COURSES SCHEME & SYLLABUS

B.E. (COMPUTER ENGINEERING)

B.E. (Computer Engineering) 2017
Course Scheme

First Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UCB008	APPLIED CHEMISTRY	3	1	2	4.5
2.	UTA017	COMPUTER PROGRAMMING	3	0	2	4.0
3.	UEE001	ELECTRICAL ENGINEERING	3	1	2	4.5
4.	UEN002	ENERGY AND ENVIRONMENT	3	0	0	3.0
5.	UMA003	MATHEMATICS-I	3	1	0	3.5
6.	UES009	MECHANICS	2	1	2 *	2.5
			17	**4**	**6**	**22.0**

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

Second Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1	UPH004	APPLIED PHYSICS	3	1	2	4.5
2.	UTA018	OBJECT ORIENTED PROGRAMMING	3	0	2	4.0
3.	UEC001	ELECTRONIC ENGINEERING	3	1	2	4.5
4.	UTA015	ENGINEERING DRAWING	2	4	0	4.0
5.	UHU003	PROFESSIONAL COMMUNICATION	2	0	2	3.0
6.	UMA004	MATHEMATICS-II	3	1	0	3.5
			16	**7**	**8**	**23.5**

Third Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UTA013	ENGINEERING DESIGN PROJECT-I (6 SELF EFFORT HOURS)	1	0	2	5.0
2.	UES012	ENGINEERING MATERIALS	3	1	2	4.5
3.	UMA007	NUMERICAL ANALYSIS	3	1	2	4.5
4.	UCS520	COMPUTER NETWORKS	3	0	2	4.0
5.	UCS405	DISCRETE MATHEMATICAL STRUCTURES	3	1	0	3.5
6.	UCS407	INVENTIONS & INNOVATIONS IN COMPUTING	2	0	0	2.0
7.	UCS303	OPERATING SYSTEMS	3	0	2	4.0
			18	**3**	**10**	**27.5**

Fourth Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UTA014	ENGINEERING DESIGN PROJECT-II (6 SELF EFFORT HOURS)	1	0	4	6.0
2.	UTA002	MANUFACTURING PROCESSES	2	0	3	3.5
3.	UMA031	OPTIMIZATION TECHNIQUES	3	1	0	3.5
4.	UCS408	PROBABILITY AND STATISTICS	3	1	2	4.5
5.		MEASUREMENT SCIENCE AND TECHNIQUES	3	1	2	4.5
6.	UCS310	DATABASE MANAGEMENT SYSTEMS	3	0	2	4.0
7.	UCS406	DATA STRUCTURES & ALGORITHMS (4 SELF EFFORT HOURS)	3	0	2	6.0

			18	**3**	**15**	**32**

Fifth Semester

S. No.	Course Number	Course Title	L	T	P	Cr
1.	UCS616	ADVANCED DATA STRUCTURES AND ALGORITHMS	3	0	2	4.0
2.	UCS521	ARTIFICIAL INTELLIGENCE	3	1	0	3.5
3.	UCS507	COMPUTER ARCHITECTURE AND ORGANIZATION	3	0	2	4.0
4.		ELECTIVE-I	3	0	2	4.0
5	UCS525	PROFESSIONAL PRACTICES#	0	1	2	1.5
6.	UCS503	SOFTWARE ENGINEERING	3	0	2	4.0
7.	UCS701	THEORY OF COMPUTATION	3	1	0	3.5
8.		GENERIC ELECTIVE	3	0	0	3.0
			21	**3**	**10**	**27.5**

#The course would consist of talks by working professionals from industry, government, academia & research organizations.

Sixth Semester

S. No.	Course Number	Course Title	L	T	P	Cr
1.	UCS794	CAPSTONE PROJECT* (STARTS) (6 SELF EFFORT HOURS)	0	0	2	-
2		ELECTIVE-II	3	0	2	4.0
3.		ELECTIVE-III	3	0	2	4.0
4.	UCS614	EMBEDDED SYSTEMS DESIGN	3	0	2	4.0
5.	UCS615	IMAGE PROCESSING	3	0	2	4.0
6.	UTA012	INNOVATION AND ENTREPRENEURSHIP (5 SELF EFFORT HOURS)	1	0	2	4.5
7.	UCS617	MICROPROCESSOR-BASED SYSTEMS DESIGN	3	0	2	4.0
			16	**0**	**14**	**24.5**

* Design / Fabrication / Implementation work under the guidance of a faculty member. Prior to registration, a detailed plan of work should be submitted by the student to the Course Coordinator for approval.

Seventh Semester

S. No.	Course Number	Course Title	L	T	P	Cr
1.	UCS794	CAPSTONE PROJECT (14 SELF EFFORT HOURS)	0	0	2	12.0
2.	UCS802	COMPILER CONSTRUCTION	3	0	2	4.0
3.		ELECTIVE-IV	3	0	2	4.0
4.	UHU005	HUMANITIES FOR ENGINEERS	2	0	2	3.0
5.	UCS781	INDEPENDENT STUDY	0	2	0	1.0
			8	**2**	**8**	**24.0**

Eight Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UCS895	PROJECT SEMESTER				20.0
		OR				
2.	UCS897	START-UP SEMESTER				
		OR				
3.	UCS896	CAPSTONE PROJECT-II (20 SELF EFFORT HOURS)	0	0	4	12.0
4.	UCS806	ETHICAL HACKING	3	0	2	4.0
5.	UCS801	SOFTWARE PROJECT MANAGEMENT	3	0	2	4.0
			6	**0**	**8**	**20**

LIST OF ELECTIVES

Based on choice of Elective Focus: High Performance Computing, Computer Animation and Gaming, Machine Learning and Data Analytics, Information and Cyber Security, Software Engineering, Mathematics and Computing

ELECTIVE-I

S.No.	**CODE**	**TITLE**	**L**	**T**	**P**	**Cr**
1	UCS608	PARALLEL AND DISTRIBUTED COMPUTING	3	0	2	4.0
2.	UCS522	COMPUTER VISION	3	0	2	4.0
3.	UML501	MACHINE LEARNING	3	0	2	4.0
4.	UCS523	COMPUTER & NETWORK SECURITY	3	0	2	4.0
5.	UCS524	ENGINEERING SOFTWARE AS A SERVICE	3	0	2	4.0
6.		MATHEMATICAL MODELING AND SIMULATION	3	0	2	4.0

Elective-II

S.No.	**CODE**	**TITLE**	**L**	**T**	**P**	**Cr**
1	UCS631	GPU COMPUTING	3	0	2	4.0
2.	UCS632	3D MODELLING AND ANIMATION	3	0	2	4.0
3.	UCS633	DATA ANALYTICS & VISUALIZATION	3	0	2	4.0
4.	UCS634	SECURE CODING	3	0	2	4.0
5.	USE401	SOFTWARE METRICS AND QUALITY MANAGEMENT	3	0	2	4.0
6.		MATRIX COMPUTATION	3	0	2	4.0

ELECTIVE-III

S.NO.	**CODE**	**TITLE**	**L**	**T**	**P**	**CR**
1	UCS641	CLOUD COMPUTING	3	0	2	4.0
2.	UCS642	AUGMENTED AND VIRTUAL REALITY	3	0	2	4.0

