transform, rate of communication, randomness, redundancy, coding, signal transmission through a linear system, Ideal and practical filters, Energy and power of a signal, Energy and power spectral density, Basic concept of data communication. **[12 Lectures]**

Principle of modulation, Generation and demodulation of Amplitude modulated signal, DSB-FC, DSB-SC, SSB-SC, VSB-SC signals, channel bandwidth, Carrier acquisition, Super heterodyne AM receiver. Frequency division multiplexing. **[8 Lectures]**

Concept of Angle modulation (frequency modulation and phase modulation), FM transmitter and receivers, Interference and bandwidth considerations in angle modulated systems, Comparison of AM and FM. **[8 Lectures]**

Overview of Sampling theorem, Baseband digital modulation - Pulse analog modulation (Pulse Amplitude Modulation, Pulse Width Modulation, Pulse Position Modulation), Pulse Digital Modulation (Pulse Code Modulation); Digital communication system, Binary signaling scheme (Amplitude Shift Keying, Phase Shift Keying, Frequency Shift Keying). **[7 Lectures]**

Text/Reference Book(s):

1. B.P. Lathi, “Modern Digital & Analog Communications Systems”, Oxford University Press
2. Behrouz A. Forouzan "Data communication and Networking", Tata McGraw Hill, 2007
3. H. Taub, D.L. Schilling, "Principles of Communication Systems", Tata McGraw Hill, 2001
4. S. Haykin,"Communication Systems", John Wiley and Sons, 2001

Course Outcome (CO):

CO1: Able to understand the basic concepts of analog communication systems.

CO2: Able to understand various linear and nonlinear continuous modulation/demodulation techniques.

CO3: Able to evaluate the performance of the analog communication system in the presence of noise.

CO4: Able to analyze various analog pulse modulation and demodulation techniques.

Course Code : MAT202	**Course Credit : 3**
Course Name : Probability and Statistics	**L-T-P : 3-0-0**

Course Prerequisite: Basic mathematics

Course Syllabus:

Probability: Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes' Theorem and independence, problems. Random Variables: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev's inequality, problems **[7 Lectures]**

Special Distributions: Discrete uniform, binomial, geometric, negative binomial, hypergeometric, Poisson, continuous uniform, exponential, gamma, Weibull, Pareto, beta, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability and hazard rate, reliability of series and parallel systems, problems, Function of a random variable. **[8 Lectures]**

Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution, problems. Transformations: functions of random vectors, distributions of order statistics, distributions of sums of random variables, problems. **[5 Lectures]**

Sampling Distributions: The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions, problems. **[4 Lectures]**

Descriptive Statistics: Graphical representation, measures of locations and variability, Estimation: Unbiasedness, consistency, the method of moments and the method of maximum likelihood estimation, confidence intervals for parameters in one sample and two sample problems of normal populations, confidence intervals for proportions, problems. **[8 Lectures]**

Testing of Hypotheses: Null and alternative hypotheses, the critical and acceptance regions, two types of error, power of the test, the most powerful test and Neyman-Pearson Fundamental Lemma, tests for one sample and two sample problems for normal populations, tests for proportions, Chi square goodness of fit test and its applications, problems. **[8 Lectures]**

Text/Reference Book(s):

1. Athanasios Papoulis, S. Unnikrishna Pillai, "Probability, Random Variables and Stochastic Processes", Tata McGraw Hill, 4th Edition
2. Pradip Kumar Gosh, "Theory of Probability and Stochastic Processes", University Press
3. Sheldon M. Ross "Introduction to Probability and Statistics for Engineers and Scientists" Elsevier Third Edition Academic Press 2008.

Course Outcome (CO):

CO1: To get the foundations of probabilistic and statistical analysis used in varied engineering applications.

CO2: To make the probabilistic models for real life situations like disease modeling, climate prediction and computer networks etc.

CO3: To evaluate and apply moments & characteristic functions and understand the concept of inequalities and probabilistic limits.

Co4: Understand the concept of random processes and determine covariance and spectral density of stationary random processes.

Course Code : CSP202	**Course Credit : 1**
Course Name : Object Oriented System Design Lab	**L-T-P : 0-0-2**

Course Prerequisite: Elementary knowledge of programing language C

Course Syllabus:

Write a program to give the examples of operators- Increment and decrement, Bitwise Complement, Arithmetic, Relational, Conditional Operators. **[1 Lab]**

Write a program to give the example of control statements- If, Else-If, Statements, Switch case, For, While loops **[1 Lab]**

Operations with 1D array, 2D array and multi-dimensional arrays. **[1 Lab]**

To find the sum of command line arguments and count the invalid integers entered. **[1 Lab]**

Write a program to create a room class, the attributes of this class is Room no, room type, room area and AC machine. In this class the member functions are set data and display data. **[1 Lab]**

Write a program to create a class 'simple object'. Using constructor and destructor to display the message given by you. **[1 Lab]**

Write a program for call by value and for call by reference. **[1 Lab]**

Write a program to give the example for 'this' operator. And also use the 'this' keyword as a return statement. **[1 Lab]**

Write a program to demonstrate static variables, methods, and blocks. Write a program to demonstrate static variables, methods, and blocks. **[1 Lab]**

Write a program to create a package named my pack and import it in circle class. Write a program to create a package named pl, and implement this package in ex1 class. **[1 Lab]**

Create a class named as 'a' and create a sub class 'b'. Which extends from class 'a'. And use these classes in 'inherit' class. **[1 Lab]**

Write a program to get the input from the user and store it into file. Using Reader and Writer file. **[1 Lab]**

Text/Reference Book(s):

1. Core Java: Vol I Fundamentals, Cay S. Horstmann. Latest Edition, Publisher – Prentice Hall
2. Java: A Beginner's Guide, Eighth Edition by Schildt, Herbert.
3. Core Java Professional for Students by Harry H. Chaudhary, 2nd Edition

Course Outcome (CO):

CO1: Practice object-oriented programs and build java applications.
CO2: Implement java programs for establishing interfaces.
CO3: Implement sample programs for developing reusable software components.
CO4: Create database connectivity in java and implement GUI applications.

Course Code : CSP204	**Course Credit : 2**
Course Name : Design and Analysis of Algorithms Lab	**L-T-P : 0-1-2**

Course Prerequisite: Data Structures and Programming

Course Syllabus:

Divide and conquer: Quick sort, randomized quick sort, Merge sort, Strassen's matrix multiplication, Closest pair of points, Matrix chain multiplication **[3 Labs]**

Detecting cycle in a graph, Kruskal and Prim's minimum spanning tree, Dijkstra's shortest path algorithm, Bellman Ford algorithm **[3 Labs]**

Fractional Knapsack problem, 0-1 Knapsack problem **[2 Labs]**

String Matching Algorithms: Naïve method, KMP algorithm, Robin-Karp algorithm, Boyer Moore algorithm, Suffix array **[2 Labs]**

Backtracking: N-queen problem, Subset sum, Graph m-coloring problem **[2 Labs]**

Text/Reference Book(s):

1. Thomas H. Cormen, et al. Introduction to Algorithms, Latest Edition, MIT Press
2. NarasimhaKarumanchi, Data Structures and Algorithms Made Easy, Latest Edition, CareerMonk Publication
3. Jon Kleinberg and Eva Tardos, Algorithm Design, Latest Edition, Pearson
4. Robert Sedgewick and Kevin Wayne, Algorithms, Latest Edition, Addison Wesley

Course Outcome (CO): At the end of this course, each student should be able:

CO1: To implement various algorithmic paradigms such as incremental approach, divide-and- conquer and backtracking algorithms.
CO2: To implement greedy and dynamic programming to solve real life problems and analyze the efficiency.
CO3: To implement and analyze the solutions based on the randomized algorithms.
CO4: Analyze the performance of various graph algorithms to find the shortest route, minimum spanning tree and other operations.
CO5: To implement the approximation algorithm to solve NP-Hard problems.

Course Code : CSP206	**Course Credit : 1**
Course Name : Operating Systems Lab	**L-T-P : 0-0-2**

Course Prerequisite: C Programming

Course Syllabus:

Basics of Unix Commands. **[1 Lab]**

Process creation (fork, wait, exec, etc.) **[2 Labs]**

Implement Scheduling Algorithms (round robin, FCFS, priority, SJF etc) **[2 Labs]**

Implement Semaphores **[1 Lab]**

Implement Banker's Algorithm for Deadlock Avoidance **[1 Lab]**

Implement an Algorithm for Deadlock Detection **[1 Lab]**

Implement concepts of memory management **[1 Lab]**

Implement all page replacement algorithms a) FIFO b) LRU c) LFU **[2 Lab]**

Implementation concepts of disk scheduling **[1 Lab]**

Text/Reference Book(s):

1. Silberschatz and Galvin: Operating System Principles, Wiley India Pvt. Ltd.
2. Tanenbaum: Modern Operating System, Prentice Hall.
3. OS – Three Easy Step by Remzi (available free online)
4. DM Dhamdhere: Operating Systems – A Concepts Based Approach, Tata McGraw Hill
5. Charles Crowly: Operating System A Design Oriented Approach, Tata McGraw Hill.

Course Outcome (CO):

CO1: At the end of the course, the student should be able to implement deadlock avoidance, and Detection Algorithms

CO2: Students will be able to compare the performance of various CPU Scheduling Algorithm
CO3: Students will be able to critically analyze the performance of the various page replacement algorithms

CO4:Students will be able to create processes and implement IPC

Course Code : CSP210	**Course Credit : 1**
Course Name : Foundation of Data Science Lab	**L-T-P : 0-0-2**

Course Prerequisite: Basic knowledge of programming and mathematics

Course Syllabus: [12 Labs]

Implementation in Python: Environment set-up, Jupyter overview, Python Numpy,
Computation on NumPy Arrays
Basics of NumPy-Computation on NumPy-Aggregations-Computation on Arrays-Comparisons, Masks and Boolean Arrays-Fancy Indexing-Sorting Arrays-Structured Data: NumPy's Structured Array
Data Manipulation with Pandas, Matplotlib, Scikit tool
Data processing, Implement different techniques to analyze dataset. Data Indexing and Selection
Operations on Data, Handling Missing Data
Vectorising different operations on Data. High-Performance Pandas: eval() and query().
Implement and analysis important statistical methods on a given data used in data science using python
Basic functions of matplotlib-Simple Line Plot, Scatter Plot-Density and Contour Plots
Histograms, Binnings and Density-Customizing Plot Legends, Colour Bars-Three-Dimensional Plotting in Matplotlib
Data visualization: Tableau. Creating charts, Mapping data in Tableau

Text/Reference Books:

1. Jake VanderPlas ,Python Data Science Handbook - Essential Tools for Working with Data, O'Reily Media,Inc, 2016
2. Zhang.Y ,An Introduction to Python and Computer Programming, Springer Publications,2016
3. Joel Grus ,Data Science from Scratch First Principles with Python, O'Reilly Media,2016
4. T.R.Padmanabhan, Programming with Python,Springer Publications,2016

Course Outcome (CO):

CO1: To learn the basic concepts of data science.
CO2: To learn the data analysis techniques and their applications.
CO3: Understanding and creating a complete analysis system

Course Code : OTP202	**Course Credit : 2**
Course Name : Entrepreneurship and Business Incubation	**L-T-P : 0-1-2**

Course Prerequisite: NIL

Course Syllabus:

Introduction to entrepreneurship and intrapreneurship
Success stories of 6-8 entrepreneurs.
Identification of critical success factors to be a successful entrepreneur
Entrepreneurship as a career option; profile of a successful entrepreneur; process of becoming an entrepreneur; personal assessment and understanding of self.
Business plan preparation; constituents of a business plan. Statuary requirements for becoming an entrepreneur. Governmental rules and regulations. Development of a Business Idea.
Start-Ups and Micro Businesses, Self-Employment. Motivations and the Process of Self-Assessment, Risks and Rewards. Dealing with business failure.
Overcoming Social, Economic and Cultural barriers to Entrepreneurships; Process of Idea Generation, Invention, Discovery, Innovation and Expansion.
Franchising and Business Partnerships; Working in teams, finding your co-founder, team dynamics. Negotiation skills; Types of Legal Entities, Incorporation and Exit.
Familiarizing with the Companies Act and other Legal Aspects of running a business. Taxes and Exemptions relevant to StartUps; Product and Service Design – Creative Problem Solving and Process of Solutions design; Opportunity Identification, Estimation and Evaluation.
Mini Project:
Student should prepare a business plan in a group and register themselves as a startup in an incubation center.
(Finalization of business plan; floating their own company; start prototype development; customer identification; market survey; demand analysis; start the enterprise after arranging funds/ finances from venture capitalists/ angle investors/ govt. agencies etc.)

