Thapar Institute of Engineering & Technology, Patiala

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COURSES SCHEME

FOR

BE (COMPUTER SCIENCE & ENGINEERING)

2019

SEMESTER-I

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCB008	APPLIED CHEMISTRY	CF	3	1	2	4.5
2	UTA003	COMPUTER PROGRAMMING	CP	3	0	2	4.0
3	UES013	ELECTRICAL & ELECTRONICS ENGINEERING	CF	3	1	2	4.5
4	UEN002	ENERGY AND ENVIRONMENT	CF	3	0	0	3.0
5	UMA010	MATHEMATICS – I	CF	3	1	0	3.5
6	UES009	MECHANICS	CF	2	1	2*	2.5
		TOTAL		**17**	**4**	**6**	**22.0**

MECHANICS (2*): 2 HOURS LAB ONCE IN SEMESTER

SEMESTER-II

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UPH004	APPLIED PHYSICS	CF	3	1	2	4.5
2	UTA018	OBJECT ORIENTED PROGRAMMING	CP	3	0	2	4.0
3	UTA026	MANUFACTURING PROCESS	CF	2	0	2	3.0
4	UTA015	ENGINEERING DRAWING	CF	2	4	0	4.0
5	UHU003	PROFESSIONAL COMMUNICATION	CF	2	0	2	3.0
6.	UMA004	MATHEMATICS – II	CF	3	1	0	3.5
		TOTAL		**15**	**6**	**8**	**22.0**

SEMESTER-III

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS303	OPERATING SYSTEMS	CP	3	0	2	4.0
2	UCS405	DISCRETE MATHEMATICAL STRUCTURES	CP	3	1	0	3.5
3	UCS301	DATA STRUCTURES	CP	3	0	2	4.0
4	UCS510	COMPUTER ARCHITECTURE AND ORGANIZATION	CP	3	0	0	3.0
5	UMA011	NUMERICAL ANALYSIS	CF	3	0	2	4.0
6	UCS311	PRACTICAL COMPUTING	CP	1	0	2	2.0
		TOTAL		**16**	**1**	**8**	**20.5**

SEMESTER-IV

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS411	ARTIFICIAL INTELLIGENCE	CP	3	0	2	4.0
2	UCS415	DESIGN AND ANALYSIS OF ALGORITHMS	CP	3	0	2	4.0
3	UCS310	DATABASE MANAGEMENT SYSTEMS	CP	3	0	2	4.0
4	UCS503	SOFTWARE ENGINEERING	CP	3	0	2	4.0
5	UCS414	COMPUTER NETWORKS	CP	2	0	2	3.0
6	UMA035	OPTIMIZATION TECHNIQUES	CF	3	0	2	4.0
		TOTAL		**17**	**0**	**12**	**23.0**

SEMESTER-V

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UML501	MACHINE LEARNING	CP	3	0	2	4.0
2	UCS410	PROBABILITY AND STATISTICS	CP	3	0	2	4.0
3	UCS531	CLOUD COMPUTING	CP	2	0	2	3.0
4	UCS413	NETWORK PROGRAMMING	CP	2	0	2	3.0
5		ELECTIVE-I	PE	2	0	2	3.0
		TOTAL		**12**	**0**	**10**	**17.0**

SEMESTER-VI

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS701	THEORY OF COMPUTATION	CP	3	1	0	3.5
2	UCS505	COMPUTER GRAPHICS	CP	3	0	2	4.0
3	UCS619	QUANTUM COMPUTING	CP	3	0	2	4.0
4		ELECTIVE-II	PE	2	0	2	3.0
5		ELECTIVE-III	PE	2	0	2	3.0
6		GENERIC ELECTIVE	GE	3	0	0	3.0
7	UCS797	CAPSTONE PROJECT* – STARTS	PR	1*	0	2	-
		TOTAL		**16+1***	**1**	**10**	**20.5**

*Alternate week

SEMESTER-VII

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS802	COMPILER CONSTRUCTION	CP	3	0	2	4.0
2	UHU005	HUMANITIES FOR ENGINEERS	CF	2	0	2	3.0
3	UCS712	COGNITIVE COMPUTING	CP	2	0	0	2.0
4		ELECTIVE-IV	PE	2	0	2	3.0
5	UCS797	CAPSTONE PROJECT	PR	0	0	2	8.0
		TOTAL		**9**	**0**	**8**	**20.0**

SEMESTER-VIII

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS898	PROJECT SEMESTER*	PR	-	-	-	15.0
		TOTAL		-	-	-	15.0

*TO BE CARRIED OUT IN INDUSTRY/RESEARCH INSTITUTION

OR

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS813	SOCIAL NETWORK ANALYSIS	CP	2	0	2	3.0
2	UCS806	ETHICAL HACKING	CP	3	0	2	4.0
3	UCS899	PROJECT	PR	-	-	0	8.0
		TOTAL		**5**	**0**	**4**	**15.0**

OR

S. N.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS900	Start- up semester**		-	-	-	15.0
		TOTAL		-	-	-	15.0

** BASED ON HANDS ON WORK ON INNOVATIONS AND ENTREPRENEURSHIP

Elective Focus

B.E. Computer Science and Engineering Program is designed to offer elective focus as soon as student clears semester IV of the program. Student has to choose EF (Elective Focus) out of the following nine choices and shall continue with this group till his study at Thapar Institute of Engineering & Technology. Choices are:

I. Computer Animation and Gaming
II. Information and Cyber Security
III. Mathematics and Computing
IV. Data Science
V. Financial Derivative (Future First Collaboration)
VI. DevOps and Continuous Delivery (Xebia Collaboration)
VII. Full Stack (Xebia Collaboration)
VIII. Conversational AI (NVIDIA Collaboration)
IX. Robotics and Edge AI (NVIDIA Collaboration)

I. Computer Animation and Gaming

1. Computer Vision
2. 3D Modelling and Animation
3. Game Design & Development
4. Augmented and Virtual Reality

II. Information and Cyber Security

1. Computer & Network Security
2. Secure Coding
3. Cyber Forensics
4. Blockchain Technology and Applications

III. Mathematics and Computing

1. Mathematic Modeling and Simulation
2. Matrix Computation
3. Financial Mathematics
4. Computational Number Theory

IV. Data Science

1. Data Science Fundamentals
2. Predictive Analytics Using Statistics
3. AI Applications – NLP, Computer Vision, IoT
4. Building Innovative Systems

V. Financial Derivative (Future First Collaboration)

1. Finance, Accounting and Valuation
2. Financial and Derivative Markets
3. Derivatives Pricing, Trading and Strategies
4. Quantitative and Statistical Methods for Finance

VI. DevOps and Continuous Delivery (Xebia Collaboration)

1. Source Code Management
2. Build and Release Management
3. Continuous Integration and Continuous Deployment
4. System Provisioning and Configuration Management

VII. Full Stack (Xebia Collaboration)

1. UI & UX Specialist
2. Database Engineer
3. Test Automation
4. Cloud & DevOps

VIII. Conversational AI (NVIDIA Collaboration)

1. Conversational AI: Data Analytics
2. Conversational AI: Data Science
3. Conversational AI: Natural Language Processing
4. Conversational AI: Speech Processing

IX. Robotics and Edge AI (NVIDIA Collaboration)

1. Edge AI and Robotics: Data Science
2. Edge AI and Robotics: Basic Computer Vision
3. Edge AI and Robotics: Advanced Computer Vision
4. Edge AI and Robotics: RL & Conversational AI

Elective-I

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS532	COMPUTER VISION	PE	2	0	2	3.0
2	UCS534	COMPUTER & NETWORK SECURITY	PE	2	0	2	3.0
3	UMC512	MATHEMATIC MODELING AND SIMULATION	PE	2	0	2	3.0
4	UCS538	DATA SCIENCE FUNDAMENTALS	PE	2	0	2	3.0
5	UCS539	FINANCE, ACCOUNTING AND VALUATION	PE	2	0	2	3.0
6	UCS537	SOURCE CODE MANAGEMENT	PE	2	0	2	3.0
7	UCS542	UI & UX SPECIALIST	PE	2	0	2	3.0
8	UCS543	CONVERSATIONAL AI: DATA ANALYTICS	PE	2	0	2	3.0
9	UCS544	EDGE AI AND ROBOTICS: DATA SCIENCE	PE	2	0	2	3.0

Elective-II

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS636	3D MODELLING AND ANIMATION	PE	2	0	2	3.0
2	UCS638	SECURE CODING	PE	2	0	2	3.0
3	UMC622	MATRIX COMPUTATION	PE	2	0	2	3.0
4	UCS654	PREDICTIVE ANALYTICS USING STATISTICS	PE	2	0	2	3.0
5	UCS657	FINANCIAL AND DERIVATIVE MARKETS	PE	2	0	2	3.0
6	UCS659	BUILD AND RELEASE MANAGEMENT	PE	2	0	2	3.0
7	UCS661	DATABASE ENGINEER	PE	2	0	2	3.0
8	UCS663	CONVERSATIONAL AI: DATA SCIENCE	PE	2	0	2	3.0
9	UCS665	EDGE AI AND ROBOTICS: BASIC COMPUTER VISION	PE	2	0	2	3.0

Elective-III

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS646	GAME DESIGN & DEVELOPMENT	PE	2	0	2	3.0
2	UCS648	CYBER FORENSICS	PE	2	0	2	3.0
3	UMC632	FINANCIAL MATHEMATICS	PE	2	0	2	3.0
4	UCS655	AI APPLICATIONS – NLP, COMPUTER VISION, IOT	PE	2	0	2	3.0
5	UCS658	DERIVATIVES PRICING, TRADING AND STRATEGIES	PE	2	0	2	3.0
6	UCS660	CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT	PE	2	0	2	3.0
7	UCS662	TEST AUTOMATION	PE	2	0	2	3.0
8	UCS664	CONVERSATIONAL AI: NATURAL LANGUAGE PROCESSING	PE	2	0	2	3.0
9	UCS666	EDGE AI AND ROBOTICS: ADVANCED COMPUTER VISION	PE	2	0	2	3.0

Elective-IV

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS752	AUGMENTED AND VIRTUAL REALITY	PE	2	0	2	3.0
2	UCS754	BLOCKCHAIN TECHNOLOGY AND APPLICATIONS	PE	2	0	2	3.0
3	UMC742	COMPUTATIONAL NUMBER THEORY	PE	2	0	2	3.0
4	UCS757	BUILDING INNOVATIVE SYSTEMS	PE	2	0	2	3.0
5	UMC743	QUANTITATIVE AND STATISTICAL METHODS FOR FINANCE	PE	2	0	2	3.0
6	UCS758	SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT	PE	2	0	2	3.0
7	UCS745	CLOUD & DEVOPS	PE	2	0	2	3.0
8	UCS746	CONVERSATIONAL AI: SPEECH PROCESSING	PE	2	0	2	3.0
9	UCS747	EDGE AI AND ROBOTICS: RL & CONVERSATIONAL AI	PE	2	0	2	3.0

SEMESTER WISE CREDITS FOR BE: COMPUTER SCIENCE AND ENGINEERING

Nature of Course	Credits to be Earned(As per Choice Based Credit System)								
	Semesters								Total
	I	II	III	IV	V	VI	VII	VIII	
Core-Foundation Courses	18	18	4	4	0	0	3	0	47
Core-Professional Courses	4	4	16.5	19	14	11.5	6	0	75
Professional & Generic Electives	0	0	0	0	3	9	3	0	15
Project Based Courses	0	0	0	0	0	0	8	15	23
								Total	160

UCB008: APPLIED CHEMISTRY

L	T	P	Cr
3	1	2	4.5

Course Objectives: The course aims at elucidating principles of applied chemistry in industrial systems, water treatment, engineering materials and analytical techniques.

Electrochemistry: Specific, equivalent and molar conductivity of electrolytic solutions, migration of ions, transference number and its determination by Hittorf's method, conductometric titrations, types of electrodes, concentration cells, liquid junction potential.

Phase Rule: States of matter, phase, component and degree of freedom, Gibb's phase rule, one component and two component systems.

Water Treatment and Analysis: Hardness and alkalinity of water: units and determination, external and internal methods of softening of water: carbonate, phosphate, calgon and colloidal conditioning, lime-soda process, zeolite process, ion exchange process, mixed bed deionizer, desalination of brackish water.

Fuels: Classification of fuels, calorific value, cetane and octane number, fuel quality, comparison of solid liquid and gaseous fuels, properties of fuel, alternative fuels: biofuels, power alcohol, synthetic petrol.

Chemistry of Polymers: Overview of polymers, types of polymerization, molecular weight determination, tacticity of polymers, catalysis in polymerization, conducting, biodegradable and inorganic polymers.

Atomic spectroscopy: Introduction to spectroscopy, atomic absorption spectrophotometry and flame photometry, quantitative methods.

Molecular Spectroscopy: Beer-Lambert`s Law, molecular spectroscopy, principle, instrumentation and applications of UV-Vis and IR spectroscopy.

Laboratory Work:

- Electrochemical measurements: Experiments involving use of pH meter, conductivity meter, potentiometer.
- Acid and Bases: Determination of mixture of bases.
- Spectroscopic techniques: Colorimeter, UV-Vis spectrophotometer.
- Water and its treatment: Determination of hardness, alkalinity, chloride, chromium, iron and copper in aqueous medium.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

The students will be able to reflect on:

1. Concepts of electrodes in electrochemical cells, migration of ions, liquid junction potential and conductometric titrations.
2. Atomic and molecular spectroscopy fundamentals like Beer`s law, flame photometry, atomic absorption spectrophotometry, UV-Vis and IR.
3. Water and its treatment methods like lime soda and ion exchange.
4. Concept of phase rule, fuel quality parameters and alternative fuels.
5. Polymerization, molecular weight determination and applications as biodegradable and conducting polymers.
6. Laboratory techniques like pH metry, potentiometry, colourimetry, conductometry and volumetry.

Text Books:

1. Ramesh, S. and Vairam S. Engineering Chemistry, Wiley India (2012) 1st ed.
2. Puri, B.R., Sharma, L.R., and Pathania, M.S. Principles of Physical Chemistry, Vishal Publishing Co. (2008).
3. Aggarwal, S. Engineering Chemistry: Fundamentals and Applications, Cambridge University Press (2015).

Reference Books:

1. Brown, H., Chemistry for Engineering Students, Thompson, 1st ed
2. Sivasankar, B., Engineering Chemistry, Tata McGraw-Hill Pub. Co. Ltd, New Delhi (2008).
3. Shulz, M.J. Engineering Chemistry, Cengage Learnings (2007) 1st ed.

UTA003: COMPUTER PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course is designed to explore computing and to show students the art of computer programming. Students will learn some of the design principles for writing good programs.

Computers Fundamentals: Binary Number System, Computer memory, Computer Software.

Algorithms and Programming Languages: Algorithm, Flowcharts, Generation of Programming Languages.

C Language: Structure of C Program, Life Cycle of Program from Source code to Executable, Compiling and Executing C Code, Keywords, Identifiers, Primitive Data types in C, variables, constants, input/output statements in C, operators, type conversion and type casting. Conditional branching statements, iterative statements, nested loops, break and continue statements.

Functions: Declaration, Definition, Call and return, Call by value, Call by reference, showcase stack usage with help of debugger, Scope of variables, Storage classes, Recursive functions, Recursion vs. Iteration.

Arrays, Strings and Pointers: One-dimensional, Two-dimensional and Multi-dimensional arrays, operations on array: traversal, insertion, deletion, merging and searching, Inter-function communication via arrays: passing a row, passing the entire array, matrices. Reading, writing and manipulating Strings, understanding computer memory, accessing via pointers, pointers to arrays, dynamic allocation, drawback of pointers.

Structures and Union: Defining a Structure, declaring a structure variables, Accessing Structure Elements, and Union.

File Handling: Defining and Opening a File, closing a File, reading from a File, Writing into a File.

Laboratory Work:
To implement Programs for various kinds of programming constructs in C Language.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Comprehend and analyze the concepts of number system, memory, compilation and debugging of the programs in C language.
2. Understanding of the fundamental data types, operators and console I/O functions as an aspect of programs.
3. Design and create programs involving control flow statements, arrays, strings and implement the concept of dynamics of memory allocations.
4. Evaluate and analyze the programming concepts based on user define data types and file handling using C language.

Text Books:

1. Brian W. Kernighan Dennis M. Ritchie, C Programming Language, 2nd ed, 2012.
2. Balagurusamy G., Programming in ANSI C, 8th ed., 2019

Reference Books:

1. Kanetkar Y., Let Us C, 16th ed.,2017

UES013 ELECTRICAL AND ELECTRONICS ENGINEERING

L	T	P	Cr
3	1	2	4.5

Course Objective: To introduce basic concepts of electrical and electronics engineering.

DC Circuits: Basic electrical quantities, Electric circuit sources and circuit elements and their behavior (Active and Passive), Kirchhoff's voltage and current laws, D.C.

Networks: Mesh and Nodal analysis, Stardelta transformation, Source transformation, Superposition theorem, Thevenin's theorem, Norton's theorem, Maximum power transfer theorem. First Order Circuits, Source free RC and RL circuits, Step response of RL and RC Circuits, time constant of RC and RL circuits.

AC Circuits: Concept of Phasors, Phasor representation of circuit elements, Complex notation representation, Series and parallel circuits, Power and power factors, Resonance in series and parallel circuits, balanced 3-phase voltage, Current and power relations, Unbalanced 3-phase circuits, 3-phase power measurement.

Measurement Science: Introduction to measurement systems, functional elements of a measurement system, Analog and digital measurement, Permanent magnet moving coil meters, current and voltage measurement, power and energy measurement.

Junction Diodes: Semiconductor materials, An n-type semiconductor, An p-type semiconductor, Junction diode, Construction and static characteristic of a junction diode, Diode as half wave and full wave rectifier, Zener diode as voltage regulator, Schottky diode.

Transistors: Bipolar junction transistor, construction, operation, Configuration of transistor in CB, CC and CE modes, Transistor as a switch and as an amplifier. Field effect transistor-construction and operation.

Operational Amplifier: Introduction to Operational amplifier, Ideal Op-Amp, Inverting Amplifier, Non inverting amplifier, summing amplifier, Difference amplifier, Applications of Op-Amp.

Laboratory Work

Kirchhoff's laws, Network theorems, A.C. series a parallel circuit, Three Phase balanced circuit power measurement, testing of components, Resonance in AC circuit, Use of diode as rectifier. BJT Characteristics.

Course Learning Outcome (CLO):

On completion of the course, the student would be able to:

1. Apply basic networks laws and theorems to solve different electric circuits.

2. Analyze transient and steady state response of DC circuits.

3. Examine the basic measurement principals of various electrical quantities.

4. Explain and analyses the behavior of different types of transistors and diodes.

5. Elucidate the principle and working of an operational amplifier.

Textbooks

1. Hughes, E., Smith, I.M., Hiley, J. and Brown, K., Electrical and Electronic Technology, Prentice

Hall (2008) 10th ed.

2. Nagrath, I.J. and Kothari, D.P., Basic Electrical Engineering, Tata McGraw Hill (2002).

3. Naidu, M.S. and Kamashaiah, S., Introduction to Electrical Engineering, Tata McGraw Hill

(2007).

4. Halkias, M., Integrated Electronics, Tata McGraw Hill (2011).

5. Doeblelin, E.O., Measurement Systems Application and Design, Tata McGraw Hill (2007).

Reference Books

1. Chakraborti, A., Basic Electrical Engineering, Tata McGraw−Hill (2008).

2. Del Toro, V., Electrical Engineering Fundamentals, Prentice−Hall of India Private Limited

(2004).

3. David Bell, Electronics Devices and Circuits, Oxford Publications (2009).

UEN002: ENERGY AND ENVIRONMENT

L	T	P	Cr
3	0	0	3.0

Course Objectives: The exposure to this course would facilitate the students in understanding the terms, definitions and scope of environmental and energy issues pertaining to current global scenario; understanding the value of regional and global natural and energy resources; and emphasize on need for conservation of energy and environment.

Introduction: Natural Resources & amp; its types, Concept of sustainability and sustainable use of natural resources, Pollution based environmental issues and case-studies.

Conventions on Climate Change: Origin of Conference of Parties (COPs), United Nations Framework Convention on Climate Change (UNFCCC) and Intergovernmental Panel on Climate Change (IPCC); Kyoto Protocol, instruments of protocol – CDM, JI and IET; Montreal Action Plan; Paris Agreement and post-Paris scenario.

Air Pollution: Origin, Sources and effects of air pollution; Primary and secondary meteorological parameters; Wind roses; Atmospheric Stability; Inversion; Plume behavior; Management of air pollution: Source reduction and Air Pollution Control Devices for particulates and gaseous pollutants in stationary and mobile sources.

Water Pollution: Origin, Sources of water pollution, Category of water pollutants, Physico-Chemical characteristics, Components of wastewater treatment systems, Advanced treatment technologies.

Solid Waste Management: Introduction to solid waste management, Sources, characteristics of municipal and industrial solid waste, Solid waste management methods: Incineration, composting, Biomethanation, landfill, E-waste management, Basal convention.

Energy Resources: Classification of Energy Resources; Conventional energy, resources-Coal, petroleum and natural gas, nuclear energy, hydroelectric power; Non-conventional energy resources- Biomass energy, Thermo-chemical conversion and biochemical conversion route; Generation of Biogas and biodiesel as fuels; Solar energy-active and passive solar energy absorption systems; Type of collectors; Thermal and photo conversion applications; Wind energy.

Facilitated through Online Platforms:

Ecology and Environment: Concept of an ecosystem; structural and functional units of an ecosystem; Food Chain, Food Web, Trophic Structures and Pyramids; Energy flow; Ecological Succession; Types, Characteristics, Biodiversity, Biopiracy.

Human Population and the Environment: Population growth, variation among nations; Population explosion – Family Welfare Programmes; Environment and human health; Human Rights; Value Education; Women and Child Welfare; Role of Information Technology in Environment and Human Health, Environmental Ethics.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Comprehend the interdisciplinary context with reference to the environmental issues and case studies.
2. Assess the impact of anthropogenic activities on the various elements of environment and apply suitable techniques to mitigate their impact.
3. Conceptualize and explain the structural and functional features of ecological systems.
4. Correlate environmental concerns with the conventional energy sources associated and assess the uses and limitations of non-conventional energy technologies.

Recommended Books:

1. Moaveni, S., Energy, Environment and Sustainability, Cengage (2018)
2. Down to Earth, Environment Reader for Universities, CSE Publication (2018)
3. Chapman, J.L. and Reiss, M.J., Ecology - Principles and Application, Cambridge University Press (LPE) (1999).
4. Eastop, T.P. and Croft, D.R., Energy Efficiency for Engineers and Technologists, Longman and Harow (2006).
5. O'Callagan, P.W., Energy Management, McGraw Hill Book Co. Ltd. (1993).
6. Peavy H.S. and Rowe D.R. Environmental Engineering, McGraw Hill (2013).

UMA010: MATHEMATICS – I

L	T	P	Cr
3	**1**	**0**	**3.5**

Course Objectives: To provide students with skills and knowledge in sequence and series, advanced calculus, calculus of several variables and complex analysis which would enable them to devise solutions for given situations they may encounter in their engineering profession.

Partial Differentiation: Functions of several variables, Limits and continuity, Chain rule, Change of variables, Partial differentiation of implicit functions, Directional derivatives and its properties, Maxima and minima by using second order derivatives

Multiple Integrals: Double integral (Cartesian), Change of order of integration in double integral, Polar coordinates, graphing of polar curves, Change of variables (Cartesian to polar), Applications of double integrals to areas and volumes, evaluation of triple integral (Cartesian).

Sequences and Series: Introduction to sequences and Infinite series, Tests for convergence/divergence, Limit comparison test, Ratio test, Root test, Cauchy integral test, Alternating series, Absolute convergence and conditional convergence.

Series Expansions: Power series, Taylor series, Convergence of Taylor series, Error estimates, Term by term differentiation and integration.

Complex analysis: Introduction to complex numbers, geometrical interpretation, functions of complex variables, examples of elementary functions like exponential, trigonometric and hyperbolic functions, elementary calculus on the complex plane (limits, continuity, differentiability), Cauchy-Riemann equations, analytic functions, harmonic functions.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
Upon completion of this course, the students will be able to

1. Examine functions of several variables, define and compute partial derivatives, directional derivatives and their use in finding maxima and minima in some engineering problems.
2. Evaluate multiple integrals in Cartesian and Polar coordinates, and their applications to engineering problems.
3. Determine the convergence/divergence of infinite series, approximation of functions using power and Taylor's series expansion and error estimation.
4. Represent complex numbers in Cartesian and Polar forms and test the analyticity of complex functions by using Cauchy-Riemann equations.

Text Books:

1. Thomas, G.B. and Finney, R.L., Calculus and Analytic Geometry, Pearson Education (2007), 9th ed.
2. Stewart James, Essential Calculus; Thomson Publishers (2007), 6th ed.
3. Kasana, H.S., Complex Variables: Theory and Applications, Prentice Hall India, 2005 (2nd edition).

Reference Books:

1. Wider David V, Advanced Calculus: Early Transcendentals, Cengage Learning (2007).
2. Apostol Tom M, Calculus, Vol I and II, John Wiley (2003).
3. Brown J.W and Chruchill R.V, Complex variables and applications, McGraw Hill, (7th edition)

UES009: MECHANICS

L	T	P	Cr
2	1	2*	2.5

(*: Two hours lab once in Semester)

Course Objectives: The objective of this module is to help students develop the techniques needed to solve general engineering mechanics problems. Students will learn to describe physical systems mathematically so that their behavior can be predicted.

Review of Newton's law of motion and vector algebra.

Equilibrium of Bodies: Free-body diagrams, conditions of equilibrium, torque due to a force, statical determinacy.

Plane Trusses: Forces in members of a truss by method of joints and method of sections.

Friction: Sliding, belt, screw and rolling.

Properties of Plane Surfaces: First moment of area, centroid, second moment of area etc.

Shear Force and Bending Moment Diagrams: Types of load on beams, classification of beams; axial, shear force and bending moment diagrams: simply supported, overhung and cantilever beams subjected to any combination of point loads, uniformly distributed and varying load and moment.

Virtual Work: Principle of virtual work, calculation of virtual displacement and virtual work.