3.	UML602	NATURAL LANGUAGE PROCESSING	3	0	2	4.0
4.	UCS643	CYBER FORENSICS	3	0	2	4.0
5.	UCS644	SOFTWARE VERIFICATION AND VALIDATION	3	0	2	4.0
6.		FINANCIAL MATHEMATICS	3	0	2	4.0

ELECTIVE-IV

S.NO.	**CODE**	**TITLE**	**L**	**T**	**P**	**CR**
1	UCS741	SIMULATION & MODELLING	3	0	2	4.0
2.	UCG731	GAME DESIGN & DEVELOPMENT	3	0	2	4.0
3.	UCS742	DEEP LEARNING	3	0	2	4.0
4.	UCS743	ADVANCED COMPUTER NETWORKS	3	0	2	4.0
5.	UCS709	ADVANCED TOPICS IN SOFTWARE ENGINEERING	3	0	2	4.0
6.		COMPUTATIONAL NUMBER THEORY	3	0	2	4.0

GENERIC ELECTIVE

SR. NO.	**CODE**	**TITLE**	**L**	**T**	**P**	**CR**
1	UHU006	INTRODUCTORY COURSE IN FRENCH	2	2	0	3.0
2.	UCS001	INTRODUCTION TO CYBER SECURITY	3	0	0	3.0
3.	UHU007	EMPLOYABILITY DEVELOPMENT SKILLS	2	2	0	3.0
4.	UEN004	TECHNOLOGIES FOR SUSTAINABLE DEVELOPMENT	3	0	0	3.0
5.	UHU008	INTRODUCTION TO CORPORATE FINANCE	3	0	0	3.0
6.	UHU009	INTRODUCTION TO COGNITIVE SCIENCE	3	0	0	3.0
7.	UPH063	NANO SCIENCE AND NANO-MATERIALS	3	0	0	3.0
8.	UMA066	GRAPH THEORY AND APPLICATIONS	3	0	0	3.0
9.	UMA061	Advanced Numerical methods	3	0	0	3.0
10.	UBTXXX	Biology for Engineers	3	0	0	3.0

FIRST SEMESTER

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, Gibbs phase rule, One component and two component systems.

Water Treatment and Analysis: Hardness and alkalinity of water: Units and determination, External and internal method 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 polymers and inorganic polymers.

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

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):

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

1. Determine ionic mobility, conductivity of electrolytes and application of electrodes.
2. Interpret phase diagram of one and two component systems.
3. Determine water and fuel quality parameters.
4. Analysetacticity and determine the molecular weight of polymers.
5. Analyse atomic/conjugated systems/functional groups using atomic/UV-Vis/IR spectroscopic techniques.
6. Carry out chemical analyses through volumetric and instrumental techniques.

Text Books:

1. *Ramesh S. and Vairam S., Engineering Chemistry, Wiley India (2012) 1st ed.*
2. *Jain C. P. and Jain M., Engineering Chemistry, DhanpatRai Publishing Co. (2005) 15thed.*
3. *Puri R.B., Sharma R. L. and Pathania S. M., Principles of Physical Chemistry, M.S. Vishal Publishing Co. (2008).*
4. *Brown S. L. and Holme A. T., Chemistry for engineering students, Cengagae Learning (2009) 2nd ed.*
5. *Shultz M. J., Chemistry for Engineers, Cengage Learnings, (2007) 1sted.*

UTA017: COMPUTER PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Course objective: 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: Classification of Computers, Application of Computers, Basic organization of computer, Input and Output Devices, Binary Number System, Computer memory, Computer Software.

Algorithms and Programming Languages: Algorithm, Flowcharts, Pseudocode, 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.
Linear and Non-Linear Data Structures: Linked lists, stacks and queues.

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. Comprehend concepts related to computer hardware and software, draw flowcharts and write algorithm/pseudocode.
2. Write, compile and debug programs in C language, use different data types, operators and console I/O function in a computer program.
3. Design programs involving decision control statements, loop control statements, case control structures, arrays, strings, pointers, functions and implement the dynamics of memory by the use of pointers.
4. Comprehend the concepts of linear and Non-Linear data structures by implementing linked lists, stacks and queues.

UEE001 ELECTRICAL ENGINEERING

L	T	P	Cr
3	1	2	4.5

Course Objectives: To introduce concepts of DC and AC circuits, electromagnetism, single-phase transformers, DC motor and generators.
DC Circuits: Kirchhoff's voltage and current laws; power dissipation; Voltage source and current source; Mesh and Nodal analysis; Star-delta transformation; Superposition theorem; Thevenin's theorem; Norton's theorem; Maximum power transfer theorem; Millman's theorem and Reciprocity theorem; Transient response of series RL and RC circuits.
Steady state analysis of DC Circuits: The ideal capacitor, permittivity; the multi-plate capacitor, variable capacitor; capacitor charging and discharging, current-voltage relationship, time-constant, rise-time, fall-time; inductor energisation and de-energisation, inductance current-voltage relationship, time-constant; Transient response of RL, RC and RLC Circuits.
AC Circuits: Sinusoidal sources, RC, RL and RLC circuits, Concept of Phasors, Phasor representation of circuit elements, Complex notation representation, Single phase AC Series and parallel circuits, power dissipation in AC circuits, power factor correction, Resonance in series and parallel circuits, Balanced and unbalanced 3-phase circuit - voltage, current and power relations, 3-phase power measurement, Comparison of single phase and three phase supply systems.
Electromagnetism: Electromagnetic induction, Dot convention, Equivalent inductance, Analysis of Magnetic circuits, AC excitation of magnetic circuit, Iron Losses, Fringing and stacking, applications: solenoids and relays.
Single Phase Transformers: Constructional features of transformer, operating principle and applications, equivalent circuit, Phasor analysis and calculation of performance indices.

Motors and Generators: DC motor operating principle, construction, energy transfer, speed-torque relationship, conversion efficiency, applications, DC generator operating principle, reversal of energy transfer, emf and speed relationship, applications.

Laboratory Work: Network laws and theorems, Measurement of R,L,C parameters, A.C. series and parallel circuits, Measurement of power in 3 phase circuits, Reactance calculation of variable reactance choke coil, open circuit and short circuit tests on single phase transformer, Starting of rotating machines, Magnetisation curve of DC generator.

Course learning outcomes (CLOs):

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

1. Learn about applications of networks laws and theorems to solve electric circuits
2. Represent AC quantities through Phasor and compute AC system behaviour during steady state.
3. Learn about principle, construction, characteristics and application of Electro-Mechanical energy conversion devices.

Text Books:

1. *Hughes E., Smith M. I., Hiley J. and Brown K., Electrical and Electronic Technology, Pearson Prentice Hall (2008) 10^{th} ed.*
2. *Nagrath J. I. and Kothari P. D., Basic Electrical Engineering, Tata McGraw Hill (2010) 3^{rd}ed .*
3. *Naidu S. M. and Kamashaiah, S., Introduction to Electrical Engineering, Tata McGraw Hill (2007).*
4. *Chakrabarti A., Nath S. and Chanda K. C., Basic Electrical Engineering, Tata McGrawHill (2009).*
5. *Toro D. V., Electrical Engineering Fundamentals, Prentice Hall of India Private Limited (2009) 2^{nd} ed.*

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.