Text/Reference Book(s):

1. Stay Hungry Stay foolish: Rashmi Bansal; CIIIE, IIM Ahmadabad, 2008.

2. Arise, Awake: The Inspiring Stories of Young Entrepreneurs Who Graduated From College Into A Business of Their Own, Westland Books Private Limited (20 January 2015)

3. Moodi, Y. (2012). Game changers: 20 extraordinary success stories of entrepreneurs from IIT Kharagpur. Noida: Random House.

4. Bansal, R. (2013). Follow every rainbow: the inspiring stories of 25 women entrepreneurs whose gentle touch created strong business. Chennai: Westland Ltd

Course Outcome (CO):

CO1: Understand the fundamentals of Entrepreneurship.

CO2: Understand the entrepreneurial behaviour.

CO3: Business Creation and StartUp Development.

CO4: Implementation of Business Plan

B.Tech. CSE 3rd Year

Semester - 5						
S. No.	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	CST301	Theory of Computation	3	1	0	4
2	CST303	Computer Networks	3	0	0	3
3	CST305	Cryptography and Cyber Security	3	0	0	3
4	CST307	Machine Learning	3	0	0	3
5	CST309	Elective - 5.1	3	0	0	3
6	CST311	Elective - 5.2	3	0	0	3
7	CST3XX	Specialization Core Course (Optional)	3	0	0	3
		Labs				
8	CSP303	Computer Networks Lab	0	0	2	1
9	CSP307	Machine Learning Lab	0	0	2	1
10	CSP309	Elective - 5.1 Lab	0	0	2	1
11	CSD301	Survey of Technical Articles	0	2	2	3
Total			**18**	**3**	**8**	**25**

Course Code : CST301	**Course Credit : 4**
Course Name : Theory of Computation	**L-T-P : 3-1-0**

Course Prerequisite: Basic Mathematics, Discrete mathematical structures, basic computer concepts

Course Syllabus:
Introduction to automata, Mathematical induction - Diagonalization principle - Pigeonhole principle - Functions - Primitive recursive and partial recursive functions - Computable and non-computable functions, Formal representation of languages - Chomsky classification. **[4 Lectures]**
Introduction to Automata Theory: Definition of Automation, Finite automata, Language acceptability by finite automata, Deterministic and nondeterministic finite automata, Regular expressions. **[5 Lectures]**
Finite automation with ϵ transitions, Conversion of NFA to DFA, Minimization of DFA, DFA to Regular expressions conversion, Pumping lemma for regular languages, Applications of finite automata, NFA with o/p (Moore / Mealy). **[5 Lectures]**
Context-Free Grammar: Simplification of CFG, Normal forms: Chomsky Normal form and Greibach Normal form, pumping lemma for Context-free languages **[4 Lectures]**
Applications of PDA, Pushdown Automata, Formal definition, Language acceptability by PDA through the empty stack and final state, Deterministic and nondeterministic PDA, designing of PDA. **[5 Lectures]**
Turing Machines: Formal definition, Language acceptability by TM, TM as acceptors, Transducers, designing of TM, Two way infinite TM, Multi tape TM, Universal Turing Machines, Church's Thesis-Godelization, Time complexity of TM, Halting Problem **[9 Lectures]**
Linear Bounded Automata, Complexity classes tractable problems, Class P, P Complete-Reduction problem, Context grammar nonempty, Intractable problems- Class NP – NP-Complete- Cook's theorem, Reduction problems SAT-Clique-Hamiltonian-TSP-Vertex Cover-NP Hard problems. Introduction to Timed-automata **[8 Lectures]**

Text/Reference Books:
1. Hopcroft, J, E; Motwani, J; Ullman, J, D (2002). Introduction to Automata Theory, Languages and Computation.Pearson Education.
2. Mishra, K, L, P; Chandrasekaran, N (2009). Theory of Computer Science.PHI. 3. Michael Sipser – Theory of Computation
3. Hopcroft, J, E; Motwani, J; Ullman, J, D (2002). Introduction to Automata Theory, Languages, and Computation. Pearson Education.

Course Outcome (CO):
CO1: Design and verify abstract models of computation for formal languages.
CO2: Design and classify any given language and apply formal mathematical properties of language, grammar and automata.
CO3: Judge the decidability of a given problem by constructing a Turing Machine.
CO4: Define various categories of automata (deterministic and nondeterministic finite state automata, and variants of Turing machines).
CO5: Define the various categories of languages and grammars in the Chomsky hierarchy

Course Code : CST303	**Course Credit : 3**
Course Name : Computer Networks	**L-T-P : 3-0-0**

Course Prerequisite: Basic understanding of computer systems

Course Syllabus:
Introduction to Protocol Layering, OSI Reference Model and TCP/IP Protocol Stack. Networking core – packet switching, circuit switching, nodal delay (processing delay, queuing delay, transmission delay, propagation delay). Introduction to interconnecting networking devices. Application layer, DNS, HTTP, SMTP, etc. **[8 Lectures]**
Transport layer, UDP, TCP, Sliding Window, sender and receiver window size, silly window syndrome, Nagle's Algorithm, packet loss detection, retransmission, RTT, RTO, Karn/Patridge Algorithm, sequence number wrap around, bandwidth delay product. **[7 Lectures]**
Resource allocation classification, best effort service v/s QoS model, Fairness, fairness index, Queuing disciplines (FIFO, FQ, WFQ). Congestion Control: AIMD, Slow Start, Fast Retransmit and Recovery, Congestion Avoidance, TCP variants (Tahoe, Reno, Vegas). **[7 Lectures]**
Network layer, IP addressing scheme, private addresses, static and dynamic assignment (DHCP), sub-netting, CIDR. Routing, Scale, avoiding loops/failures, Distance Vector routing – RIP (15 hops), IGRP (255 hops). Link State Routing (OSPF). Brief introduction to multi-cast routing, MPLS, QoS,IPv6, etc. **[12 Lectures]**
Link layer (OSI – physical layer, MAC, LLC), Physical layer – bit stream, cables, hubs, repeaters, switches. Error detection – parity, CRC, checksum. MAC, Ethernet, CSMA/CD, ARP, ICMP, ARQ, bridging concepts. Introduction to Mobile Networks, Wi-Fi and Mobile IP. **[6 Lectures]**

Text/Reference Books:
1. Computer Networks A Systems Approach by Peterson and Davie
2. Computer Networking A Top-Down Approach by Kurose and Ross.
3. An Engineering Approach to Computer Networking by S. Keshav.

Course Outcome (CO):
CO1: Students will understand fundamental underlying principles of computer networking
CO2: Students will understand details and functionality of layered network architecture.
CO3: Students will apply mathematical foundations to solve computational problems in computer networking
CO4: Students will analyze performance of various communication protocols.
CO5: Students will compare routing algorithms

Course Code : CST305	**Course Credit : 3**
Course Name : Cryptography and Cyber Security	**L-T-P : 3-0-0**

Course Prerequisite: Basic concepts of Computer Science and Mathematics

Course Syllabus:
Course Introduction and Terminology, Security Trends, Security Attacks, Security Mechanism. Conventional Cryptography: Definitions, Classical Encryption Techniques i.e. Substitution Techniques, Transposition Techniques, Rotor Machines and Steganography. **[8 Lectures]**
Finite Fields: Groups, Rings, Fields, Modulo Arithmetic GCD (Euclids Algorithm); Symmetric Cryptography: DES, AES and other Symmetric Cryptography. **[9 Lectures]**
Asymmetric Cryptography: Number Theory, Public Key Cryptography: RSA, Elgamal, and Elliptic Curve Cryptography, Key Management. **[8 Lectures]**
Authentication: Message Authentications and Hash Functions, Hash Algorithms, Digital Signatures and Authentication Protocols. **[5 Lectures]**
History of Internet, Cyber Crime, Information Security, Computer Ethics and Security Policies, Securing web browser, Antivirus, Email security, secure password and wi-fi security, Smartphone Security, Firewall, Defensive Programming, Counter Cyber Security Initiatives in India **[10 Lectures]**

Text/Reference Books:
1. W Stallings, “Cryptography and Network Security: Principles and Practice, 5/e”, Prentice Hall.
2. B A Forouzan, “Cryptograpgy and Network Security”, Tata McGraw Hill, 2007.
3. Singer PW, Friedman A. “Cybersecurity: What everyone needs to know”, Oxford University Press India, 2014.
4. C Kaufman, R Perlman, M Speciner, Network Security, 2/e” , Pearson Education, 2006.
5. Alfred J. Menezes, et al, Handbook of Applied Cryptography, CRC Press

Course Outcome (CO):
CO1: Identify information security goals, classical encryption techniques and acquire fundamental knowledge on the concepts of finite fields and number theory.
CO2: Understand, compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication
CO3: Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes
CO4: Apply different digital signature algorithms to achieve authentication and create secure applications.
CO5: Analyze and resolve security issues in networks and computer systems to secure a digital infrastructure.
CO6: Develop policies and procedures to manage security risks.

Course Code : CST307	**Course Credit : 3**
Course Name : Machine Learning	**L-T-P : 3-0-0**

Course Prerequisite: Basics of Artificial Intelligence

Course Syllabus:
Introduction to supervised, unsupervised and reinforcement learning, Nearest Neighbours Algorithm – KNN, K-d Tree. Decision Tree – Decision tree in classification, Gini impurity, Entropy and Information gain, ID3 algorithm, C4.5 algorithms, CART algorithm, Decision tree in regression, Problem of Underfitting and overfitting, Decision tree pruning **[10 Lectures]**
Support Vector Machine – Basics of SVM, Mathematical modelling for linear classification, Maximum margin, Non-linear classification, Kernel functions. Soft-max classification – Softmax function, Training Softmax classifier, Role of Softmax in Neural Networks **[8 Lectures]**
Random Forest Classifier – Bagging, Bootstrap aggregating, Random sample with replacement, Cross-validation, From bagging to random forests, Types of Random Forest models: Random Forest Prediction for a classification problem, Random Forest Prediction for a regression problem, Features and Advantages of Random Forest **[7 Lectures]**
Semi-supervised learning – how to handle the combination of labelled and unlabeled data, Assumptions, Self-training, Generative methods, Graph-based methods, Adversarial training, Applications **[7 Lectures]**
Unsupervised learning, K-means clustering: partitions data into k distinct clusters based on distance to the centroid of a cluster, Dimension reduction, Matrix Factorization – Covariance, Principal component analysis, Independent component analysis, Introduction to Reinforcement Learning, Q-Learning, State-Action-Reward-State-Action (SARSA) **[8 Lectures]**

Text/Reference Books:
1. T. M. Mitchell, “Machine Learning”, McGraw Hill.
2. S. Marsland, “Machine Learning: An Algorithmic Perspective”, CRC Press.
3. K. P. Murphy, “Machine Learning: A probabilistic perspective”, MIT Press.
4. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer.
5. D. Barber, “Bayesian Reasoning and Machine Learning”, Cambridge University Press.
6. M. Mohri, A. Rostamizadeh, and A. Talwalkar, “Foundations of Machine Learning”, MIT Press.