Experimental Project Assignment/ Micro Project:
Students in groups of 4/5 will do project on Model Bridge Experiment: This will involve construction of a model bridge using steel wire and wood.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
The students will be able to:

1. Determine resultants in plane force systems
2. Identify and quantify all forces associated with a static framework
3. Draw Shear Force Diagram and Bending Moment Diagram in various kinds of beams subjected to different kinds of loads

Text Books:

1. Shames, I. H. Engineering Mechanics: Dynamics, Pearson Education India (2006).
2. Beer, Johnston, Clausen and Staab, Vector Mechanics for Engineers, Dynamics, McGraw-Hill Higher Education (2003).

Reference Books:

1. Hibler, T.A., Engineering Mechanics: Statics and Dynamics, Prentice Hall (2012).
2. Timoshenko and Young, Engineering Mechanics, Tata McGraw Hill Education Private Limited, (2006).

UPH004: APPLIED PHYSICS

L	T	P	Cr
3	1	2	4.5

Course Objectives: To introduce the student to the basic physical laws of oscillators, acoustics of buildings, ultrasonics, electromagnetic waves, wave optics, lasers, and quantum mechanics and demonstrate their applications in technology. To introduce the student to measurement principles and their application to investigate physical phenomena

Oscillations and Waves: Oscillatory motion and damping, Applications - Electromagnetic damping – eddy current; ***Acoustics***: Reverberation time, absorption coefficient, Sabine's and Eyring's formulae (Qualitative idea), Applications - Designing of hall for speech, concert, and opera; ***Ultrasonics***: Production and Detection of Ultrasonic waves, Applications - green energy, sound signaling, dispersion of fog, remote sensing, Car's airbag sensor.

Electromagnetic Waves: Scalar and vector fields; Gradient, divergence, and curl; Stokes' and Green's theorems; Concept of Displacement current; Maxwell's equations; Electromagnetic wave equations in free space and conducting media, Application - skin depth.

Optics: ***Interference***: Parallel and wedge-shape thin films, Newton rings, Applications as Non-reflecting coatings, Measurement of wavelength and refractive index. ***Diffraction***: Single and Double slit diffraction, and Diffraction grating, Applications - Dispersive and Resolving Powers. ***Polarization***: Production, detection, Applications – Anti-glare automobile headlights, Adjustable tint windows. ***Lasers***: Basic concepts, Laser properties, Ruby, HeNe, and Semiconductor lasers, Applications – Optical communication and Optical alignment.

Quantum Mechanics: Wave function, Steady State Schrodinger wave equation, Expectation value, Infinite potential well, Tunneling effect (Qualitative idea), Application - Quantum computing.

Laboratory Work:
1. Determination of damping effect on oscillatory motion due to various media.
2. Determination of velocity of ultrasonic waves in liquids by stationary wave method.
3. Determination of wavelength of sodium light using Newton's rings method.
4. Determination of dispersive power of sodium-D lines using diffraction grating.
5. Determination of specific rotation of cane sugar solution.
6. Study and proof of Malus' law in polarization.
7. Determination of beam divergence and beam intensity of a given laser.
8. Determination of displacement and conducting currents through a dielectric.
9. Determination of Planck's constant.

Micro project:
Students will be given physics-based projects/assignments using computer simulations, etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
Upon completion of this course, students will be able to:
1. Understand damped and simple harmonic motion, the role of reverberation in designing a hall and generation and detection of ultrasonic waves.
2. Use Maxwell's equations to describe propagation of EM waves in a medium.
3. Demonstrate interference, diffraction and polarization of light.

4. Explain the working principle of Lasers.
5. Use the concept of wave function to find probability of a particle confined in a box.

Text Books:

1. Beiser, A., Concept of Modern Physics, Tata McGraw Hill (2007) 6^{th} ed.
2. Griffiths, D.J., Introduction to Electrodynamics, Prentice Hall of India (1999) 3^{rd} ed.
3. Jenkins, F.A. and White, H.E., Fundamentals of Optics, McGraw Hill (2001) 4^{th} ed.

Reference Books:

1. Wehr, M.R, Richards, J.A., Adair, T.W., Physics of The Atom, Narosa Publishing House (1990) 4^{th} ed.
2. Verma, N.K., Physics for Engineers, Prentice Hall of India (2014)1^{st} ed.
3. Pedrotti, Frank L., Pedrotti, Leno S., and Pedrotti, Leno M., Introduction to Optics, Pearson Prentice Hall TM (2008) 3^{rd} ed.

UTA018: OBJECT ORIENTED PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To become familiar with object oriented programming concepts and be able to apply these concepts in solving diverse range of applications.

Objects and Classes: Structure in C and C++, Class specification, Objects, Data hiding, Encapsulation and abstraction, namespaces, Array of objects, Passing objects as arguments, Returning object from a function, inline functions, Static data member and member function, ‘const’ member function.

Constructor and Destructor: Constructors, Parameterized Constructors, Constructor Overloading, Constructors in array of objects, Constructors with default arguments, Dynamic Initialization, Pointer to objects, this pointer, Dynamic memory allocation, Array of pointer to objects, Copy Constructor, Static objects, Friend function, and Friend classes.

Operator Overloading and Type Conversion: Syntax of operator overloading, Overloading Unary operator and Binary operator, Overloading arithmetic operator, relational operator, Overloading Unary operator and Binary operator using friend function, Data conversion, Overloading some special operators like (), []

Inheritance: Derived Class declaration, Public, Private and Protected Inheritance, friend function and Inheritance, Overriding member function, Forms of inheritance, virtual base class, Abstract class, Constructor and Inheritance, Destructor and Inheritance, Advantage and disadvantage of Inheritance.

Polymorphism: Classification of Polymorphism, Compile time and Run time Polymorphism, Pointers to derived class object, Virtual functions, Pure virtual functions.

File handling: Formatted I/O, Hierarchy of file stream classes, Opening and closing a file, Working with multiple files, file modes, file pointers, Text vs Binary Files.

Templates: Need of template, Function templates, Function template with non-type parameter, Overloading function templates, Class templates, Class template with non-type parameter.

Exception Handling: Exception handling mechanism, Multiple Catch Blocks, Catch All exceptions, Throw an exception, Exception Specification.

Standard Template Library: Fundamental idea about string, iterators, hashes and other types, The String and Vector classes vs C-style pointers

Laboratory work:
To implement Programs for various kinds of programming constructs in C++ Language.

Course Learning Outcomes (CLOs):

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

1. Understand the basic concept of Classes, objects and Object Orientation, with basic layout of an object oriented program.
2. Comprehend the concept of constructors and destructors.
3. Demonstrate the prime concepts *viz.* overloading, polymorphism, abstraction and Inheritance of an object oriented paradigm.
4. Grasp the File handling concepts and be able to use files.
5. Use template and Exception handling in an object oriented programming.

Text Books:

1. Schildt H., C++: The Complete Reference, Tata McGraw Hill (2003) 4th ed.
2. Lippman B. S., Lajoie J., and Moo E. B., C++ Primer, Addison-Wesley Professional (2013) 5th ed.

Reference books:

1. Lafore R., Object-Oriented Programming in C++, Pearson Education (2002) 4th ed.
2. E Balagurusamy, Object Oriented Programming with C++ (2017) 7th ed.
3. Stroustrup B., The C++ programming language, Pearson Education India (2013) 4th ed.

6.

UTA026: MANUFACTURING PROCESSES

L	T	P	Cr
2	0	2	3

Course Objectives: This course introduces the basic concepts of manufacturing via machining, forming, joining, casting and assembly, enabling the students to develop a basic knowledge of the mechanics, operation and limitations of basic machining tools. The course also introduces the concept of metrology and measurement of parts.

Machining Processes: Principles of metal cutting, Cutting tools, Cutting tool materials and applications, Geometry of single point cutting tool, Introduction to multi-point machining processes – milling, drilling and grinding, Tool Life, Introduction to computerized numerical control (CNC) machines, G and M code programming for simple turning and milling operations, introduction of canned cycles.

Metal Casting: Principles of metal casting, Introduction to sand casting, Requisites of a sound casting, Permanent mold casting processes.

Metal Forming: Forging, Rolling, Drawing, Extrusion, Sheet Metal operations. Joining Processes: Electric arc, Resistance welding, Soldering, Brazing.

Laboratory Work:
Relevant shop floor exercises involving practices in Sand casting, Machining, Welding, Sheet metal fabrication techniques, CNC turning and milling exercises, Experiments on basic engineering metrology and measurements to include measurements for circularity, ovality, linear dimensions, profiles, radius, angular measurements, measurement of threads, surface roughness.

Basic knowledge and derivations related to above measurements, uncertainties, statistical approaches to estimate uncertainties, Line fitting, static and dynamic characteristics of instruments will be discussed in laboratory classes.

Assignments:
Assignments for this course will include the topics: Manufacturing of micro- chips used in IT and electronics industry and use of touch screens. Another assignment will be given to practice numerical exercises on topics listed in the syllabus.

Micro Project:
Fabrication of multi-operational jobs using the above processes as per requirement by teams consisting of 4-6 members. The use of CNC machines must be part of micro project. Quality check should be using the equipment available in metrology lab.

Course Learning outcomes (CLOs) / Course Objectives (COs):

After the completion of this module, students will be able to:

1. Develop simple CNC code, and use it to produce components while working in groups.
2. Analyse various machining processes and calculate relevant quantities such as velocities, forces.
3. Recognise cutting tool wear and identify possible causes and solutions.
4. Understand the basic principle of bulk and sheet metal forming operations for analysis of forces.
5. Analyse various shearing operations for tooling design.
6. Apply the knowledge of metal casting for different requirements.
7. Analyse and understand the requirements to achieve sound welded joint while welding different similar and dissimilar engineering materials.

Text Books:

1. Degarmo, E. P., Kohser, Ronald A. and Black, J. T., Materials and Processes in Manufacturing, Prentice Hall of India (2008) 8^{th} ed.
2. Kalpakjian, S. and Schmid, S. R., Manufacturing Processes for Engineering Materials, Dorling Kingsley (2006) 4^{th} ed.

Reference Books:

1. Martin, S.I., Chapman, W.A.J. , Workshop Technology, Vol.1 & II, Viva Books (2006) 4^{th} ed.
2. Zimmer, E.W. and Groover, M.P., CAD/CAM - Computer Aided Designing and Manufacturing, Dorling Kingsley (2008).
3. Pandey, P.C. and Shan, H. S., Modern Machining Processes, Tata McGraw Hill (2008).
4. Mishra, P. K., Non-Conventional Machining, Narosa Publications (2006).
5. Campbell, J.S., Principles of Manufacturing, Materials and Processes, Tata McGraw Hill Company (1999).
6. Lindberg, Roy A., Processes and Materials of Manufacture, Prentice Hall of India (2008) 4^{th} ed.

UTA015: ENGINEERING DRAWING

L	T	P	Cr
2	4	0	4.0

Course Objectives: This module is dedicated to graphics and includes two sections: manual drawing and AutoCAD. This course is aimed at to make the student understand dimensioned projections, learn how to create two-dimensional images of objects using first and third angle orthographic projection as well as isometric, perspective and auxiliary projection, to interpret the meaning and intent of tolerance dimensions and geometric tolerance symbolism and to create and edit drawings using drafting software AutoCAD.

Engineering Drawing

1. Introduction
2. Orthographic Projection: First angle and third angle projection system
3. Isometric Projections
4. Auxiliary Projections
5. Perspective Projections
6. Introduction to Mechanical Drawing
7. Sketching engineering objects
8. Sections, dimensions and tolerances

AutoCAD

1. Management of screen menus commands
2. Introduction to drawing entities
3. Co-ordinate systems: Cartesian, polar and relative coordinates
4. Drawing limits, units of measurement and scale
5. Layering: Organizing and maintaining the integrity of drawings
6. Design of prototype drawings as templates.
7. Editing / modifying drawing entities: Selection of objects, object snap modes, editing commands
8. Dimensioning: Use of annotations, dimension types, properties and placement, adding text to drawing

Micro Projects/Assignments:

1. Completing the views – Identification and drawing of missing lines in the projection of objects.
2. Missing views – Using two views to draw the projection of the object in the third view, primarily restricting to Elevation, Plan and Profile views.
3. Projects related to orthographic and isometric projections.
   a. Using wax blocks or soap bars to develop three dimensional object from given orthographic projections.
   b. Using wax blocks or soap bars to develop three dimensional object, section it and color the section.

c. Use of AUTOCAD as a complementary tool for drawing the projections of the objects created in (1) and (2).

4. Develop the lateral surface of different objects involving individual or a combination of solids like Prism, Cone, Pyramid, Cylinder, Sphere etc.
5. To draw the detailed and assembly drawings of simple engineering objects / systems with due sectioning (wherever required) along with bill of materials e.g. Rivet joints, simple bearing, wooden joints, Two plates connected with nut and bolt etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

1. Upon completion of this module, students will be able to:
2. Creatively comprehend geometrical details of common engineering objects.
3. Draw dimensioned orthographic and isometric projections of simple engineering objects.
4. Draw sectional views of simple engineering objects.
5. Interpret the meaning and intent of toleranced dimensions and geometric tolerance symbolism.
6. Create and edit dimensioned drawings of simple engineering objects using AutoCAD.
7. Organize drawing objects using layers and setting up of templates in AutoCAD.

Text Books:

1. Jolhe, D.A., Engineering Drawing, Tata McGraw Hill, 2008.
2. Davies, B. L., Yarwood, A., Engineering Drawing and Computer Graphics, Van Nostr and Reinhold (UK), 1986.

Reference Books:

1. Gill, P.S., Geometrical Drawings, S.K. Kataria & Sons, Delhi (2008).
2. Gill, P.S., Machine Drawings, S.K. Kataria & Sons, Delhi (2013).
3. Mohan, K.R., Engineering Graphics, Dhanpat Rai Publishing Company (P) Ltd, Delhi (2002).
4. French, T.E., Vierck, C.J. and Foster, R.J., Fundamental of Engineering Drawing & Graphics Technology, McGraw Hill Book Company, New-Delhi (1986).
5. Rowan, J. and Sidwell, E. H., Graphics for Engineers, Edward Arnold, London (1968).

UHU003: PROFESSIONAL COMMUNICATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: To introduce the students to effective professional communication. The student will be exposed to effective communication strategies and different modes of communication. The student will be able to analyze his/ her communication behavior and that of the others. By learning and adopting the right strategies, the student will be able to apply effective communication skills, professionally and socially.

Effective communication: Meaning, Barriers, Types of communication and Essentials. Interpersonal Communication skills.

Effective Spoken Communication: Understanding essentials of spoken communication, Public speaking, Discussion Techniques, Presentation strategies.

Effective Professional and Technical writing: Paragraph development, Forms of writing, Abstraction and Summarization of a text; Technicalities of letter writing, internal and external organizational communication. Technical reports and proposals.

Effective non verbal communication: Knowledge and adoption of the right non verbal cues of body language, interpretation of the body language in professional context. Understanding Proxemics and other forms of non verbal communication.

Communicating for Employment: Designing Effective Job Application letter and resumes. Communication Networks in organizations: Types, barriers and overcoming the barriers.

Laboratory work:

1. Needs-assessment of spoken and written communication and feedback.
2. Training for Group Discussions through simulations and role plays.
3. Technical report writing on survey based projects.
4. Project based team presentations.

Course learning outcome (CLO) / Course Objectives (COs):

1. Apply communication concepts for effective interpersonal communication.
2. Select the most appropriate media of communication for a given situation.
3. Speak assertively and effectively.
4. Write objective organizational correspondence.
5. Design effective resumes, reports and proposals.

Text Books:

1. Lesikar R.V and Flately M.E., Basic Business Communication Skills for the Empowering the Internet Generation. Tata Mc Graw Hill. New Delhi (2006).

2. Raman, M & Sharma, S., Technical Communication Principles and Practice, Oxford University Press New Delhi.(2011).
3. Mukherjee H.S., Business Communication – Connecting at Work, Oxford University Press New Delhi, (2013).

Reference Books:

1. Butterfield, Jeff., Soft Skills for everyone, Cengage Learning New Delhi, (2013).
2. Robbins, S.P., & Hunsaker, P.L., Training in Interpersonal Skills, Prentice Hall of India, New Delhi, (2008).
3. DiSianza, J.J. & Legge, N.J., Business and Professional Communication, Pearson Education India New Delhi, (2009).

UMA004: MATHEMATICS – II

L	T	P	Cr
3	1	0	3.5

Course Objectives: To introduce students the theory and concepts of differential equations, linear algebra, Laplace transformations and Fourier series which will equip them with adequate knowledge of mathematics to formulate and solve problems analytically.

Linear Algebra: Row reduced echelon form, Solution of system of linear equations, Matrix inversion, Linear spaces, Subspaces, Basis and dimension, Linear transformation and its matrix representation, Eigen-values, Eigen-vectors and Diagonalisation, Inner product spaces and Gram-Schmidt orthogonalisation process.

Ordinary Differential Equations: Review of first order differential equations, Exact differential equations, Second and higher order differential equations, Solution techniques using one known solution, Cauchy - Euler equation, Method of undetermined coefficients, Variation of parameters method, Engineering applications of differential equations.

Laplace Transform: Definition and existence of Laplace transforms and its inverse, Properties of the Laplace transforms, Unit step function, Impulse function, Applications to solve initial and boundary value problems.

Fourier Series: Introduction, Fourier series on arbitrary intervals, Half range expansions, Applications of Fourier series to solve wave equation and heat equation.

Course learning outcome (CLO) / Course Objectives (COs):
Upon completion of this course, the students will be able to:

1. Solve the differential equations of first and second order and basic application problems described by these equations.
2. Find the Laplace transformations and inverse Laplace transformations for various functions. Using the concept of Laplace transform students will be able to solve the initial value and boundary value problems.
3. Find the Fourier series expansions of periodic functions and subsequently will be able to solve heat and wave equations.
4. Solve systems of linear equations by using elementary row operations.
5. Identify the vector spaces/subspaces and to compute their bases / orthonormal bases. Further, students will be able to express linear transformation in terms of matrix and find the Eigen values and Eigen vectors.

Text Books:

1. Simmons, G.F., Differential Equations (With Applications and Historical Notes), Tata McGraw Hill (2009).
2. Krishnamurthy, V.K., Mainra, V.P. and Arora, J.L., An introduction to Linear Algebra, Affiliated East West Press (1976).

Reference Books:

1. Kreyszig Erwin, Advanced Engineering Mathematics, John Wiley (2006), 8^{th} ed.
2. Jain, R.K. and Iyenger, S.R.K., Advanced Engineering Mathematics, Narosa Publishing House (2011), 11^{th} ed.

UCS303: OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: To understand the role, responsibilities, and algorithms involved for achieving various functionalities of an Operating System.

Introduction and System Structures: Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments, Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure.

Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, Multi-threaded programming: Multi-core Programming, Multithreading Models, Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Algorithm Evaluation.

Deadlock: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Memory Management: Basic Hardware, Address Binding, Logical and Physical Address, Dynamic linking and loading, Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table, Virtual Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Allocating Kernel Memory.

File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

Disk Management: Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Protection and Security: Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, The Security Problem, Program Threats, System and Network Threats, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks.

Concurrency: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

Laboratory work:
To explore detailed architecture and shell commands in Linux / Unix environment, and to simulate CPU scheduling, Paging, Disk-scheduling and process synchronization algorithms.

Course learning outcome (CLO) / Course Objectives (COs):
After the completion of the course, the student will be able to:
1. Explain the basic of an operating system viz. system programs, system calls, user mode and kernel mode.
2. Select a particular CPU scheduling algorithms for specific situation, and analyze the environment leading to deadlock and its rectification.
3. Explicate memory management techniques viz. caching, paging, segmentation, virtual memory, and thrashing.
4. Understand the concepts related to file systems, disk-scheduling, and security, protection.
5. Comprehend the concepts related to concurrency.

Text Books:
1. Silberschatz A., Galvin B. P. and Gagne G., Operating System Concepts, John Wiley & Sons Inc (2013) 9^{th} ed.
2. Stallings W., Operating Systems Internals and Design Principles, Prentice Hall (2018) 9^{th} ed.

Reference Books:
1. Bovet P. D., Cesati M., Understanding the Linux Kernel, O'Reilly Media (2006), 3^{rd} ed.
2. Kifer M., Smolka A. S., Introduction to Operating System Design and Implementation: The OSP 2 Approach, Springer (2007).

UCS405: DISCRETE MATHEMATICAL STRUCTURES

L	T	P	Cr
3	1	0	3.5

Course Objectives: Detailed study of various discrete and algebraic structures, basic logic, basics of counting and proof techniques.

Sets, Relations, and Functions: Sets: Operations on set, Inclusion-exclusion principle, Representation of Discrete Structures, Fuzzy set, Multi-set, bijective function, Inverse and Composition of functions, Floor and Ceiling functions, Growth of functions: Big-O notation, Big-Omega and Big-Theta Notations, Determining complexity of a program, Hashing functions, Recursive function, Functions applications.

Relations: Reflexivity, symmetry, transitivity, Equivalence and partial-ordered relations, Asymmetric, Irreflexive relation, Inverse and complementary relations, Partition and Covering of a set, N-ary relations and database, Representation relation using matrices and digraph, Closure of relations, Warshall's algorithm, Lexicographic ordering, Hasse diagram, Lattices, Boolean algebra, Application of transitive closure in medicine and engineering. Application: Embedding a partial order.

Graphs Theory: Representation, Type of Graphs, Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism, Graph Coloring, Covering and Partitioning, Max flow: Ford-Fulkerson algorithm, Application of Graph theory in real-life applications.

Basic Logic: Propositional logic, Logical connectives, Truth tables, Normal forms (conjunctive and disjunctive), Validity of well-formed formula, Propositional inference rules (concepts of modus ponens and modus tollens), Predicate logic, Universal and existential quantification.

Proof Techniques and counting: Notions of implication, equivalence, converse, inverse, contra positive, negation, and contradiction, The structure of mathematical proofs, Direct proofs, Disproving by counter example, Proof by contradiction, Induction over natural numbers, Structural induction, Weak and strong induction, The pigeonhole principle, Solving homogenous and heterogeneous recurrence relations.

Algebraic Structures: Group, Semi group, Monoids, Homomorphism, Congruencies, Ring, Field, Homomorphism, Congruencies, Applications of algebra to control structure of a program, the application of Residue Arithmetic to Computers.

Course learning outcome (CLO) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Perform operations on various discrete structures such as set, function and relation.
2. Apply basic concepts of asymptotic notation in analysis of algorithm.
3. Illustrate the basic properties and algorithms of graphs and apply them in modeling and solving real-world problems.
4. Comprehend formal logical arguments and translate statements from a natural language into its symbolic structures in logic.
5. Identify and prove various properties of rings, fields and group.

Text Books:

1. Rosen H. K., Discrete Mathematics and its Applications, McGraw Hill (2011) 7th ed.
2. Tremblay P. J. and Manohar, R., Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill (2008).

Reference Books:

1. Gallian A. J., Contemporary Abstract Algebra, Cengage Learning (2017) 9th ed.
2. Lipschutz S., Lipson M., Discrete Mathematics, McGraw-Hill (2007) 3rd ed.

UCS301: DATA STRUCTURES

L	T	P	Cr
3	0	2	4.0

Course Objectives: To become familiar with different types of data structures and their applications.

Analysing algorithms: Importance of efficient algorithms, Order arithmetic, time and space complexity.

Linear Data Structures: Arrays, Records, Strings and string processing, References and aliasing, Linked lists (Singly, Doubly, Circular), Strategies for choosing the appropriate data structure, Abstract data types, their implementation and applications: Stacks (using Arrays and Linked-list), Queues (using Arrays and Linked-list), Hash tables, including strategies for avoiding and resolving collisions, Dictionaries, Sets, Maps.

Searching and Sorting: Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Merge Sort, Counting Sort, Radix Sort. Introduction to internal, external, and distribution sorting techniques.

Trees and their applications: Binary search trees, AVL Tree, Splay Tree, Red-Black Tree, B Tree and B+ Tree, Common operations on these trees such as select min, select max, insert, delete, traversals, iterate over tree. Heaps, Heap Sort Priority Queue, Fibonacci heaps and Binomial Heaps.

Graphs and their applications: Graphs and graph algorithms, Representations of graphs, Depth- and breadth-first traversals, Shortest-path algorithms (Dijkstra and Floyd), Data Structures for Disjoint Sets, Minimum spanning tree (Prim and Kruskal).

Problem Classes: Introduction to P, NP, NP- Hard and NP-complete.

Laboratory work:
Implementation of Arrays, Recursion, Stacks, Queues, Lists, Binary trees, AVL trees, Splay trees, Sorting techniques, Searching techniques.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On completion of this course, the students will be able to
1. Implement basic data structures in solving fundamental problems.
2. Implement various searching and sorting techniques.
3. Implement tree and graph data structures along with their related operations.
4. Evaluate and apply appropriate data structure(s) for real-world problems.

Text Books:

1. Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press (2009) 3rd ed.
2. Sahni S., Data Structures, Algorithms and Applications in C++, Universities Press (2005) 2nd ed.

Reference Books:

1. Karumanchi N., Data Structures and Algorithms Made Easy, Career Monk Publications (2017) 5th ed.

UCS510: COMPUTER ARCHITECTURE AND ORGANIZATION

L	T	P	Cr
3	**0**	**0**	**3.0**

Course Objectives: Focus is on the architecture and organization of the basic computer modules viz. controls unit, central processing unit, input-output organization and memory unit.

Basics of Computer Architecture: Number System and code conversion , Logic gates, Flip flops, Registers, Counters, Multiplexer, De-multiplexer, Decoder, Encoder etc.

Register Transfer and Micro operations: Register transfer Language, Register transfer, Bus & memory transfer, Arithmetic micro operations, Logic micro operations, Shift micro operations, Design of ALU.

Basic Computer Organization: Instruction codes, Computer instructions, Timing &control, Instruction Cycles, Memory, register, and input-output reference instructions, Interrupts, Complete computer description & design of basic computer.

Central Processing Unit: General register organization, Stack organization, Instruction format, Addressing modes, Data transfer & manipulation, Program control, RISC, CISC. Pipelining and hazards.

Computer Arithmetic: Addition &Subtraction, Multiplication Algorithms, Division algorithms.