Environment pollution, global warming and climate change: Air pollution (local, regional and global); Water pollution problems; Land pollution and food chain contaminations; Carbon cycle, greenhouse gases and global warming; Climate change – causes and consequences; Carbon footprint; Management of greenhouse gases at the source and at the sinks

Ecology, Structure and functioning of natural ecosystems: Ecology, ecosystems and their structure, functioning and dynamics; Energy flow in ecosystems; Biogeochemical cycles and climate; Population and communities

Natural resources: Human settlements and resource consumption; Biological, mineral and energy resources; Land, water and air; Natural resources vis-à-vis human resources and technological resources; Concept of sustainability; Sustainable use of natural resources

Agricultural, industrial systems and environment: Agricultural and industrial systems vis-à-vis natural ecosystems; Agricultural systems, and environment and natural resources; Industrial systems and environment

Energy technologies and environment: Electrical energy and steam energy; Fossil fuels, hydropower and nuclear energy; Solar energy, wind energy and biofuels; Wave, ocean thermal, tidal energy and ocean currents; Geothermal energy; Future energy sources; Hydrogen fuels; Sustainable energy

Course learning outcomes (CLOs):

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

1. Correlate major local and regional environmental issues with changes in ecology and human health
2. Monitor and document the development and dynamics of ecosystems in experimental or natural microcosms
3. Define and document local resource consumption patterns and conservation strategies

4. Define opportunities available for energy conservation and for use of renewable energy resources in local and regional entities.

Text Books:

1. *Bharucha E., Textbook for Environmental Studies, Universities Press (2005).*
2. *Chapman L. J. and Reiss J. M., Ecology- Principles and Application, Cambridge University Press (LPE) (1999) 2nd ed.*
3. *Joseph B., Environmental Studies, Tata McGraw-Hill (2009) 2nd ed.*
4. *Eastop D. T. and Croft R. D., Energy Efficiency for Engineers and Technologists, Pearson Education (2006).*
5. *Miller T. G., Environmental Science- Working with Earth, Thomson Learning (2006) 11th ed.*
6. *Wright T. R. and Boorst F. D., Environmental Science-Towards a sustainable Future, Pearson Prentice Hall (2017) 13th ed.*
7. *O'Callagan W. P., Energy Management, McGraw Hill Book Co. Ltd. (1993).*

UMA003 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 and calculus of several variables which would enable them to devise solutions for given situations they may encounter in their engineering profession.

Applications of Derivatives: Mean value theorems and their geometrical interpretation, Cartesian graphing using first and second order derivatives, Asymptotes and dominant terms, Graphing of polar curves, Applied minimum and maximum problems.

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.

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: Change of order of integration, Change of variables, Applications of multiple integrals.

Course learning outcomes (CLOs):

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

1. Apply the knowledge of calculus to plot graphs of functions and solve the problem of maxima and minima.
2. Determine the convergence/divergence of infinite series, approximation of functions using power and Taylor's series expansion and error estimation.
3. Evaluate multiple integrals and their applications to engineering problems.
4. Examine functions of several variables, define and computer partial derivatives, directional derivatives and their use in finding maxima and minima.
5. Analyze some mathematical problems encountered in engineering applications.

Text Books:

1. *Thomas B. G. and Finney L. R., Calculus and Analytic Geometry, Pearson Education (2007) 9th ed.*
2. *Stewart J., Essential Calculus, Thomson Publishers (2007) 6th ed.*

3. *Apostol M. T., Calculus, Vol I: One-Variable Calculus, with an introduction to Linear Algebra, John Wiley (2003) 2^{nd} ed.*
4. *Apostol M. T., Calculus, Vol II: Multi Variable Calculus and Linear Algebra, with applications to Differential Equations and Probability, John Wiley (2003) 2^{nd} ed.*

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 behaviour can be predicted.

Equilibrium of bodies: Free-body diagrams, conditions of equilibrium, torque due to a force, statistical 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.

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

Work and energy: Work and energy, work-energy theorem, principle of conservation of energy, collisions, principles of momentum etc.

Dynamics of Rigid Bodies: Newton's Laws, D'Alembert's Principle, Energy Principles.

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):

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

1. Determine resultants in plane force systems.
2. Identify and quantify all forces associated with a static framework.
3. Solve problems in kinematic and dynamic systems.

Text Books:

1. *Shames H. I., Engineering Mechanics: Dynamics, Pearson Education India (1996) 4^{th} ed.*
2. *Beer P. F., Johnston R. E. and Clausen E. W., Vector Mechanics for Engineers-Dynamics, McGraw-Hill Higher Education (2004) 7^{th} ed.*
3. *Hibler C. R., Engineering Mechanics: Statics and Dynamics, Prentice Hall (2012) 11^{th} ed.*
4. *Timoshenko S., Young H. D, Rao V. J. and Pati S., Engineering Mechanics, Tata McGraw Hill Education Private Limited (2000).*

SECOND SEMESTER

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 - skindepth.

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

Course learning outcomes (CLOs):

After the completion of the course, the student 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 (2003) 6^{th}ed.*
2. *Griffiths J. D., Introduction to Electrodynamics, Prentice Hall of India (1999) 3^{rd}ed.*
3. *Jenkins A. F. and White E. H., Fundamentals of Optics, McGraw Hill (2001) 4^{th}ed.*
4. *Wehr R. M., Richards A. J. and Adair W. T., Physics of The Atom, Narosa Publishing House (1990) 4^{th}ed.*
5. *Verma K. N., Physics for Engineers, Prentice Hall of India (2017) 2^{nd}ed.*
6. *Pedrotti L. F., Pedrotti S. L. and Pedrotti M. L., Introduction to Optics, Pearson Education (2007) 3^{rd}ed.*

UTA018: OBJECT ORIENTED PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Object Oriented Programming with C++: Class declaration, creating objects, accessing objects members, nested member functions, memory allocation for class, objects, static data members and functions. Array of objects, dynamic memory allocation, this pointer, nested classes, friend functions, constructors and destructors, constructor overloading, copy constructors, operator overloading and type conversions.

Inheritance and Polymorphism: Single inheritance, multi-level inheritance, multiple inheritance, runtime polymorphism, virtual constructors and destructors.

File handling: Stream in C++, Files modes, File pointer and manipulators, type of files, accepting command line arguments.

Templates and Exception Handling: Use of templates, function templates, class templates, handling exceptions.

Introduction to Windows Programming in C++: Writing program for Windows, using COM in Windows Program, Windows Graphics, User Input

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. Write, compile and debug programs in C++, use different data types, operators and I/O function in a computer program.
2. Comprehend the concepts of classes, objects and apply basics of object oriented programming, polymorphism and inheritance.
3. Demonstrate use of file handling.
4. Demonstrate use of templates and exception handling.
5. Demonstrate use of windows programming concepts using C++.

UEC001 ELECTRONIC ENGINEERING

L	T	P	Cr
3	1	2	4.5

Course Objectives: To enhance comprehension capabilities of students through understanding of electronic devices, various logic gates, SOP, POS and their minimization techniques, various logic families and information on different IC's and working of combinational circuits and their applications.