Course Outcome (CO):
CO1: To learn the basic concepts of supervised and unsupervised learning
CO2: To learn the regression techniques and data interpretation
CO3: Dimensionality reduction and reducing the complexity of the solution
CO4: Understanding and implementing an intelligent system

Course Code : CSP303	**Course Credit : 1**
Course Name : Computer Networks Lab	**L-T-P : 0-0-2**

Course Prerequisite: Basic knowledge of programming

Course Syllabus: [12 Labs]

Running and using services/commands like ping, trace route, nslookup, arp, telnet, ftp, etc.
Socket programming using UDP and TCP (e.g., simple DNS, data & time client/server, echo client/server, iterative & concurrent servers)
Connecting hosts in a LAN
Implementation of DLL framing schemes viz bit stuffing / character stuffing
Implementation of Dijkstra's Algorithm to compute shortest path having given a path.
Obtaining a routing table while each node uses Distance Vector routing algorithm, given a subnet and weights quantifying delay between nodes of a subnet.
Implementation of checksum to detect errors during transmission
Implementation of CRC given a generator polynomial
Implementation of sliding window protocol
Simulating wired/wireless network functions and protocols using NS2/NS3

Text/Reference Books:

1. Computer Networks A Systems Approach by Peterson and Davie
2. Computer Networking A Top-Down Approach by Kurose and Ross.
3. NS2 manual

Course Outcome (CO):

CO1: Students will understand the concepts of networking thoroughly.
CO2: Students will be able to analyse the performance of the network.
CO3: Students will be able to implement networking protocols.
CO4: Students will learn to connect client and server through socket creation.

Course Code : CSP307	**Course Credit : 1**
Course Name : Machine Learning Lab	**L-T-P : 0-0-2**

Course Prerequisite: Basics of Artificial Intelligence and Programming

Course Syllabus: [12 Labs]
Implementation of Linear regression, Linear regression with L1 and L2 regularization
Implementation of Logistic Regression
Implementation of K-Nearest Neighbours
Decision Tree using different algorithms (ID3, C4.5, CART)
Linear classification using support vector machine, Non-linear classification using SVM with the help of kernel functions
Implementation of random forest classifier and its basics
Implementation of K-means clustering
Implementation of principal component analysis and Independent component analysis
Basic reinforcement learning, Q-Learning, Implementation of State-Action-Reward-State-Action (SARSA)

Text/Reference Books:
1. T. M. Mitchell, “Machine Learning”, McGraw Hill.
2. S. Marsland, “Machine Learning: An Algorithmic Perspective”, CRC Press.
3. K. P. Murphy, “Machine Learning: A probabilistic perspective”, MIT Press.
4. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer.
5. D. Barber, “Bayesian Reasoning and Machine Learning”, Cambridge University Press.
6. M. Mohri, A. Rostamizadeh, and A. Talwalkar, “Foundations of Machine Learning”, MIT Press.

Course Outcome (CO):
CO1: To learn the basic concepts of supervised and unsupervised learning
CO2: To learn the regression techniques and data interpretation
CO3: Dimensionality reduction and reducing the complexity of the solution
CO4: Understanding and implementing an intelligent system

Course Code : CSD301	**Course Credit : 3**
Course Name : Survey of Technical Articles	**L-T-P : 0-2-2**
Course Prerequisite: Basic knowledge of programming	
Course Objectives: 1. Understanding and evaluating the existing technologies in a domain 2. To prepare the students to take and work on major projects 3. To understand the general description, terminologies and an idea of research work	
Text/Reference Books: Relevant research papers.	
Course Outcome (CO): CO1: To prepare the students to assess the available solutions and compare CO2: To interact with the people professionally to collect data and other things CO3: To work in a team at any level of the project CO4: To write the report of the developed system	

B.Tech CSE 3rd Year

Semester - 6						
S. No.	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	CST302	Compiler Design	3	1	0	4
2	CST304	Elective - 6.1	3	0	0	3
3	CST306	Elective - 6.2	3	0	0	3
4	HST302	Professional Development	2	0	0	2
5	CST3XX	Specialization Core Course (Optional)	3	0	0	3
6	CST3XX	Specialization Core Course (Optional)	3	0	0	3
		Labs				
7	CSD302	Project – I	0	2	6	5
8	CSP302	Compiler Design Lab	0	0	2	1
9	CSP304	Elective - 6.1 Lab	0	0	2	1
10	HSP302	Professional Development Lab	0	0	2	1
11	CSP3XX	Specialization Core Course Lab (Optional)	0	0	2	1
Total			**11**	**3**	**12**	**20**

Course Code : CST302	**Course Credit : 4**
Course Name : Compiler Design	**L-T-P : 3-1-0**

Course Prerequisite: Theory of Automata

Course Syllabus:
Overview of Compilation: Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator. **[8 Lectures]**
Top down Parsing: Context free grammars, Top down parsing – Backtracking, LL (1), recursive descent parsing, Predictive parsing, Preprocessing steps required for predictive parsing. Bottom up parsing: Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing, handling ambiguous grammar, YACC – automatic parser generator. **[9 Lectures]**
Semantic analysis: Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes. Attributed grammars, Syntax directed translation, Conversion of popular Programming languages language Constructs into Intermediate code forms, Type checker. **[9 Lectures]**
Symbol Tables: Symbol table format, organization for block structures languages, hashing, tree structures representation of scope information. Block structures and non-block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.
Code optimization: Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation. **[8 Lectures]**
Object code generation: Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation. **[6 Lectures]**

Text/Reference Books:
1. Principles of compiler design -A.V. Aho .J.D.Ullman; Pearson Education.
2. Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University Press.
3. lex&yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
4. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, Wiley dreamtech.
5. Engineering a Compiler-Cooper & Linda, Elsevier.
6. Compiler Construction, Louden, Thomson.

Course Outcome (CO):
CO1: To give students hands-on experience with crafting a simple compiler.
CO2: To understand the language translation and compiler design and to develop an awareness of the function and complexity of modern compilers.
CO3: To implement lexical analyzer using Lex tool & Syntax Analyzer or parser using YACC Tool.
CO4: To implement NFA and DFA from a given regular expression.
CO5: To implement the front end of the compiler by means of generating Intermediate codes and code optimization techniques.

Course Code : HST302	**Course Credit : 2**
Course Name : Professional Development	**L-T-P : 2-0-0**
Course Prerequisite: NIL	

Course Syllabus:
Basics of Professional Communication: Purpose; Audience; Clarity and Precision; Cohesion and Coherence; Tone and style; Using visuals; Ethical issues.
Writing a Statement of Purpose: Introduction and Importance of the SOP, Essential Components, Style, Errors to be avoided, Drafting an effective SOP **[6 Lectures]**
Personal SWOT Analysis for Professional Development: Introduction and Importance of Personal SWOT Analysis; Identifying one's Strengths, Weaknesses, Opportunities and Threats; Using the Findings to Develop a Short-term and Long-Term Personal Development Plan
Preparing the Cover Letter and Resume: Introduction, Significance and Basic Components of a Cover letter and a Resume; Common errors; Drafting a Good Resume **[5 Lectures]**
Gearing up for the Interview: Significance and types of Interviews (Face to face, video telephonic,), Interview preparation: company background, refreshing one's theoretical knowledge.
Interview Skills: Personal introduction; Dress code and Personal grooming; Punctuality and Listening skills; Interview Procedure; Important questions; Situation, Task, Approach and Response (STAR Approach) for acing an interview; Errors to be avoided. **[5 Lectures]**
Group Discussion Skills: Introduction and significance; Procedure of conducting GD; Importance of Preparation and Practice; Attitude and Etiquette; Body language during a GD.
Team Skills: Effective Listening; Brainstorming; Negotiation; Communicating in Teams.
Presentation Skills: Introduction and Significance; Planning and Preparing Presentations; Presentation Strategies; Using technology effectively; Handling questions **[6 Lectures]**
Report-writing: Introduction & Importance, Basic features & components, Types, Structure, Drafting the Report, Using visual elements.
Drafting Executive Summaries: Importance and basic elements; Format and Style.
Writing Emails: Drafting Professional Electronic Mails. Writing Positive, Negative. Persuasive messages. Sending notices, agenda and minutes of a meeting through mails **[5 Lectures]**

Text/Reference Books:
1. Wentz, Fredrick H. *Soft skills Training*. Amazon Digital Services, 2012
2. Mitra, Barun K. *Personality Development and Soft Skills*. Oxford University Press, 2016.
3. Sharma, R.C. & Krishna Mohan. *Business Correspondence and Report Writing*. Tata McGraw Hill, 2020.
4. Desarda, Sheetal. *Master the Group Discussion and Personal Interview*. Notion Press, 2015.
5. Rizvi, M. Ashraf. *Effective Technical Communication,* McGraw Hill, 2009.

Course Outcome (CO):
CO1: Understand the basics of professional communication.
CO2: Learn the technique of identifying one’s professional talents and weaknesses.
CO3: Understand the steps to developing the professional Cover Letter and Resume.
CO4: Learn the techniques to prepare for employment and internship Interviews, and participate in Group Discussions, prepare and deliver effective presentations.
CO5: Learn the essential components and features of reports, executive summaries, and professional emails, and steps to drafting them.

Course Code : CSD302	**Course Credit : 5**
Course Name : Project – I	**L-T-P : 0-2-6**

Course Prerequisite: Basic knowledge of programming

Course Objective:
1. Experiential learning through hands-on experiments on cutting edge technologies
2. To prepare the students to tackle real life technical challenges efficiently by providing cheap and feasible solutions
3. To prepare competitive industry ready students

Text/Reference Books:
Relevant research papers

Course Outcome (CO):
CO1. To understand the problem in engineering point of view.
CO2. To design the system and analyse mathematical modelling of the system
CO3. To work individually and in team at any level of the project
CO4. To interact and troubleshoot
CO5. To write the report of the developed system

Course Code : CSP302	**Course Credit : 1**
Course Name : Compiler Design Lab	**L-T-P : 0-0-2**
Course Prerequisite: NIL	

Course Syllabus: [12 Labs]
Introduction to compilers, translators, and interpreters, compilation process. Compare two complier front ends - GCC and Clang
Design and implement a lexical analyzer for given language using C
Implementation of Lexical Analyzer using Lex Tool
To write a Yacc program to valid arithmetic expression using Yacc
Implementation of Calculator Using Lex & Yacc
Lexical Analysis, Syntax Analysis – create parsers using Lex and Yacc (Bison).
Use GCC to understand code optimization: Basic blocks, Control Flow Graphs, Global data flow analysis.
Use GCC for implementing Loop optimization
Understanding of code generation: Compilation of expression and control structures

Text/Reference Books:
1. Compilers: Principles, Techniques and Tools, by Alfred V. Aho , Monika, Ravi Sethi , D. Jeffrey Ullman
2. Compilers Principles and Practice, D M Dhamdhere

Course Outcome (CO):
CO1: Hands-on to create a basic compiler with basic functionalities
CO2: To implement the different phases of compiler.
CO3: To implement and test simple optimization techniques.

Course Code : HSP302	**Course Credit : 1**
Course Name : Professional Development Lab	**L-T-P : 0-0-2**

Course Prerequisite: NIL

Course Syllabus: [12 Labs]

Exercises based on Basics of Professional Communication: Purpose; Audience; Clarity & Precision; Cohesion & Coherence; Tone & Style; Using Visuals; Ethical issues [2 Labs]

Preparing the Cover Letter and Resume: Drafting an Effective Cover Letter and Resume; Common errors to be avoided [2 Labs]

Gearing up for the Interview - Preparing for different types of Interviews (Face to face, telephonic, video), Body language and personal etiquettes.

Interview Skills: Personal introduction; Honing Listening skills; Responses to Important questions; Applying Situation, Task, Approach and Response (STAR Approach) for acing an interview; Errors to be avoided. Mock interview drills [3 Labs]

Group Discussion Skills: Preparation and Practice; Focus on Attitude, Group dynamics and Body language during a GD. Group Discussions for problem solving and decision-making.

Presentation Skills: Practice of Planning and Preparing Presentations; Presentation Strategies; Using technology effectively; Handling questions.

Team Skills: Practice of Brainstorming; Negotiation; Communicating in Teams; Effective Listening; Team-building activities [3 Labs]

Report-writing: Drafting different types of Reports, Using visual elements.