Memory Unit: Memory hierarchy, Processor vs. memory speed, High-speed memories, Main Memory, Cache memory and mapping schemes, Associative memory, Interleaving, Virtual memory, Memory management techniques.

Multiprocessors: Characteristics of multiprocessors, Interconnection structures, Inter-processor arbitration, Inter-processor communication & synchronization. Peripheral devices, I/O interface Data transfer schemes, Program control, Synchronous and asynchronous data transfer, Interrupt, DMA transfer, I/O processor.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On completion of this course, the students will be able to

1. Illustrate various elementary concepts of computer architecture including, syntax of register transfer language, micro operations, instruction cycle, and control unit.
2. Describe the design of basic computer with instruction formats & addressing modes
3. Explore various memory management techniques and algorithms for performing addition, subtraction and division etc.
4. Interpret the concepts of pipelining, multiprocessors, and inter processor communication.

Text Books:

1. Mano, Morris M., Computer System Architecture, Prentice Hall (1991) 3rd ed.
2. Hayes, J.P., Computer Architecture and Organization, McGraw Hill (1998) 3rd ed.

Reference Books:

1. Hennessy, J.L., Patterson, D.A, and Goldberg, D., Computer Architecture A Quantitative Approach, Pearson Education Asia (2006) 4th ed.
2. Leigh, W.E. and Ali, D.L., System Architecture: software and hardware concepts, South Wester Publishing Co. (2000) 2nd ed.

UMA011: NUMERICAL ANALYSIS (For all branches except ELE and EIC)

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objectives: The main objective of this course is to motivate the students to understand and learn various numerical techniques to solve mathematical problems representing various engineering, physical and real-life problems.

Floating-Point Numbers: Floating-point representation, rounding, chopping, error analysis, conditioning and stability.

Non-Linear Equations: Bisection, secant, fixed-point iteration method and its convergence analysis, Newton method for simple and multiple roots, and order of convergence.

Linear Systems and Eigen-Values: Gauss elimination method using pivoting strategies, LU decomposition, Gauss-Seidel and successive-over-relaxation (SOR) methods and their convergence, Rayleigh's power method for Eigen-values and Eigen-vectors.

Interpolation and Approximations: Finite differences, Newton's forward and backward interpolation, Lagrange and Newton's divided difference interpolation formulas with error analysis, least square approximations.

Numerical Integration: Newton-Cotes quadrature formulae (Trapezoidal and Simpson's rules) and their error analysis, Gauss-Legendre quadrature formulae.

Differential Equations: Solution of initial value problems using Taylor series, Euler's and Runge-Kutta methods of order four, system of first-order differential equations.

Laboratory Work: Lab experiments will be set in consonance with materials covered in the theory. Implementation of numerical techniques using **MATLAB**.

Course learning outcomes (CLOs): Upon completion of this course, the student will be able to:

1. understand the errors, source of error and its effect on any numerical computations.
2. learn how to obtain numerical solution of nonlinear equations using bisection, secant, Newton, and fixed-point iteration methods.
3. solve system of linear equations numerically using direct and iterative methods.
4. understand how to approximate the functions using interpolating polynomials.
5. learn how to solve definite integrals and initial value problems numerically.

Text Books:

1. Gerald F. C. and Wheatley O. P., Applied Numerical Analysis, Pearson, (2003) 7thEdition, 2. Jain K. M., Iyengar K. R. S. and Jain K. R., Numerical Methods for Scientific and Engineering Computation, New Age International Publishers (2012), 6thedition.

2. Steven C. Chappra, Numerical Methods for Engineers, McGraw-Hill Higher Education; 7th edition (1 March 2014)

Reference Books:

1. Mathew H. J., Numerical Methods for Mathematics, Science and Engineering, Prentice Hall, (1992) 2nd edition.

2. Burden L. R. and Faires D. J. Numerical Analysis, Brooks Cole (2011), 9thedition.

3. Atkinson K. and Han H., Elementary Numerical Analysis, John Willey & Sons (2004), 3rd edition.

UCS311: PRACTICAL COMPUTING

L	T	P	Cr
1	**0**	**2**	**2.0**

Course Objectives: The objective of this course is to provide exposure to the students on the basic operating system handing, code debugging and secure coding practices

Operating System Basics: Files, Directories, File utilities; Basic filters; Creating your own shell, Shell programming; Inter-process communication using pipes, System programming for file handling, file locking and data management, Process control,, Signal Handling.

Network Administration: Understanding NFS and NIS.

Code Debugging: Introduction to GDB, Code execution procedure in the memory, viewing stack and other registers.

Secure Coding Practices: Understanding stack and heap overflow

Laboratory Work:
To implement the various concepts

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of this course the student will be able to:

1. Understand Linux file system, process control and communication.
2. Debug code and identify reasons of abnormal code behavior
3. Implement Linux shell using C and obtain skills to set up and use NFS and NIS
4. Develop secure coding practices by acquiring knowledge about security vulnerabilities and their exploitation

Text Books:

1. Terry Dawson, Olaf Kirch, Linux Network Administrator's Guide, O Reilly, 3rd edition.
2. James C. Foster and Jason Deckard, Buffer Overflow Attacks: Detect, Exploit, Prevent, Syngress.
3. Sumitava Das, Unix Concepts and Application.

Reference Books:

1. Richard M. Stallman, Roland Pesch , Stan Shebs , Debugging with GDB: The GNU Source-Level Debugger, 9th Edition
2. Andrew Mallett, Mastering Linux Shell Scripting
3. Richard Stones Neil Matthew, Beginning Linux Programming, 4th Edition

UCS521: ARTIFICIAL INTELLIGENCE

L	T	P	Cr
3	0	2	4.0

Course Objectives: To be familiar with the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, machine learning, knowledge acquisition and learning methods in solving particular engineering problems.

Overview: foundations, scope, problems, and approaches of AI.

Intelligent agents: reactive, deliberative, goal-driven, utility-driven, and learning agents

Problem-solving through Search: forward and backward, state-space, blind, heuristic, problem-reduction, A, A*, AO*, minimax, constraint propagation, neural, stochastic, and evolutionary search algorithms, sample applications.

Knowledge Representation and Reasoning: ontologies, foundations of knowledge representation and reasoning, representing and reasoning about objects, relations, events, actions, time, and space; predicate logic, situation calculus, description logics, reasoning with defaults, reasoning about knowledge, sample applications.

Planning: Planning as search, partial order planning, construction and use of planning graphs, existing expert systems like MYCIN, RI, Expert system shells.

Representing and Reasoning with Uncertain Knowledge: probability, connection to logic, independence, Bayes rule, Bayesian networks, probabilistic inference, sample applications.
Decision-Making: basics of utility theory, decision theory, sequential decision problems, elementary game theory, sample applications.

Machine Learning and Knowledge Acquisition: learning from memorization, examples, explanation, and exploration. Learning nearest neighbor, naive Bayes, and decision tree classifiers, Q-learning for learning action policies, applications.

Languages for AI problem solving: Introduction to PROLOG syntax and data structures, representing objects and relationships, built-in predicates. Introduction to LISP- Basic and intermediate LISP programming.

Expert Systems: Architecture of an expert system.

Laboratory work:
Programming in C/C++/Java/LISP/PROLOG: Programs for Search algorithms- Depth first, Breadth first, Hill climbing, Best first, A* algorithm, Implementation of games: 8-puzzle, Tic-Tac-Toe, tower of Hanoi and water jug problem using heuristic search, Designing expert system using logic in PROLOG, Implementing an intelligent agent.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Learn the basics and applications of artificial intelligence and categorize various problem domains, basic knowledge representation and reasoning methods.
2. Analyze basic and advanced search techniques including game playing, evolutionary search algorithms, and constraint satisfaction.
3. Learn and design intelligent agents for concrete computational problems.
4. Understand and implement the basic concepts of programming languages like Prolog and LISP.
5. Acquire knowledge about the architecture of an expert system and design new expert systems for real life applications.

Text Books:

1. Rich E., Knight K. and Nair B. S., Artificial Intelligence, Tata McGraw Hills (2009) 3^{rd} ed.
2. Luger F. G., Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009) 6^{th} ed.

Reference Books:

1. Patterson W. D., Introduction to Artificial Intelligence and Expert Systems, Pearson (2015) 1^{st} ed.
2. Russel S., Norvig P., Artificial Intelligence: A Modern Approach, Prentice Hall (2014) 3^{rd} ed.

UCS415: DESIGN AND ANALYSIS OF ALGORITHMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of the course is to teach techniques for effective problem solving in computing. It covers good principles of algorithm design, elementary analysis of algorithms, and advanced data structures.

Introduction and Complexity Analysis: Basics of data structures such as stacks, queues, trees, heaps, Algorithm Definition, Analysing algorithms, Complexity classes, order arithmetic, Time and space trade-offs in algorithms, Recurrence relations, Analysis of iterative and recursive algorithms, Analysis of Search and Traversal in trees, graphs etc., Amortized Analysis.

Algorithm Design Techniques and Analysis
Divide and Conquer: General method, Applications such as binary search, merge sort, quick sort etc.
Greedy algorithms: General method, Elements of greedy strategy, Applications such as activity selection, job sequencing, fractional knapsack problem etc.
Dynamic Programming: General method, Elements of dynamic programming, Use of table instead of recursion, Applications such as matrix multiplication, 0/1 knapsack, optimal binary search tree, longest common subsequence etc.
Backtracking: General method, Applications such as N queen problem, sum of subsets, graph coloring, knapsack problem etc.
Branch and Bound Algorithm: General method, Applications such as0/1 knapsack problem, Traveling salesperson problem etc.

Graphs & Algorithms: Introduction to graphs, Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism, Covering and Partitioning, Strongly connected component, Topological sort, Max flow: Ford Fulkerson algorithm, max flow- min cut, Dynamic Graphs.

String Matching Algorithms: Suffix arrays, Suffix trees, tries, Rabin-Karp, Knuth-Morris-Pratt, Boyer Moore algorithm.

Lower Bound Theory: Comparison trees for sorting and searching, Oracles and adversary arguments, techniques for algebraic problems.

Problem Classes: P, NP, NP-Hard and NP-complete, deterministic and non deterministic polynomial time algorithm approximation, solutions for some NP complete problems using Approximation, Randomized, Online, and Genetic Algorithms.
Laboratory work:
Implementation of various advanced data structures and algorithms techniques for solving common engineering problems.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Analyze the complexity of algorithms, to provide justification for the selection, and to implement the algorithm in a particular context.
2. Apply various algorithmic design paradigms such as greedy, dynamic, backtracking etc. to solve common engineering problems.
3. Identify basic properties of graphs and apply their algorithms to solve real life problems.
4. Demonstrate the application of algorithms and selection of appropriate data structures under several categories such as string matching, randomized algorithms and genetic algorithms.

Text Books:

1. Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press (2009) 3rd ed.
2. Horwitz E., Sahni S., Rajasekaran S., Fundamentals of Computers Algorithms, Universities Press (2008) 2nd ed.

Reference Books:

1. Levitin A., Introduction to the design and analysis of algorithms, Pearson Education (2008) 2nd ed.
2. Aho A.V., Hopcraft J. E., Dulman J. D., The Design and Analysis of Computer Algorithms, Addsion Wesley (1974) 1st ed.
3. Sedgewick R. and Wayne K., Algorithms, Addison-Wesley Professional (2011), 4th ed.

UCS310 DATABASE MANAGEMENT SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: Emphasis is on the need of database systems. Main focus is on E-R diagrams, relational database, concepts of normalization and de-normalization and SQL commands.

Introduction: Data, data processing requirement, desirable characteristics of an ideal data processing system, traditional file based system, its drawback, concept of data dependency, Definition of database, database management system, 3-schema architecture, database terminology, benefits of DBMS.

Relational Database: Relational data model: Introduction to relational database theory: definition of relation, keys, relational model integrity rules.

Database Analysis: Conceptual data modeling using E-R data model -entities, attributes, relationships, generalization, specialization, specifying constraints, Conversion of ER Models to Tables, Practical problems based on E-R data model.

Relational Database Design: Normalization- 1NF, 2NF, 3NF, BCNF, 4NF and 5NF. Concept of De-normalization and practical problems based on these forms.

Transaction Management and Concurrency control: Concept of Transaction, States of Transaction and its properties, Need of Concurrency control, concept of Lock, Two phase locking protocol.

Recovery Management: Need of Recovery Management, Concept of Stable Storage, Log Based Recovery Mechanism, Checkpoint.

Database Implementation: Introduction to SQL, DDL aspect of SQL, DML aspect of SQL – update, insert, delete & various form of SELECT- simple, using special operators, aggregate functions, group by clause, sub query, joins, co-related sub query, union clause, exist operator. PL/SQL - cursor, stored function, stored procedure, triggers, error handling, and package.

Laboratory work:
Students will perform SQL commands to demonstrate the usage of DDL and DML, joining of tables, grouping of data and will implement PL/SQL constructs. They will also implement one project.

Project:

It will contain database designing & implementation, should be given to group of 2-4 students. While doing projects emphasis should be more on back-end programming like use of SQL, concept of stored procedure, function, triggers, cursors, package etc. Project should have continuous evaluation and should be spread over different components.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Analyze the Information Systems as socio-technical systems, its need and advantages as compared to traditional file-based systems.
2. Analyze and design database using E-R data model by identifying entities, attributes and relationships.
3. Apply and create Relational Database Design process with Normalization and De-normalization of data.
4. Comprehend the concepts of transaction management, concurrence control and recovery management.
5. Demonstrate use of SQL and PL/SQL to implementation database applications.

Text Books:

1. Silverschatz A., Korth F. H. and Sudarshan S., Database System Concepts, Tata McGraw Hill (2010) 6^{th} ed.
2. Elmasri R. and Navathe B. S., Fundamentals of Database Systems, Pearson (2016) 7^{th} ed.

Reference Books:

1. Bayross I., SQL, PL/SQL the Programming Language of Oracle, BPB Publications (2009) 4^{th} ed.
2. Hoffer J., Venkataraman, R. and Topi, H., Modern Database Management, Pearson (2016) 12^{th} ed.

UCS503: SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To plan and manage large scale software and learn emerging trends in software engineering.

Software Engineering and Processes: Introduction to Software Engineering, Software Evolution, Software Characteristics, Software Crisis: Problems and Causes, Software process models -Waterfall, Iterative, Incremental and Evolutionary process models

Requirements Engineering: Problem Analysis, Requirement Elicitation and Validation, Requirement Analysis Approaches- Structured Analysis Vs Object Oriented Analysis, Flow modeling through Data Flow Diagram and Data Dictionary, Data Modeling through E-R Diagram, Requirements modeling through UML, based on Scenario, Behavioral and Class modeling, documenting Software Requirement Specification (SRS)

Software Design and construction: System design principles like levels of abstraction, separation of concerns, information hiding, coupling and cohesion, Structured design (top-down or functional decomposition), object-oriented design, event driven design, component-level design, test driven design, data design at various levels, architecture design like Model View Controller, Client – Server architecture. Coding Practices: Techniques, Refactoring, Integration Strategies, Internal Documentation.

Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit and Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Overview of Debugging.

Agile Software Development: Agile Manifesto, Twelve Practices of eXtreme Programming (XP), XP values, XP practices, velocity, spikes, working of Scrum, product backlog, sprint backlog, Adaptive Software Development(ASD), Feature Driven Development (FDD), Test Driven Development, Dynamic System Development Method(DSDM), and Crystal Methodology,Agile Requirement and Design: User Stories, Story Boards, UI Sketching and Story Cards.

Software Project Management: Overview of Project Management: Scope, Time and Cost estimations.

Laboratory work:
Implementation of Software Engineering concepts and exposure to CASE tools like Rational Software Suit through projects.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Analyze software development process models for software development life cycle.
2. Elicit, describe, and evaluate a system's requirements and analyze them using various UML models.
3. Demonstrate the use of design principles in designing data, architecture, user and component level design.
4. Test the system by planning appropriate test cases and applying relevant test strategies.
5. Comprehend the use of agile development methodologies including UI sketching, user stories, story cards and backlog management.

Text Books:

1. Pressman R., Software Engineering, A Practitioner’s Approach, McGraw Hill International, 7^{th} ed. (2010).
2. Sommerville I., Software Engineering, Addison-Wesley Publishing Company, 9^{th} ed. (2011).

Reference Books:

1. Jalote P.,An integrated Approach to Software Engineering, Narosa, 3^{rd} ed. (2005).
2. Booch G.,Rambaugh J.,Jacobson I.,The Unified Modeling Language User Guide, 2^{nd} ed. (2005).

UCS414: COMPUTER NETWORKS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The subject will introduce the basics of computer networks to students through a study of layered models of computer networks and applications.

Introduction: Computer Network and criteria, Classification of networks, Network performance and Transmission Impairments. Networking Devices, OSI and TCP/IP Protocol Suite, Layering principles, Line Encoding, Switching and Multiplexing techniques.

Local Area Networks: Networking topologies: Bus, Star, Ring, Token passing rings, Ethernet, IEEE standards 802.3, 802.5. Wireless LANs: IEEE 802.11 and Bluetooth

Reliable Data Delivery: Error control (retransmission techniques, timers), Flow control (Acknowledgements, sliding window), Multiple Access, Performance issues (pipelining).

Routing and Forwarding: Routing versus forwarding, Static and dynamic routing, Unicast and Multicast Routing. Distance-Vector, Link-State, Shortest path computation, Dijkstra's algorithm, Network Layer Protocols (IP, ICMP), IP addressing, IPV6, Address binding with ARP

Process-to-Process Delivery: UDP, TCP and SCTP, Multiplexing with TCP and UDP, Principles of congestion control, Approaches to Congestion control, Quality of service, Flow characteristics, Techniques to improve QoS.

Self Learning Contents: Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.), Distributed applications (client/server, peer-to-peer, etc.), HTTP, Electronic mail, File transfer, Telnet.

Laboratory work:
To design conceptual networks using E-Draw, Visual Studio etc. and to implement topologies BUS, RING, STAR, Mesh and configuring Router using Packet tracer or GNS3 platform.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Conceptualize and explain the functionality of the different layers within a network architecture
2. Understand the concept of data communication, error detection and correction, access and flow control.
3. Demonstrate the operation of various routing protocols, subnetting and their performance analysis.

4. Illustrate design and implementation of datalink, transport and network layer protocols within a simulated/real networking environment.

Text Books:

1. Forouzan A. B., Data communication and Networking, McGraw Hill (2012) 5th ed.
2. Tanenbaum S. A. and Wetherall J. D., Computer Networks, Prentice Hall (2013) 5th ed.

Reference Books:

1. Kurose J. and Ross K., Computer Networking: A Top Down Approach, Pearson (2017) 7th ed.
2. Stallings W., Computer Networking with Internet Protocols and Technology, Pearson (2004).

UMA035: OPTIMIZATION TECHNIQUES

L	T	P	Cr
3	0	2	4.0

Course Objective: The main objective of the course is to formulate mathematical models and to understand solution methods for real life optimal decision problems. The emphasis will be on basic study of linear and non-linear programming problems, Integer programming problem, Transportation problem, Two person zero sum games with economic applications and project management techniques using CPM.

Scope of Operations Research: Introduction to linear and non-linear programming formulation of different models.

Linear Programming: Geometry of linear programming, Graphical method, Linear programming (LP) in standard form, Solution of LP by simplex method, Exceptional cases in LP, Duality theory, Dual simplex method, Sensitivity analysis.

Integer Programming: Branch and bound technique, Gomory's Cutting plane method.

Network Models: Construction of networks, Network computations, Free Floats, Critical path method (CPM), optimal scheduling (crashing). Initial basic feasible solutions of balanced and unbalanced transportation problems, optimal solutions, assignment problem.

Multiobjective Programming: Introduction to multiobjective linear programming, efficient solution, efficient frontier.

Nonlinear Programming:

Unconstrained Optimization: unimodal functions, Fibonacci search method, Steepest Descent method.

Constrained Optimization: Concept of convexity and concavity, Maxima and minima of functions of n-variables, Lagrange multipliers, Karush-Kuhn-Tucker conditions for constrained optimization.

Laboratory Work: Lab experiments will be set in consonance with materials covered in the theory using **Matlab**.

Course learning outcome: Upon Completion of this course, the students would be able to:

1) formulate the linear and nonlinear programming problems.
2) solve linear programming problems using Simplex method and its variants.
3) construct and optimize various network models.
4) construct and classify multiobjective linear programming problems.
5) solve nonlinear programming problems.

Text Books:

1) Chandra, S., Jayadeva, Mehra, A., Numerical Optimization and Applications, Narosa Publishing House, (2013).
2) Taha H.A., Operations Research-An Introduction, PHI (2007).

Recommended Books:

1) Pant J. C., Introduction to optimization: Operations Research, Jain Brothers (2004)
2) BazaarraMokhtar S., Jarvis John J. and ShiraliHanif D., Linear Programming and Network flows, John Wiley and Sons (1990)
3) Swarup, K., Gupta, P. K., Mammohan, Operations Research, Sultan Chand & Sons, (2010).
4) H.S. Kasana and K.D. Kumar, Introductory Operations research, Springer publication, (2004)
5) Ravindran, D. T. Phillips and James J. Solberg: Operations Research- Principles and Practice, John Wiley & Sons, Second edn. (2005).

UML501: MACHINE LEARNING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course provides a broad introduction to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications.

Introduction: Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning and Reinforcement learning.

Data Pre-processing: Need of Data Pre-processing, Data Pre-processing Methods: Data Cleaning, Data Integration, Data Transformation, Data Reduction; Feature Scaling (Normalization and Standardization), Splitting dataset into Training and Testing set.

Classification: Need and Applications of Classification, Logistic Regression, Decision tree, Tree induction algorithm – split algorithm based on information theory , split algorithm based on Gini index; Random forest classification, Naïve Bayes algorithm; K-Nearest Neighbours (K-NN), Support Vector Machine (SVM), Evaluating Classification Models Performance (Sensitivity, Specificity, Precision, Recall, etc).

Clustering: Need and Applications of Clustering, Partitioned methods, Hierarchical methods, Density-based methods.

Association Rules Learning: Need and Application of Association Rules Learning, Basic concepts of Association Rule Mining, Naïve algorithm, Apriori algorithm.

Artificial Neural Network: Need and Application of Artificial Neural Network, Neural network representation and working, Activation Functions.

Genetic Algorithms: Basic concepts, Gene Representation and Fitness Function, Selection, Recombination, Mutation and Elitism.

Laboratory Work:
Implement data preprocessing, Simple Linear Regression, Multiple Linear Regression, Decision Tree, Random forest classification, Naïve Bayes algorithm; K-Nearest Neighbors (K-NN), Support Vector Machine , k-Means, Apriori algorithm, ANN, and GA in Python/R/MATLAB/Mathematica/Weka.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Analyze methods and theories in the field of machine learning and provide an introduction to the basic principles, techniques, and applications of machine learning, supervised, unsupervised and reinforcement learning.
2. Comprehend and apply different classification and clustering techniques.
3. Comprehend and apply different association mining techniques.
4. Understand the concept of Neural Networks and its implementation in context of Machine Learning.

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1st Edition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rd Edition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018)

Reference Books:

1. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2nd Edition.
2. Michie D., Spiegelhalter J. D., Taylor C. C., Campbell, J., Machine Learning, Neural and Statistical Classification. Overseas Press (1994).

UCS410: PROBABILITY AND STATISTICS

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course shall make the students familiar with the concepts of Probability and Statistics useful in implementing various computer science models. One will also be able to associate distributions with real-life variables and make decisions based on statistical methods.

Introduction to Statistics and Data Analysis: Introduction to Statistical Inference, Samples, Populations and Experimental Design, Collection of Data, Measures of location and variability, Graphical representation of data.

Probability: Sample space, Events, Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Baye's Theorem.

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, Function of a random variable.

Special Distributions: Discrete uniform, binomial, geometric, negative binomial, Poisson, continuous uniform, exponential, gamma, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability of series and parallel systems.

Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution.

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.

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.

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.

Laboratory Work:

Implementation of statistical techniques using statistical packages viz. SPSS/R including evaluation of statistical parameters and data interpretation, regression analysis, covariance, hypothesis testing and analysis of variance.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Analyze the data using different descriptive measures and present graphically.
2. Compute the probabilities of events along with an understanding of the random variables.
3. Comprehend the concept of statistical distributions, their properties and relevance to real-life data.
4. Understand the estimation of mean and variance and their respective hypothesis tests.

Text Books:

1. Probability & Statistics for Engineers & Scientists by R.E. Walpole, R.H. Myers, S.L. Myers & Keying Ye, Prentice Hall, (2016), 9th edition.
2. An Introduction to Probability and Statistics by V.K. Rohatgi & A.K. Md. E. Saleh, Wiley, (2008), 2nd edition

Reference Books:

1. Miller and Freund's – Probability and Statistics for Engineers by R. A. Johnson, Person Education, (2017), 9th edition.
2. Introduction to Probability and Statistics for Engineers and Scientists by S.M. Ross, Elsevier, (2014), 4th edition.

UCS531: CLOUD COMPUTING

L	T	P	Cr
2	0	2	3.0

Course Objectives: To learn the concepts of cloud infrastructure and services in addition with its implementation for assessment of understanding the course by the students.

Introduction and Evolution of Computing Paradigms: General Benefits and Architecture, Business Drivers, Main players in the Field, Overview of Existing Hosting Platforms and its architecture, Cluster Computing, Grid Computing, XaaS Cloud Based Service Offerings, Overview of Security Issues

Classification of Cloud Implementations: Key Amazon offerings-Amazon Web Services, The Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queuing Services (SQS), Bundling Amazon instances, AWS Identity Management and Security in the Cloud, Messaging in the Cloud, RESTFul Web Services.

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, Classic Data Center, Virtualized Data Center (Compute, Storage, Networking and Application), Business Continuity in VDC. VMware vCloud – IaaS, Network virtualization through Software Defined Networks

Cloud based Data Storage: Introduction to Hadoop, Hadoop Ecosystem (Pig, Hive, Cassandra and Spark), Introduction No-SQL databases, Map- Reduce framework for Simplified data processing on Large clusters using Hadoop, Data Replication, Shared access to data stores.