Semiconductor Devices: p- n junction diode: Ideal diode, V-I characteristics of diode, Diode small signal model, Diode switching characteristics, Zener diode

Electronics Devices and Circuits: PN Diode as a rectifier, Clipper and clamper, Operation of Bipolar Junction Transistor and Transistor Biasing, CB, CE, CC (Relationship between α, β, γ) circuit configuration Input-output characteristics, Equivalent circuit of ideal and real amplifiers, Low frequency response of amplifiers, Introduction to Field Effect Transistor and its characteristics

Operational Amplifier Circuits: The ideal operational amplifier, The inverting, non-inverting amplifiers, Op-Amp Characteristics, Frequency response of op-amp, Application of op-amp

Digital Systems and Binary Numbers: Introduction to Digital signals and systems, Number systems, Positive and negative representation of numbers, Binary arithmetic, Definitions and basic theorems of boolean Algebra, Algebraic simplification, Sum of products and product of sums formulations (SOP and POS), Gate primitives, AND, OR, NOT and Universal Gate, Minimization of logic functions, Karnaugh maps.

Combinational and Sequential Logic: Code converters, multiplexors, decoders, Addition circuits and priority encoder, Master-slave and edge-triggered flip-flops, Synchronous and Asynchronous counters, Registers

Logic families: N and P channel MOS transistors, CMOS inverter, NAND and NOR gates, General CMOS Logic, TTL and CMOS logic families, and their interfacing

Laboratory Work: Familiarization of CRO and Electronic Components, Diodes characteristics Input-Output and Switching characteristics, BJT and MOSFET Characteristics, Zener diode as voltage regulator, Transistorized Series voltage regulator. Half and Full wave Rectifiers with and without filter circuit, Half and full adder circuit implementation, Decoder, DMUX and MUX, Binary/BCD up/down counters.

Course learning outcomes (CLOs):

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

1. Demonstrate the use of semiconductor diodes in various applications.
2. Discuss and explain the working of transistors and operational amplifiers, their configurations and applications.
3. Recognize and apply the number systems and Boolean Algebra.
4. Reduce Boolean Expressions and implement them with Logic Gates.
5. Analyze, design and implement combinational and sequential circuits.
6. Analyze and differentiate logic families, TTL and CMOS.

Text Books:

1. *Mano M. M. and Ciletti D. M., Digital Design, Pearson Prentice Hall (2013) 5th ed.*
2. *Milliman J., Halkias C.C. and Jit S., Electronic Devices and Circuits, Tata McGraw Hill, 2007 (2015) 4th ed.*
3. *Givone D. D., Digital Principles and Design, McGraw-Hill (2003) 1st ed.*
4. *Wakerly F. J., Digital Design: Principles and Practices, Pearson (2006) 4th ed.*

5. *Storey N., Electronics: A Systems Approach, Pearson, Prentice Hall (2009) 4th ed.*
6. *Boylestad L. R. and Nashelsky L., Electronic Devices & Circuit Theory, Pearson (2009) (2013) 11th 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 toleranced dimensions and geometric tolerance symbolism and to create and edit drawings using drafting software AutoCAD.

Engineering Drawing

Introduction

1. Orthographic Projection: First angle and third angle projectionsystem
2. IsometricProjections
3. AuxiliaryProjections
4. PerspectiveProjections
5. Introduction to MechanicalDrawing
6. Sketching engineeringobjects
7. Sections, dimensions andtolerances

AutoCAD

1. Management of screen menuscommands
2. Introduction to drawing entities
3. Co-ordinate systems: Cartesian, polar and relativecoordinates
4. Drawing limits, units of measurement andscale
5. Layering: organizing and maintaining the integrity ofdrawings
6. Design of prototype drawings astemplates.
7. Editing/modifying drawing entities: selection of objects, object snap modes, editing commands.

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 thirdview, primarily restricting to Elevation, Plan and Profileviews
3. Projects related to orthographic and isometricprojections
    - Using wax blocks or soap bars to develop three dimensional object from given orthographicprojections.
    - Using wax blocks or soap bars to develop three dimensional object, section itand color thesection.
    - 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 combinationof solids like Prism, Cone, Pyramid, Cylinder, Sphereetc.
5. To draw the detailed and assembly drawings of simple engineering objects/systems with due sectioning (where ever required) along with bill ofmaterials.e.g. Rivet joints, simple bearing, wooden joints, Two plates connected with nut and bolt etc.

Text Books:

1. *Jolhe, D.A., Engineering Drawing, Tata McGraw Hill,2008*
2. *Davies, B. L., Yarwood, A., Engineering Drawing and Computer Graphics,* Van Nostrand 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, DhanpatRai 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, proposals and papers.

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; Success strategies for Group discussions and Interviews.

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. Training for effective presentations.
4. Project based team presentations.
5. Proposals and papers-review and suggestions.

Minor Project (if any): Team projects on technical report writing and presentations.

Course learning outcomes (CLOs):

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

1. Understand and appreciate the need of communication training.
2. Use different strategies of effective communication.
3. Select the most appropriate mode of communication for a given situation.
4. Speak assertively and effectively.
5. Correspond effectively through different modes of written communication.
6. Write effective reports, proposals and papers.
7. Present himself/ herself professionally through effective resumes and interviews.

Text Books:

1. *Lesikar V. R. 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 (2015) 3rd ed.*
3. *Mukherjee S. H., Business Communication-Connecting at Work, Oxford University Press, New Delhi (2013).*
4. *Butterfield J., Soft Skills for everyone, Cengage Learning New Delhi (2011) 1st ed.*
5. *Robbins P. S. and Hunsaker L. P., Training in Interpersonal Skills, Pearson Prentice Hall of India (2012) 6th ed.*
6. *DiSanza J. and Legge J. N., Business and Professional Communication, Pearson Education India (2012) 5th ed.*

UMA004 MATHEMATICS - II

L	T	P	Cr
3	1	0	3.5

Course Objectives: The course is intended as a basic course in the linear algebra, complex analysis and differential equations. The objective is to give a perspective into theory and numerical techniques to solve differential equations and lap lace transform. The topics included in this course have their direct impact on engineering applications.

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 outcomes (CLOs):

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

1. Solve the differential equations of first and 2nd 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 Eigenvectors.

Text Books:

1. *Simmons F. G., Differential Equations: With Applications and Historical Notes, CRC Press (2016) 3rd ed.*
2. *Krishnamurthy V., Mainra P. V. and Arora L. J., An introduction to Linear Algebra, Affiliated East West Press (1976).*
3. *Kasana S. H., Complex Variables: Theory and Applications, Prentice Hall of India (2005) 2nd ed.*
4. *Kreyszig E., Advanced Engineering Mathematics, John Wiley (2011) 10th ed.*
5. *Babu R., Engineering Mathematics, Pearson Education (2009).*
6. *Apostol M. T., Calculus, Vol I: One-Variable Calculus, with an introduction to Linear Algebra, John Wiley (2003)1967 2nd ed.*
7. *Apostol M. T., Calculus, Vol II: Multi Variable Calculus and Linear Algebra, with applications to Differential Equations and Probability, John Wiley (2003)1969 2nd ed.*

THIRD SEMESTER

UTA013: ENGINEERING DESIGN PROJECT-I

L	T	P	Cr
1	0	2	5

Course Objectives: To develop design skills according to a Conceive-Design-Implement-Operate (CDIO) compliant methodology. To apply engineering sciences through learning-by-doing project work. To provide a framework to encourage creativity and innovation. To develop team work and communication skills through group-based activity. To foster self-directed learning and critical evaluation.