Executive Summaries: Preparing Executive Summaries of documents

Professional Emails: Drafting professional Emails. Writing Positive, Negative, Persuasive messages. Sending notices, agenda & minutes of a meeting through mails. [2 Labs]

Text/Reference Books:

1. Raman, Meenakshi, Sangeeta Sharma. *Professional Communication*. Oxford University Press, 2018
2. Mitra, Barun K. *Personality Development and Soft Skills*. Oxford University Press, 2016.
3. Sharma, RC & Krishna Mohan. *Business Correspondence and Report Writing*. Tata McGraw Hill, 2020.
4. Desarda, Sheetal. *Master the Group Discussion and Personal Interview*. Notion Press, 2015.
5. Rizvi, M. Ashraf, *Effective Technical Communication,* McGraw Hill, 2009.
6. Prince, Emma Sue. *Practical Business Communication* (Macmillan Study Skills), 2017.

Course Outcome (CO):

CO1: Understand the basics of professional communication.

CO2: Learn the technique of identifying one’s professional talents and weaknesses.

CO3: Understand the steps to developing the professional Cover Letter and Resume.

CO4: Learn the techniques to prepare for employment and internship Interviews, participate in Group Discussions, prepare effective Presentations

B.Tech CSE 4th Year

7th Semester On Campus with 8th On/Off Campus / 8th Semester On Campus with 7th Off Campus						
S. No	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	CST401 / CST402	Elective - 7.1 / Elective - 8.1	3	0	0	3
2	CST403 / CST404	Elective - 7.2 / Elective - 8.2	3	0	0	3
3	OTT401 / OTT402	Open Elective – 7.1 / Open Elective – 8.1	3	0	0	3
4	HST401 / HST402	Living Ethics	2	0	0	2
5	CST4XX / CST4XX	Specialization Elective Course (Optional)	3	0	0	3
6	CST4XX / CST4XX	Specialization Elective Course (Optional)	3	0	0	3
		Labs				
4	CSP401 / CSP402	Elective - 7.1 Lab / Elective - 8.1 Lab	0	0	2	1
5	CSD401 / CSD402	Project - II	0	2	6	5
Total			**11**	**0**	**8**	**17**

7th Semester Off Campus / 8th Semester Off Campus						
S. No	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	OTD401	Internship	0	0	20	10
Total			**0**	**0**	**20**	**10**

8th Semester On Campus with 7th On Campus						
S. No	**Subject Code**	**Subject**	**Scheme**			
			L	**T**	**P**	**Credits**
1	CST402	Elective - 8.1	3	0	0	3
2	CST404	Elective - 8.2	3	0	0	3
3	OTT402	Open Elective - 8.1	3	0	0	3
		Labs				
5	CSP402	Elective - 8.1 Lab	0	0	2	1
Total			**9**	**0**	**2**	**10**

Course Code : HST401 / HST402	**Course Credit : 2**
Course Name : Living Ethics	**L-T-P : 2-0-0**

Course Prerequisite: NIL

Course Syllabus:
Introduction to Ethics- What Does It Mean to Be "Ethical"? The rising urgency for ethics in a pluralistic society, Sustainable Development and the Need for Ethics in Action **[4 Lectures]**
Professional Ethics & Personal Value- Dealing with ethical dilemmas at workplace, Leadership and Ethics, Ethics and Responsibility for all stakeholders at workplace, Breaking the glass ceiling- Gender and Ethics at workplace **[7 Lectures]**
Academic Ethics- From working 'on' the community to working 'with' the community, Ethical consideration in research, Dealing with the stakeholders with Ethics, Ethics in publication **[8 Lectures]**
Ethics in the age of social media- Discussing social media use, which takes the form of hate speech, increased surveillance, lack of anonymity and questionable use of our data. What ethical considerations do we need to follow to use social media spaces effectively, in a way that is safe and productive for all? **[8 Lectures]**

Text/Reference Books:

1. Adams, R. B., & Funk, P. (2012). Beyond the glass ceiling: Does gender matter? *Management science*, *58*(2), 219-235.
2. Biedenweg, K., Monroe, M. C., & Oxarart, A. (2013). The importance of teaching ethics of sustainability. *International Journal of Sustainability in Higher Education*.
3. Hoke, T. (2012). A question of ethics: The importance of understanding engineering ethics. *Civil Engineering Magazine Archive*, *82*(5), 40-41.
4. Iphofen, R., & Tolich, M. (Eds.). (2018). *The SAGE handbook of qualitative research ethics*. Sage.
5. Macfarlane, B., Zhang, J., & Pun, A. (2014). Academic integrity: a review of the literature. *Studies in Higher Education*, *39*(2), 339-358.
6. Resnik, D. B. (2015, December). What is ethics in research & why is it important. In *ideas*.
7. Singer, P. (2011). *Practical ethics*. Cambridge university press.
8. Wellman, M. L., Stoldt, R., Tully, M., & Ekdale, B. (2020). Ethics of authenticity: social media influencers and the production of sponsored content. *Journal of Media Ethics*, *35*(2), 68-82.

Course Outcome (CO):
CO1: Understand the fundamentals of living ethics
CO2: Learn the central issues of living ethics and also considers how these can be applied to several contemporary issues.
CO3: Understand interdisciplinary perspectives and thematic issues in the fields of ethics and enable learners to reflect on major ethical dilemmas arising in everyday life.

Course Code : CSD401 / CSD402	**Course Credit : 5**
Course Name : Project - II	**L-T-P : 0-2-6**
Course Prerequisite: Basic knowledge of programming	
Course Objective: 1. Experiential learning through hands-on experiments on cutting edge technologies 2. To prepare the students to tackle real life technical challenges efficiently by providing cheap and feasible solutions 3. To prepare competitive industry ready students	
Text/Reference Books: Relevant research papers	
Course Outcome (CO): CO1. To understand the problem in engineering point of view. CO2. To design the system and analyse mathematical modelling of the system CO3. To work individually and in team at any level of the project CO4. To interact and troubleshoot CO5. To write the report of the developed system	

Specialization Core Courses

Course Code : CST3XX	**Course Credit : 3**
Course Name : Linear Algebra and Mathematical Modeling	**L-T-P : 3-0-0**
Course Prerequisite: Engineering Math	

Course Syllabus:
Review of vector spaces over arbitrary fields and linear transformation. Characteristic and minimal polynomials. Diagonalization of linear transformations, the primary decomposition theorem, the rational and Jordan canonical forms and some applications. Linear functional and dual spaces. Bilinear, Quadratic and Hermitian forms. Best approximation, Cauchy –Schwarz inequality, structure theory for normal operators: adjoint, self-adjoint, normal, unity and positive definite operator and their properties **[10 Lectures]**
Mathematical modelling concepts: Concepts of mathematical modelling; open and closed systems; limitations of mathematical modelling; properties of mathematical modelling; needs and techniques used; areas of applications; discussion on non-uniqueness of models **[5 Lectures]**
Classification of Mathematical modelling: Classification of mathematical models in terms of areas of application; Classification in terms of the types of mathematics used: Graphical models, models using algebra, models using differential equations (ordinary and partial both); models using difference equations; models using calculus of variations and dynamic programming, etc. **[5 Lectures]**
Procedure and techniques of Mathematical modelling: Real problems, identification of parameters, significant parameters, parameters of importance, reduction of an open problem to a closed form, conversion of a real problem into a mathematical problem; identification of problem to be modelled; quest for a mathematical technique for solution; importance of numerical techniques; computer simulation; physical interpretation; illustrations **[8 Lectures]**
Mathematical models in different fields Classical and continuous models, Deterministic, probabilistic and stochastic models; Case studies in problems of physics, chemistry, engineering, biological sciences, genetics, economics, defence, meteorology, music, languages and literature, chaos, synchronization, sports etc. **[12 Lectures]**

Text/Reference Books:
1. Linear Algebra by Kenneth Hoffman, Ray Kunze, PHI learning
2. J. N. Kapoor, Mathematical Modelling, Wiley Eastern Limited.
3. J. N. Kapoor, Mathematical Modelling in biology and medicine, Affiliated East-West Press Pvt. Ltd.
4. Edward A. Bender., An Introduction to Mathematical Modelling.
5. C. Fowler, Mathematical Models in Applied Sciences, Cambridge University Press.

Course Outcome (CO):
CO1. Linear Algebra & Its Applications by Gilbert Strang
CO2. Linear Algebra ,Schum’s outline series.
CO3. Advanced Linear Algebra ,Steven Roman, Third edition, Springer.

Course Code : CST3XX	**Course Credit : 3**
Course Name : Reinforcement Learning	**L-T-P : 3-0-0**
Course Prerequisite: Basics of Machine Learning	

Course Syllabus:

Elementary Reinforcement Learning- Introduction and Characteristics of Reinforcement Learning (RL), Reward Hypothesis, Agent and Environment, Fully and Partially Observable Environment, Problems in Sequential Decision Making, RL Agent Policy, Value Function, Reward Prediction Model, RL Agent Taxonomy **[8 Lectures]**

Markov Process- Markov Decision Process (MDP), State-Value Function, Markov Reward Process (MRP), Bellman Equation, Bellman Expectation Equation, Optimal Value Function, Bellman Optimality Equation, Extensions to MDPs, Infinite and Continuous MDP, Partially Observable Markov Decision Process (POMDP), Ergodic MDP, Average Reward MDP **[8 Lectures]**

Iterative Policy Evaluation, Principle of Optimality, Synchronous and Asynchronous Dynamic Programming, Contraction Mapping, Planning by Dynamic Programming **[8 Lectures]**

Model Free Prediction and Control- Monte-Carlo Learning, Temporal-Difference (TD) Learning, Bootstrapping and Sampling, Forward and Backward TD (lambda), Model Free Control, On Policy Monte-Carlo Control, On Policy Temporal-Difference Learning, Off Policy Control, Q-Learning, Value Function Approximation, Batch Reinforcement Learning, Least Squares Prediction, Policy Gradient Methods, Integrating Learning and Planning **[10 Lectures]**

Exploration and Exploitation- Multi-Armed Bandits, Counting Regrets, Greedy Algorithm, Hoeffding's Inequality, Contextual Bandits, Case Study of Reinforcement Learning in Practices **[6 Lectures]**

Text/Reference Books:

R. S. Sutton and A. G. Barto, "An Introduction to Reinforcement Learning", 1998, MIT Press.
Szepesvari, "Algorithms for Reinforcement Learning", Morgan and Claypool, 2010

Course Outcome (CO):

CO1: Will be able to recognize the problems which can be solved using reinforcement learning
CO2: Understand and implement the basic concepts of neural network and reinforcement learning
CO3: Will be able to design an architecture to solve many real-world problems
CO4: Realize the power and importance of various reinforcement learning capabilities

Course Code : CST3XX	**Course Credit : 3**
Course Name : Deep Learning and Applications	**L-T-P : 3-0-0**

Course Prerequisite: Machine Learning

Course Syllabus:
Neural Networks Overview and its Representation, Neuron Model and Network Architecture, Transfer Function, Single layer and Multiple layers of Neurons **[6 Lectures]**
Role of Hidden layers, Computing a Neural Network's Output, Activation Functions, Derivative of Activation Function, Need of Non-linear Activation **[6 Lectures]**
Perceptron, Gradient Descent and its role in Neural Networks, Feedforward and Backpropagation, Perceptron Learning Rules **[8 Lectures]**
Deep Learning Models: Convolutional Neural Network, Recurrent Neural Network and LSTM **[8 Lectures]**
Deep Learning Libraries: Keras, PyTorch, and TensorFlow; Practical Training Issues **[6 Lectures]**
Case Study of Deep Learning Research **[6 Lectures]**

Text/Reference Books:
1. Martin T. Hagan, et al. "Neural Network Design", Latest Edition
2. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook", Springer
3. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press book
4. François Chollet, "Deep Learning with Python", Latest Edition
5. Aurelien Gero, "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques", O'Reilly

Course Outcome (CO):
CO1: Will be able to recognize the problems which can be solved using deep learning
CO2: Understand and implement the basic concepts of neural network and deep learning
CO3: Will be able to design an architecture to solve many real-world problems
CO4: Realize the power and importance of various neural networks and deep learning capabilities