Related Technologies: Introduction to Fog Computing and Edge Computing, Usage of Cloud for IoT and Big data analytics, Overview of Google AppEngine - PaaS, Windows Azure

Self-learning Content:
Cloud Issues and Challenges: Cloud models, Cloud computing issues and challenges like Security, Elasticity, Resource management and Scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management and Cloud bursting.

Laboratory work:
To implement Cloud, Apache and basics of Hadoop framework, an open source implementation of MapReduce, and its Java API, Hadoop Distributed File System (HDFS). Implementation of RESTFul Web Services. To understand various concepts about virtualization and data storage. To implement few algorithms with the help of MapReduce and some high-level language.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the basic concepts and architecture of Cloud computing.
2. Implement Cloud Services through AWS offerings and Restful web services.
3. Apply the knowledge of virtualization through different virtualization technologies.
4. Perform operations on data sets using Map Reduce framework, SQL and NO SQL databases.

Text Books:

1. Buyya K, R., Broberg J. and Goscinski M. A., Cloud Computing: Principles and paradigms, MIT Press (2011) 4th ed.
2. Kai Hwang, Geoffrey Fox and Jack Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann (2012) 2nd ed.
3. Miller M., Cloud Computing, Que Publishing (2008) 1st ed.
4. Puttini R. and Mahmood Z., Cloud Computing: Concepts, Technology & Architecture, Service Tech press (2013) 1st ed.

Reference Books:

1. Velte A., Velte T., and Elsenpeter R., Cloud Computing: A practical Approach, Tata McGrawHill (2009) 1st ed.
2. Hurwitz J., Bllor R., Kaufman M. and Halper F., Cloud Computing for dummies (2009) 1st ed.

UCS413: NETWORK PROGRAMMING

L	T	P	Cr
2	0	2	3.0

Course Objectives: The course introduces programming applications that use computer networks. The focus is on problem solving with emphasis on network programming. The operation and characteristics of major computer networks are studied because of their strong influence on programming interfaces (APIs) and application design.

OSI Model Introduction: OSI Model Layers Functions, TCP IP Stack real World Analogy, Data Encapsulation and Decapsulation – Introduction, Data Encapsulation, Data Decapsulation, Data Encapsulation and Decapsulation on Forwarding nodes, Multi Node Topology, Local And Remote Subnets, L3 Routing Information L3 Routes

Subnetting: Data Delivery, Mac and IP Address, Network ID, Broadcast Addresses, Max Value and Control Bits, IP Address Configuration, Point to Point Links Mask, Broadcast Addresses In Detail

Layer 2 and Layer 3 Routing: Routing Introduction: Basics, Ethernet Header format, Layer 2 Routing, ARP Goals, ARP Standard Message Format, Address Resolution Protocol, Address Resolution Protocol Demonstration, Layer 2 Switch Concept and Functioning, Need of L3 Routes, Semantics of Layer 3 Routes, Routing table Look up, L3 Routing Topology, Layer 3 Operations, Loopback interfaces – Introduction, properties, Routing using Loopback IP Address as Destination Address

Broadcast Domain and Collision Domain: Introduction, Collision Domain reduction by L2 Switches, Broadcast Domain reduction by L3 router, Application Layer: Introduction, HTTP Server Design and Implementation from Scratch, HTTP Server Demonstration, HTTP Server Code Walk

LANs and VLANs: LANs and Use Cases, LANs Problems: Immobility and Security Issues, Introduction to VLANs: Access and Trunk Ports of L2 Switch, 802.1Q VLAN Header, VLAN Tagging Rules, VLAN L2 Routing Example, VLAN Benefits such as Segmentation, Resolve Thrashing, Reduced Broadcast Domain, Mobility

Router - VLAN Routing: Introduction and Problem Statement, Concept of SVIs, L3 Router Configuration for VLAN forwarding Router to VLAN Forwarding – Example, Inter VLAN Routing Basics and Routing methodology, Ping, Linux TCP dump utility: Capture packets, TCP Dump and ping assignment

Transport Layer: Introduction, Header Stacking, Transport Layer Port Numbers, System Call Interface

Socket Programming: Introduction to Socket Programming, Server Designing, Accept system call, Select System Call Implementing Multiplexing with Accept & Select System Calls , TCP Server Example, TCP Server Design Observation, TCP Client Design and Implementation, TCP Server Client Demonstration, TCP Server With Multiplexing: High Level Design, Implementation and Demonstration

Domain Name Server (DNS): Introduction, DNS Architecture and Geographical Distribution, A Hierarchical and Decentralized System, Hosting your Own website, Website Domain Name and FQDN, Top Level Domain Servers Classification, DNS Resolver, DNS Query types: Recursive, Iterative and Reverse DNS Query

Packet Encapsulation: IP in IP Encapsulation – Introduction, Heterogeneous Networks, IPv6-in-IP Encapsulation Problem Statement and Solution, IP Encapsulation Problem Statement and Solution, IP Encapsulation Applications

Type Length Value: Introduction, Need of TLVs, Understanding TLVs, TLV Addressing the problem of Heterogeneity, TLV Addressing the problem of Software Upgrade, Data Structure – STREAMS, TLV De-Serialization using STREAMS

Self - Learning Content: Application Layer Protocols: Telnet, FTP, HTTP, SSL, IPFS, SMTP, POP3, IMAP, Client Server architecture, P2P architecture, Secure Socket Layer (SSL).

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Analyse the requirements of a networked programming environment and identify the issues to be solved;
2. Create conceptual solutions to those issues and implement a programming solution;
3. Understand the key protocols that support the Internet;
4. Apply several common programming interfaces to network communication;
5. Apply advanced programming techniques such as Broadcasting, Multicasting

Text Books:

1. Harold E. R., Java Network Programming (2013), 4th ed. O'Reilly.
2. James F. K., Keith W. R., Computer Networking: A Top-Down Approach (2009) 6th Ed.

Reference Book:

1. Stevens W. R., Fenner B., Rudoff A.M., Unix Network Programming The Socket Networking API (2004) 3rd Ed, Pearson Education.

UCS701: THEORY OF COMPUTATION

L	T	P	Cr
3	1	0	3.5

Course Objectives: This course introduces basic theory of computer science and formal methods of computation. The course exposes students to the computability theory, as well as to the complexity theory.

Regular Languages: Alphabets, Language, Regular Expression, Definitions of Finite State Machine, Transition Graphs, Deterministic & Non-deterministic Finite State Machines, Regular Grammar, Thompson's Construction to Convert Regular Expression to NDFA & Subset Algorithm to convert NDFA to DFA, Various recent development in the Conversion of Regular Expression to NFA, Minimization of DFA, Finite State Machine with output-Moore machine and Melay Machine, Conversion of Moore machine to Melay Machine & Vice-Versa.

Properties of Regular languages: Conversion of DFA to Regular Expression, Pumping Lemma, Properties and Limitations of Finite state machine, Decision properties of Regular Languages, Application of Finite Automata.

Context Free Grammar and Push Down Automata: Context Free Grammar, Derivation tree and Ambiguity, Application of Context free Grammars, Chomsky and Greibach Normal form, Properties of context free grammar, CKY Algorithm, Decidable properties of Context free Grammar, Pumping Lemma for Context free grammar, Push down Stack Machine, Design of Deterministic and Non-deterministic Push-down stack.

Turing Machine: Turing machine definition and design of Turing Machine, Variations of Turing Machines, combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem, Halting problem, Turing decidability.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend regular languages and finite automata and develop ability to provide the equivalence between regular expressions, NFAs, and DFAs.
2. Disambiguate context-free grammars and understand the concepts of context-free languages and pushdown automata.
3. Analyse and design efficient Turing Machines.
4. Distinguish different computing languages and classify their respective types.

Text Books:

1. Hopcroft E. J., Ullman D. J. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education (2007) 3rd ed.
2. Martin C. J., Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011) 4th ed.
3. Lewis R. H., Papadimitriou H. C., Elements of the Theory of Computation, Prentice Hall (1998) 2nd ed.

Reference Books:

1. Cohen A. I. D., Introduction to Computer Theory, Wiley (1997) 2nd ed.
2. Sipser M., Introduction to the Theory of Computation, Cengage Learning (2013) 3rd ed.

UCS505: COMPUTER GRAPHICS

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course provides an introduction to the principles of computer graphics. It covers detailed study of computer graphics fundamentals, 2-D & 3-D geometric transformations, curve design, visible surface detection and illumination models.

Fundamentals of Computer Graphics: Applications of computer Graphics in various fields, Evolution of computer Graphics, Graphical Input-Output Devices, Random scan displays, Raster scan displays.

Graphics Primitives: Algorithms for drawing various output primitives - Line, circle, ellipse, arcs & sectors, Boundary Fill & Flood Fill algorithm, Color Tables.

2-D & 3-D Geometrical Transformations: Translation, Rotation, Scaling, Shear, Reflection, Homogenous coordinate system, Composite transformations.

Viewing & Clipping in 2-D: Window to View port transformation, Cohen Sutherland, Liang Barsky, Nicholl-Lee-Nicholl Line clipping algorithms, Sutherland Hodgeman, Weiler Atherton Polygon clipping algorithm.

Three Dimensional Viewing & Clipping: 3-D Viewing, Projections, Parallel and Perspective projections, Clipping in 3-D.

Curves & Surfaces: Curved Lines & surfaces, Interpolation & Approximation splines, Parametric & Geometric Continuity conditions, Bezier Curves & surfaces, B-spline curves & surfaces.

Visible Surface Detection Methods: Classification of visible surface detection algorithms, Depth buffer method, Scan-line method, Depth-Sorting method, Subdivision Algorithm.

Illumination Models & Surface Rendering: Light sources, Illumination models, Surface Rendering methods, Basic Ray tracing algorithm.

Laboratory work:
Laboratory work should be done in OpenGL (version 3+). Covers all the basic drawing, filling, 2D & 3D transformations, clipping, and curve generation.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the concepts related to basics of computer graphics and its applications in various fields.
2. Apply algorithms to scan convert various output primitives and alters the coordinate descriptions of objects using 2-D & 3-D geometric transformations.
3. Understand and apply various concepts of viewing & clipping in 2-D & 3-D.
4. Comprehend the concepts related to curve design and identify visible surfaces in three dimensional scene using visible surface detection methods.
5. Apply OpenGL to create various primitives of computer graphics.

Text Books:

1. Donald D Hearn, M. Pauline Baker, "Computer Graphics, C version", 2nd Edition, Pearson Education (1997).
2. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, "Computer Graphics: Principles & Practice in C", 2nd Edition, Addison Wesley Longman (1995).

Reference Books:

1. Donald Hearn and M Pauline Baker, "Computer Graphics with OpenGL", Pearson education, 2004.
2. Zhigang Xiang, Roy A Plastock, "Computer Graphics", Schaums Outline, TMH (2007).
3. Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis, "OpenGL Programming Guide: The Official Guide to Learning OpenGL" (2013).

UCS619: QUANTUM COMPUTING

L	T	P	Cr
3	0	2	4

Course Objectives: The objective of this course is to provide the students an introduction to quantum computation after covering the concepts of linear algebra, vector space and quantum mechanics.

Mathematics and Quantum Mechanics foundation: Basics of Vector, Inner and outer product, Linear and complex vector space, Hilbert spaces (finite dimensional), Tensor Products, Trace of a matrix, Dirac's notation, Probabilities and measurements, Axioms of quantum probability, Quantum vs Classical probability, Basics of quantum mechanics, Postulates of quantum mechanics, Measurements in bases other than computational basis, Introduction of qubit, Bloch sphere representation of qubit, Quantum Superposition and entanglement, Super-dense coding, Density operators, EPR paradox, Bell's inequality, Euler identity.

Quantum Computing: classical gates, Single qubit gates, multiple qubit gates, quantum gates, universal quantum gates, Quantum circuits, design of quantum circuits.

Quantum Fourier Transform: The quantum Fourier transform, Phase estimation, Fourier Sampling, Applications: order-finding and factoring, General applications of the quantum Fourier transform, Period-finding, Discrete logarithms, The hidden subgroup problem.

Quantum Cryptography: Difference between classical and quantum cryptography, Basics of BB84 and E91 protocol.

Quantum Error Correction: Graph states and codes, Quantum error correction, fault-tolerant computation.

Lab: Implementation of Quantum concepts on any quantum programming language/ quantum simulator/MATLAB/ Julia environment.

Course learning outcomes (CLOs):

After the completion of the course, the student will be able to:

- Comprehend the basic concepts of quantum computing.
- Illustrate the concepts of quantum gates and quantum circuits.
- Explore quantum Fourier transformation and quantum error correction mechanism.
- Acquire basic knowledge of quantum protocols.

Text Books:

1. *NielsenM. A., Chuang I. L., Quantum Computation and Quantum Information, Cambridge University Press (2010) 10th Anniversary ed.*
2. *Benenti G, Casati G., Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific (2007*

Reference Books:

1. *PeresA., Quantum Theory: Concepts and Method, Kluwer Academic Publishers (2002) 1st ed..*
2. *Pittenger A. O., An Introduction to Quantum Computing Algorithm, Springer Science + BuisnessMedia(2000) 1st ed.*

UCS794: CAPSTONE PROJECT

L	T	P	Cr
0	0	2	8.0

Course Objectives: To facilitate the students learn and apply an engineering design process in electrical engineering, including project resource management. As a part of a team, the students will make a project, that emphasizes, hands-on experience, and integrates analytical and design skills. The idea is to provide an opportunity to the students to apply what they have learned throughout the course of graduate program by undertaking a specific problem.

Course Description: Capstone Project is increasingly interdisciplinary and requires students to function on multidisciplinary teams. It is the process of devising a system, component or process to meet desired needs. It is a decision-making process (often iterative), in which the basic sciences, mathematics, and the engineering sciences are applied to convert resources optimally to meet these stated needs." It typically includes both analysis and synthesis performed in an iterative cycle. Thus, students should experience some iterative design in the curriculum. As part of their design experience, students have an opportunity to define a problem, determine the problem scope and To list design objectives. The project must also demonstrate that students have adequate exposure to design, as defined, in engineering contexts. Engineering standards and realistic constraints are critical in engineering design. The program must clearly demonstrate where standards and constraints are taught and how they are integrated into the design component of the project. Each group will have 4-5 students. Each group should select their team leader and maintain daily diary. Each Group will work under mentorship of a Faculty supervisor. Each group must meet the assigned supervisor (2hrs slot/week) till the end of the semester (record of attendance will be maintained), as per the time slot which will be provided to them by the respective supervisor. This is mandatory requirement for the fulfilment of the attendance as well as the successful completion of the project. The faculty supervisor of the project will continuously assess the progress of the works of the assigned groups.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Develop skills necessary for structuring, managing, and executing the projects.
2. Design, develop, debug, document, and deliver a project and learn to work in a team environment.
3. Develop written and oral communication skills.
4. Become proficient with software development tools and environments
5. Apply interdisciplinary knowledge to engineering design solutions, taking into account professional and ethical issues.

UCS712: COGNITIVE COMPUTING

L	T	P	Cr
2	0	0	2

Course Objectives: This course will provide advanced students in cognitive science and computer science with the skills to develop computational models of human cognition, giving insight into how people solve challenging computational problems, as well as how to bring computers closer to human performance.

Overview: Introduction to Artificial Intelligence, Machine Learning, NLP, Computer Vision, Chatbots, Deep learning, Human Computer interaction, Cognitive computing and it's elements. Applications of cognitive computing like Emotion Modeling, Cognitive IoT.
Cognitive System Fundamentals: Introduction to knowledge based AI, Semantic nets, Generate and tests, Mean end analysis, Production Systems, Case based approach.
Learning: Concept learning, Classification logic, understanding, common sense based learning, Explanation based learning.
Reasoning: Case based reasoning, Analogical reasoning, constraint reasoning and Meta reasoning.
Cognitive System Design principles: Building the corpus, Considering data in cognitive systems, Machine learning based design: Supervised, Unsupervised, Reinforcement learning; Hypothesis generation and scoring, Natural language processing, Representing Knowledge, Taxonomies and Ontologies.
Advance analytics: Big data and Cognitive computing, Predictive Analytics, Text Analytics, Image Analytics, Speech Analytics.
Case-Study: Introduction to IBM Watson, Components of Deep QA: Building dataset, Question Analysis, Hypothesis generation, scoring and estimation in Watson services.

Self Learning Content: Introduction to IBM Watson, Components of Deep QA: Building dataset, Question Analysis, Hypothesis generation, scoring and estimation in Watson services.

Laboratory Work: Use the knowledge gained in theory to design and implement a cognitive application. Implement various machine learning, NLP, AI algorithms in R or Python. Explore IBM Watson Studio and Microsoft Azure.

Course learning outcomes (CLOs):

After the completion of the course, the student will be able to:

1. Understand and discuss what cognitive computing is, and how it differs from traditional approaches.
2. Understand various design principles for cognitive computing.
3. Plan and use the primary tools associated with cognitive computing.
4. Plan and use various analytics in related areas of computer science.

5. Learn about the underlying technology and techniques of IBM Watson, and assess the current state of the art in Cognitive Systems and develop an intuition about its future direction.

Text Books:

1. Judith S., Kaufman M., and Bowles A., Cognitive Computing and Big Data Analytics, Wiley,(2005), 1st ed.
2. Fingar P., Cognitive Computing: A Brief guide for Game Changers, , Meghan-Kiffer Press,(2014),1st ed.

Reference Books:

1. Miller J., Learning IBM Watson Analytics, Packt Publishers,(2016),1st ed.
2. Hashmi A. and Masood A., Cognitive Computing Recipes: Artificial Intelligence Solutions using Microsoft Cognitive Services and Tensorflow, Apress, (2019), 1st ed.
3. Kashyap P., Machine Learning for Decision Makers: Cognitive Computing Fundamentals for better decision making,,Apress,(2018), 1st ed.

UCS802: COMPILER CONSTRUCTION

L	T	P	Cr
3	0	2	4.0

Course Objectives: To Gain the working knowledge of the major phases of compilation and develop the ability to use formal attributed grammars for specifying the syntax and semantics of programming languages. Learn about function and complexities of modern compilers and design a significant portion of a compiler.

Introduction to compiling: Compilers, Analysis of the source program, the phases of Compiler, Compilation and Interpretation, Bootstrapping and Cross compiler.

Lexical Analysis: Need of Lexical analyzer, Tokens and regular expressions, Generation of lexical analyzer from DFA, Introduction to LEX and program writing in LEX.

Syntax Analysis: Need for syntax analysis and its scope, Context free grammar, Top down parsing, bottom up parsing, backtracking and their automatic generation, LL(1) Parser, LR Parser, LR(0) items, SLR(1), LALR(1), Canonical Parsing, Introduction to YACC and Integration with LEX.

Error Analysis: Introduction to error analysis, detection, reporting and recovery from compilation errors, Classification of error-lexical, syntactic and semantic.

Static semantics and Intermediate Code generation: Need for various static semantic analyses in declaration processing, name and scope analysis, S-attribute def. and their evaluation in different parsing, Semantic analysis through S-attribute grammar, L-attribute def. and their evaluation.

Run time Environment: Need for runtime memory management, Address resolution of runtime objects at compile time, Type checking, Language features influencing run time memory management, Parameter passing mechanism, Division of memory into code, stack, heap and static, Activation record, Dynamic memory management, garbage collection.

Code Generation: Code generation for expressions, Issues in efficient code generation, Sethi Ullman algorithm.

Code Optimization: Need for code optimizations, Local and global optimization, Control flow analysis, Data flow analysis, performing global optimizations, Graph coloring in optimization, Live ranges of run time values.

Laboratory work: Construct a lexical analyzer using Flex. Construct a parser using Bison/ any programming language. Build simple compilers from parsing to intermediate representation to code generation and simple optimization.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the working of major phases of compiler.
2. Apply top-down and bottom-up parsing techniques for the Parser construction.
3. Classify various parameters passing scheme, explain memory management techniques.
4. Apply code optimization techniques on HLL.

Text Books:

1. Aho V. A., Ullman D. J., Sethi R. and Lam S. M., Compilers Principles, Techniques and Tools, Pearson Education (2007), 2^{nd} ed.
2. Levine J., Mason T., Brown D., Lex and Yacc, O'Reilly (2012), 2^{nd} ed.

Reference Books:

1. Kenneth C. L., Compiler Construction and Practices, Thomson Publication (1997), 2^{nd} ed.
2. Dhamdhere, Compiler Construction, Macmillan Publication (2008), Edition 2^{nd} ed.

UHU005: HUMANITIES FOR ENGINEERS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of this course is to introduce values and ethical principles, that will serve as a guide to behavior on a personal level and in professional life. The course is designed to help the students to theorize about how leaders and managers should behave to motivate and manage employees; to help conceptualize conflict management strategies that managers can use to resolve organizational conflict effectively. It also provides background of demand and elasticity of demand to help in devising pricing strategy; to make strategic decisions using game theory and to apply techniques of project evaluation.

Unit 1: Human Values and Ethics
Values: Introduction to Values, Allport-Vernon-Lindzey Study of Values, Rokeach Value Survey, Instrumental and Terminal Values.
Moral and Ethical Values: Types of Morality, Kant's Principles of Morality, Factors for taking ethical decisions, Kohlberg's Theory of Moral Development
Professional Ethics: Profession: Attributes and Ethos, Whistle-blowing.

Unit 2: Organizational Behavior
Introduction to the Field of Organizational Behaviour: Individual Behaviour, Personality, and Values, Perceiving Ourselves and Others in Organizations, Workplace Emotions, Attitudes, and Stress, Foundations of Employee Motivation and Leadership, Performance Appraisal, Conflict and Negotiation in the Workplace.

Unit 3: Economics
Demand, Supply & Elasticity – Introduction to Economics, Demand & its Determinants, Elasticity and its types
Production & Cost Analysis – Short run & Long Run Production Functions, Short run & Long run cost functions, Economies & Diseconomies of Scale
Competitive Analysis & Profit Maximization – Perfect competition, Monopoly, Monopolistic & Oligopoly Markets
Strategy & Game Theory – Pure Strategy & Mixed Strategy Games, Dominance, Nash Equilibrium, & Prisoner's Dilemma
Capital Budgeting – Capital Projects, Net Present Value (NPV) & IRR techniques.

Practical:

1. Practical application of these concepts by means of Discussions, Role-plays and Presentations,
2. Analysis of Case Studies on ethics in business and whistle-blowing, leadership, managerial decision- making.
3. Survey Analysis
4. Capital Budgeting assignment

Course Learning Outcomes (CLOs) / Course Objectives (COs):

The student after completing the course will be able to:

1. Comprehend ethical principles and values and apply them as a guide to behavior in personal and professional life.
2. Apply tools and techniques to manage and motivate employees.
3. Analyse and apply conflict management strategies that managers can use to resolve organizational conflict effectively.
4. Devise pricing strategy for decision-making.
5. Apply techniques for project evaluation.

Text Books:

1. N. Tripathi, Human Values, New Age International (P) Ltd. (2009).
2. Robbins, S. P/ Judge, T. A/ Sanghi, S Organizational Behavior Pearson, New Delhi, (2009).
3. Petersen, H.C., Lewis, W.C. and Jain, S.K., Managerial Economics, Pearson (2006).

Reference Books:

1. McKenna E. F. Business psychology and organisational behaviour. Psychology Press, New York (2006).
2. Furnham A. The Psychology of Behaviour at Work: The Individual in the organization. Psychology Press, UK (2003).
3. Salvatore, D and Srivastava, R., Managerial Economics, Oxford University Press (2010).
4. Pindyck, R and Rubinfiled, D., Microeconomics, Pearson (2017).

UCS813: SOCIAL NETWORK ANALYSIS

L	T	P	Cr
2	0	2	3.0

Course Objectives: To enable students to put Social Network Analysis projects into action in a planned, informed and efficient manner.

Preliminaries: Graphs, Types of graphs, Representation, Bipartite graphs, Planar networks, The graph Laplacian, Random Walks, Maximum Flow and Minimum Cut Problem, Introduction to Approximation Algorithms, Definitions. Approximation algorithms for vertex cover and TSP.

Introduction to Social Networks: Types of Networks: General Random Networks, Small World Networks, Scale-Free Networks; Examples of Information Networks; Static Unweighted and weighted Graphs, Dynamic Unweighted and weighted Graphs, Network Centrality Measures; Strong and Weak ties.

Walks: Random walk-based proximity measures, Other graph-based proximity measures. Clustering with random-walk based measures, Algorithms for Hitting and Commute, Algorithms for Computing Personalized Pagerank and Sim- rank.

Community Detection: Basic concepts, Algorithms for Community Detection: Quality Functions, The Kernighan-Lin algorithm, Agglomerative/Divisive algorithms, Spectral Algorithms, Multi-level Graph partitioning, Markov Clustering; Community Discovery in Directed Networks , Community Discovery in Dynamic Networks, Community Discovery in Heterogeneous Networks, Evolution of Community.

Link Prediction: Feature based Link Prediction, Bayesian Probabilistic Models, Probabilistic Relational Models, Linear

Algebraic Methods: Network Evolution based Probabilistic Model, Hierarchical Probabilistic Model, Relational Bayesian Network, Relational Markov Network.

Event Detection: Classification of Text Streams, Event Detection and Tracking: Bag of Words, Temporal, location, ontology based algorithms. Evolution Analysis in Text Streams, Sentiment analysis.

Social Influence Analysis: Influence measures, Social Similarity - Measuring Influence, Influencing actions and interactions. Homophily, Influence maximization.

Laboratory work:
Implementation of various concepts taught in the course using Python/R Programming

Text Books / Reference Books:

1. Charu C. Aggarwal, Social Network Data Analytics, Springer; 2011.
2. S.Wasserman, K.Faust: Social Network Analysis: Methods and Applications, Cambridge Univ Press, 1994
3. Scott, J. (2007). Social network analysis: A handbook (2^{nd} Ed.). Newbury Park, CA: Sage.
4. Knoke (2008). Social Network Analysis, (2^{nd} Ed). Sage.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Formalize different types of entities and relationships as nodes and edges and represent this information as relational data.
2. Plan and execute network analytical computations.
3. Use advanced network analysis software to generate visualizations and perform empirical investigations of network data.
4. Interpret and synthesize the meaning of the results with respect to a question, goal, or task.
5. Collect network data in different ways and from different sources while adhering to legal standards and ethics standards.