To provide a basis for the technical aspects of the project a small number of lectures are incorporated into the module. As the students would have received little in the way of formal engineering instruction at this early stage in the degree course, the level of the lectures is to be introductory with an emphasis on the physical aspects of the subject matter as applied to the 'Mangonel' project. The lecture series include subject areas such as Materials, Structures, Dynamics and Digital Electronics delivered by experts in the field.

This module is delivered using a combination of introductory lectures and participation by the students in 15 “activities”. The activities are executed to support the syllabus of the course and might take place in specialised laboratories or on the open ground used for firing the Mangonel. Students work in groups throughout the semester to encourage teamwork, cooperation and to avail of the different skills of its members. In the end the students work in sub-groups to do the Mangonel throwing arm redesign project. They assemble and operate a Mangonel, based on the lectures and tutorials assignments of mechanical engineering they experiment with the working, critically analyse the effect of design changes and implement the final project in a competition. Presentation of the group assembly, redesign and individual reflection of the project is assessed in the end.

Text Books:

1. *Michael McRoberts, Beginning Arduino, Technology in action publications.*
2. *Alan G. Smith, Introduction to Arduino: A piece of cake, CreateSpace Independent Publishing Platform (2011)*

Reference Book:

1. *John Boxall, Arduino Workshop - A Hands-On Introduction with 65 Projects, No Starch Press (2013)*

UES012 ENGINEERING MATERIALS

L	T	P	Cr
3	**1**	**2**	**4.5**

Course Objectives: The objective of the course is to provide basic understanding of engineering materials, their structure and the influence of structure on mechanical, chemical, electrical and magnetic properties.

Structure of solids: Classification of engineering materials, Structure-property relationship in engineering materials, Crystalline and non-crystalline materials, Miller Indices, Crystal planes and directions, Determination of crystal structure using X-rays, Inorganic solids, Silicate structures and their applications. Defects; Point, line and surface defects.

Mechanical properties of materials: Elastic, Inelastic and Viscoelastic behaviour, Engineering stress and engineering strain relationship, True stress - true strain relationship, review of mechanical properties, Plastic deformation by twinning and slip, Movement of dislocations, Critical shear stress, Strengthening mechanism and Creep.

Equilibrium diagram: Solids solutions and alloys, Gibbs phase rule, Unary and binary eutecticphase diagram, Examples and applications of phase diagrams like Iron - Iron carbide phase diagram.

Electrical and magnetic materials: Conducting and resistor materials, and their engineering application; Semiconducting materials, their properties and applications; Magnetic materials, Soft and hard magnetic materials and applications; Superconductors; Dielectric materials, their properties and applications. Smart materials: Sensors and actuators, piezoelectric, magnetostrictive and electrostrictive materials.

Corrosion process: Corrosion, Cause of corrosion, Types of corrosion, Protection against corrosion.

Materials selection: Overview of properties of engineering materials, Selection of materials for different engineering applications.

Laboratory Work and Micro-Project:

Note: The micro-project will be assigned to the group(s) of students at the beginning of the semester. Based on the topic of the project the student will perform any of the six experiments from the following list:

1. To determine Curie temperature of a ferrite sample and to study temperature dependence of permeability in the vicinity of Curie temperature.
2. To study cooling curve of a binary alloy.
3. Determination of the elastic modulus and ultimate strength of a given fiber strand.
4. To determine the dielectric constant of a PCB laminate.
5. Detection of flaws using ultrasonic flaw detector (UFD).
6. To determine fiber and void fraction of a glass fiber reinforced composite specimen.
7. To investigate creep of a given wire at room temperature.
8. To estimate the Hall coefficient, carrier concentration and mobility in a semiconductor crystal.
9. To estimate the band-gap energy of a semiconductor using four probe technique.
10. To measure grain size and study the effect of grain size on hardness of the given metallic specimens.

Course learning outcomes (CLOs):

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

1. Classify engineering materials based on its structure.
2. Draw crystallographic planes and directions.
3. Distinguish between elastic and plastic behavior of materials.
4. Distinguish between isomorphous and eutectic phase diagram.
5. Classify materials based on their electrical and magnetic properties.
6. Propose a solution to prevent corrosion.

Text Books:

1. *Callister D. W and Rethwisch G. D., Materials Science and Engineering; John Wiley & Sons, Singapore (2013) 9th Edition.*
2. *Smith F. W., Principles of Materials Science and Engineering: An Introduction; Tata Mc-Graw Hill, (2008) 3rd Edition.*
3. *Raghavan V., Introduction to Materials Science and Engineering, Prentice Hall (2015) 6th Edition.*

Reference Books:

1. *Kasap S., Principles of Electronic Engineering Materials; Tata Mc-Graw Hill, 2007.*

2. *Vlack V. H. L., Elements of Material Science and Engineering, Pearson, India, (1989) 6th Edition.*
3. *Budinski G. K. and Budinski K. M., Engineering Materials – Properties and selection, Prentince Hall India (2002) 7th Edition.*

UMA007 NUMERICAL ANALYSIS

L	T	P	Cr
3	1	2	4.5

Course Objectives:

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

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

Linear Systems and Eigen-Values: Gauss elimination method using pivoting strategies, LU decomposition, Gauss-Seidel and successive-over-relaxation (SOR) iteration methods and their convergence, ill and well conditioned systems, 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 Picard, Taylor series, Euler's and Runge-Kutta methods (up to fourth-order), 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):

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

1. Understand the errors, source of error and its effect on any numerical computations and also analysis the efficiency of any numerical algorithms.
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.*
3. *Steven C. Chappra, Numerical Methods for Engineers, McGraw-Hill Higher Education; 7thedition (1 March 2014)*
4. *Mathew H. J., Numerical Methods for Mathematics, Science and Engineering, Prentice Hall, (1992) 2ndedition,*
5. *Burden L. R. and Faires D. J. Numerical Analysis, Brooks Cole (2011), 9thedition.*
6. *Atkinson K. and Han H., Elementary Numerical Analysis, John Willey & Sons (2004), 3rdEdition*

UCS520 COMPUTER NETWORKS

L	T	P	Cr
3	0	2	4.0

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: Organization of the Internet, ISP, Network criteria, Categories of networks, Network performance and Transmission Impairments. Network Devices, OSI Model, TCP/IP Protocol Suite, Layering principles, Line Encoding, Switching technique and Multiplexing.

Local Area Networks: LAN topologies: Bus topology, Ring topology, Token passing rings, FDDI, Star topologies, Asynchronous transfer mode, Ethernet, IEEE standards 802.3, 802.5. Wireless LANs: IEEE 802.11 and Bluetooth, introduction to Virtual circuit switching including frame relay, X.25, and ATM.

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, Scalability issues (hierarchical addressing).

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.

Network Applications: Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.), Distributed applications (client/server, peer-to-peer, cloud, etc.), HTTP as an application layer protocol, Electronic mail, File transfer, Remote login.