List of Electives
B. Tech. CSE

Electives for 3rd Year

Internet of Things
Nature Inspired Computing
Information Theory and Coding
Information Retrieval
Quantum Computing
Principles of Programming Languages
Wireless Sensor Networks
Network Reliability & Performance Models
Malware Analysis and Detection
Random Variables and Stochastic Processes
Database Security & Privacy
Data Warehousing and Data Mining
Advanced Topics in Theory of Computation
Internet Security & Computer Forensics
Graph Theory and Algorithms
Digital Image Processing
Advanced Software Engineering
Specialization Electives

Electives for 4th Year

Neural Networks and Deep Learning
Blockchain Technology and Applications
Robotics and Intelligent Systems
Game Theory and Applications
Biometrics
Computer Vision and Pattern Recognition
Natural Language Processing
Cloud Computing
Advanced Cryptography Techniques
Cyber Forensics & Security
Parallel and Distributed Systems
High Performance Computing
Human Computer Interaction
Big Data Analytics
Mobile Computing
Pervasive Computing
Software Project Management
Multi Agent Systems
System-Level-Design and Modeling
Advanced Algorithms

Course Code : CST3XX	**Course Credit : 3**
Course Name : Cyber Physical Systems	**L-T-P : 3-0-0**

Course Prerequisite: Basic knowledge of programming

Course Syllabus:

Cyber-Physical Systems (CPS) in the real world, Industry 4.0, AutoSAR, IIOT implications, Continuous Dynamics, Feedback Control, Discrete Systems, Hybrid Systems, Composition of State Machines, Concurrent Models of Computation, Building Automation, Medical CPS and mathematical modeling **[7 Lectures]**

Design and Implementation: Sensors and Actuators, Embedded Processors, Memory Architectures, Input and Output Interface, Multitasking, Scheduling. **[5 Lectures]**

Dynamical Systems and Stability, Controller Design Techniques, Performance under Packet drop and Noise **[5 Lectures]**

Intelligent CPS: Safe Reinforcement Learning (Robot motion control, Autonomous Vehicle control), Gaussian Process Learning(Smart Grid Demand Response, Building Automation) **[6 Lectures]**

Analysis and Verification: Invariants and Temporal Logic, Equivalence and Refinement, Reachability Analysis, Model Checking, Timing Analysis **[5 Lectures]**

CPS Control: Event triggered Control, Receding Horizon Control, Anytime Control **[5 Lectures]**

Secure Deployment of CPS: Secure Task mapping and Partitioning, State estimation for attack detection, Automotive Case study : Vehicle ABS hacking, Power Distribution Case study : Attacks on SmartGrids **[7 Lectures]**

Text/Reference Books:

1. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, http://LeeSeshia.org, ISBN 978-1-312-42740-2, 2015.
2. Rajeev Alur. Principles of Cyber-Physical Systems. MIT Press. 2015.
3. K. J. Astrom and R. M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Prince- ton University Press, 2009. http://www.cds.caltech.edu/~murray/amwiki/index.php/Main_Page.
4. Relevant research papers.

Course Outcome (CO):

CO1: Implement the basic concepts of cyber physical systems

CO2: System modelling, real-time scheduling and real-time resource utilization.

CO3: Verify and validate a model mathematically.

Course Code : CSP3XX	**Course Credit : 1**
Course Name : Cyber Physical Systems Lab	**L-T-P : 0-0-2**

Course Prerequisite: Basic knowledge of programming

Course Syllabus: [12 Labs]
Dynamic Modeling -- Disease spreading models, Cruise Control, rocket and aircraft dynamics, Water & waste management, Agriculture, MPC [3 Labs]
Smart Energy System-- Smart Grid, Smart buildings, Smart Cities,EnergyPlus- HCV [3 Labs]
Medical CPS: Model verification using UPPAAL- Heart, Pacemaker, drugs and tissues model and verification [5 Labs]
CPS for Automotives- Application ---demo --Introduction to driverless car [1 Lab]

Text/Reference Books:
1. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, 2015.
2. Rajeev Alur. Principles of Cyber-Physical Systems. MIT Press. 2015.
3. K. J. Astrom and R. M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, 2009.
4. Relevant research papers.

Course Outcome (CO):
CO1: Implement the basic concepts of cyber physical systems
CO2: System modelling, real-time scheduling and real-time resource utilization.
CO3: Verify and validate a model mathematically.

Course Code : ECT3XX	**Course Credit : 3**
Course Name : Embedded Systems	**L-T-P : 3-0-0**

Course Prerequisite: Digital Logic Design and Synthesis, Computer Architecture and Organization, Algorithms, and OS

Course Syllabus:

Embedded Computing Requirements: Characteristics and applications of embedded systems, Components of Embedded Systems, challenges in Embedded System Design and design process, Formalism for system design. **[3 Lectures]**

Embedded processor technology: General-purpose processors, Single-purpose processors, Application-specific processors **[3 Lectures]**

IC technology: Full-custom/VLSI, Semi-custom ASIC, PLD. Design technology: Compilation/Synthesis, Libraries/IP, Test/Verification, Other productivity improvers. Software-Hardware co-design **[3 Lectures]**

Communication and Interfacing: Introduction, Timing diagrams. Hardware protocol basics: Concepts, Master - Servant, Control methods: strobe, handshake.

Interfacing with a general-purpose processor: I/O addressing: port and Bus- based I/O, concept, memory-mapped I/O, standard I/O, parallel I/O. Interrupts and DMA **[7 Lectures]**

Arbiter: Priority arbiter, Daisy-chain arbitration, Network-oriented arbitration methods, Multi-level bus architectures, Advance Communication principles and protocols: Parallel communication-PCI and ARM bus **[5 Lectures]**

Serial communication-I2C, USB, CAN and FireWire, Wireless Protocols- Bluetooth, IrDA, IEEE 802.11 **[3 Lectures]**

Embedded Software Analysis and Design: Software design pattern for Embedded Systems; Model programs – data flow graphs and control/data flow graphs; Assembly and linking; Compilation techniques, Analysis and optimization of execution time, energy, power and program size **[4 Lectures]**

Processor accelerators, accelerated system design. Networks - Distributed embedded architectures, networks for embedded systems, network-based design, Internet-enabled systems **[4 Lectures]**

Real-Time operating: - Multiple tasks and multiple processes, context switching, real-time scheduling policies - EDF, Rate Monotonic , inter-process communication mechanisms **[4 Lectures]**

Study RTOS: Case Study of VxWorks and Free RTOS. Different Layers of work -Device Drivers, HAL and Low-power RTOS designing. Burning of RTOS **[4 Lectures]**

Text/Reference Books:

1. Wayne Wolf, “Computers As Components - Principles of Embedded Computing System Design”. Morgan Kaufman Publishers, 2nd Edition
2. Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley, 3 rd Edition.
3. Steve Heath, “Embedded System Design”, Newnes, 3 rd Edition.
4. Steve Furber, “ARM System-on-Chip Architecture”, Pearson, 2014

Course Outcome (CO):

CO1: This course will introduce the fundamentals of Embedded System Design.
CO2: Students will be able to understand the communication protocols and TLM modeling.
CO3: This course introduces the detail of hardware-software codesign and prototyping of ES.
Students will be able to understand the fundamentals of RTOS and application writing on it.

Course Code : ECP3XX	**Course Credit : 1**
Course Name : Embedded Systems Lab	**L-T-P : 0-0-2**

Course Prerequisite: Digital Logic Design and Synthesis, Computer Architecture and Organization, Algorithms, and OS

Course Syllabus: [12 Labs]
Free RTOS: Installation of frame-work and study of different components. Task distribution, multi-threading and thread scheduling, blocking and nonblocking, memory management, interrupt handling and low-power support with real-time scheduling
Prototyping with ARM Kit, xilinx Platform studio with xiline kernel
Designing of Bus protocols using TLM (SystemC), arbiter, interface unit and others
IP designing using systemC / system Verilog

Text/Reference Books:
Getting started with the FreeRTOS kernel
Getting started with FreeRTOS Plus Libraries
Getting started with FreeRTOSIoT Libraries
System C source : https://www.accellera.org/downloads/standards/systemc
SystemC: From the Ground Up, Second Edition Authors: Black, D.C., Donovan, J., Bunton, B., Keist, A.

Course Outcome (CO):
CO1: Know what an embedded system is.
CO2: Basic understanding of General System Theory how this applies to embedded system engineers, and how this differs from the traditional mechanistic theory.
CO3: Understand the general process of embedded system development
CO4: Comprehend important embedded system terminology
CO5: Experience common aspects of embedded system development
CO6: Understanding of what an embedded system R&D project is, and the activities it involves

Course Code : CST3XX	**Course Credit : 3**
Course Name : Principles of Programming Languages	**L-T-P : 3-0-0**

Course Prerequisite: Basic Knowledge of Computer Programing languages like C, C++, Java

Course Syllabus:
Introduction: Preliminary Concepts: Reasons for studying, concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms– Imperative, Object Oriented, functional Programming , Logic Programming. **[4 Lectures]**
Programming Language Implementation – Compilation and Virtual Machines, programming environments. Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features. **[6 Lectures]**
Procedure based languages: General features, Data types, Abstract Data Types (ADT), Structuring, Syntax, Semantics, RAM model of computation, Example: C language **[5 Lectures]**
Object based languages: Concepts of objects, Class vs ADT, control structures, methods, General features-inheritance, polymorphism, derived classes & information hiding, Example: C++ and Java, Difference with C. **[4 Lectures]**
Concurrent programming languages: Concurrency structure for message passing, loosely coupled system, shared memory, PRAM, monitor, semaphore, Example: Java RMI, Parallel Java, Parallel C. Declarative languages **[5 Lectures]**
Logic programming: Predicate calculus- Logical operators, Propositional forms, Rules of inference, Logical equivalence, Quantification, Well-formed formula, Disproofs; Prolog- Syntax, Lists, Operators and arithmetic, Control, i/o, data structures **[8 Lectures]**
Functional programming: Lambda calculus and computability; Lisp- Control constructs, List processing, Files and i/o, Generic functions, Objects, Exceptions **[8 Lectures]**

Text/Reference Books:
1. Concepts of Programming Languages by Robert W. Sebesta, Pearson Education.
2. Programming Languages: Concepts and Constructs by Ravi Sethi, Pearson Education.
3. Benjamin C Pierce: types of programming Languages. MIT Press 2002.
4. Programming Language Concepts by Carlo Ghezzi and Mehdi Jazayeri, John Wiley & Sons.
5. Programming Languages: Paradigm and Practices by Doris Appleby and J. J. Vandekopple, McGraw Hill.

Course Outcome (CO):
CO1: Able to understand the fundamental concepts of most programming languages and the trade off between language design and implementation.
CO2: Able to compare programming languages, assess programming languages critically and scientifically
CO3: Able to understand the use of formal description for a programming language and the essence of program execution by evaluators: interpreter, compiler.
CO4: Able to understand different programming paradigms: analyze the principles of imperative, object oriented, functional and logic programming.