UCS806: ETHICAL HACKING

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course is designed to impart a critical and theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools and the services related to Ethical Hacking.

Introduction: Understanding the importance of security, Concept of ethical hacking and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in hacking.

Footprinting: Introduction to footprinting, Understanding the information gathering methodology of the hackers, Tools used for the reconnaissance phase.

Scanning: Detecting live systems-on the target network, - Discovering services running listening on target systems, Understanding port scanning techniques, Identifying TCP and LIDP services running on the target network, Understanding active and passive fingerprinting.

System-Hacking: Understanding Sniffers, Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing.

Session Hijacking: Understanding Session Hijacking, Phases involved in Session Hijacking, Types of Session Hijacking, and Session Hijacking Tools.

Hacking Wireless Networks: Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

Cryptography: Symmetric and Asymmetric Cryptography, Classical Encryption techniques, Substitution techniques, Block Ciphers Principles, Fiestel Structure, DES, Double and Triple DES, AES, Public Key Cryptography, RSA, Diffie-Hellman Key Exchange, Cryptographic Hash Functions and Digital Signatures.

Laboratory Work:
Lab Exercises including using scanning tools like IPEYE, IPsecScan, SuperScan etc. and Hacking Tools likes Trinoo, TFN2K, Zombic, Zapper etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Understand the different phases involved in hacking.
2. Utilize the scanning tools used for the information gathering.
3. Recognize the phases in session hijacking and use the tools for counter-measuring the various sniffing attacks.
4. Analyse different types of attacks on the wireless networks.
5. Describe and apply different types of algorithms for securing the data.

Text Books:

1. Simpson T. M., Backman K., Corley J., Hands-On Ethical Hacking and Network Defense, Delmar Cengage Learning (2011) 2nd edition.
2. Fadia A. and Zacharia M., Network intrusion alert: an ethical hacking guide to intrusion detection, Boston, MA: Thomas Course Technology 3rd edition (2008).

Reference Books:

1. Mathew T., Ethical Hacking, OSB Publication (2003). 2nd edition
2. McClure S., Scambray J. and Kurtz G., Hacking Exposed 7: Network Security Secrets and Solutions, McGrawHill (2012) 7th Edition.

UCS532: COMPUTER VISION

L	T	P	Cr
2	0	2	3.0

Course Objectives: To understand the basic concepts of Computer Vision. The student must be able to apply the various concepts of Computer Vision in other application areas.

Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Image Representation & Description: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, LBP and its variants, Gabor Filters and DWT.

Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Pattern Analysis: Clustering: K-Means, Fuzzy C-means; Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Dimensionality Reduction: PCA, LDA, ICA.

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Self-Learning Content:

Miscellaneous: Applications: CBIR, CBVR, Activity Recognition, computational photography, Biometrics, stitching and document processing; Modern trends - super-resolution; GPU, Augmented Reality; cognitive models, fusion and SR&CS.

Laboratory Work:

To implement various techniques and algorithms studied during course.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Understand the fundamental problems of computer vision.
2. Implement various techniques and algorithms used in computer vision.
3. Analyse and evaluate critically the building and integration of computer vision algorithms and systems.
4. Demonstrate awareness of the current key research issues in computer vision.

Text Books:

1. Szeliski, R., Computer Vision: Algorithms and Applications, Springer-Verlag London Limited (2011), 1st Edition.

2. Forsyth, A., D. and Ponce, J., Computer Vision: A Modern Approach, Pearson Education (2012) 2nd Edition.

Reference Books:

1. Hartley, R. and Zisserman, A., Multiple View Geometry in Computer Vision Cambridge University Press (2003) 2nd Edition.
2. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, Morgan Kaufmann (1990) 2nd Edition.
3. Gonzalez, C., R. and Woods, E., R. Digital Image Processing, Addison- Wesley (2018) 4th Edition.

UCS534: COMPUTER & NETWORK SECURITY

L	T	P	Cr
2	**0**	**2**	**3.0**

Course Objectives: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools.

Introduction: Security Attacks, Security Services, Security Mechanisms and Principles, Security goals, Malicious software, Worms, Viruses, Trojans, Spyware, Botnets, Life cycle of a vulnerability: CAN and CVE.

Computer Security: Set-UID programs, privileged programs, environment variables: hidden inputs, capability leaking, invoking other programs, principle of least privileges. Environment variables and attacks, attacks via dynamic linker, external program and library. Shellshock attack, exploiting shellshock vulnerability. Buffer overflow attacks: program memory layout, stack and function invocation. Writing a shell code, injecting code into buffer, address space layout randomization, Stack Guard.

Network Security: Packet sniffing and spoofing, Attacks on TCP protocol, SYN flood, TCP reset attack, session hijacking attack, Firewalls: Packet filter, Stateful firewall, Application firewall. IP tables, DNS poisoning, Authoritative replies, ARP poisoning, Heartbleed Bug and Attack, Public key infrastructure and Transport Layer Security.

Laboratory work:
Demonstrate use of Environment variables and privileged programs, Demonstrate Buffer Overflow and showcase EIP and other register status, insert malicious shell code into a program file and check its malicious or benign status, perform ARP poisoning, implement stateful firewall using IPTables.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:
1. Identify software vulnerabilities and apply various security mechanisms to protect against security attacks.
2. Demonstrate shellshock attack and its countermeasure.
3. Demonstrate buffer-overflow attack, locate and fix security leaks in a computer software.
4. Implement firewall and its variants.
5. Implement PKI and TLS.

Text Books:

1. Stallings, W., Network Security Essentials, Prentice Hall (2017) 6th Edition.
2. Cheswick, R., W., Bellovin, M., S., and Rubin, D., A., Firewalls and Internet Security, Addison-Wesley Professional (2003) 2nd Edition.
3. Wenliang Du, Computer Security: A hands-on approach, CreateSpace (2017).

Reference Books:

1. Graves, K., Certified Ethical Hacking Study Guide,Sybex (2010) 1st Edition.
2. Stallings, W., Cryptography and Network Security, Prentice Hall (2013), 6th Edition.

UMC512: MATHEMATIC MODELING AND SIMULATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: The primary goal is to provide students a basic knowledge of mathematical modeling. The students will be able to construct different mathematical models using various mathematical techniques. The course introduces computer simulations and techniques, provides the foundations for the student to understand computer simulation needs.

Mathematical Modeling: Modeling and its principles, some methods of mathematical modeling: problem definition, dimensional homogeneity and consistency, abstraction and scaling, conservation and balance principles, system characterization, constructing linear models, discrete versus continuous modelling, determinstic versus stochastic.

Approximating and Validating Models: Review of Taylor's formula and various trigonometric expansions, validating the model, error analysis, fitting curves to the data.

Basic Simulation Approaches: Methods for simulation and data analysis using MATLAB, statistics for simulations and analysis, random variates generation, sensitivity analysis.

Model and its Different Types: Linear and nonlinear population models, traffic flow models, transport phenomena, statistical models, Poisson process, stochastic models, stock market, option pricing, Black-Scholes model, modeling engineering systems.

Software Support:MATLAB.

Lab Experiment:
Implementation of numerical techniques using MATLAB based on course contents.
Projects: The projects will be assigned according the syllabus covered.
Text Books / References Books:

1. Clive L. Dym, Principles of Mathematical Modelling, Elsevier Press, Second Edition, 2004.
2. Edward A. Bender, An Introduction to Mathematical Modeling, Dover, 2000.
3. D Kincaid and W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, Third Edition, American Mathematical Society, 2009.
4. J. Nathan Kutz, Data-Driven Modeling& Scientific Computation: Methods for Complex Systems & Big Data, Oxford University Press, 2013.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
At the end of the course, the student will be able to
1. Formulate various mathematical models based on modeling tools and techniques.
2. Derive and use various simulation techniques.
3. Simulate examples based to realistic models using appropriate modeling tools.

4. Implement statistical simulation for various models.

UCS636: 3D MODELLING AND ANIMATION

L	T	P	Cr
2	**0**	**2**	**3.0**

Course Objectives: To develop the skill & knowledge in 3D Modeling & Animation. Students will understand the know-how and can function either as an entrepreneur or can take up jobs in the multimedia and animation industry, video studios, edit set-up and other special effects sectors.

3D Object Modelling: Basic modelling concepts, vertices, edges, and faces, basic transformations, pivot points, duplication and merging, extrusion, insetting, modifiers, loop cuts and face loops, subdivision methods, coordinate system and exporting, model rendering.

Low Poly Models: Triangular meshes, objects and mesh data, cursor and origins hidden geometry, Boolean modifiers, geometry from curve, curve resolution, non-planner geometry.

3D Character Modelling: Introduction, character modelling, unwrapping UVs & mapping texture, texture painting, armatures, character rigging, constrained movements, forward and inverse kinematics, time-line, keyframes, character animation, animation rendering.

Physically Based Modelling and Animation: Introduction, Simulation Foundation, Particle based Models, Collision detection and response, Particle System, Particle Simulation, Particle Rendering, Numerical Integration in Particle System, Deformable Meshes, Rigid Bodies and Constrained Dynamics, Fluid Simulation.

Self-Learning Content: Real Time Animation: Splines and curves, Key-frame techniques, Quaternions for rotations / orientations, Blending and interpolation, Kinematics, Motion capture systems, Motion graphs and character control, Animation data representations, Behavioural Animation, Facial Animation, Perception in animation.

Laboratory Work

This course covers beginner to intermediate 3D Modeling and Animation. In this Lab the students will be able to model the 3D character and objects, its UV Mapping, Texture Painting, Rigging, and Animation. Evaluation will be mainly via projects and assignments taking a creative approach to expressive 3D modelling and Animation.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Apply modelling concepts in order to implement 3D objects. (Blender / Max).
2. Understand the basic geometry and triangulation techniques behind low poly models.
3. Implement 3D humanoid characters and to apply the concept of rigging for animating the character using key frames.
4. Illustrate the theoretical and practical aspects of 3D Modelling, Key Frame Animation, Simulation & effects.
5. Demonstrate different types of animation and its effects in the real world.
6. Analyse the different processes, post processes involved in computer animation field.

Text Books:

1. House, H., D. and Keyser, C., J., Foundations of Physically Based Modeling and Animation, CRC Press (2017) 1st Edition.
2. Chopine, A., 3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation, Focal Press (2011) 1st Edition.
3. Zeman, B., N., Essential Skills for 3D Modeling, Rendering, and Animation, A K Peters / CRC Press (2017) 1st Edition.

Reference Books:

1. Villar, O., Learning Blender: A Hands-On Guide to Creating 3D Animated Characters, Addison Wesley (2017) 2nd Edition.
2. Kerlow, I., The Art of 3D Computer Animation and Effects, Wiley, (2009) 4th Edition.
3. Flavell, L., Beginning Blender: Open Source 3D Modelling, Animation, and Game Design, Apress, (2010) 1st Edition.
4. Boardman, T., 3dsmax 7 Fundamentals, New Riders, (2005) 1st Edition.

UCS637: IMAGE PROCESSING

L	T	P	Cr
2	0	2	3.0

Course Objectives: To learn the advanced concepts of image processing and its implementation.

Introduction: Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system. Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels.

Image enhancement in the spatial domain: Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

Image restoration: A model of the image degradation/restoration process, noise models, and restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function.

Color Image Processing: Color fundamentals, color models, pseudo color image processing, basics of full–color image processing, color transforms, smoothing and sharpening, color segmentation.

Image Compression: Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards.

Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

Image Segmentation: Detection of discontinuous, edge linking and boundary detection, thresholding, region–based segmentation.

Object Recognition: Patterns and patterns classes, recognition based on decision–theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods – matching shape numbers, string matching.

Laboratory work:
Demonstrate the use of Image Processing Toolbox on MATLAB to create interactive image processing applications like image enhancement, image compression, image segmentation, feature extraction etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the need and usage of concepts of image processing.
2. Enhance the visual quality of given grey/color image using well known transformations and filters.
3. Distinguish between lossy and lossless image compression prototypes.
4. Segment the regions of given image using various feature extraction algorithms in order to recognize object.
5. Demonstrate the use of MATLAB to create correlative image processing applications.

Text Books:

1. Gonzalez C. R., Woods E. R., Digital Image Processing, Pearson Education (2008) 3rd ed.
2. Sonka M., Hlavac V. and Boyle R., Image Processing, Analysis and Machine Vision, Thomson Learning, (1993) 1st ed.

Reference Books:

1. McAndrew A., Introduction to Digital Image Processing with Matlab, Thomson Course Technology (2004)
2. Low A., Introductory Computer Vision and Image Processing, McGraw-Hill (1991), 1st ed.

UCS638: SECURE CODING

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course aims to provide an understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities. It gives an outline of the techniques for developing a secure application.

Introduction: Security, CIA Triad, Viruses, Trojans, and Worms, Security Concepts-exploit, threat, vulnerability, risk, attack, Rootkits, Trapdoors, Botnets, Key loggers, Honeypots. Active and Passive Security Attacks.

Need for secure systems: Proactive Security development process, Secure Software Development Cycle (SSDLC), Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline.

Threat modelling process and its benefits: Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques and Security Best Practices. Security techniques, authentication, authorization. Defense in Depth and Principle of Least Privilege.

Software & Web Security: Return-to-libc attack, format string vulnerability. Race condition vulnerability, Dirty COW, PE Code injection. Cross site request forgery: CSRF attacks on HTTP GET and POST services & countermeasures. XSS attack: self-propagating XSS worm, preventing XSS attacks, SQL injection attack & countermeasures. Client-side attacks

Laboratory Work:
In this Lab, student shall learn to recognize and remove common coding errors that lead to vulnerabilities. This lab also gives an outline of the techniques for developing a secure application code, implementing different types of attacks and protection schemes for both software and web security. Evaluation will be mainly based on projects and assignments.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Demonstrate skills needed to deal with common programming errors and develop secure applications.
2. Implement PE Code injection and demonstrate control hijacking via EIP manipulation
3. Demonstrate client-side attacks and identify nature of threats to software and incorporate secure coding practices throughout the planning and development of software product.
4. Demonstrate SQL injection, XSS attack and suggest countermeasures for the same.

Text Books:

1. Howard, M. and LeBlanc, D., Writing Secure Code, Howard, Microsoft Press (2002) 2^{nd} Edition.
2. Deckard, J., Buffer Overflow Attacks: Detect, Exploit, Syngress (2005) 1^{st} Edition.
3. Wenliang Du, Computer Security: A hands-on approach, CreateSpace (2017).

Reference Books:

1. Swiderski, F. and Snyder, W., Threat Modeling, Microsoft Professional, (2004) 1^{st} Edition.
2. Salt, C., J., SQL Injection Attacks and Defence, Elsevier (2012), 2^{nd} Edition.

UCS639: IT PROJECT MANAGEMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: Learn and explore SPM activities through knowledge of software project management and project planning.

Course Prerequisite: Software Engineering

Introduction to Project Management: The characteristics of software projects, Objectives of project management: time, cost and quality, Basics of Software Project Management, Project Management Processes and Framework, Project Stakeholders, Stages of Project Planning, Project Management Knowledge areas, Project Management Tools & Techniques, Project success factors, role of project manager.

Project Initiation: Project Pre- Initiation, business case, Feasibility Study, Strategic planning and project selection, Project Charter, Project Management Plan.

Measurement and Control: Measurements for project monitoring, what and when to measure, Plan versus Control, managing the plan, Reviews, feedback and reporting mechanisms, revisiting the plan.

Project Scope Management: Scope Planning & Scope management processes, Measuring Project size, Lines of Code (LOC), Function point calculation (FP), Scope definitions & project scope statement, Work Breakdown Structure (WBS), WBS dictionary, scope verification, scope control, Scope Baseline.

Project Time Management: Project time management processes, activity sequencing, network diagrams, activity duration estimation, schedule development, Gantt Charts, Critical path method(CPM), Program evaluation & review technique (PERT), concept of slack time, schedule control, schedule baseline.

Project Cost management: Basic principles of cost management, Cost estimation techniques, type of project costs, cost estimation tools & techniques, COCOMO, Putnam/ SLIM model Estimating by Analogy, cost budgeting, cost control, Earned Value Management (EVM), project cost baseline.

Project Quality Management: Quality Planning, quality Assurance, Quality control, Tool &techniques for quality control, Defect Removal vs. Defect prediction, Pareto Analysis, Six Sigma, CMM, ISO Standards

Project Human Resource Management: Human resource planning, project organizational charts, responsibility assignment metrics, acquiring project team, resource assignment, resource loading, resource leveling, Different team structures, developing project teams.

Project Communication Management: Communication Planning, Performance reporting, managing stakeholders, improving project communication.

Project Risk Management: Risk Management planning, types of risk, risk identification, risk register, qualitative risk analysis, using probability impact matrixes, expert judgment, qualitative risk analysis, decision trees & expected monetary value, simulation, sensitivity analysis, risk response planning, risk monitoring & control.

Self Learning Material:
Project Procurement Management: Procurement management plans, contract statement of work, make or buy solution, planning contracts, requesting seller responses, selecting sellers, administrating the contract, closing the contract, Software Configuration Management: Why versions exist, why retain versions, SCI, Releases vs. version. Change Control and Change Management Process and Change Control Board.

Assignment work:
1. Preparing Project Plan for a Software Project for any Project or case study.
2. Using Function Point calculation tools for estimation, comparing with COCOMO estimates.
3. Implementation of PERT, CPM methods for preparing schedule.
4. Resource allocation etc. using MS Project or OpenProj tool.
5. Preparing RMMM Plan for a case study.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:
1. Apply project validation techniques for project selection in software organizations
2. Estimate the project scope, schedule and cost effectively and with proper documentation.
3. Implement Quality control, Quality assurance and Risk management in software projects through various quality standards.
4. Formulate efficient plans for effective Communication, Human Resource Management and overall Software project management.

Text Books:
1. Kathy Schwalbe, Introduction to Project Management, Cengage Learning; 8th edition (2015).
2. Jalote P. Software Project Management in Practice, Pearson; 1st Edition, (2016).

Reference Books:

1. Stellman, A. and Greene, J., Applied Software Project Management, O'Reilly Media, Inc.; 1st edition (2005).
2. Futrell, R. T., Shafer, D. F. and Shafer, L. I., Quality Software Project Management, Prentice Hall; 1st edition (2014).
3. Hughes, B. and Cotterell, M., Mall, R., Software Project Management, Tata McGraw Hill; 6th edition (2017).
4. Pressman, R., A practitioner's Guide to Software Engineering, Tata McGraw Hill; 8th edition (2019).

UMC622: MATRIX COMPUTATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course aims to provide a platform for the students to use linear algebra in real life. Most of the real life problems are based on computation of eigenvalues and singular values. In this course we stress on the computational methods to compute the same. The Matlab implementation of the methods will be insightful for better understanding. The students are expected to have taken basic and a continuation course in numerical analysis or acquired equivalent knowledge in a different way.

Matrix Analysis:

***Review of matrices and vector spaces*:** rank of a matrix, linear dependence and independence, bases and dimensions, linear transformations, range and null space of a matrix, rank-nullity theorem.

***Inner product space*:** Gram Schmidt orthogonalization, dual space and invariant space.

***Matrix transformations*:** similarity transformation, diagonalization of matrices, Householder transformation, QR factorization.

Conditioning of matrices: vector and matrix norms, convergent matrices, condition number of a matrix.

Techniques for finding eigen values: Eigen value problems, spectral stability of matrices, reduction to Hessenberg or tridiagonal form, iterative techniques using Krylov subspace concepts for eigen value problems.

Spectral theory of matrices: spectral decompositions, Gersgorin bounds on eigenvalues, spectrum of perturbed matrices, Schur decomposition theorem.

Singular value decomposition: SVD and their applications.

Real life applications of eigen values and singular values: Discussion of real life problems based on eigen values and SVDs and their application in image processing and big data analysis.

Laboratory assignments:
Matlab experiments will be designed to implement algorithms from the syllabus.

Text Books / References Books:

1. Kenneth Hoffman and Ray Kunze, Linear Algebra, Pearson India, second edition, 2015.
2. Derek J. S. Robinson, A course in linear algebra with application, World Scientific Press, second edition, 2006.
3. Gene H. Golub and Charles F. Van Loan, Matrix Computations, Johns Hopkins University Press, fourth edition, 2012.
4. Roger A. Horn and Charles R. Johnson, Matrix Analysis, second Edition, Cambridge University Press, 2012.
5. L. N. Trefethen and David Bau, Computational Linear Algebra, SIAM, 1997.
6. Gilbert Strang, Linear algebra and its Applications, fourth edition, CENGAGE, 2014.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After completing the course, a student will be able to:

1. Explain and apply fundamental linear algebra concepts,
2. Evaluate norms of vectors and matrices,
3. Solve eigen value problems using theoretical and computational methods,
4. Apply singular value decomposition,
5. Implement linear algebra algorithms using Matlab.

UCS646: GAME DESIGN & DEVELOPMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: To become familiar with various fundamental and advanced gaming concepts including basic maths and physics used behind the game engine.

Introduction: Types of games, History, Impact of Games on Society , Game life cycle, Game loop, Components of game, Model and scene rendering, State Management, Scene management, Texture compression, Level of details, Frustum culling, Occlusion culling, Game as a software, Steps for Game Design, Data Structure for Game, CPU vs.GPU, Game Engine, Components of game engine, Linear Transformation. Composite transformation.

Fundamental Gaming concepts: Static and Dynamic Game objects, Vectors, Concept of Time, Lighting, Particle System, Collider, Collision handles, Materials, Texture mapping, Input Process, Object replication, Instantiation, Special Effects, Terrain, Audio design and production, Ray Casting.

Maths behind Game Engines: Introduction to Vectors- Addition & Subtraction, Vector length, Scaling, Unit length vectors, Dot & Cross product, Linear Interpolation, Euler Angles, Intersection, Matrices, Coordinate systems, Projections, Triangle Meshes, Optimizations, Quaternion.

Advanced Games: Augmented Reality, Virtual Reality, Mixed Reality, AR & VR based Games, Artificial Intelligence based Game, Networking based game, Android based games, Debugging mode, Understanding of Screen and World Coordinate system, Raycasting, Touch & Swipe Input: Touch in Orthographic view, Touch in Perspective view, Accelerometer input, Scaling of Game screen, AR/VR/Android/iOS/Windows Game Deployment methods.

Self-Learning Content: Game Physics: Mathematical concepts, Basic transformations, Collision Detection and response, Newton's law of motion, Modeling gravity, Air resistance, Unstable rotation, Inertia tensor, Moment of Inertia, Applying torque to rigid body, The Magnus effect, Overview of friction, Critical angle, Dynamic Friction.

Laboratory work:
2D and 3D game development for windows and android platform using Unity 3D Game Engine and C# language.

Course Learning Outcomes (CLOs)/ Course Objectives (COs):

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

1. Illustrate the basic concepts, requirements and processes of game design and development
2. Implement the fundamental gaming concepts to create a game.
3. Understand the physics and mathematics behind the game engine.
4. Demonstrate the advanced gaming concepts such as AR, VR, Android etc.
5. Develop a 2D/3D game using C# and Unity 3D Game engine.

Text Books:

1. Eberly H. D., Game Physics, Morgan Kaufmann Publisher (2010), 2nd ed.
2. Bond G. J., Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#, Addison-Wesley (2015), 2nd ed.

Reference Books:

1. House H. D., Keyser C. J, Foundations of Physically Based Modeling and Animation, CRC Press (2017), 1st ed.
2. Okita. A., Learning C# Programming with Unity 3D, CRC Press (2014), 1st ed.

UCS647: NATURAL LANGUAGE PROCESSING

L	T	P	Cr
2	0	2	3.0

Course Objectives: To understand the basic concepts of Natural Language Processing (NLP). The student must be able to apply the various concepts of NLP in other application areas.

Introduction: Origin of Natural Language Processing (NLP), Challenges of NLP, NLP Applications, Processing Indian Languages.

Words and Word Forms: Morphology fundamentals; Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Named Entities.

Parsing: Definite clause grammars; shift-reduce parsing; chart parsing' Shallow Parsing, Statistical Parsing, Maximum Entropy Models; Random Fields, Scope Ambiguity and Attachment Ambiguity resolution, Approaches to discourse, generation.

Language Modeling and Part of Speech Tagging: Markov models, N-grams, estimating the probability of a word, and smoothing, Parts-of-speech, examples and its usage.

Machine Translation: Need of MT, Problems of Machine Translation, MT Approaches, Direct Machine Translations, Rule-Based Machine Translation, Knowledge Based MT System, Statistical Machine Translation.

Meaning: Lexical Knowledge Networks, WorldNet Theory; Semantic Roles; Word Sense Disambiguation; WSD and Multilinguality; Metaphors.

Other Applications: Sentiment Analysis; Text Entailment; Question Answering in Multilingual Setting; NLP in Information Retrieval, Cross-Lingual IR. Text-classification.

Laboratory Work:
To implement Natural language concepts and computational linguistics concepts using popular tools and technologies. To implement key algorithms used in Natural Language Processing. To implement various machine translations techniques for Indian languages.

Self-leaning Content:
Morphological Diversity of Indian Languages, Phrase structure and constituency models: phrase structure grammar; dependency grammar; formal language theory. Indian Language Word Nets and Multilingual Dictionaries.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the concept of natural language processing, its challenges and applications.
2. Comprehend the concepts of words form using morphology analysis.
3. Acquire the knowledge of syntax and semantics related to natural languages.
4. Ability to design and analyze various NLP algorithms.
5. Acquire knowledge of machine learning techniques used in NLP.

Text Books:

1. Jurafsky D. and Martin H. J, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall (2014), 2nd ed.
2. Manning D. C. and Schütze H., Foundations of Statistical Natural Language Processing MIT Press (1999) 1st ed.

Reference Books:

1. Dale R., Moisl H. and Somers H., Handbook of Natural Language Processing, CRC Press (2010), 2nd ed.
2. Bird S., Klein E. and Loper E., Natural Language Processing with Python, Oreilly Publication (2009), 2nd ed.

UCS648: CYBER FORENSICS

L	T	P	Cr
2	0	2	3.0

Course Objectives: To maintain an appropriate level of awareness, knowledge and skill required to understand and recreate the criminal terminology and Cyber Forensics investigation process.