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):

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. Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies, subnetting and routing mechanism.
3. Demonstrate the operation of various routing protocols 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) 5thed.*
2. *Tanenbaum S. A. and Wetherall J. D., Computer Networks, Prentice Hall (2013) 5thed.*

Reference Books:

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

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, GraphColoring, 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 outcomes (CLOs):

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.*

UCS407 INVENTIONS AND INNOVATIONS IN COMPUTING

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

Course Objectives:

The passion for invention - profile of great inventors in computing history, their creations and impacts, Technological creativity in idea generation, Creating ideas based on needs (Application Pull), Creating ideas based on observation of phenomena (Technology Push), Understanding the role and use of Space, Time, Matter, and Energy in invention, Recognition and effective use of Resources in invention, Using analogy and feature transfer for invention, Recognition of patterns of technological evolution and their use in invention, Turning ideas into meaningful inventions.

Computing devices, The Language Before the Hardware, The Earliest Processors, Dawn of Modern Computers, Transitioning Toward Transistors, Invention of semiconductor materials; Examples of simple and complex CPUs.

Programming Paradigms and Languages, Compilers and Algorithms

Operating Systems; Internet and distributed computing; Social networks; Numerical methods for the approximate computer solution of otherwise intractable problems;

Databases; Data Analytics; Computer graphics and animation; Graphics Processor Unit;

Computer and data security; Program Verification, Testing, Reliability and Correctness.

Top Computing machines, Top Green Computing machines, their ranking system.

Internet of Things, Smart devices, Smart cities (requirement, design and implementations), Case study: Smart street lighting and smart traffic management, use of technology and open data, Interpreting Technology Hype, five key phases of a technology's life cycle.

Course learning outcomes (CLOs):

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

1. Generalize the important inventions in computing and technological evolution.
2. Demonstrate the trade-off of time, space and technology used in invention.
3. Summarizes the chronological development in computing in terms of hardware and software.
4. Relate computing to technology advancement

Text Books:

1. *Raum E., The History of the Computer, Heinemann-Raintree (2007).*
2. *Woodford C., Communication and Computers, Facts on File Inc. (2004).*

Reference Books:

1. *Ahmad I. and Ranka S., Handbook of Energy-Aware and Green Computing, CRC Press (2012) 1st ed.*
2. *Fortino G. and Trunfio P., Internet of things based on smart objects, Springer Science & Business Media (2014).*

UCS303 OPERATING SYSTEMS

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: Role and purpose of the operating system, Functionality of a typical operating system, managing atomic access to OS objects.

Operating System Principles: Structuring methods (monolithic, layered, modular, microkernel models), processes, and resources, Concepts of APIs, Device organization, interrupts: methods and implementations, Concept of user/system state and protection, transition to kernel mode.

Concurrency: Implementing synchronization primitives, Multiprocessor issues (spin locks, re-entrancy).

Scheduling and Dispatch: Dispatching and context switching, Pre-emptive and non-pre-emptive scheduling, Schedulers and policies, Processes and threads.

Memory Management: Review of physical memory and memory management hardware, Working sets and thrashing, Caching, Paging and virtual memory, Virtual file systems.

File Systems: Files: data, metadata, operations, organization, buffering, sequential, non-sequential, Directories: contents and structure, Naming, searching, access, backups, Journaling and log-structured file systems.

Deadlock: Introduction, Analysis of conditions, Prevention & avoidance, Detection & recovery.

Security and Protection: Overview of system security, Security methods and devices, Protection, access control, and authentication.

Virtual Machines: Types of virtualization (including Hardware/Software, OS, Server, Service, Network).

Device Management: Characteristics of serial and parallel devices, Buffering strategies, Direct memory access, Disk structure, Disk scheduling algorithms.

Laboratory work: To explore different operating systems like Linux, Windows etc. To implement main algorithms related to key concepts in the operating systems.

1. Detailed architecture of Linux commands and flow of command execution.
2. Detailed commands related to basics of Linux, file handling, process management.
3. Shell program having sequential, decision and loop control constructs.
4. CPU Scheduling Algorithms
5. Threaded programming in Linux (Eg. POSIX threads in LINUX)

Course learning outcomes (CLOs):

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

1. Explain basic operating system concepts such as overall architecture, interrupts, APIs, user mode and kernel mode.

2. Explication of the concepts related to concurrency including, synchronization primitives, race conditions, critical sections and multi-threading.
3. Analyze and apply CPU scheduling algorithms, deadlock detection and prevention algorithms.
4. Explicate various memory management techniques like caching, paging, segmentation, virtual memory, and thrashing.
5. Untangle operating systems concepts such as file systems, security, protection, virtualization and device-management, disk-scheduling algorithms and various file systems.

Text Books:

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

Reference Books:

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

FOURTH SEMESTER
UTA014 ENGINEERING DESIGN PROJECT-II
(Includes project with 6self effort hours)

L	T	P	Cr
1	0	4	6.0

Course Objective: Understanding of Arduino microcontroller architecture and programming, Interfacing of Arduino board with various I/O devices. Serial data transmission using Arduino board.Learning of ARM processor Instruction set and programming concepts.

Arduino Microcontroller:
Features of Ardunio Microcontroller, Architecture of Arduino, Different boards of Arduino, Arduino Interfacing and Applications, Anatomy of an Interactive Device like Sensors and Actuators, A to D converters and their comparison, Blinking an LED, LCD Display, Driving a DC and stepper motor, Temperature sensors, Serial Communications, Sending Debug Information from Arduino to Your Computer, Sending Formatted Text and Numeric Data from Arduino, Receiving Serial Data in Arduino, Sending Multiple Text Fields from Arduino in a Single Message, Receiving Multiple Text Fields in a Single Message in Arduino. Light controlling with PWM.

Introduction to ARM processor: Features of ARM processor, ARM Architecture, Instruction set, ARM Programming

Programming of Arduino: The Code designing step by step. Taking a Variety of Actions Based on a Single Variable, Comparing Character and Numeric Values, Comparing Strings, Performing Logical Comparisons, Performing Bitwise Operations, Combining Operations and Assignment, Using Embedded techniques to program Arduino microcontroller, Understanding the libraries of Arduino programming language and applying for circuit design

Laboratory work: Introduction to Arduino board.Programming examples of Arduino board. Interfacing of LED, seven segment display, ADC and DAC with Arduino board. Introduction to ARM processor kit.

Projects: Arduino and ARM based projects to be allocated by concerned faculty.

Course Learning Outcomes:
The student should be able to:

1. Understand of features of Arduino board.
2. Analyze of internal Architecture of Arduino board.
3. Apply Arduino board programming concepts.
4. Design and implement Buggy project based on different goals and challenges defined.

Text Books:

1. *Michael McRoberts, Beginning Arduino, Technology in action publications.*
2. *Alan G. Smith, Introduction to Arduino: A piece of cake, CreateSpace Independent Publishing Platform (2011)*

Reference Book:

1. *John Boxall, Arduino Workshop - A Hands-On Introduction with 65 Projects, No Starch Press; 1 edition (2013).*

UTA002 MANUFACTURING PROCESSES

L	T	P	Cr
2	0	3	3.5

Course Objectives: To introduce basic manufacturing processes used in industry. To identify, analyze, and solve problems related to basic manufacturing processes both independently and as a part of a team.

Introduction: Common engineering materials and their important mechanical and manufacturing properties, General classification of manufacturing processes.