Course Code : CST4XX	**Course Credit : 3**
Course Name : Digital System Synthesis	**L-T-P : 3-0-0**

Course Prerequisite: Digital Logic Design, Computer Architecture and Organization and Algorithms

Course Syllabus:

Introduction: Microelectronics, semiconductor technology, and circuit taxonomy, microelectronic design styles, CAD and optimization. Graph Theory and optimization problems and techniques: the shortest path, longest path, vertex cover, coloring, clique covering and partitioning. Algorithms: Greedy, dynamic, Branch and Bound, Backtracking, tractable and intractable problems. Basic digital electronics. Y-Chart **[7 Lectures]**

Boolean Algebra and Application: Computational Boolean Algebra: Basics, Boolean Difference, Quantification Operators, Application to Logic Network Repair, Recursive Tautology **[5 Lectures]**

Recursive Tautology—URP Implementation.BDD Basics, BDD Sharing, BDD Ordering, Satisfiability (SAT), Part, Boolean Constraint Propagation (BCP) for SAT, Using SAT for Logic **[5 Lectures]**

Introduction to Digital VLSI Design Flow Specification, High-level Synthesis, RTL Design, Logic Optimization, Verification, and Test Planning, Design Representation, Hardware Specific Transformations **[6 Lectures]**

Scheduling, Allocation, and Binding: Problem Specification:

Scheduling, Allocation, and Binding, Basic Scheduling Algorithms (Time constrained and Resource-Constrained), Allocation Steps: Unit Selection, Functional Unit Binding, Storage Binding, Interconnect Binding, and Allocation Techniques: Clique Partitioning, Left-Edge Algorithm, Iterative Refinement **[7 Lectures]**

Logic Optimization and Synthesis: 2-Level Logic: Basic, The Reduce-Expand-Irredundant Optimization Loop, Details for One Step: Expand, Multilevel Logic, and the Boolean Network Model **[6 Lectures]**

Multilevel Logic: Algebraic Model for Factoring, Algebraic Division, Role of Kernels and Co-Kernels in Factoring, Finding the Kernels. **[4 Lectures]**

Text/Reference Books:

1. Synthesis and Optimization of Digital synthesis by Giovanni De Micheli.
2. High-Level Synthesis: from Algorithm to Digital Circuit by Philippe Coussy, Adam Morawiec
3. Introduction to Logic Synthesis using Verilog HDL (Synthesis Lectures on Digital Circuits and Systems)
4. Finite State Machine Datapath Design, Optimization, and Implementation (Synthesis Lectures on Digital Circuits and Systems)

Course Outcome (CO):

CO1: This course is about the automatic generation of digital circuits from high-level descriptions.

CO2: Modern electronic systems are specified in Hardware Description Languages and are converted automatically into digital circuits.

CO3: This course introduces the VHDL Hardware Description Language, and follows it up with a discussion of the basics of synthesis topics including High-level Synthesis, FSM Synthesis, Retiming, and Logic Synthesis.

CO4: Students will be able to understand the fundamentals of HDL and synthesis.

Course Code : OTT4XX	**Course Credit : 3**
Course Name : Numerical Computations	**L-T-P : 3-0-0**

Course Prerequisite: Basic Mathematics

Course Syllabus:
Computational errors: Error definition, Absolute and relative errors, Truncation errors, Round off errors with examples and implementation in MATLAB.
Solutions of system of linear equations: LU decomposition method, Gauss-Seidal method.
Roots of non-linear equations: Bisection method vs Regula-Falsi method, geometrical interpretations, Newton-Raphson method vs Modified Newton-Raphson method, geometrical interpretations and MATLAB implementations **[10 Lectures]**
Finite Differences: operators, forward and backward differences, central differences. Relation between them. Use MATLAB to compute numerically.
Interpolation: Newton-Gregory formula for forward interpolation with error, Newton-Gregory formula for backward interpolation with error, Stirling's formula for central interpolation. Lagrange's interpolation formula, Relationship among various interpolation formulae. Use MATLAB for computation. **[10 Lectures]**
Numerical integration with MATLAB implementation: General quadrature formula, Trapezoidal rule with geometrical interpretation and error, Simpson's $1/3^{rd}$ and $3/8^{th}$ rules with errors. **[10 Lectures]**
Numerical solution of Ordinary differential equations of first order: Picard's method for successive approximations, Euler's method with its geometrical interpretations, Modified Euler's method with error analysis, Runge-Kutta IV order method. Use MATLAB to execute the above methods **[10 Lectures]**

Text/Reference Books:
1. M.K. Jain, S.R.K Iyenger and R.K. Jain; Numerical methods for scientific and engineering computation, New age international publishers
2. L. N. Trefethen and D. Bau III, Numerical Linear Algebra, SIAM, Philadelphia, 1997.
3. J. H. Mathews and K.D. Fink, Numerical methods using MATLAB, Pearson Education.
4. Balagurusamy: Numerical Methods, Scitech.
5. Baburam: Numerical Methods, Pearson Education.

Course Outcome (CO):
CO1: To understand the numerical methods of solving the non-linear equations, interpolation, differentiation, and integration.
CO2: To improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.
CO3: To provide a basic understanding of the derivation, analysis, and use of these numerical methods.

Course Code : CST4XX	**Course Credit : 3**
Course Name : Software Planning and Management	**L-T-P : 3-0-0**

Course Prerequisite: Basic knowledge of software engineering

Course Syllabus:
The Basics of Measurement: Measurement in software engineering, The scope of software metrics, the representational theory of measurement, Measurement and models, Measurement scales and scale types, Meaningfulness' in measurement **[8 Lectures]**
Goal-based framework for software measurement: Classifying software measures, determining what to measure, applying the framework, Software measurement validation **[8 Lectures]**
Empirical investigation: Principles of investigation, planning formal experiments, planning, case studies **[4 Lectures]**
Measuring internal product attributes: Aspects of software size, Length, reuse
Measuring internal product attributes; Types of structure measures, Control-flow structure, and Modularity and information flow attributes, Object-oriented metrics **[4 Lectures]**
Measuring external product attributes: Modeling software quality, measuring aspects of quality **[6 Lectures]**
Making process predictions: Good estimates, Cost estimation - problems and approaches, Models of effort and cost, Problems with existing modeling methods, Dealing with problems of current estimation methods, Implications for process prediction **[6 Lectures]**
Software Project Management: General management, introduction to software project management, Conventional software management, project initiation, feasibility study, project CO1 Determine the software measurement attributes and metrics **[4 Lectures]**

Text/Reference Books:
1. R. S. Pressman, Software Engineering
2. P. Jalote, Software Project Management in Practice.
3. B. Hughest& M. Cotterell, Software Project Management.
4. Norman E. Fenton, Shari Lawrence Pfleeger, Software Metrics - A Rigorous and Practical Approach, 2nd Edition, PWS Pub, 1996.
5. Walker Royce, Software Project Management, Addison Wesley, 1998

Course Outcome (CO):
CO1: Determine the software measurement attributes and metrics
CO2: Plan and evaluate software projects.
CO3: Analyze factors involved in implementation of software projects.
CO4: Understand project monitoring and control techniques, planning, project evaluation, resource allocation, project monitoring, project control, case studies.

Course Code : CST4XX	**Course Credit : 3**
Course Name : Digital Image Processing	**L-T-P : 3-0-0**

Course Prerequisite: Basic concepts of computer science and mathematics

Course Syllabus:

Digital Image Fundamentals: A simple image model, Sampling and Quantization, Imaging Geometry, Image Acquisition Systems, Different types of digital images.

Binarization and Segmentation of Grey level images: Histogram of grey level images, Optimal thresholding using Bayesian classification, multilevel thresholding, Segmentation of grey level images **[9 Lectures]**

Detection of edges and lines in 2D images: First order and second order edge operators, multi-scale edge detection, Cannys edge detection algorithm, Hough transform for detecting lines and curves **[8 Lectures]**

Images Enhancement: Point processing, Spatial Filtering, Frequency domain filtering, multi-spectral image enhancement, image restoration.

Image Registration and depth estimation: Registration Algorithms, Setreo Imaging, Computation of disparity map **[8 Lectures]**

Bilevel Image Processing: Basic concepts of digital distances, distance transform, medial axis transform, component labeling, thinning, morpho-logical processing, extension to grey scale morphology **[8 Lectures]**

Color Image Processing: Color Representation, Laws of color matching, chromaticity diagram, color enhancement, color image segmentation, color edge detection, color demosaicing.

Image compression: Lossy and lossless compression schemes, prediction based compression schemes, vector quantization, sub-band encoding schemes, JPEG compression standard **[7 Lectures]**

Text/Reference Books:

1. R C Gonzalez & R E Woods, Digital Image Processing, 3rd /4th Ed, PHI 2. B A Forouzan, “Cryptograpgy and Network Security”, Tata McGraw Hill, 2007.
2. A. K. Jain, Fundamentals of DIP, PHI
3. Wiliam K Pratt, DIP, Wiley Student Publishers, 3ed.
4. R C Ganzalez, R E Woods & S L Eddins, DIP using MATLAB, 2 nd Ed.

Course Outcome (CO):

CO1: Students will be able to compare different methods for image acquisition, storage and representation in digital devices and computers.

CO2: Students will be able to appreciate role of image transforms in representing, highlighting, and modifying image features.

CO3: Students will be able to interpret the mathematical principles in digital image enhancement and apply them in spatial domain and frequency domain.

CO4: Students will be able to apply various methods for segmenting image and identifying image components.

CO5: Students will be able to summarise different reshaping operations on the image and their practical applications.

Course Code : CST4XX	**Course Credit : 3**
Course Name : Graph Theory	**L-T-P : 3-0-0**
Course Prerequisite: Basic mathematics	

Course Syllabus:
Basic definitions of graphs and multigraphs: adjacency matrices, isomorphism, decompositions, independent sets and cliques, graph complements, important graph like cubes and the Petersen graph.
Paths, cycles, and trails: Eulerian circuits, hamiltonian graph and circuit.
Vertex degrees and counting: large bipartite subgraphs, the handshake lemma.
Directed graphs: weak connectivity, connectivity, strong components **[10 Lectures]**
Trees Basics: equivalent characterizations of trees, forests, Spanning trees and 2-switches, Distance and center.
Optimization: Kruskal’s Theorem, Prim’s Theorem and Dijkstra’s Theorem **[6 Lectures]**
Matching and covering: Bipartite matching, vertex cover, edge cover, independent set, M-alternating path, Hall’s Theorem, König-Egeváry Theorem, Gallai’s Theorem.
Connectivity: Vertex cuts, separating sets, vertex and edge connectivity, block-cutpoint tree, Menger’s Theorem: undirected vertex and edge versions **[8 Lectures]**
Planarity: Embeddings, dual graphs, Euler's formul, Kuratowski's Theorem
Network flow: Ford-Fulkerson Labeling algorithm, flow integrality, Max-flow/Min-cut Theorem **[7 Lectures]**
Coloring: Chromatic number, lower bounds from clique number and maximum independent set, upper bounds from greedy coloring (& Welsh-Powell), k-critical graphs, cartesian product of graphs, and interval graphs, k- Chromatic graphs: Mycielski's construction, Edge coloring, line graphs **[9 Lectures]**

Text/Reference Books:
1. Douglas B. West, Introduction to Graph Theory, 2nd ed., Prentice Hall.
2. Narasingh Deo, Graph theory, PHI, 1979.
3. Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.,

Course Outcome (CO):
CO1: Students will achieve command of the fundamental definitions and concepts of graph theory.
CO2: Students will understand and apply the core theorems and algorithms, generating examples as needed, and asking the next natural question.
CO3: Students will achieve proficiency in writing proofs, including those using basic graph theory proof techniques such as bijections, minimal counterexamples, and loaded induction.
CO4: Students will become familiar with the major viewpoints and goals of graph theory: classification, externality, optimization and sharpness, algorithms, and duality.
CO5: Students will be able to apply their knowledge of graph theory to problems in other areas, possibly demonstrated by a class project.

Course Code : CST4XX	**Course Credit : 3**
Course Name : Natural Language Processing	**L-T-P : 3-0-0**

Course Prerequisite: Artificial Intelligence

Course Syllabus:
Introduction to NLP: Introduction natural language processing, stop word removal, stemming, lemmatization. Language Modeling: N-grams, chain rule, Markov assumption, Evaluating Language Models, Smoothing: Laplace Smoothing, Add-k smoothing, interpolation, backoff methods **[6 Lectures]**
Classification, Learning representation: Text classification, Naïve bayes, Evaluation: Precision, Recall, F-measure. Vector space model, Term weighting schemes, Term Frequency, Term Frequency-Inverse Document Frequency, Binary.
Vector Semantics: Embeddings, Cosine for measuring similarity, Point wise Mutual Information (PMI). **[8 Lectures]**
Dimensionality reduction for NLP: Latent semantics, Singular value decomposition, Principal Component Analysis. Distributional semantics, Word Embeddings, Word2Vec, skipgram, continuous bag of words (CBOW), Embeddings using SVD. **[10 Lectures]**
Neural Networks and Neural Language Models: Gradient descent, convolution, Convolutional neural network for NLP applications. Sequence Processing with Recurrent Networks: Recurrent neural network, Long short term memory, GRU **[10 Lectures]**
Unsupervised Approaches: Topic Models: Latent semantic analysis, Latent Dirichlet allocation (LDA). NLP Applications – Sentiment Analysis, Spam Detection, Abusive language detection, Fake news detection etc **[6 Lectures]**

Text/Reference Books:
1. Dan Jurafsky and James Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, Second Edition, 2009.
2. Chris Manning and HinrichSchütze. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA: May 1999.
3. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
4. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.