Introduction to Cybercrime: Defining Cybercrime, Understanding the Importance of Jurisdictional Issues, Quantifying Cybercrime, Differentiating Crimes That Use the Net from Crimes That Depend on the Net, working toward a Standard Definition of Cybercrime, Categorizing Cybercrime, Developing Categories of Cybercrimes, Prioritizing Cybercrime Enforcement, Reasons for Cybercrimes.

Understanding the People on the Scene: Understanding Cybercriminals, Profiling Cybercriminals, Categorizing Cybercriminals, Understanding Cyber victims, Categorizing Victims of Cybercrime, Making the Victim Part of the Crime-Fighting Team, Understanding Cyber investigators, Recognizing the Characteristics of a Good Cyber investigator, Categorizing Cyber investigators by Skill Set.

Computer Investigation Process: Demystifying Computer/Cybercrime, Investigating Computer Crime, How an Investigation Starts, Investigation Methodology, Securing Evidence, Before the Investigation, Professional Conduct, Investigating Company Policy Violations, Policy and Procedure Development, Policy Violations, Warning Banners, Conducting a Computer Forensic Investigation, The Investigation Process, Assessing Evidence, Acquiring Evidence, Examining Evidence, Documenting and Reporting Evidence, Closing the Case.

Acquiring, Duplicating and Recovering Deleted Files: Recovering Deleted Files and Deleted Partitions, recovering "Deleted" and "Erased" Data, Data Recovery in Linux, Recovering Deleted Files, Recovering Deleted Partitions, Data Acquisition and Duplication, Data Acquisition Tools, Recovering Data from Backups, Finding Hidden Data, Locating Forgotten Evidence, Defeating Data Recovery Techniques.

Collecting and Preserving Evidence: Understanding the Role of Evidence in a Criminal Case, Defining Evidence, Admissibility of Evidence, Forensic Examination Standards, Collecting Digital Evidence, Evidence Collection, Preserving Digital Evidence, Preserving Volatile Data, Special Considerations, Recovering Digital Evidence, Deleted Files, Computer Forensic Information, Understanding Legal Issues, Searching and Seizing Digital Evidence

Building the Cybercrime Case: Major Factors Complicating Prosecution, Difficulty of Defining the Crime, Jurisdictional Issues, The Nature of the Evidence, Human Factors, Overcoming Obstacles to Effective Prosecution, The Investigative Process, Investigative Tools, Steps in an Investigation, Defining Areas of Responsibility.

Self-Learning Contents:

Acquiring, Duplicating and Recovering Deleted Files: Deleted Partition Recovery Tools, Deleted File Recovery Tools, Data Acquisition and Duplication Tools, Defeating Data Recovery Techniques.

Collecting and Preserving Evidence: Data Recovery Software and Documentation, Computer Forensic Resources, Computer Forensic Training and Certification, Computer Forensic Equipment and Software, Computer Forensic Services.

Laboratory Work:

Hands with open source tools for forensic investigation process models (from Item confiscated to submitting evidence for lawful action), such as FTK, Sleuth Toolkit (TSK), Autopsy, etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Familiarize with cybercrime& forensics ontology
2. Analyse& demonstrate the crime scene and criminology.
3. Redesign the crime scene using digital investigation process
4. Recovery of evidence and creating document for judicial proceedings.

Text Books:

1. Shinder L. D., Cross M., Scene of the Cybercrime, Syngress (2008) 2nd ed.
2. Marcella J. A. and Guillossou F., Cyber Forensics: From Data to Digital Evidence, Wiley (2012).
3. Nina Godbole, Sunit Belapure, Cyber Security, Wiley (2011).

Reference Books:

1. Marcella J. A. and Menendez D., Cyber Forensics: A Field Manual for Collection, Examining and preserving Evidence of computer crimes. Auerbach Publication (2010), 2nd ed.
2. Dejey, Murugan, Cyber Forensics, Oxford (2018).

UCS649: ENGINEERING SOFTWARE AS A SERVICE

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course introduces standard concepts of software engineering and exposes students to the process of writing good and robust software to be used as a service.

Introduction to SaaS and Agile Development: Introduction, Software Development Processes: Plan and Document, Software Development Processes: The Agile Manifesto, Service Oriented Architecture, Software as a Service, Cloud Computing, Beautiful vs. Legacy Code, Productivity: Conciseness, Synthesis, Reuse and Tools.

The Architecture of SaaS Applications: Client-Server Architecture, Communication – HTTP and URIs, Template Views, 3-Tier Architecture & Horizontal Scaling, Model-View-Controller Architecture, Active Record for Models, Routes, Controllers, and REST, Representation – HTML and CSS.

Introduction to Ruby: Overview and Three Pillars of Ruby, Classes, Methods, and Inheritance, Meta-programming, Blocks: Iterators, Functional Idioms, and Closures, Mix-ins and Duck Typing, Make Your Own Iterators Using Yield, Fallacies and Pitfalls, Idiomatic Language Use.

Introduction to Rails: Rails Basics: From Zero to CRUD, Databases and Migrations, Models: Active Record Basics, Controllers and Views, Debugging, Form Submission: New and Create, Redirection and the Flash, Finishing CRUD: Edit/Update and Destroy, Designing for SOA, Perspectives on SaaS and SOA.

Advanced Rails: DRYing Out MVC: Partials, Validations and Filters, Single Sign-On and Third-Party Authentication, Associations and Foreign Keys, Through-Associations, RESTful Routes for Associations, Composing Queries With Reusable Scopes.

SaaS Client Framework: JavaScript Introduction: JavaScript: The Big Picture, Client-Side JavaScript for Ruby Programmers, Functions and Constructors, The Document Object Model and jQuery, Events and Callbacks, AJAX, Testing JavaScript and AJAX, Single-Page Apps and JSON APIs.

Requirements: BDD and User Stories: Introduction to Behavior-Driven Design and User Stories, Points, Velocity, and Pivotal Tracker, SMART User Stories, Lo-Fi User Interface Sketches and Storyboards.

Testing: Test-Driven Development: A RESTful API and a Ruby Gem, FIRST, TDD, and Red--Green—Refactor, Seams and Doubles, Expectations, Mocks, Stubs, Setup, Fixtures and Factories, Implicit Requirements and Stubbing the Internet, Coverage Concepts and Unit vs. Integration Tests, Other Testing Approaches and Terminology.

Maintenance: Legacy, Refactoring, and Agile: Exploring a Legacy Codebase, Establishing Ground Truth with Characterization Tests, Comments, Metrics, Code Smells, and SOFA, Method-Level Refactoring, The Plan-And-Document Perspective.

Laboratory Work:

This includes introduction and assignments related to Ruby on Rails, Ruby, ruby gems, jQuery and configuring database.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Explain the Agile Software Development concepts, Software as a Cloud Service and SaaS architecture
2. Construct a SaaS Application using Model–View–Controller (MVC) framework.
3. Design SaaS Client Framework using Java Script
4. Demonstrate the use of Behavior Driven Design (BDD) and User Stories for analyzing the requirements and designing the solution of Web Service
5. Apply Test Driven Development (TDD) approach to test the expected behavior of the functionality.

Text Books:

1. Fox, A., Patterson, D. and Joseph, S., Engineering Software as a Service: An Agile Approach Using Cloud Computing (2013), 1st Edition.
2. Eric Matthes, "Python Crash Course: A Hands-On, Project-Based Introduction to Programming", 2019, 2nd Edition, Packt Publishing.
3. Miguel Grinberg "Flask Web Development", 2018, 2nd Edition, O'Reilly.
4. Jake Kronika, Aidas Bendoraitis, "Django 2 Web Development Cookbook", 2018, 3rd Edition.

UMC632: FINANCIAL MATHEMATICS

L	T	P	Cr
2	0	2	3.0

Course Objectives: This is an introductory course in finance to equip with a framework and basic techniques necessary for financial engineering. The main focus is on valuation of financial assets and more specifically derivative products. The course will introduce the concept of risk and relation between risk and return. The knowledge of risk and valuation will be integrated in optimal decision-making. The models will be studied in discrete-time scenario.

Basics of Financial Mathematics: Financial markets, terminologies, basic definitions and assumptions, Interest rate, present value, future value, NPV, annuity and perpetuity, Market structure, no arbitrage principle, derivative products, forwards, futures – their valuation, dividend and non divided cases, options, swap, valuation concept, purpose and working of these products.

Theory of Option Pricing: Options-calls and puts, pay-off, profit diagrams, hedging and speculation properties of options, valuation of options using pricing and replication strategies, mathematical properties of their value functions, put-call parity, Risk neutral probability measure (RNPM) (discrete case), existence of RNPM, Binomial lattice model, Binomial formula for pricing European style and American style options, dividend and non-divided cases, CRR model, Black-Scholes formula derivation, Examples. Greeks and their role in hedging, delta-neutral portfolio, delta-gamma neutral portfolio

Portfolio Optimization: Introduction, risk, return, two-assets portfolio, Markowitz curve, efficient frontier, Multi-assets all risky portfolio, mean-variance Markowitz model, two fund theorem.

Laboratory activities:
Extraction of data from various online resources like NSE, moneyconrol.com etc. Implementation and validation of various models studied in the course for option and portfolio valuation using Matlab/R/Excel.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After completion of the course, students will be able to:

1. Understand basic quantities that are reported in everyday life such as interest rates, periodic payments of money, dividends, shares, bonds, forwards, futures etc.
2. Evaluate call and put option prices using binomial and CRR models.
3. Construct a portfolio which is optimal in a given market scenario.

Text Books / Reference Books:

1. D.G. Luenberger, Investment Science, Oxford University Press, 1999 (new edn. 2013).
2. S. Chandra, S. Dharmaraja, A. Mehra, R. Khemchandani, Financial Mathematics: An Introduction, Narosa, 2012.
3. M. Capinsky and T. Zastawniak, Mathematics for Finance: An Introduction to Financial Engineering, Springer, 2004 (new edn, 2011).
4. J C Hull, Options, Futures and other Derivatives, Prentice Hall, 8th edn, 2011.
5. J H Cochrane, Asset Pricing, Princeton University, 2000 (new edn 2005).

UCS752: AUGMENTED AND VIRTUAL REALITY

L	T	P	Cr
2	0	2	3.0

Course Objectives: To become familiar with the concept and applications of augmented & virtual reality and learn different types of algorithmic techniques and strategies.

Introduction of Augmented Reality (AR): Definition and Applications, History, Types of AR, Suitable devices, Holograms, Mixed reality, Ubiquitous computing, AR Displays: Method of Augmentation, Spatial Display Model.

Tracking in AR: Basic steps of AR, Tracking, Occlusion, Calibration, Registration, Co-ordinate Systems: Model-View-Projective Transformation, Frame of reference, Characteristics of Tracking Technology: Physical Phenomenon, GPS, D-GPS, Triangulation, Trilateration, Measurement Principles, Degree of Freedom, Stationary Tracking System, Mobile Tracking, Optical Tracking, Sensor Fusion.

Computer Vision for AR: Marker Tracking, Thresholding, Contour detection, Hough Transformation, Quadrilateral fitting, SIFT, Pose Estimation, Homography, Incremental Tracking, SLAM: Bundle Adjustment, Parallel Tracking and Mapping, Outdoor Tracking, STML.

Virtual Reality: Definition, History, Application, Types of VR, Components of VR,VR-HMDs and their working, Augmented virtuality, Geometric modeling, Modeling Transformation, Viewing transformation Chain and Rendering Pipeline, Light and Optical System, Rendering Problems in VR, Shading Models, Rasterization, Depth, Motion and Auditory Perception, Rendering, Post Rendering Image Warping.

Self-Learning Content: Calibration and Registration: Camera Representation, Camera Calibration, Display Calibration, Registration, Visual Coherence, Photometric Registration, Common Illumination, Diminished Reality, Camera Simulation, Stylized Augmented Reality.

Laboratory work:
To implement various types of AR and VR applications using Unity, Vuforia and Google VR SDK.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

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

1. Analyze the components of AR systems, its current and upcoming trends, types, platforms, and devices.
2. Understand the basic steps and technologies required to achieve AR system.
3. Apply various well-known computer vision algorithms in order to implement the AR.
4. Understand the various components, applications, latest devices and working model of VR systems.
5. Develop interactive augmented and virtual reality applications for PC and Mobile based devices using a variety of input devices.

Text Books:

1. Dieter Schmalstieg, Tobias Höllerer, Augmented-Reality-Principles-and-Practice-Usability-, Addison-Wesley (2016) 1st ed.
2. Parisi T., Learning Virtual Reality, O'Reilly (2016) 1st ed.
3. Gerard Jounghyun Kim, Designing Virtual Reality Systems: The Structured Approach, Springer (2005) 1st ed.

Reference Books:

1. Whyte J., Virtual Reality and the Built Environment, Architectural Press (2002).
2. Aukstakalnis S., Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR, Addison-Wesley (2016).

UCS753: DEEP LEARNING AND COMPUTER VISION

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of this course is to provide exposure to the students on the advances in learning theories and their applications to real life problems.

Deep Learning Networks: Deep networks for unsupervised and supervised learning, Hybrid deep networks, Deep auto-encoders including variational auto-encoders and its relationship with PCA, Pre-trained CNNs for classification and object detection.

Sequence Modelling: Recurrent Neural Networks (RNNs), BPTT, Truncated BPTT, Gated Recurrent Units, Long Short Term Memory.

Deep Generative Models: Basics of generative adversial networks (GANs), GAN training, Synthesizing and manipulating images with GANs.

Self Learning Content:
Machine Learning Basics: Learning, Under fitting, Over fitting, Estimators, Bias, Variance, Maximum likelihood estimation, Bayesian Statistics, Supervised learning, Unsupervised learning, Reinforcement learning, Stochastic gradient decent and its variants for Back-propagation, Regularization techniques.

Laboratory Work:
To implement the models included in this syllabus using open source libraries. The students will be encouraged to work on a project related with NLP/Speech Processing/Computer Vision etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of this course the student will be able to:
1. Analyze the advanced machine learning techniques.
2. Compare and explain various deep learning architectures and algorithms for auto-encoders and CNNs.
3. Experiment the working of sequence and generative models.
4. Apply deep learning specific open source libraries for solving real life problems.

Text Books:

1. Ian Goodfellow and YoshuaBengio and Aaron Courville, “Deep Learning”, MIT Press, 2016.
2. Michael Nielsen, “Neural Network and Deep Learning”, Online Book 2016.

Reference Books:

1. Le Deng and Dong Yu, “Deep Learning: Methods and Applications”, Foundations and Trends in Signal Processing, 2013.
2. Charu C. Aggarwal, “Neural Networks and Deep Learning”, Springer; 1st ed. 2018.

UCS754: BLOCKCHAIN TECHNOLOGY AND APPLICATIONS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of this course is to provide exposure on blockchain technology and its real-time applications.

Basic Cryptography: Introduction to cryptography and cryptanalysis, Cryptographic issues, cryptographic components, cryptographic techniques, cryptographic categories: symmetric key and asymmetric key cryptography, traditional ciphers, modern ciphers, message integrity, message authentication, key management, digital signatures, entity authentication, ECDSA, ECC, Ring, One time signature, Hashing: SHA-356, SHA-512, TLS and SSL, Timestamp, Public and Private keys, Merkle root hash.

Bitcoin Cryptocurrencies: What is Bitcoin, Brief history of Bitcoin, Bitcoin mining and supply, Bitcoin cryptocurrency (BTC), Traditional centralized vs. decentralized, Bitcoin's blockchain: evolution of blockchain, block header, genesis block, hash generation, Bitcoin address: formats, hash generation, address structure, transactions: multi-signatures, generating transactions, storing data, block verification and validation, block mining.

Smart Contracts: Introduction to smart contracts, smart contracts used in a centralized and decentralized systems, Blockchain platforms using smart contracts: Ethereum, architecture of Ethereum virtual machine, token- ETH, Mining process, ERC- standards, transactions in Ethereum, Hyperledger fabric, Sidechains, NXT, Stellar, R3Conda, Litecoin, Quorum, IBM, Openchain, Eris:db.

Consensus Mechanisms: Double spending problem, BFT, PBFT, PoW, PoS, DPoS, PoA, PoB, PoR, PoET, PoI, PoO, PoSp, PoC, Ripple, Tendermint.

Applications of Blockchain: Financial system, smart grid, healthcare, smart transportation system, e-Governance, education, exchange and trading, online market place, commercial supply chain, food production, drug manufacturing, safety and security.

Laboratory Work:
Experiments on creating of blockchain, implementation of smart contract on Python, Conda and Ethereum, Solidity.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:
1. Create their own blockchain using Block creation and verification
2. Create the smart contracts for transaction execution
3. Evaluate the performance of blockchain in presence of various attacks
4. Develop and validate various security models for real-life applications.

Text Book:

1. Melanie Swan, “Blockchain: Blueprint for a new economy”, Oreilly publications.

Reference books:

1. Bellaj Badr, Rcihcard Horrocks and Xun Brian Wu, “Blokchain by example”, Packt Publications.
2. Fatima Castiglione Maldonado, “Introduction to Blockchain and Ethereum”, Packt Publications.

UCS755: SOFTWARE VERIFICATION AND VALIDATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course enables students to understand the concepts and theory related to the software verification and validation testing.

Basics of Software Testing: Error, Faults, Failures, Testing and debugging, Test Metrics, Testing and verification, Test generation strategies, Static testing, Execution History, Model based testing and model checking, Control flow graph, dominators and Post Dominators, The saturation effect.

Test Generation from Requirements: The test selection problem, Equivalence partitioning, Boundary value analysis, Category partition method, Cause effect graphing, Test generation from predicates.

Test Generation from finite-state Models: Software design and testing, Finite state machine, Conformance testing, A fault model, Characterization set, The W-method, The partial W-method.

Test Adequacy assessment using control flow and data flow: Basics of Test adequacy, Adequacy criteria based on control flow, Data flow concepts, Adequacy criteria based on data flow, Control flow Vs Data flow.

Test Adequacy assessment using program mutation: Mutation and mutants, Test assessment using mutation, Mutation operations, Design of mutation operations, Types of mutants, Equivalent mutants, founding principles of mutation testing, Mutation operator in C and Java.

Self-Learning Content:
Test Oracles, Dependence graphs, Cyclomatic Complexity, Types of Software Testing Techniques, NUnit testing tool.

Laboratory Work:
1. Creating Adjacency matrix and adjacency list for a particular source code program written in (C/C++/Java).
2. Creating dependency (data as well as control) graph for the specific source code program written in (C/C++/Java).
3. Implementing genetic algorithm to generate mutants for taking a specific source code program written in C/C++/Java.
4. Exploring Selenium tool for testing.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Understand the basics of software verification and validation testing.
2. Design test cases from software requirements, and finite state models.
3. Assess the test adequacy using control flow and data flow constructs.
4. Grasp the concept of mutants and the process of mutation for test assessment.

Text Books:

1. Jorgensen C. P., Software Testing: A Craftsman's Approach, CRC Press (2014), 4th ed.
2. Mathur P. A., Foundations of Software Testing, Pearson (2013), 2nd ed.
3. Fisher S. M., Software Verification and Validation: An Engineering and Scientific Approach, Springer (2007).

Reference Books:

1. Beizer B., Software Testing Techniques, Van Nostrand Reinhold (1983), 1st ed.
2. Rakitin R. S., Software Verification and Validation for Practitioners and Managers, Artech House (2001), 2nd ed.

UMC742: COMPUTATIONAL NUMBER THEORY

L	T	P	Cr
2	**0**	**2**	**3.0**

Course Objective: The course intents to provide an introduction to elementary number theory, including theory of congruences, prime modulo, quadratic residues. The focus is then on to computational aspects and finding applications in cryptography that deals with secure encryption methods for communication.

Divisibility and Primes: Twin primes, Goldbach conjecture, Fermat and Mersenne primes, Primality testing and factorization.

Congruences: Linear congruences, Chinese Remainder Theorem, congruences with a prime-power modulus, Fermat's little theorem, Wilson's Theorem, Euler function, Quadratic Residues, Legendre Symbol, Euler's criterion.

Cryptography Basics: Symmetric and asymmetric key cryptography, Pseudo-primes, Pseudo-primality Testing, Randomized Primality test & Deterministic Polynomial Time Algorithm, Pollard-Rho Method.

Public key Cryptosystems: RSA, Diffie Hellmann key exchange, different attacks and Remedies, Digital Signature, Elliptic curve cryptography and its application in cryptography.

Laboratory work:
Implementation of various traditional ciphers, symmetric ciphers and asymmetric ciphers using C-programming language.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On successful completion of this course, students will have the knowledge and skills to:

1. Find the greatest common factor using the Euclidean Algorithm and investigate different factorization methods and primes
2. Solve linear and simultaneous congruences
3. Apply Wilson's and Fermat's Little Theorem as the basis for primality tests and factoring algorithms.
4. Apply and analyse elementary number theory concepts to symmetric and asymmetric key cryptography for encrypting and decrypting a message.

Text Books:

1. Neal Koblitz, A course in Number Theory and Cryptography, Springer, 2006
2. Niven, H.S. Zukermann, H.L. Montgomery, An introduction to theory of numbers, Willey, 2015
3. D. Burton, Elementary Number Theory, McGraw-Hill, 2012

Reference Books:

1. Behrouz A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, McGraw Hill, 2015.
2. J. Pipher, J. Hoffstein and J.H. Silverman, An introduction to Mathematical Cryptography, Springer-verlag 2014.

UCS538: Data Science Fundamentals

L	T	P	Cr
2	0	2	3.0

Course Objective: To elaborate the basics of data science and provide a foundation for understanding the challenges and applications.

Introduction to Python: Basic syntax, variables, Operators (Arithmetic Operators, Bitwise Operators, Assignment Operators, Comparison Operator, Logical Operators, Identity Operators, Membership Operators), Data types (Numbers, Booleans, Strings), Control Flow (if-else, for loop, while loop, break/continue), Sequence Generation (range function), String Operations (length, upper/lower, formatting, sub string, indexing, comparison, strip, split, count, search), Random Numbers, Functions.

Data Structures in Python: List, Tuple, Sets, Dictionary, Operations on Data Structures (Declarations, Iterations, Adding/deleting element, min/max/sorting, merge, select).

More of Python: Exception Handling, Command Line Arguments, Use of Libraries, File Handling (Read, Write, Merge, etc).

OOPs in Python: Classes, Objects, Inheritance, Create and upload packages on pypi.org.

Advance Topics in Python: Working with Numpy, Working with Scipy.

Plotting and Visualization in Python: Plotting using Matplotlib library (Histogram, Box Plot, Scatter Plot, Bar Graphs, Line Graph, etc)

Basics of Data Science: Handling of CSV files (Read, Write, Update, Transform), Measures of Central Tendency (Mean, Median, and Mode), Measures of Variability (Range, Interquartile Range, Variance, and Standard Deviation),

Visualization: -Traditional Visualization, Multivariate Data Visualization, Principles of Perception, Color, Design, and Evaluation, Text Data Visualization, Network Data Visualization, Temporal Data Visualization and visualization Case Studies.

Advances in Data Science: Basics of Correlation, Regression, Working with Pandas, Working Scikit-Learn, Featuring Engineering, Probability and Random Variables, Correlation, Regression, Models for Structured and Unstructured Data, Storage and Retrieval of Structured Data, Predictive Analytics of Structured Data using Python, Big Data and Distributed Databases, Storage and Retrieval of Unstructured Data, The HDFS File System, Basics of Machine Learning (Decision Trees and Neural Networks).

Laboratory work: Implementation of various data analytics techniques such as classification clustering on real world problems using Python.

Course Learning Outcomes (CLO):
On completion of this course, the students will be able to:

1. To analyse the need and usage of analytics and visualization techniques.
2. To implement how to manage, manipulate, cleanse and analyse data.
3. To develop dashboards for real-time data sets.
4. To visualized the dataset using different visualization techniques.
5. To demonstrate the use of Python on real-life problems.

Text Books:

1. Jiawei Han, Micheline Kamber, Jian Pei ,Data Mining Concepts and Techniques, (3^{rd} Ed.),Morgan Kaufmann
2. Roger D. Peng R Programming for Data Science

Reference Books:

Trevor Hastie Robert Tibshirani Jerome Friedman, The Elements of Statistical Learning, Springer

UCS654: PREDICTIVE ANALYTICS USING STATISTICS

L	T	P	Cr
2	0	2	3.0

Course Objectives: Advanced analytics requires the use of unstructured data. Uncertainty is a primary characteristic of unstructured data. Statistical methods that relate to correlating information, finding patterns, predictive modeling are essential in dealing comprehensively with data so that it can used as information to make decisions. This course will provide an overview of statistical methods relevant in the world of business analytics. This will be demonstrated through the use of case studies and statistical software.

Descriptive statistics, Introduction to Analytics, Business Understanding, Introduction to R-Basics, The R Environment, Inductive and Deductive Logic, Installation of R.

Distributions, Continuous Distributions, Probability, Decisions Trees, Machine Learning, Probability and Distributions, Descriptive Statistics and Graphs.

Sampling-Distributions, Parameter-Estimations, Hypothesis-Testing, Two-population, Linear Regression, Tests, Regression and Correlation, Block Chain.

UniVariate-Analysis, Multi-Variate, ANOVA and Tabular Data, Fake News Detection.

Hierarchical-KMeans Clustering, Neural Networks, Genetic Algorithms, Power, Sample size, Data Handling, Multiple Regression, Introcution Modeling, Visualization, Real-time Decision Making.

Survival Analysis and Nonlinear Curves, Digital Genesis, Big Data and Analytics. Nonlinear Curve Fitting, Affective Computing and Sentiment Analysis.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Demonstrate the ability to use basic probability concepts with descriptive statistics.
2. Analyze statistical data graphically using central tendencies with the help of R programming language.
3. Demonstrate the use of statistical inferential methods to estimate population characteristics from the given data.
4. Explain and demonstrate the use of predictive analytics in the field of data science.

Text Books:

1. Peter Dalgaard, Introductory Statistics with R, Springer, Second Edition, ISBN: 978-0-387-79053-4
2. Brett Lantz, Machine Learning with R (2nd Edition), www.PacktPub.com.