Metal Casting: Principles of metal casting, Patterns, Their functions, Types, Materials and pattern allowances, Characteristics of molding sand, Types of cores, Chaplets and chills, their materials and functions, Moulds and their types, Requisites of a sound casting, Introduction to Die Casting.

Metal Forming and Shearing: Forging, Rolling, Drawing, Extrusion, Bending, Spinning, Stretching, Embossing and Coining, Die and Punch operation in press work, Shearing, Piercing and blanking, Notching, Lancing.

Machining Processes: Principles of metal cutting, Cutting tools, their materials and applications, Geometry of single point cutting tool, Cutting fluids and their functions, Basic machine tools and their applications, Introduction to non-traditional machining processes (EDM, USM, CHM, ECM, LBM, AJM, and WJM).

Joining Processes: Electric arc, Gas, Resistance and Thermit welding, Soldering, Brazing and Braze welding, Adhesive bonding, Mechanical fastening (Riveting, Screwing, Metal stitching, Crimping etc.).

Plastic Processing: Plastics, their types and manufacturing properties, Compression molding, Injection molding and Blow molding, Additives in Plastics.

Modern Trends In Manufacturing: Introduction to numerical control (NC) and computerized numerical control (CNC) machines.

Laboratory Work: Relevant Shop Floor Exercises Involving Practice in Pattern Making, Sand Casting, Machining, Welding, Sheet Metal Fabrication Techniques, Fitting Work, and Surface Treatment of Metals. Demonstration of Forge Welding, TIG/MIG/GAS/Spot/Flash Butt Welding, Demonstration on Shaper, Planer, and Milling Machine.

Course learning outcomes (CLOs):

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

1. Analyze various machining processes and calculate relevant quantities such as velocities, forces, powers etc.;
2. Suggest appropriate process parameters and tool materials for a range of different operations and workpiece materials;
3. Understand the basic mechanics of the chip formation process and how these are related to surface finish and process parameters;
4. Recognize cutting tool wear and identify possible causes and solutions;
5. Develop simple CNC code, and use it to produce components while working in groups.
6. Perform calculations of the more common bulk and sheet forming, casting and welding processes and given a particular component.
7. Select the most appropriate manufacturing process to achieve product quality through the efficient use of materials, energy and process.

Text Books:

1. *Degarmo E. P., Kohser, A. R. and Black, T. J., Materials and Processes in Manufacturing, Wiley (2010) 10th Edition.*
2. *Kalpakjian S. and Schmid S. R., Manufacturing Processes for Engineering Materials, Pearson Education Asia (2007) 5th Edition.*

Reference Books:

1. *Chapman J. A. W., Workshop Technology Part 1, CBS Publishers (2001) 5thEdition.*
2. *Zimmer W. E. and Groover P. M., Computer Aided Designing and Manufacturing, Pearson (2003) 1stEdition.*
3. *Pandey C. P. and Shan, S. H., Modern Machining Processes, Tata McGraw Hill (2004).*
4. *Mishra K. P., Non Conventional Machining, Narosa Publications (2006).*
5. *Campbell S. J., Principles of Manufacturing, Materials and Processes, Tata McGraw Hill Company (1995).*
6. *Lindberg A. R., Process and Materials of Manufacture, Prentice Hall of India (2015) 4thEdition.*

UMA031 OPTIMIZTION TECHNIQUES

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

Course Objectives:

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.

Transportation and Assignment Problem: Initial basic feasible solutions of balanced and unbalanced transportation/assignment problems, Optimal solutions.

Project Management: Construction of networks, Network computations, Floats (free floats and total floats), Critical path method (CPM), Crashing.

Game Theory: Two person zero-sum game, Game with mixed strategies, Graphical method and solution by linear programming.

Course learning outcomes (CLOs):

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

1. Formulate and solve linear programming problems.
2. Solve the transportation and assignment problems
3. Solve the Project Management problems using CPM
4. Solve two person zero-sum games

Text Books:

1. *Chandra S., Jayadeva, Mehra A., Numerical Optimization and Applications, Narosa Publishing House, (2013).*
2. *Taha A. H., Operations Research-An Introduction, Pearson (2013) 9th Edition.*

Reference Books:

1. *Pant C. J., Introduction to optimization: Operations Research, Jain Brothers (2004) 7th Edition.*
2. *Bazaarra S. M., Jarvis J. J. and Shirali D. H., Linear Programming and Network flows, John Wiley and Sons (2010) 4th Edition.*
3. *Taha A. H., Operations Research-An Introduction, Pearson (2016) 10thedition.*

UCS408: PROBABILITY AND STATISTICS

L	T	P	Cr
3	1	2	4.5

Algebra of Sets: sets and classes, limit of a sequence of sets, rings, sigmarings, fields, sigma-fields, monotone classes.
Probability: Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes' Theorem and independence, problems.
Random Variables: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev's inequality, problems, Function of a random variable, problems.
Special Distributions: Discrete uniform, binomial, geometric, negative binomial, hypergeometric, Poisson, continuous uniform, exponential, gamma, Weibull, Pareto, beta, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability and hazard rate, reliability of series and parallel systems, problems.
Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution, problems.
Sampling Distributions: The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions, problems.
Descriptive Statistics: Graphical representation, measures of locations and variability.
Estimation: Unbiasedness, consistency, the method of moments and the method of maximum likelihood estimation, confidence intervals for parameters in one sample and two sample problems of normal populations, confidence intervals for proportions, problems.
Testing of Hypotheses: Null and alternative hypotheses, the critical and 8 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, Chisquare goodness of fit test and its applications, problems.
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: After completion of this course, the students will be able to:

- Analyze the data using different descriptive measures and present graphically.
- Compute the probabilities of events along with an understanding of the random variables, expectation, variance and distributions.
- Understand the estimation of mean and variance and their respective one-sample and k-sample hypothesis tests.
- Understand the law of large numbers and the central limit theorem and how these concepts are used to model various random phenomena

Text Books:

1. *An Introduction to Probability and Statistics by V.K. Rohatgi & A.K. Md. E. Saleh, Wiley, (2008), 3rd ed.*
2. *Introduction to Probability and Statistics by J.S. Milton & J.C. Arnold, Boston, London, McGraw-Hill, (2006), 4th ed.*

Reference Books:

1. *Introduction to Probability Theory and Statistical Inference by H.J. Larson, John Wiley & Sons, (2005) 3^{rd} ed.*
2. *Introduction to Probability and Statistics for Engineers and Scientists by S.M. Ross, Elsevier, (2014), 4^{th} ed.*

MEASUREMENT SCIENCE AND TECHNIQUES

L	T	P	Cr
3	1	0	3.5

Introduction: Definition, Application and types of measurements, Instrument classification, Functional elements of an instrument, Input-output configuration of measuring instruments, Methods of correction for interfering and modifying inputs, Standards, Calibration, Introduction to Static characteristics and Dynamic characteristics, measurement errors and error analysis.

Sensors and Transducers: Definition, types, Basic principle and applications of Resistive, Inductive, Capacitive transducers. LVDT, Strain gauge. Signal Conditioning Circuits (Bridges, Op-Amps, Instrumentation Amplifiers, Filters).