Course Outcome (CO):
CO1: To understand the fundamental concepts and state-of-the-art techniques of natural language processing.
CO2: To analyze the challenges of empirical methods for NLP applications and hand-on experience to implement state-of-the-art techniques using text analysis tools.
CO3: To gain an in-depth understanding of the applied natural languages processing based on state-of-the-art neural models.

Course Code : CST4XX	**Course Credit : 3**
Course Name : System Level Design and Modeling	**L-T-P : 3-0-0**

Course Prerequisite: Digital Logic Design and Synthesis, Computer Architecture and Organization, Algorithms, RTOS

Course Syllabus:
Introduction: System Stack- Application to Physics, exploration of all layers, System-Design Challenges, System Design Methodology, System-Level Models. Modeling- Models of Computation, System Design Languages, System Modeling, Processor Modeling, Communication Modeling, System Models **[7 Lectures]**
System Synthesis: TLM Based Design, Automatic TLM Generation, Automatic Mapping, Platform Synthesis **[6 Lectures]**
Software Synthesis: Target Languages for Embedded Systems and RTOS, Software Synthesis Overview, Code Generation, Multi-Task Synthesis, Internal and External Communication **[6 Lectures]**
Hardware Synthesis: RTL Architecture, Estimation and Optimization, Register Sharing, Functional Unit Sharing, Connection Sharing Register Merging **[7 Lectures]**
Hardware Synthesis: Chaining, and Multi-Cycling, Functional-Unit Pipelining, Datapath Pipelining, Control, and Datapath Pipelining, Scheduling **[7 Lectures]**
System Design with SystemC: SystemC library, Concurrency, Module, Channel, shared data communication and protocols design, blocking and non-blocking communication FIFo design, Bus design, critical section protocol design **[7 Lectures]**

Text/Reference Books:
1. Embedded System Design: Modeling, Synthesis and Verification. Author: Daniel D. Gajski, Samar Abdi, Andreas Gerstlauer, ISBN-13: 9781489985309, Publisher: Springer, 2009.
2. Specification and Design of Embedded Systems by Gajski, Daniel D., Vahid, Frank, Narayan, Sanjiv, Gong, Jie Prentice Hall, 1994.

Course Outcome (CO):
CO1: This course presents information on how to design a future multiprocessor system consisting of several processors and other components.
CO2: Design methodology, modeling techniques, software and hardware synthesis methods and techniques for verification of such multi-processor systems.

Specialization Elective Courses

List of Electives
B.Tech CSE Specialization in AI and Machine Learning

Elective Courses:

Human-AI Interaction
Computational Intelligence
Intelligent Agents
Multi-Agent Systems
Automated Reasoning
AI in Robotics
Game Theory
Knowledge Representation and Reasoning
Decision Making and Planning Techniques
Design of Artificial Intelligence Products
Business Intelligence
Recommender Systems
Natural Language Processing
Computer Vision

Course Code: CST4XX	**Course Credit: 3**
Course Name: Human-AI Interaction	**L-T-P: 3-0-0**

Course Prerequisite: Basics of programming

Course Syllabus:

Introduction: Perspectives on human-AI interaction, History of humans interacting with AI, Artificial intelligence (AI) vs. intelligence augmentation (IA), Designing AI/ML user experience, Matchmaking needs and risks for adding AI/ML, AI ethics, fairness, social acceptability, and trust design of human-AI systems, AI-infused system, Human-AI teams understanding and addressing the performance/compatibility tradeoff. [6 Lectures]

AI and mental models: Designing AI for different stakes, AI and coadaptation, Designing for failure- failure and feedback with users, Communicating predictions and recommendations with users.[7 Lectures]

Data and visualization: Data and knowledge, Data ethics and laws, Using human-centric data in an ML pipeline, Data visualization and data communication, Visualizations to improve human-AI interaction, Intelligible artificial intelligence. [10 Lectures]

Improving fairness in AI/ML systems: Limits/Pitfalls of post-hoc, algorithmic de-biasing, Metrics to measure human-AI performance, Modeling of humans interaction with risk predictions, Intelligible Models for healthcare- predicting pneumonia. [10 Lectures]

Human in the loop with AI/ML and recommendations: Chatbots, Natural language and speech applications, Vision, Images, and Art vision with GANs, personalized context aware health interventions, case studies.
[7 Lectures]

Text/Reference Books:

1. Illah Reza Nourbakhsh and Jennifer Keating, AI and Humanity. MIT Press. 2020.
2. MIT Press Bites: The Future of AI and Human Interaction. https://mitpress.mit.edu/blog/mitp-bites-future-ai-and-human-interaction
3. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S. T., Bennett, P., Inkpen, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. (2019) Guidelines for Human-AI Interaction. Glasgow, Scotland, Uk.
4. Kocielnik, R., Amershi, S., and Bennett, P. (2019) Will You Accept an Imperfect AI? Exploring Designs for Adjusting End-User Expectations of AI Systems.

Course Outcome (CO):

CO1: This course bridges the gap between the two fields of HCI and AI.
CO2: Students will understand how human-AI interaction can explore various dimensions, including ethics, explainability, design process involving AI.
CO3: Students will learn to improve visualization, human-AI collaboration through user feedback and recommender systems.

Course Code: CST4XX	**Course Credit: 3**
Course Name: Computational Intelligence	**L-T-P: 3-0-0**

Course Prerequisite: Basics of programming

Course Syllabus:

Introduction: Intelligence machines, why computational intelligence, paradigms, computational intelligence concept and importance [4 lectures]

Fuzzy Expert Systems: Rule-based expert system. Uncertainty management. Fuzzy Logic and Fuzzy Relationships. Fuzzy sets and operations of fuzzy sets. Fuzzy rules and fuzzy inference. Fuzzy expert systems. Adaptation of fuzzy systems. Case Studies. [10 lectures]

Artificial Neural Networks: Fundamental neuro computing concepts: artificial neurons, activation functions, neural network architectures, learning rules. Supervised learning neural networks: multi-layer feed forward neural networks, simple recurrent neural networks, time-delay neural networks, supervised learning algorithms. Unsupervised learning neural networks: self-organizing feature maps. Radial basis function networks. Deep neural networks and learning algorithms. Case studies. [12 lectures]

Evolutionary computation: Chromosomes, fitness functions, and selection mechanisms. Genetic algorithms: crossover and mutation, Genetic programming. Evolution strategies. Case studies [8 lectures]

Swarm Intelligence: Foundations. cAnts, Termites, Gnats, Birds. Applications. Case Studies. Hybrid Intelligent Systems: Neural expert systems. Neuro-fuzzy systems. Evolutionary neural networks. [6 lectures]

Text/Reference Books:

1. Computational Intelligence - Concepts to Implementations by Eberhart& Shi
2. Konar A., Computational Intelligence: Principles, Techniques and Applications, Springer Verlag, 2005
3. A.P. Engelbrecht, Computational Intelligence: An Introduction, 2nd Edition, John Wiley & Sons, 2012.
4. H.K. Lam, S.S.H. Ling, and H.T. Nguyen, Computational Intelligence and Its Applications: Evolutionary Computation, Fuzzy Logic, Neural Network and Support Vector Machine, Imperial College Press, 2011.

Course Outcome (CO):

CO1: Understand the various searching techniques, constraint satisfaction problem and example problems-game playing techniques.

CO2: Apply these techniques in applications which involve perception, reasoning and learning.

CO3: Explain the role of agents and how it is related to the environment and the way of evaluating it and how agents can act by establishing goals.

CO4: Acquire the knowledge of real world Knowledge representation

Course Code: CST4XX	**Course Credit: 3**
Course Name: Intelligent Agents	**L-T-P: 3-0-0**

Course Prerequisite: Introductory course on AI

Course Syllabus:

Intelligent agent(s), Role of intelligent agents in real-world, Basic models and algorithms for individual agents, Differences between agents and conventional computer programs, Single vs multi-agents [7 lectures]

Agent-based computing, need and motivation, goal-oriented agents, reactive agents, exploration-exploitation tradeoff, AI planning methods [10 lectures]

Design intelligent agents, investigate different types of agent architectures, Applications of intelligent agents to solve some real-world problems such as (i) searching (ii) games (iii) self-driving cars (iv) face recognition (v) web search (vi) autonomous robot (vii) missile guidance, and other problems. [15 lectures]

Agent interactions, modeling cooperative behaviour, modeling competitive behaviour, Structuring agent models in code, Deploying agents within a simulated environment [8 lectures]

Text/Reference Books:

1. Michael Wooldridge : An Introduction to MultiAgent Systems - Latest Edition, John Wiley & Sons.
2. Yoav Shoham, Kevin Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Latest Edition, Cambridge University Press
3. Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall, 2009.

Course Outcome (CO):

After successfully completed this course, students will be able to:
CO1: Deploy an agent within a simulated agent trading environment
CO2: Analyse and critique the performance of a deployed agent
CO3: Agent models in use today and their grounding in artificial intelligence research
CO4: Motivations for, and appropriate use of, agent-based computing
CO5: Main agent decision making frameworks for cooperative and competitive environments

Course Code: CST4XX	**Course Credit: 3**
Course Name: Multi-Agent Systems	**L-T-P: 3-0-0**

Course Prerequisite: Basics of AI

Course Syllabus:

Introduction of Multi-agents: agents and objects, agents and distributed systems, agents and expert systems, typical application areas for multi-agent systems, the design of multi-agent systems - reasoning agents, agents as reactive systems, hybrid agents [8 lectures]

Multi-Agent Systems: program multi-agent systems, societies and working together, Task Sharing, practical reasoning, Distributed Problem Solving and Planning, Modeling and design of Multi-Agent Systems, Environments: Support for Defining Simulated Environments, Running a System of Multiple Situated Agents, Applications of Distributed Artificial Intelligence in Industry [12 lectures]

Multiagent Interactions: Classifying multi-agent interactions - cooperative versus non-cooperative, Nash Equilibrium, Competitive and Zero-Sum Interactions, The Prisoner's Dilemma, Axelrod's Tournament, The Game of Chicken and other symmetric 2 x 2 games [10 lectures]

Multiagent communication: Communication and interaction, Available Performatives, Contract Net Protocol, Issues for Implementing Contract Net, communication between two agents, adversarial communication among multiple agents, Learning communication for multi-agent systems, targeted multi-agent communication [10 lectures]

Text/Reference Books:

1. Michael Wooldridge, An Introduction to MultiAgent Systems - Second Edition. (Wiley, 2009)
2. Rafael H. Bordini, Jomi Fred Hubner and Michael Wooldridge, Programming Multi-agent Systems in AgentSpeak. (Wiley, 2007)
3. Y. Shoham and K. Leyton-Brown, Multiagent Systems, Cambridge University Press

Course Outcome (CO):

CO1: Understand the notion of multi-agent and their characteristics

CO2: Design an agent for competitive environment key issues associated with constructing agents capable of intelligent autonomous action

CO3: Designing societies of agents that can effectively cooperate in order to solve problems, including an understanding of the key types of multi-agent interactions possible in such systems

CO4: Understand the main application areas of agent-based solutions, and be able to develop a meaningful agent-based system using a contemporary agent development platform.