Reference Books:

1. Online Resources: (e.g., http://r-statistics.co/)
2. Introduction to Machine Learning in R
https://www.kaggle.com/camnugent/introduction-to-machine-learning-in-r-tutorial

UCS655: AI Applications – NLP, Computer Vision, IoT

L	T	P	Cr
2	0	2	3.0

Course Objectives: There have been many recent advances in the field of Artificial Intelligence. The objective of the course is to provide exposure to these advances and facilitate in depth discussion on chosen topics.

Introduction to Artificial Intelligence: Introducing AI and its importance in transforming human life, Canonical architecture of an AI system, AI systems in practice, Machine Learning basics, Five tribes of Machine Learning, Selection of Machine Learning method, Machine learning applications in sales, marketing and information security.

Fundamentals of Natural Language Processing and its applications: What is NLP, Difficulties in NLP, Basics of text processing and spelling correction, Introduction to language modeling, Application of NLP in sentiment analysis.

Fundamentals of Computer Vision and its applications: Introduction and goal of computer vision, Basics of image processing and formation, Convolutional neural network, Application of computer vision in face recognition.

Artificial Intelligence and IoT: Basic architecture of an IoT system, Role of AI in IoT, Blockchain and AI for an Intelligent IoT system, Applications in Smart City and Smart Energy Grids.

Ethics of AI and AI systems: Robustness and transparency, data bias and fairness, accountability, privacy and Human-AI interaction.

Laboratory Work: To implement models and use cases using python and google open source library Tensorflow.

Course Learning Outcomes (CLOs)

After the completion of the course, the student will be able to:

1. Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems.
2. Demonstrate awareness and a fundamental understanding of various applications of AI techniques.
3. Comprehend the advancements in machine learning techniques.
4. Acquire knowledge to apply open source libraries for solving real life problems.

Text books:

1. Artificial Intelligence: A Modern Approach, by Stuart Russell and Peter Norvig, Pearson.
2. Speech and Language Processing, by M. Jurafsky, & J. Martin, New York: Prentice-Hall (2000).
3. Computer Vision: Algorithms and Applications, by Richard Szeliski, Springer.
4. Deep Learning, by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016.

Reference Books:

1. Artificial Intelligence theory and practice, by T. Dean, J. Allen & Y. Aloimonos, New York: Benjamin Cummings (1995).
2. The Internet of Things by Samuel Greengard, MIT Press Essential Knowledge series, (2015).

UCS757: BUILDING INNOVATIVE SYSTEMS

L	T	P	Cr
2	0	2	3.0

There will be three projects that will be required to be completed for this course.

- The first project will be in predictive analytics using statistics.
- The second project will involve the use of machine learning and/or AI technologies.
- The third project will involve the building of a prototype for a real-world application using the course content learned during this Elective Focus basket.

Course Learning Outcomes (CLOs)

After the completion of the course the student will be able to:

1. Work on development of exploratory data analysis on the given dataset using statistical methods.
2. Develop a proof of concept for machine learning project covering data selection, preprocessing, modeling, optimization, and deployment.
3. Apply hyper parameter tuning on the machine learning models under development.
4. Illustrate the knowledge of machine learning and statistics to build an end-to-end project for solving real-world problems.
5. Comprehend the development of the project in a well-documented report.

Text books:

1. Freedman, D., Pisani, R., and Purves, R., Statistics, W. W. Norton & Company; Fourth edition (2007).
2. Müller, A. C., and Guido, S., Introduction to machine learning with Python: a guide for data scientists, O'Reilly Media, Inc. (2016).

Reference Books:

1. Bruce, P., Bruce, A., and Gedeck, P., Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python, O'Reilly Media, (2020).
2. Géron, A., Hands-on machine learning with Scikit-Learn, Keras, and Tensor Flow: Concepts, tools, and techniques to build intelligent systems, O'Reilly Media, (2019).

UCS539: Finance, Accounting and Valuation

L	T	P	Cr
3	0	2	4.0

Course Objectives: Understanding relationship of finance, accounting and valuation of securities.

Introduction to Accounting: Meaning of accounting, the accounting process, fundamental equation, types of accounts, accounting statements, recording of transactions, conceptual framework

Summary Statements: Types of summary statements, preparation of the statements, relationship between the statements, introduction to financial statement analysis

Basics of Finance: Meaning of finance, process of financial decision making, types of financial decisions

Basic Corporate Finance: Capital budgeting decisions, cost of capital, capital structure decisions

Time Value of Money: Meaning, principle, calculations, interest rates, importance of interest rates

Valuation: Introduction to valuation, valuation of stocks, valuation of bonds, methods and techniques

Practical sessions: To explore several formula in Microsoft Excel and to gain an understanding of a proprietary software

1. Introduction to proprietary software
2. Detailed understanding of basic features of proprietary software
3. Details of formulae (intermediate and advanced) related to Microsoft Excel, syntax, troubleshooting
4. Other advanced concepts related to Microsoft Excel not including VBA programming

Recommended Prerequisites: Basics of Microsoft Excel

Course learning outcomes (CLOs): After the completion of the course, the student will be able to:

1. Explain the basic accounting concepts and apply the fundamental equation in basic business transactions
2. Explicate and apply the techniques learnt for doing financial statement analysis
3. Explain various financial decisions and evaluate some of them
4. Explicate the principle of time value of money (TVP) and importance of interest rates in TVP
5. Apply the methods learnt for valuation of securities

Reference Books:

1. Jamie Pratt. (8th Edition). Financial Accounting in an Economic Context.
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

UCS657: Financial and Derivative Markets

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding various financial and derivative markets and their relationships.

Banks and Financial Institutions: Types of financial institutions, evolution of financial system, flow of money, creation of money

Monetary System: Monetary authority, monetary policy framework, policy tools, comparison of different countries

Risk and Return: Meaning of risk, meaning of return, estimation of risk and return, introduction to derivatives, role of derivatives in risk reduction

Capital and Money Markets: Meaning, types, role and functions, instruments

Portfolio Theory: Meaning of portfolio, theoretical principles, choices, optimal choice, introduction to pricing models

Practical sessions:
1. To use the proprietary software in live international derivatives
2. Introduction to an international derivatives product
3. Introduction to technical analysis
4. Practice of analytical methods on the proprietary software
5. Using Excel to evaluate some tools and techniques

Recommended Prerequisites: Course – Finance, Accounting and Valuation; Excel (intermediate)

Course learning outcomes (CLOs):
After the completion of the course, the student will be able to:
1. Explain the role of the financial system and the process of creation of money
2. Explicate various monetary policy tools
3. Explain relationship between risk and return, and role of derivatives in risk reduction
4. Explicate the types of capital markets and money markets including the market instruments
5. Apply the portfolio theory to choose an optimal portfolio

Text Books:

1. Financial Markets and Institutions – Anthony Saunders & Marcia Millon Cornett
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

Reference Books:

1. Financial Markets and Institutions – Anthony Saunders & Marcia Millon Cornett
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

UCS658: Derivatives Pricing, Trading and Strategies

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding methods of valuation and strategies of trading derivative instruments

Law of One Price: Meaning, implication of the law of one price, no arbitrage model, usage in pricing of securities and derivative instruments

Pricing and Valuation: Basic principles, building blocks, assumptions, difference between price and value, pricing and valuation of basic derivative instruments

Basics of Stochastic Processes: Introduction to stochastic processes, types, relevance to finance, need for stochastic processes

Basics of Option Pricing: Meaning of options, types of options, difference between options and basic derivative instruments

Practical sessions: To use the proprietary software in live international derivatives

5. Introduction to fixed income derivatives
6. Introduction to an international fixed income derivatives product
7. Learning and creating trading strategies
8. Practicing the strategies on the proprietary software
9. Using Excel to evaluate the strategies

Recommended Prerequisites: Courses – Finance, Accounting and Valuation + Financial and Derivative Markets; Excel (intermediate)

Course learning outcomes (CLOs):

After the completion of the course, the student will be able to:

6. Explain and apply the law of one price in pricing models
7. Explicate basics of stochastic processes
8. Explain and apply formulae to calculate price and value of basic derivative instruments
9. Explicate the meaning and types of options
10. Apply the basic methods to calculate option prices

Reference Books:

1. Fundamentals of Futures and Options Markets - John C. Hull
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

UMC743: QUANTITATIVE AND STATISTICAL METHODS FOR FINANCE

L	T	P	Cr
2	0	2	3.0

Course Objectives: Detailed study of probability distributions, financial time series, Principal components, Pricing and risk management models

Refresher on Statistics: Correlation, OLS regression, probability distributions and moments, using Microsoft Excel for statistical calculations and interpretations.

Financial Time Series: Introduction to time series, types, univariate and multivariate time series models, autocorrelation, AR models, MA models, ARMA models, ARIMA models, stationary series, unit-root, cross-correlation, cointegration, vector auto regression (VAR).

Principal Component Analysis: Meaning of PCA, methods and interpretation, usage

Option Pricing and Risk Management: Introduction to option pricing models, formulae and derivation, option Greeks, risk management using options

Volatility: Meaning of volatility, types, methods of calculation, volatility models, estimation.

Practical sessions: To use the proprietary software in live international derivatives

1. Introduction to other derivative products
2. Refining trading strategies created in previous courses
3. Practicing the strategies on the proprietary software
4. Evaluating the strategies using quantitative and statistical methods

Recommended Prerequisites: Courses – Finance, Accounting and Valuation + Financial and Derivative Markets + Derivatives Pricing, Trading and Strategies; Excel (intermediate)

Course Learning Outcomes (CLOs):

After the completion of the course the student will be able to:

1. Comprehend the concept of probability distributions, their properties and relevance to real-life data.
2. Analyze methods and theories in the field of finance and provide an introduction to the basic principles, techniques, and applications of financial time series
3. Understand the concept of PCA, volatility and its implementation in context of financial domain.
4. Comprehend the art of pricing and risk management.

Text Books:

1. Ruey S. Tsay, Analysis of Financial Time Series, Wiley (2010); 3rd edition.
2. Sheldon Natenberg, Option Volatility and Pricing: Advanced Trading Strategies and Techniques, McGraw-Hill; 2nd edition.

Reference Books:

1. John L. Teall and Iftekhar Hasan, Quantitative Methods for Finance and Investments, Wiley-Blackwell; (2002) 1st ed.
2. I.T. Jolliffe, Principal Component Analysis (Springer Series in Statistics), Springer; 2nd ed.

UCS537: SOURCE CODE MANAGEMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to combine software development and IT operations using DevOps. It helps to understand faster software development practices with higher quality.

Traditional Software Development: The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict.
Rise of Agile methodologies: Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, working software over comprehensive documentation, Customer collaboration over contract negotiation, responding to change over following a plan
Definition and Purpose of DevOps: Introduction to DevOps, DevOps and Agile, Minimum Viable Product, Application Deployment, Continuous Integration, Continuous Delivery
CAMS (Culture, Automation, Measurement and Sharing): CAMS – Culture, Automation, Measurement, Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning.
Typical Toolkit for DevOps: Introduction to continuous integration and deployment, Version control system
Source Code Management History and Overview: Examples - SVN, Mercury and Git, History - Linux and Git by Linus Torvalds,
Version Control System: Version control system vs Distributed version control system: Local repository, Advantages of distributed version control system, The Multiple Repositories Models, completely resetting local environment, Revert - cancelling out changes.

Laboratory work:
Basic structure and Implementation of various distributed version control systems for source code management.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:

1. Identify the need for migrating from traditional software development to Agile model and then to DevOps.
2. Define and understand the basic principles and need of DevOps and Continuous Delivery.
3. Understand the history and overview of Source Code Management, along with real-time examples.
4. Differentiate between centralized and distributed version control systems and basic operations in version control systems and Demonstrate the use of various version control systems.

Text Books:

1. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis Willis.
2. Pro Git – Book by Scott Chacon and Ben Straub (available at https://git-scm.com/book/).

Reference Books:

1. What is DevOps? - by Mike Loukides.

UCS659: Build and Release Management

L	T	P	Cr
2	0	2	3.0

Course Objectives: The main objective of this course to help students understand the process of build and release management.

Introduction to Build and Release Management: Introduction to build, understanding different phases of build and release management, introduction to release management, best practices for build and release management, concept of build abstraction and dependency abstraction.

Dependency Management: Introduction to dependency management, how to use source code repositories, managing transitive dependencies, dependency scope and discussion of various tools like Ant, Maven and Gradle.

Document and Reporting: Introduction to build document and reporting, different types of documentation, understanding site life cycle, advance site configurations and reports, generation of unit test reports, generation of code coverage reports, code coverage tools, code coverage pros and cons.

Release Cycle: To understand project release life cycle, different stages of release lifecycle, source code repositories, how to install and configure source code repositories and deploying build to production goals- prepare, perform, clean and rollback.

Laboratory Work:

- Setting up Maven environment and understanding POM hierarchy, creation of a project using Maven and its configurations.
- Handling Azure DevOps environment using several artifacts.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Able to understand the concepts of Build and Release management by learning build abstraction and declarative dependency management.
2. Grasp the knowledge about dependency management and its related concepts such as repositories, transitive dependencies and the tools for building the build and release pipeline.
3. Understand the process of documentation and reporting using code coverage reports.
4. Understand the complete cycle of release pipeline.

Text Books:

1. Kent B., Test Driven development, McGraw Hill (2011) 1st Ed.
2. Humble J., Farley D., Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Ed.
3. Sharma S., The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise Wiley; 1st Ed., 2017.
4. Relan K, Building REST APIs with Flask: Create Python Web Services with MySQL, Apress, 1st Ed., 2019.

UCS660: Continuous Integration and Continuous Deployment

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to automate the process of integration and deployment software product. It covers prerequisites, anatomy and framework/tools used for the automated process of continuous integration and continuous deployment.

DevOps Automation: Phases in software development-delivery pipeline, components of automated software delivery, RAD model and model driven architecture.
Automation Benefits: advantages of automation, Time and efforts saving scenarios, error preventing scenarios.
Continuous Integration and Continuous Deployment Introduction: Overview and practices of continuous integration, working mechanism and benefits of continuous integration; continuous delivery's introduction and pipeline. Prerequisites and benefits, introduction and business drivers of continuous deployment, benefits of continuous deployment.
Stages and Anatomy of CI CD: Core continuous integration process and advanced continuous integration process, release process, continuous delivery engineering practices, continuous testing & promotion of builds, continuous monitoring of delivery pipeline, understanding continuous feedback process.
Testing, Debugging and Refactoring: Understanding test-driven development (TDD), categories of TDD, Junit framework, need for code refactoring, its process and strategies.
Understanding Framework and Tools: Common frameworks and code architectures, third party code, IDEs (Eclipse, Netbeans and IntelliJ), common mistakes and avoiding them, issues with making code IDE dependent.

Laboratory work:
Setting up Jenkins, Jenkins job, parameters, build, post-build actions and pipeline; Jenkins plugins, using Jenkins as a continuous integration server; Configuring Jenkins with git plugin; Jenkins pipeline to poll the feature branch.

.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:
1. Understand the phases of software development-delivery pipeline and automation benefits.
2. Identify and apply continuous integration and deployment prerequisites, process and benefits.
3. Understand and implement the continuous delivery engineering practices and release process.
4. Identify & use the test-driven deployment and various tools/frameworks used for continuous integration and delivery in DevOps

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", IT revolution Press (2016) 1^{st} ed.

Reference Books:

1. Sander Rossel, "Continuous Integration, Delivery, and Deployment: Reliable and Faster Software Releases with Automating Builds, Tests, and Deployment", Packt Publishing (2017) 1^{st} ed.
2. Online material available at:https://digitallearn.xebiaacademyglobal.com/

UCS758: System Provisioning and Configuration Management

L	T	P	Cr
2	0	2	3.0

Understanding Containers: Transporting goods analogy and its problems, Containerization platform, images and runtime, comparison with virtual machine, chroot system call, FreeBSD Jails, LinuX containers (LXC), Docker.

Introduction to Containerization: Docker architecture, different environments (Dev, QA and Prod), overcoming issued with different environments, virtual machine for dev/deployments, containers for dev/deployments, advantages and drawbacks of containerization.

Orchestration Tools: Orchestration: its definition and need, Docker swarm and Kubernetes, AWS (ECS and EKS), Kubernetes on cloud, monitoring containers and its process.

Introduction to Provisioning: Basic and software definition, provisioning concepts, reason for exclusive provisioning, configuration management definition and tools, difference between provisioning and configuration management, provisioning tools, test machines for provisioning, deployment and its relationship with provisioning.

On Premise Provisioning: Understanding and Defining On Premise, On Premise provisioning infrastructure, Templating, server templating and its challenges.

Provisioning on Cloud: defining cloud provisioning, types of cloud provisioning, life-cycle of provisioning on cloud, On Premise cloud mitigation strategies, network security enablement from On Premises to cloud, micro-services management in cloud.

Provisioning and Configuration Management: State of tools in provisioning and configuration, definition and need for configuration management, its benefits and drawbacks in DevOps, need for monitoring in DevOps, reasons for using provisioning and configuration tools, automation, preventing errors and tracking changes, examples of tools and their capabilities.

Laboratory Work:

System Provisioning: Automation of infrastructure, AWS configuration for Terraform, create IAM User, security group, spinning up with EC2 instance, variables,, resources, modules, state management, VPC, IAM policy, S3 bucket and its variables.

Containers Lab: Playing with Vagrant and understanding its file, Docker machine, Dockerfile, Docker extras, DTR, Docker compose and swarm, Kubernetes -Minikube, deploying Pods and services on Minikube.

UCS542: UI & UX SPECIALIST

L	T	P	Cr
2	0	2	3.0

Course Objectives: The main objective of this course is to impart concepts related with web technology which are essential for the development of web applications. The key technology components include web markup languages, client and server side programming.

HTML& CSS: Introduction to HTML, Introduction, HTML Page Structure, Create HTML document, Understand the various elements available in HTML, HTML Use, Attributes in HTML, Need of Attributes, Common Attributes, HTML forms, Apply validations to the form elements, Creating web pages with HTML5, HTML5 introduced features, HTML5 form validate/no validate, HTML5 canvas, embedding audio, and video in a webpage, drag and drop, HTML5 Local Storage, HTML5 web workers and server sent events. What is CSS, how to insert CSS in HTML, How CSS adds value to HTML, Difference between Semantic and HTML mark-up, CSS3, CSS Selectors, Buttons, CSS float and clear, CSS align - horizontal and center, CSS Padding, CSS Links, CSS Lists, CSS Tables.

JavaScript: What is JavaScript, Importance of JavaScript, What can JavaScript Do?, JavaScript with HTML Attributes, JavaScript with CSS, Operators, JavaScript Syntax, JavaScript Data Types, JavaScript Functions, Setting up Environment, Variables, Control flow, if. Else, switch, loops, JavaScript HTML DOM Elements, JavaScript Syntax, Operators, Data Types, JavaScript String Methods, JavaScript Functions, Arrays, Sorting, Joins, Reduce map.

Frontend Architecture: Introduction to Frontend Development, History, MVC, MVP, MVVM& Web Apps, Development of AJAX, Introduction to DOM, Basic DOM Manipulation, Reactive Programming, Refreshing ES6 Specifications and Features ECMA Script, ES6 let and const, The arrow functions, New Literal Syntax, Classes, Inheritance using extends, Default Parameter Values, Spread Operator (...), Iterators and Generators, Features of React, Practical Application, Why need React, How React Works, Leveraging Virtual DOM, Setting up React.

REST API, JSX: Why JSX, Embedding JavaScript Expression in JSX, JSX Attributes, JSX Comments, Styling and Representation as Object, The State of the Component, Changing the State, Props of Component, Using Props, Props Validation in React, Similarities Between State & Props RESTAPI : Intro to API, History of API Development, Development of AJAX, CRUD; GraphQL; HTTP ,HTTP 1.1,HTTP/2, Stream prioritization, Introduction to React Native, Setting up React Native, The Expo Client, Working up on the First Project, Style, Fexbox Layout.

Node.js: Introduction to Node.js, History, Why Node.js, Node.js Architecture, Features, Working of Node.js, Installation & Setting Up Node, setting up React, REPL Environment, REPL Commands, Variable, Components of Node.js, Local Modules, Module Exports:

Export Object, Export Class, Loading Module from Separate Folder, Operating System, File Systems.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able:

1. To acquire knowledge and skills for creation of full web framework considering both client and server side programming.
2. To develop a dynamic and interactive WebPages by the use of HTML, XHTML, CSS, Java script and DHTML.
3. To build web apps with fully functional frontend with database connectivity.
4. To undertake transfer of multimedia content over the Internet between the multi platforms web applications through communication protocols and the learner will also be able to build web API.

Text Books:

1. M. Srinivasan, Web Technology: Theory and Practice, Pearson (2012).
2. A.Kumar, Web Technology: Theory and Practice, CRC Press (2018).

Reference Books:

1. Web programming with HTML, XHTML and CSS, 2e, Jon Duckett, Wiley India.
2. Web technologies: A Computer Science Perspective, Jeffrey C. Jackson ,1st ed., Pearson (2007).

UCS661: DATABASE ENGINEER

L	T	P	Cr
2	0	2	3.0

Course Objectives: Basic concepts of database, Mongo DB, SQL, and Java script.

Getting started with MongoDB: No SQL Databases, Features of MongoDB, Installation overview, Documents, Collections, Databases, What isthe NoSQL approach? Why Use the NoSQL Approach, Benefits of No SQL, Types of Databases, Key-Value Stores, Wide-column Stores/ Columnar Databases, Document/Document-store/Document-oriented Databases, Graph-based Databases, Starting and stopping MongoDB

Javascript in MongoDB: Javascript in MongoDB, Execution of a JavaScript file in MongoDB, Making the output of find readable in shell, Complementary Terms, Installation, Basic commands on mongo shell, HelloWorld, Create, Update Delete, Read, Update of embedded documents, more update operators, Updating multiple documents.

Collections: List all collections in the database, List all databases, Find(), FindOne( ), limit, skip, sort and count the results of the find() method, Query Document – Using AND, OR and IN Conditions, find() method with Projection, Find() method with Projection, $set operator to update specified field(s) in document(s), Insert a document, Create a Collection, Drop Collection, Aggregation

Indexes: Indexes, Index Creation Basics, Dropping/Deleting an Index, Sparse indexes and Partial indexes, Get Indices of a Collection, Compound, Unique Index, Single field, Delete, List, Mongoas Shards

Sharding Environment Setup: Managing Database for Availability and Performance, Database Scaling, Database Distribution Models, Database Replication, Types of Database Replication, Master-Slave Replication, Peer-to-Peer Replication, Advantages and Disadvantages of Peer-to-Peer Replication, Introduction to Sharding, Why Sharding, The Lookup Strategy, Basic configuration with three nodes, Mongo as a Replica Set, Mongoose.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Learn NoSQL, Mongo DB, and Javascript.
2. Apply basic concepts of indexing and collections using MongoDB.
3. Formulate the basic usage of database such as graph-based database, creating and dropping of collections, and indexing and solving real-world problems.
4. Able to handle and update multiple documents using Mongo DB.
5. Able to manage database for availability and performance, and scaling of database.

Text Books:

1. Andreas Kretz, The Data Engineering Cook Book, 6^{th} ed. (2019)
2. Alex Petrov, Database Internals: A deep dive into how distributed data systems work, O'REILLY Publication (2021).

Reference Books:

1. S. Bradshaw, Eoin Brazil, and Kristina Chodorow, MongoDB: The definite guide: Powerful and Scalable Data Storage, O'REILLY Publication (2021).
2. Dan Sullivan, NoSQL for Mere Mortals, O'REILLY Publication (2021).

UCS662: TEST AUTOMATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: The course provides understanding of software testing and how to use various tools (like Selenium and TestNG etc.) used for automation of software testing.

Introduction to Software Testing: Seven principles of Software Testing, SDLC vs STLC, Testing Life Cycle, Usability Testing, why do we need Usability Testing, how to do Usability testing, Advantages & Disadvantages, Functional Testing, End to End Testing, Methods, Advantages & Disadvantages, Compatibility Testing, Types GUI testing, Techniques API testing, Advantages

Test Automation: Selenium: Selenium components, Selenium Architecture, TestNG: Installing TestNG in Eclipse, TestN Gannotations – Understanding usage, setting priority of execution for test cases, Hard Assertion, Soft Assertion, TestNG Reports, ANT-Downloading & Configuring, XSLT report generation using TestNG and Ant.

Introduction to Selenium 3.x: Describe Selenium 3.x advantages and implementation, Define drivers for Firefox, IE, chrome, IPhone, Android etc. Analyze first Selenium Code, differentiate between Close and Quit, Describe Firepath and firebug Add-ons installation in Mozilla, inspect elements in Mozilla, Chrome and IE, Identifying Web Elements using id, name, class, Generate own CSS Selectors. Differentiate between performance of CSS Selectors as compared to Xpaths, define class attribute, Handle Dynamic objects/ids on the page, Analyze whether object is present on page or not

Manual Testing: Manual Testing, Manual Testing – How to Approach? Manual Testing – Myth and fallacy, Defect Life Cycle, Qualities of a good Manual Tester, Manual Testing Vs Automation Testing, Types, System Testing, Acceptance Testing, Unit Testing, Techniques, Integration Testing, Smoke- Sanity Testing

Introduction to Test Design: Test Scenario, Test Case Design, Test Basis Traceability Matrix

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able to:

1. Understand the concepts related to software testing and test automation.
2. Take into account the different considerations when planning automated tests vs. manual tests
3. Architect the test project and fit it to the architecture of the tested application
4. Design and implement highly reliable automated tests
5. Understand how different types of automated tests will fit into your testing strategy, including unit testing, load and performance testing, visual testing, and

more

Text Books:

1. Axelrod Arnon, Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects, A press (2018).
2. Gundecha U. and Cocchiaro C., Learn Selenium: Build data-driven test frameworks for mobile and web applications with Selenium Web Driver 3, Packt (2019).

Reference Books:

1. Diego Molina, Selenium Fundamentals, Packt (2018).
2. Aditya P. Mathur, Foundations of Software Testing, Pearson Education(2008).

UCS745: CLOUD & DEVOPS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to automate the process of integration and deployment software product.