Course Code: CST4XX	**Course Credit: 3**
Course Name: Automated Reasoning	**L-T-P: 3-0-0**

Course Prerequisite: Discrete mathematics

Course Syllabus:

Reasoning, Introduction to automated reasoning and history of automated theorem provers. SAT/SMT basics, SAT examples, SMT applications. [6 lectures]

Mathematical logic: representations, propositional, predicate logic, semantics. First-order Logic, Higher-Order Logic. [10 lectures]

Proof and correctness: formalization of proof, inference rules, and resolution, unification, equational reasoning, combinatorial explosion, search algorithms. Inductive Proofs. [8 lectures]

Guidance techniques: rewrite rules, human proofs, decision procedures, meta-level inference.
Inductive theorem proving, heuristic guidance, rippling, proof planning. [8 lectures]

Applied uses of automated reasoning: diagrammatic reasoning, ontology/semantic web, Reasoning Under Uncertainty, Program Verification with few examples. [8 lectures]

Text/Reference Books:

1. J. Harrison, Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009.
2. M. Huth and M. Ryan: Logic in Computer Science: Modelling and Reasoning about Systems, 2nd Edition, Cambridge University Press, 2004.
3. Bundy, A.: The computer modelling of mathematical reasoning. Latest Edition.

Course Outcome (CO):

CO1: Represent mathematical and other knowledge using logical formalism;

CO2: Explain the history of formalizing mathematical knowledge;

CO3: Illustrate the advantages and limitations of the main approaches and techniques in automated reasoning of mathematical knowledge;

CO4: Apply different automated reasoning techniques to new problems;

Course Code: CST4XX	**Course Credit: 3**
Course Name: AI in Robotics	**L-T-P: 3-0-0**

Course Prerequisite: Some programming experience and mathematical background

Course Syllabus:

What are intelligent robots, role and importance of AI in robotics, autonomous robots, path planning, motion planning, environment perception and reaction, manipulation. robot architecture and modeling in simulation environment [8 lectures]

Configuration space, rigid body motion planning, understanding kinematics and dynamics of a robot (wheeled mobile robot or robot manipulator or unmanned aerial vehicle) [6 lectures]

Goal based robots, path planning and search based algorithms - Bug algorithms, A* search, grid based planning, sampling based planning, artificial potential field methods [6 lectures]

Target tracking, vision based system, target pose estimation, target estimation using kalman filter, vision based environment perception, target identification and localization, object recognition [10 lectures]

Vision based object detection and recognition, vision based navigation, vision based object pick and place, other vision based applications [10 lectures]

Text/Reference Books:

1. Howie Choset, Seth Hutchinson, et al., Principles of Robot Motion: Theory, Algorithms, and Implementations, Latest Edition, MIT Press
2. Francis X. Govers, Artificial Intelligence for Robotics, Latest Edition, Packt Publishing
3. Robin R. Murphy, Introduction to AI Robotics, Second Edition, The MIT Press

Course Outcome (CO):

Upon successfully completing this course, students will be able to:
CO1: Understand and implement search based AI techniques in autonomous systems
CO2: Design robot structure with its dynamical systems as per requirement
CO3: Implement motion planning for target tracking and identification in crowd
CO4: Implement arm manipulator to make service robot

Course Code: CST4XX	**Course Credit: 3**
Course Name: Game Theory	**L-T-P: 3-0-0**

Course Prerequisite: Algorithms

Course Syllabus:

Introduction to game theory, routing games and mechanism design; Strategies, costs, and payoffs;Prisoner's dilemma, Nash Equilibrium, Strategic games; Best response; Dominant strategies; Pure strategy v/s Mixed strategy [10 Lectures]

Routing games; Selfish routing; Quantifying inefficiency of equilibrium; Price of Anarchy; Social optimum; Price of stability; Scheduling games [8 Lectures]

Repeated games; Bayesian games, Population games; Evolutionary game theory; Evolutionary stable strategy; Replicator dynamics [10 Lectures]

Non-cooperative games; Cooperative game theory; Nash bargaining [5 Lectures]

Mechanism design, Algorithmic mechanism design, Distributed algorithmic mechanism design [7 Lectures]

Text/Reference Books:

1. D. Fudenberg and J. Tirole, Game Theory, MIT Press, Latest Edition.
2. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Algorithmic Game Theory, Cambridge University Press, Latest Edition.

Course Outcome (CO):

After this course student should be able to
CO1: distinguish a game situation from a pure individual’s decision problem,
CO2: explain concepts of players, strategies, payoffs, rationality, equilibrium,
CO3: find dominant strategy equilibrium, pure and mixed strategy Nash equilibrium,
CO4: explain concepts of asymmetric information, and to analyze simple signaling games,
CO5: analyze repeated games, and to explain the folk-theorem.

Course Code: CST4XX	**Course Credit: 3**
Course Name: Knowledge Representation and Reasoning	**L-T-P: 3-0-0**

Course Prerequisite: Some exposure to formal languages, logic and programming

Course Syllabus:
Introduction, Propositional Logic, Syntax and Semantics ,Proof Systems, Natural Deduction, Tableau Method, Resolution Method [8 lectures]

First Order Logic (FOL), Syntax and Semantics, Unification, Forward Chaining , The Rete Algorithm, Rete example, Programming Rule Based Systems , Representation in FOL, Categories and Properties, Reification, Event Calculus, Deductive Retrieval [12 lectures]

Backward Chaining, Logic Programming with Prolog , Resolution Refutation in FOL, FOL with Equality, Complexity of Theorem Proving , Description Logic (DL), Structure Matching, Classification, Extensions of DL, The ALC Language, Inheritance in Taxonomies, Default Reasoning, Circumscription [12 lectures]

The Event Calculus Revisited , Default Logic, Autoepistemic Logic, Epistemic Logic, Multi Agent Scenarios [8 lectures]

Text/Reference Books:
1. Chitta Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press.
2. Nilson, Artificial Intelligence: A New Synthesis, Latest Edition.
3. Soldek, Jerzy, Drobiazgiewicz, Leszek, Artificial Intelligence and Security in Computing Systems, Allied Publishers, 2004.

Course Outcome (CO):
CO1: be able to model simple application domains in a logic-based language;
CO2: understand the notion of a reasoning service;
CO3: master the fundamentals of the reasoning algorithms underlying current systems;
CO4: understand the fundamental trade-off between representation power and computational properties of a logic-based representation language;

Course Code: CST4XX	**Course Credit: 3**
Course Name: Decision Making and Planning Techniques	**L-T-P: 3-0-0**

Course Prerequisite: Basic programming

Course Syllabus:

Introduction to Decision Making: Decision Making, Rational Decision Making, Flexibility-Bounded Decision Making, Correlation Machine, Causal Machine, Brief Introduction to Neural Networks, Fuzzy Sets, Rough Sets [6 lectures]

Causal Function for Rational Decision Making: Causality Description, Models of Causality, Transmission Theory, Probability Theory, Projectile Theory, Causal Calculus, Manipulation Theory, Process Theory, Counterfactual Theory, Structural Learning, Causal Function for Relation Decision Making, Interstate Conflict, Rough Sets Casual Function, Rough Set Causal Machine [10 lectures]

Correlation Function for Rational Decision Making: Correlation Description, Correlation Function, Modelling Epileptic Activity, SVM Correlation Function, Application to Epileptic Activity [5 lectures]

Decision Making and Flexibility -Bounded Rationality: Basic Fuzzy Logic Theory, Neuro-Fuzzy Model, Inference Making, Rational Choice, Bounded Rational Decision Making, Flexibility – Bounded Rational Decision Making, Missing Data Estimation [8 lectures]

Irrelevant Information Filtering: Cocktail Party Problem, Marginalization of Irrationality Theory, Automatic Relevance Determination Theory, Principal Component Analysis, Blind Source Separation [6 lectures]

Group Decision Making: Types of Group Decision Making, Artificial Intelligence for Group Decision Making – Equality Weighted Ensemble, Statically Weighted Ensemble, Dynamically Weighted Mixtures [5 lectures]

Text/Reference Books:

1. Russell and Norvig, Artificial Intelligence: A Modern Approach, Latest Edition, Prentice Hall.
2. Tshilidzi Marwala, Artificial Intelligence Techniques for Rational Decision Making
3. Ritch and Knight, Artificial Intelligence

Course Outcome (CO):

CO1:Develop knowledge of decision making and learning methods
CO2:Learn extensively of rational decision making
CO3: Filter out irrelevant information

Course Code: CST4XX	**Course Credit: 3**
Course Name: Design of Artificial Intelligence Products	**L-T-P: 3-0-0**

Course Prerequisite: Software Engineering

Course Syllabus:
AI contribution to user experience, Intelligent UI, Computationally intelligent systems. Introduction to the Artificial Intelligence Design Process, Stages of AI product design, Technical and Operational requirements, Cost metrics for an AI software development plan. Matchmaking (Capabilities, Activities, Domain, Target). [8 lectures]

Artificial Intelligence Technology Fundamentals: Unsupervised and Semi-supervised methods of machine learning algorithms. Bayesian and Regression models. Basics of Deep Learning, Neural networks, Artificial Neurons, and Simulation of complex networks. [6 lectures]

Designing Artificial Machines to Solve Problems, Identify superhuman intelligence used in an AI product. Compare and contrast the advantages and disadvantages of using AI technology. [5 lectures]

Designing Intelligent Human-Machine Interfaces (HMI): Techniques, Application areas, Benefits, and drawbacks of HMI. An appropriate level of machine involvement in interactions with humans. [5 lectures]

Superminds: Designing Organizations that Combine Artificial and Human Intelligence, the concept of superminds, and compare and contrast the different types of superminds. Analyze how humans and machines can work together to surpass the sum of their parts. Cognitive processes to various organizations and community problems. [8 lectures]

Case studies on Marketplace Frontiers of AI Design: Artificial intelligence and Generative Adversarial Networks (GANs) to generate fake images and videos from real data. Assess the technical, social, and economic impact of AI technologies. [8 lectures]

Text/Reference Books:
1. Denis Rothman, Artificial Intelligence By Example: Acquire advanced AI, machine learning, and deep learning design skills, 2nd Edition, 2020.
2. Max Tegmark, Life 3.0 Being Human in the Age of Artificial Intelligence, 2017.
3. Wilbert O. Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, 3rd Edition, 2013.
4. Stuart Russell and Peter Norvig,: Artificial Intelligence: A Modern Approach, 4th US ed. 2021.

Course Outcome (CO):
CO1: Learn the stages of AI product design
CO2: Analyze technical and operational requirements to build AI models
CO3: Differentiate between various machine learning algorithms
CO4: Learn to apply machine learning methods to practical problems
CO5: Learn about challenges you may encounter when designing AI products

Course Code: CST4XX	**Course Credit: 3**
Course Name: Business Intelligence	**L-T-P: 3-0-0**

Course Prerequisite: Basics of programming and AI

Course Syllabus:

Introduction: Definition, Evolution, Business Intelligence Segments, Difference between Information and Intelligence, Defining Business Intelligence Value Chain, Factors of Business Intelligence System, Real time Business Intelligence, Development Stages and Steps, Business Intelligence Applications. [7 lectures]

Business Intelligence Essentials: Creating Business Intelligence Environment, Business Intelligence Landscape, Business Intelligence Platform, Dynamic roles in Business Intelligence, Roles of Business Intelligence in Modern Business, Business Intelligence Framework, Challenges [7 lectures]

Business Intelligence Types: Multiplicity of Business Intelligence Tools, Business Intelligence Tools, Modern Business Intelligence, Enterprise Business Intelligence, Architecting the Data: Types of Data, Enterprise Data Model, Enterprise Conceptual Model, Data Reporting and Query, Data Partitioning, Metadata, Total Data Quality Management [10 lectures]

Business Intelligence Strategy and Road Map: Planning to implement a Business Intelligence Solution, Understand Limitations of Business Intelligence. Business Intelligence Implementation: Business Intelligence Platform, Capability Matrix, Target Databases, Data Mart [10 lectures]

Business Intelligence Issues and Challenges: Critical Challenges, Cross-Organizational Partnership, Business Intelligence Application Development Methodology, Planning the BI Projects, Business Analysis and Data Standardization [6 lectures]

Text/Reference Books:

1. Enterprise Business Intelligence and Data Warehousing: Program Management Essentials
2. Cindi Howson, Successful Business Intelligence: Unlock the Value of BI & Big Data", 2nd Edition, Kindle Edition.
3. Larissa T. Moss and Shaku Atr, Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications, Addison-Wesley Professional.

Course Outcome (CO):

CO1: business intelligence purposes and for working as a business intelligence developer.
CO2: The course gives an overview of how business intelligence technologies can support decision making across any number of business sectors.