Introduction to DevOps: Definition of DevOps, Challenges of traditional IT systems &processes, History and emergence of DevOps, DevOps definition and principles governing DevOps, DevOps and Agile, The need for buildinga business use case for DevOps, Purpose of DevOps, Application Deployment, Automated Application Deployment, Application Release Automation (ARA), Components of Application Release Automation (ARA), Continuous Integration, Best Practices of CI, Benefits of CI, Continuous Delivery

Typical Toolkit for DevOps: DevOps, An Overview, Achieving DevOps, Continuous Practices, Continuous Integration (CI), How does CI Work? Continuous Integration Practices, Benefits of Continuous Integration A Quick Recap of Continuous Delivery, Continuous Delivery Process, Benefits of Continuous Delivery, Continuous Deployment

Source Code Management: History of Version Control Systems (VCS), Basic operations in a VCS, Examples of version control systems, Subversion (SVN), Features and Limitations, Mercurial, Git, Overview, History - Linux and Git by Linus Torvalds, Advantages of Git, Explain how local version control works, Centralized Version Control Systems (CVCS), Distributed Version Control Systems (DVCS), advantages of DVCS, Private Workspace, Easier merging, Easy to scale horizontally, List the disadvantages of DVCS, Explain how CVCS and DVCS compare with each other, Describe the working of the multiple repositories model Unit IV Application Containerization Understanding Containers: Transporting Goods Analogy, Problems in Shipping Industry before Containers, Shipping Industry Challenges, Container: Virtualization Introduction, Hypervisor, Scope of Virtualizations, Containers vs Virtual Machines, Understanding Containers, Containerizations Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, LinuX Containers (LXC), Docker

Introduction to Containerization: Docker architecture, Docker Daemon (Container Platform), Docker Rest API, CLIDifferent environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment Docker Swarm and Kubernetes, Architecture, AWS (ECS, EKS), AWS Elastic Container Services Architecture, Azure Kubernetes Services, Openshift, Kuberneteson cloud, Monitoring of container

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the benefits of DevOps over other software development processes.
2. Understand the phases of software development-delivery pipeline and automation benefits.
3. Identify and apply continuous integration and deployment prerequisites, process and benefits.
4. Understand and implement the continuous delivery engineering practices and release process.
5. Identify & use the test-driven deployment and various tools/frameworks used for continuous integration and delivery in DevOps.
6. Demonstrate the different DevOps Tools like Git, Docker, and Kubernetes etc.

Text Books:

1. Arundel, J., & Domingus, J., Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the cloud. O'Reilly Media, (2019).
2. Kim, G., Humble, J., Debois, P., & Willis, J., The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT revolution Press 1st ed(2016).
3. Bass, L., Weber, I., & Zhu, L., DevOps: A software architect's perspective. Addison-Wesley Professional (2015).

Reference Books:

1. Fox, A., Patterson, D. and Joseph, S., Engineering Software as a Service: An Agile Approach Using Cloud Computing, 1st Edition (2013).
2. Rossel, S., Continuous Integration, Delivery, and Deployment: Reliable and Faster Software Releases with Automating Builds, Tests, and Deployment. Packt Publishing, 1st ed (2017).

UCS543: Conversational AI: Data Analytics

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understand the basic concepts of data mining processes, algorithms, and systems. Impart knowledge about how Data Mining provides techniques for collecting information from distributed Databases and then performing Data Analysis.

Introduction: Introduction to Probability and Statistics, Linear Algebra, Differentiation, GPU Computing, Data Science: Data Collection, Preprocessing and Analytics.

Introduction to Probability and Statistics: Fundamentals of Probability: Counting, Random Variables, Conditional probability, Bayes' theorem, Joint distributions, covariance, correlation, independence, Central Limit Theorem, Statistics: Bayesian inference, Bernoulli and Poisson processes, Markov Chains.

Fundamentals of Data Science: Linear Algebra, What is Data Science and its Importance, Analytical Building Blocks, Traditional Data Science packages, GPU Computing.

Data Collection: Collecting Data, Scraping Data, Popular Scraping libraries, Data Annotation and Data Quality, SQLits as Simple, SQL Refresher, Missing Indexes in Datasets.

Data Pre-processing (ETL): Introduction to Data-preprocessing, Data cleaning, Statistical pre- processing, Data Cleaners: OpenRefine and Wrangler, Feature Selection: Introduction to Filter Methods, Introduction to Model- based methods, Feature Reduction: PCA.

Data Analytics and Visualization: Bias and Fairness Measures, Knowledge Graph, Data deduplication, What is info viz, Exploratory Data Analytics.

Distributed Computing: Hadoop, Spark, Dask and UCX: Hadoop: Big data is common, How to store Big data, Hive Overview and Comparing pig, Spark: Overview of Spark SQL and other libraries, RAPIDS & Spark Integration, Dask and UCX with RAPIDS and BlazingSQL.

Laboratory Work:

- Introduction to Dockers and Containers, Introduction to NVIDIA GPU Container Store (NGC), Getting started with Docker Platform.
- Practical on Traditional Data Science packages (Numpy, Pandas, Scipy, Scikit-Learn, SQL) Getting started with Accelerated Data Science with RAPIDS AI and Spark. (cuPy, cuDF, cuSignal, cuML, cuStrings).
- Data Collection via API: GPU-accelerated SQL with BlazingSQL, Data Annotation in Active Learning.
- Data Wrangling with OpenRefine, Outlier Detection with IQR, Feature Reduction with PCA.
- Visual Exploratory Data Analytics using cuXFilter, Fixing common Bar charts, Line charts and Tables.
- Working with GPU accelerated frameworks: Apache Spark 3.0.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. To analyse the need and usage of analytics and visualization techniques.
2. To implement how to manage, manipulate, cleanse and analyse data.
3. To implement various data clustering and classification approaches in R/Python.
4. To demonstrate the use of R/Python on real life problems.

Text Books:

1. Jiawei Han Data Mining: Concepts and Techniques Morgan Kaufmann Publishers 2011.
2. Vipin Kumar Introduction to Data Mining Pearson India 2016.
3. Roger D. Peng, R Programming for Data Science.

Reference Books:

1. Foster Provost and Tom Fawcett Data Science for Business O'Reilly Media, Inc 2013.
2. Gordon S. Linoff, Michael J. A. Berry Data Mining Techniques Wiley 3 rd 2012.
3. Charu C. Aggarwal Data Mining Springer International Publishing 2015.
4. Trevor Hastie Robert Tibshirani Jerome Friedman, the Elements of Statistical Learning, Springer.

UCS663: Conversational AI: Data Science

L	T	P	Cr
2	0	2	3.0

Course Objectives: Detailed study of various data science algorithms/techniques and introduction to text analytics, basic neural network models and advanced deep learning models

Introduction: Introduction to Natural Language Processing, Text Analytics, Supervised and Unsupervised ML for NLP, Getting started with Deep Learning and Various NLP approaches: RNNs/CNNs/Attention etc.

Introduction to Machine Learning- Supervised: Introduction to Supervised Learning, Linear Regression, RAPIDS acceleration: Linear Regression, Linear Classification, Overfitting and Cross Validation, Introduction to Tree-based Methods, Decision Tree, Visualizing Classification: ROC, AUC, Confusion Matrix, Bagging, Random Forests, RAPIDS Acceleration: Random Forest, Boosting, XGBoost, RAPIDS acceleration: K-NN, XGBoost.

Introduction to Machine Learning- Unsupervised: Revamping Supervised Learning, Introduction to Unsupervised Learning, Kmeans, Affinity Propagation, Hierarchical Clustering, RAPIDS acceleration: K-Means, DBSCAN, Singular Value Decomposition (SVD), Principal Component Analysis (PCA), t-SNE, UMAP, Visualizing Clusters, RAPIDS acceleration.

Natural Language Processing: Basics: Preprocessing, Representation, Word Importance, Introduction to NLP, Stages, Approaches, Sequence Labelling and Noisy Channels, Probability Parsing and Speech Tagging, Morphology Analysis.

Fundamentals of Text Analytics: Language Models: n-grams, smoothing, class-based, brown clustering, Sequence Labeling: HMM, MaxEnt, CRFs, Latent Semantic Indexing, SVD: Dimensionality Reduction, Text Visualization, Wordnet, and Word Sense Disambiguation: Supervised and Unsupervised, Parsing: CFG, Lexicalized CFG, PCFGs, Dependency parsing.

Introduction to Neural Networks: Introduction to Artificial Neural Network, Artificial Neurons, Layers, Perceptron, Multilayer Perceptron, Advanced Deep Neural Networks, Going from DS to DL and back with RAPIDS.

Laboratory Work:

- Projects on End-to-End Machine Learning pipelines with RAPIDS.
- Latent semantic indexing for text via singular value decomposition (cuML).
- Machine Learning project: Implementing Word2vec, Glove, FastText.
- Hands-on practicals on forward and backward pass using python numpy and cuPy.
- Basic Deep learning project on Natural Language Processing using ANNs - cuPy.
- Introduction to DL Frameworks: PyTorch, Tensorflow, Pytorch Lightning.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Learn the basics of data science and its applications in different domains such as finance, image processing etc.
2. Learn and analyze the basic and advanced methods/algorithms in the field of data science, including supervised, unsupervised and semi-supervised techniques.
3. Learn and understand the basics of text analytics/natural language processing including preprocessing, and modeling techniques.
4. Acquire knowledge about the architecture of fundamental neural network models and advanced deep learning models.

Text Books:

1. Tan PN, Steinbach M, Kumar V. Introduction to data mining. Pearson Education India; 2016.
2. Han, J., Kamber, M. and Pei, J., 2011. Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems, 5(4), pp.83-124.
3. Blum, A., Hopcroft, J. and Kannan, R., 2016. Foundations of data science. Vorabversioneines Lehrbuchs, 5.

Reference Books:

1. Sarkar, D., 2019. Text analytics with Python: a practitioner's guide to natural language processing. A press.

UCS664: Conversational AI: Natural Language Processing

L	T	P	Cr
2	0	2	3.0

Course Objectives: Detailed study of various machine learning and deep learning models such as word embeddings, BERT, ELMo, RNNs, LSTMs, encoders and their applications on various NLP tasks.

Introduction: Advanced NLP and its Applications, Fundamentals of NVIDIA Toolkits, SDKs and platforms, Introduction to Speech Signal Processing.

Text Analytics with Deep Learning: Applications: Named Entity Recognition, Coreference Resolution, text classification, toolkits e.g. Spacy, etc. Distributional Semantics: distributional hypothesis, vector space models, etc. Improving Deep Neural Networks, Hyperparameter tuning, Activation Functions, Metrics, Optimization, Regularization, Back propagation, CUDA basics.

Introduction to NLP using Deep Learning: Bias and Fairness Measures, Knowledge Graph, Data deduplication, What is info viz, Exploratory Data Analytics. Word Vectors: Feed forward NN, Word2Vec, GloVE, Contextualization (ELMo etc.), Sub- word information (FastText, etc.), Deep Models: RNNs, LSTMs, Attention, CNNs, applications in language, etc.

Advanced NLP using Deep Learning: Introduction to NeMo and Jarvis, Sequence models RNNs/LSTMs, Attention, BERT, Megatron, GPT etc. Accelerated Text Processing: Tokenization, Vectorization etc. Working with open source datasets: GLUE Benchmarks.

Applications of NLP: Exploring NLP Problem Statements:- Information Retrieval, Intent Slot Filling, Machine Translation, Punctuation & Capitalization, Q&A Machine, Relation Extraction, Sentiment Analysis, Token Classification.

Introduction to NVIDIA Toolkits and SDKs: Transfer Learning Toolkit, DALI for pre-processing acceleration, TensorRT Optimization for model inference, Triton Inference Server Integration, Various Visualization Tools, Kubernetes Deployment, Kube-Flow integration with Triton.

Laboratory Work:

1. Deep learning project on Natural Language Processing using RNNs/LSTMs - PyTorch/Tensorflow.
2. Practical Hands-on with AMP and Horovord on Pytorch/Tensorflow Deep learning based NLP project.
3. Neural Modules (NeMo) for Training Conv AI Models, Exploring NeMo Fundamentals, Exploring NeMo Model Construction, Nemo Swap App Demo.
4. NeMo Information Retrieval.
5. NeMo Intent Slot Filling for ChatBot using Joint Bert Model.
6. NeMo Machine Translation.
7. NeMo Punctuation & Capitalization and Q& A Machine.
8. NeMo Relation Extraction, Sentiment Analysis, Token Classification.
9. Hands-on practical on TLT, TensorRT Optimization, Triton Inference Server.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Analyze methods and theories in the field of deep learning for text analytics.
2. Comprehend and apply different word embedding models.
3. Comprehend and apply advanced deep learning and language models such as RNNs/LSTMs, Attention, BERT, ELMo, etc.
4. Acquire knowledge of various open-source tools, datasets, benchmark metrics for various NLP applications.

Text Books:

1. Lane H., Howard C. and Hapke H.M, Natural Language Processing in Action, Manning Publications (2019).
2. Raaijmakers S., Deep Learning for Natural Language Processing, Manning Publications (2021).

Reference Books:

1. Manning C. and Schuetze H., Foundations of Statistical Natural Language Processing, Cambridge (1999)
2. Goyal P., Pandey S. and Jain K., Deep Learning for Natural Language Processing, Apress (2018).
3. Beysolow T., Applied Natural Language Processing with Python, Apress (2018)

UCS746: Conversational AI: Speech Processing

L	T	P	Cr
2	0	2	3.0

Course Objectives: Detailed study of various statistical methods and deep learning models for automatic speech processing and their applications to test-to speech and speech-to-text systems

Introduction: Advanced NLP and its Applications, Fundamentals of NVIDIA Toolkits, SDKs and platforms, Introduction to Speech Signal Processing.

Fundamentals of Speech Processing: Introduction to Statistical Speech Processing, HMMs for Acoustic Modeling, WFTS for Automatic Speech Recognition (ASR), Basics of Speech Production, Tied State HMMs, Introduction to NNs in Acoustic Modeling (Hybrid/TDNN/Tandem).

Introduction to Deep Learning in ASR: Introduction to RNN based models, Language modelling, Acoustic feature analysis for ASR, End-to-end neural architectures for ASR, Search and Decoding, Multilingual and low-resource ASR.

Advanced ASR: Automatic Speech Processing: End-to-End ASR training with NeMo (Metrics-CTC, WER, Models-Jasper/ QuartzNet), Offline ASR, CTC Segmentation.

Use-cases of Speech-to-Text: Speech Commands: Speech Commands Recognition using MatchboxNet. Overview of Noise Augmentation, Voice Activity Recognition and Speaker Recognition

Speech Synthesis: Text Normalization: Preparing Dataset and Text Normalization for input to Speech Synthesis model. Introduction to Text-to-Speech (TTS) Models:- Mel Spectrogram Generator: - Tacotron-2, Glow-TTS, Audio Generators:- WaveGlow, SqueezeWave.

Module 6 Jarvis: Jarvis Integration, Overview of Jarvis ASR, NLU and TTS APIs, Introduction to Jarvis Dialog Manager. Jarvis Deployment: - Nemo model deployment using TLT.

Laboratory Work:

1. Practical Exercise on Statistical Speech Processing.
2. NeMo Automatic Speech recognition using Jasper/QuartzNet on English Dataset.
3. NeMo Automatic Speech recognition using Jasper/QuartzNet on Indic Language Dataset.
4. NeMo Speech Commands Recognition using MatchboxNet, Noise Augmentation, and Speaker Recognition, Q& A Machine.
5. NeMo Text to Speech using Tacotron-2 and WaveGlow.
6. End-to-End Conversational AI Model (Any Language): - ASR/NLP/TTS using NeMo and Jarvis integration.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand basic approaches for statistical speech processing.
2. Comprehend and apply deep learning architectures for Automatic Speech Recognition.
3. Understand and apply different models for Text-to-Speech and Speech-to-text Synthesis.
4. Acquire knowledge of various open-source tools, datasets, benchmark metrics for Automatic Speech Processing applications.

Text Books:

1. Jurafsky D. and Martin J.H., Speech and Language Processing, Pearson (2018), 2^{nd} ed.
2. Rabiner L., Juang B. and Yegnanarayana B., Fundamentals of Speech Recognition, Pearson (2018), 1^{st} ed.

Reference Books:

1. Kamath U., Liu J. and Whitaker J., Deep Learning for NLP and Speech Rcognition, Springer (2019) 1^{st} ed.
2. Jelinek F., Statistical methods for Speech Recognition, Bradford (1998), 4^{th} ed.

UCS544: Edge AI and Robotics: Data Science

L	T	P	Cr
2	0	2	3.0

Course Objectives: To be familiar with the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, machine learning, knowledge acquisition and learning methods in solving particular engineering problems. This course provides a broad introduction to machine learning and statistical pattern recognition.

Introduction: Overview of Robotics and Kinematics, Edge Computing with Jetson Devices, GPU Computing and Dockers, Introduction to Machine Learning with RAPIDS.

Basics of Robotics: Introduction to Robotics, Homogenous Motion, Kinematic Chains, Forward Kinematics, Inverse Kinematics: Analytical Method.

Advanced Robotics: Differential kinematics: Jacobian computation, singular configurations, Configuration space operation, Mobile Robots, Motion Planning.

Edge AI: AI at the Edge & IoT, Jetson Architecture, Introduction to Deep Learning Framework like Tensor Flow/PyTorch, Getting Started with Jetpack, Dockers and Containers, Introduction to NVIDIA GPU Container Store (NGC), Getting started with Docker Platform on Jetson.

Fundamentals of Machine Learning: Introduction to Supervised Learning, Linear Regression, RAPIDS acceleration: Linear Regression, Linear Classification, Overfitting and Cross Validation, Introduction to Tree-based Methods, Decision Tree, Visualizing Classification: ROC, AUC, Confusion Matrix, Bagging, Random Forests, RAPIDS Acceleration: Random Forest, Boosting, XGBoost, RAPIDS acceleration: K-NN, XGBoost.

Laboratory Work:

Introduction to Dockers and Containers, Introduction to NVIDIA GPU Container Store (NGC), Practical on Traditional Data Science packages (Numpy, Pandas, Scipy, Scikit-Learn, SQL), Getting started with Accelerated Data Science with RAPIDS AI and Spark. (cuPy, cuDF, cuSignal, cuML, cuStrings), Getting Started with AI on Jetson Nano.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Comprehend the basics of Artificial Intelligence and representing various problem domains using knowledge representation schemes.
2. Apply various artificial intelligence techniques for obtaining solutions to real-life problems.
3. Design of programs in AI language(s)/Machine learning programs.
4. Acquire knowledge about the architecture of robotics system.
5. Analyze methods and theories in the field of machine learning and provide an introduction to the basic principles, techniques, and applications of machine learning, classification tasks, decision tree learning.
6.

Text Books:

1. Rich E., Knight K. and Nair B. S., Artificial Intelligence, Tata McGraw Hills (2009) 3rd ed.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009).
3. Mitchell T.M., Machine Learning, McGraw Hill (1997).
4. Alpaydin E., Introduction to Machine Learning, MIT Press (2010).

Reference Books:

1. Patterson D.W, Introduction to AI and Expert Systems, Mc GrawHill (1998).
2. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006).
3. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009).

UCS665: Edge AI and Robotics: Basic Computer Vision

L	T	P	Cr
2	0	2	3.0

Course Objectives: To understand the basic concepts of Computer Vision and formulating Computer Vision Problem Statements. The student must be able to apply the various concepts of Computer Vision in other application areas.

Introduction: Basics of Machine Learning, GPU Computing, Formulating Computer Vision Problem Statements, Image Classification using CNN Architectures like VGG, Inception, ResNet(18/34/50/152), ResNext and EfficientNet. Utilizing Jetpack SDK and other NVIDIA Toolkits to deploy CNN models on Jetson.

Introduction to Machine Learning- Unsupervised: Revamping Supervised Learning, Introduction to Unsupervised Learning, Kmeans, Affinity Propagation, Hierarchical Clustering, RAPIDS acceleration: K-Means, DBSCAN, Singular Value Decomposition (SVD), Principal Component Analysis (PCA), t-SNE, UMAP, Visualizing Clusters, RAPIDS acceleration.

Introduction to Deep Learning and GPU Computing: Introduction to GPU Computing and CUDA Basics, Introduction to Neural Networks (Data-Preprocessing, Neural Network Architecture like MLP, CNN, Optimization like SGD, Adam).

Transfer Learning in Computer Vision: Introduction to Transfer Learning using Transfer Learning Toolkit. Image Classification using Deep CNN Architecture like VGG, ResNet18/34/50 using Transfer Learning Toolkit with Model Pruning.

Introduction to NVIDIA Toolkits and SDKs: Transfer Learning Toolkit, DALI for pre-processing acceleration, TensorRT Optimization for model inference, Triton Inference Server Integration, Various Visualization Tools, Deploying CNN Models on Jetson Devices.

Laboratory Work:
Hello AI World (Classification) on public datasets (MNIST and CIFAR) datasets using CNNs and Deep Learning frameworks, student application (custom model, analytics, ect), Utilizing TLT, TensorRTfor training and inference of Classification Models, Working with Triton Deployment and Visualization Tools.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the fundamental problems of computer vision.
2. Apply basic concepts of computer vision in other application areas such as robotics, healthcare, and graphics.
3. Analyze and evaluate critically the building and integration of computer vision algorithms and systems.
4. Demonstrate awareness of the current key research issues in computer vision.

Text Books:

1. Szeliski, R., Computer Vision: Algorithms and Applications, Springer-Verlag London Limited (2011), 1st Edition.
2. Forsyth, A., D. and Ponce, J., Computer Vision: A Modern Approach, Pearson Education (2012) 2nd Edition.

Reference Books:

1. Hartley, R. and Zisserman, A., Multiple View Geometry in Computer Vision Cambridge University Press (2003) 2nd Edition.
2. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, Morgan Kaufmann (1990) 2nd Edition.
3. Gonzalez, C., R. and Woods, E., R. Digital Image Processing, Addison- Wesley (2018) 4th Edition.

UCS666: Edge AI and Robotics: Advanced Computer Vision

L	T	P	Cr
2	0	2	3.0

Course Objectives: To understand the basic concepts of Computer Vision and formulating Computer Vision Problem Statements. The student must be able to apply the various concepts of Computer Vision in other application areas.

Introduction: Basics of Machine Learning, GPU Computing, Formulating Computer Vision Problem Statements, Image Classification using CNN Architectures like VGG, Inception, ResNet(18/34/50/152), ResNext and EfficientNet. Utilizing Jetpack SDK and other NVIDIA Toolkits to deploy CNN models on Jetson.

Introduction to Machine Learning- Unsupervised: Revamping Supervised Learning, Introduction to Unsupervised Learning, Kmeans, Affinity Propagation, Hierarchical Clustering, RAPIDS acceleration: K-Means, DBSCAN, Singular Value Decomposition (SVD), Principal Component Analysis (PCA), t-SNE, UMAP, Visualizing Clusters, RAPIDS acceleration.

Introduction to Deep Learning and GPU Computing: Introduction to GPU Computing and CUDA Basics, Introduction to Neural Networks (Data-Preprocessing, Neural Network Architecture like MLP, CNN, Optimization like SGD, Adam).

Transfer Learning in Computer Vision: Introduction to Transfer Learning using Transfer Learning Toolkit. Image Classification using Deep CNN Architecture like VGG, ResNet18/34/50 using Transfer Learning Toolkit with Model Pruning.

Introduction to NVIDIA Toolkits and SDKs: Transfer Learning Toolkit, DALI for pre-processing acceleration, Tensor RT Optimization for model inference, Triton Inference Server Integration, Various Visualization Tools, Deploying CNN Models on Jetson Devices.

Laboratory Work:
Working with Jetbot Kits, Handling Camera Sensors for Data Collection, Training and Deployment of Object Detection (SSD, YOLOv3), Segmentation (Unet), Collision Avoidance, and Path following (DriveNet). Robotics Simulations using Isaac SDK and Gazebo.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the fundamental problems of computer vision.
2. Apply basic concepts of computer vision in other application areas such as robotics, healthcare, and graphics.
3. Analyze and evaluate critically the building and integration of computer vision algorithms and systems.
4. Demonstrate awareness of the current key research issues in computer vision.

Text Books:

1. Szeliski, R., Computer Vision: Algorithms and Applications, Springer-Verlag London Limited (2011), 1st Edition.
2. Forsyth, A., D. and Ponce, J., Computer Vision: A Modern Approach, Pearson Education (2012) 2nd Edition.

Reference Books:

1. Hartley, R. and Zisserman, A., Multiple View Geometry in Computer Vision Cambridge University Press (2003) 2nd Edition.
2. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, Morgan Kaufmann (1990) 2nd Edition.
3. Gonzalez, C., R. and Woods, E., R. Digital Image Processing, Addison- Wesley (2018) 4th Edition.

UCS747: Edge AI and Robotics: RL & Conversational AI

L	T	P	Cr
2	0	2	3.0

Introduction: GPU Computing, Formulating Advanced Vision problem statements specific towards Robotics such as pose estimation, depth calculation, Kinematics and Task Learning, Fundamentals of Reinforcement Learning and Conversational Ai for Robotics and Edge AI.

Advanced Vision & SLAM: Pose Recognition: Human Pose Estimation and its TensorRT optimization, Depth Estimation: Mono/stereo depth + point cloud extraction, Visual Odometry: Camera pose estimation from DNNs.

Manipulation: Introduction to Inverse Kinematics and Control, Gripping & Task Learning. Reinforcement Learning: Introduction to RL: RL agents, Dynamic Programming, Monte Carlo's and Temporal-Difference Methods, OpenAI Gym, RL in Continuous Spaces.

Deep Reinforcement Learning: Policy Based Methods, Deep RL for Vision: DQN, Deep RL for Robotics, actor/critic, PPO, DDPG, Introduction to Multi-RL Agent.

Conversational AI: Automated Speech Recognition: Introduction to ASR, Architectures: Jasper/QuartzNet, Natural Language Processing: Introduction to NLP, BERT, NMT (seq2seq), Text to Speech: TTS-Tacotron2/WaveGlow and Jarvis Deployment.

Practical Labs:
Implementing SLAM on Jetbot, Training and Deploying TensorRT pose estimation models on Jetson, Manipulation Lab: Building Pick-n-place, Object Assembly models, Game Agent: Open AI Gym (Jetbot in simulation), Conversational AI: Verbal JetBot commands/feedback (optional mic/speaker).