IoT Sensors and applications: Smart sensors, MEMS based sensors, Piezoelectric, Hall-Effect, Photo-Diode, Photovoltaic, Photoresistive (LDR), Pyroelectric PIR motion detector, Ultrasonic range finder, Gas sensor, 3-axis gyro sensor module. Digital transducers, Encoders, Touchpad. Finger print scanner.

Measurement of Parameters: Measurement of Temperature, Pressure, Speed, Force, Vibrations, Voltage, Current and Power consumption related in computer/laptops. CRO as a measurement device: Basic block diagram, Deflection sensitivity, Application: Voltage, Current, Frequency and phase angle measurement.

Computer aided instrumentation: Data acquisition, PC based instrumentation, Virtual Instrumentation, SCADA, PLC their architecture, programming and case study.

Text Books

1. *Doeblin, E.O., Measurement systems, Tata McGraw Hill (2007).*
2. *Nakra, B.C. and Chaudhry, K.K., Instrumentation Measurement and Analysis, Tata McGraw−Hill (2003) 2nd ed.*

Reference Books

1. *Murthy, D.V.S., Transducers and Instrumentation, Prentice Hall of India (2008) 2nd ed.*
2. *Sawhney, A.K. and Sawhney, P., A Course in Electrical and Electronic Measurements and Instrumentation, Dhanpat Rai (2008) 18th ed.*

Course Learning outcomes: After the successful completion of course, the students will be able to:

1. Explain the concepts of measurement and its analysis
2. Elucidate the working principle of various sensors and transducers
3. Exhibit the knowledge of sensor modules for IoT applications
4. Exhibit the knowledge of computers in measurement

UCS310 DATABASE MANAGEMENT SYSTEM

L	T	P	Cr
3	0	2	4.0

Course Objectives: Emphasis is on the need of information 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, Database development process - conceptual data modeling, logical database design, physical database design, database implementation, database maintenance.

Database Analysis: Conceptual data modeling using E-R data model -entities, attributes, relationships, generalization, specialization, specifying constraints. 5 – 6 practical problems based on E-R data model.

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

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

Indexing of Data: Impact of indices on query performance, basic structure of an index, creating indexes with SQL, Types of Indexing and its data structures.

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 learn SQL and other database concepts. One project, which should include database designing & implementation.

Project: It will contain a Project which should include 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. There should be a formal project report. Evaluation components may include a poster, video presentation as well as concept of peer evaluation and reflection component.

Course learning outcomes (CLOs):

After the completion of the course, the student 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. Comprehend architecture of DBMS, conceptual data modelling, logical database design and physical database design.
3. Analyze and design database using E-R data model by identifying entities, attributes and relationships.
4. Apply and create Relational Database Design process with Normalization and De-normalization of data.
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) 6th ed.*
2. *Elmasri R. and Navathe B. S., Fundamentals of Database Systems, Pearson (2016) 7th ed.*

Reference Books:

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

UCS406 DATASTRUCTURES AND ALGORITHMS

L	T	P	Cr
3	0	2	6.0

(4 Self effort hours)

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

Linear Data Structures: Arrays, Records, Strings and string processing, References and aliasing, Linked lists, Strategies for choosing the appropriate data structure, Abstract data types and their implementation: Stacks, Queues, Priority queues, Sets, Maps.

Basic Analysis: Differences among best, expected, and worst case behaviours of an algorithm, Asymptotic analysis of upper and expected complexity bounds, Big O notation: formal definition and use, Little o, big omega and big theta notation, Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential, Time and space trade-offs in algorithms, Recurrence relations, Analysis of iterative and recursive algorithms.

Searching and Sorting: Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Heap Sort, Merge Sort, Counting Sort, Radix Sort.

Algorithmic Strategies with examples and problem solving: Brute-force algorithms with examples, Greedy algorithms with examples, Divide-and-conquer algorithms with examples, Recursive backtracking, Dynamic Programming with examples, Branch-and-bound with examples, Heuristics, Reduction: transform-and-conquer with examples.

Non-Linear Data Structures and Sorting Algorithms: Hash tables, including strategies for avoiding and resolving collisions, Binary search trees, Common operations on binary search trees such as select min, max, insert, delete, iterate over tree, Graphs and graph algorithms, Representations of graphs, Depth- and breadth-first traversals, Heaps, Graphs and graph algorithms, Shortest-path algorithms (Dijkstra and Floyd), Minimum spanning tree (Prim and Kruskal)

Problem Clauses: P, NP, NP- Hard and NP-complete, deterministic and non-deterministic polynomial time algorithm approximation and algorithm for some NP complete problems. Introduction to parallel algorithms, Genetic algorithms, intelligent algorithms.

Laboratory work: Implementation of Arrays, Recursion, Stacks, Queues, Lists, Binary trees, Sorting techniques, Searching techniques. Implementation of all the algorithmic techniques.

Project: It will contain a Project which should include designing a new data structure/algorithm/ language/tool to solve new problems & implementation. It can also involve creating visualizations for the existing data structures and algorithms. Quantum of project should reflect at least 60 hours of Work excluding any learning for the new techniques and technologies. It should be given to group of 2-4 students. Project should have continuous evaluation and should be spread over different components. There should be a formal project report. Evaluation components may include a poster, video presentation as well as concept of peer evaluation and reflection component.

Course learning outcomes (CLOs):

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

1. Implement the basic data structures and solve problems using fundamental algorithms.
2. Implement various search and sorting techniques.
3. Analyze the complexity of algorithms, to provide justification for that selection, and to implement the algorithm in a particular context.
4. Analyze, evaluate and choose appropriate data structure and algorithmic technique to solve 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, CareerMonk Publications (2017) 5th ed.*

FIFTH SEMESTER

UCS616 ADVANCED DATA STRUCTURES AND ALGORITHMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: To learn the advanced concepts of data structure and algorithms and its implementation.

Advanced Data Structures: Importance and need of good data structures and algorithms Heaps, AVL Trees (Search, Insertion, Deletion) Red-Black Trees (Search, Insertion and Deletion), Splay Trees (Search, Insertion and Deletion), B-trees, B+ Trees (Search, Insertion and Deletion), Fibonacci heaps, Data Structures for Disjoint Sets, Augmented Data Structures, Self-Adjusting Data Structures, Temporal data structures, Succinct data structures, Dictionaries and cuckoo hashing.

Algorithms Complexity and Analysis: Probabilistic Analysis with example, Amortized Analysis with example, Competitive Analysis wit example, Internal and External Sorting algorithms like external merge sort, distribution sorts.

Graphs & Algorithms: 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, Topological sort, Max flow: Ford-Fulkerson algorithm, max flow – min cut, Dynamic Graphs, Few Algorithms for Dynamic Graphs, Union-Find Algorithms.

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

Approximation algorithms: Need of approximation algorithms: Introduction to P, NP, NP-Hard and NP-Complete; Deterministic, non-Deterministic Polynomial time algorithms; Knapsack, TSP, Set Cover, Open Problems.

Randomized algorithms: Introduction, Type of Randomized Algorithms, Quick Sort, Min- Cut, 2-SAT; Game Theoretic Techniques, Random Walks.

Online Algorithms: Introduction, Online Paging Problem, Adversary Models, k-server Problem.

Genetic Algorithm: Introduction to GA, implementation in Python, problem solving using GA such as subset problem, TSP, Knapsack.

Advance Data Structure in Python: List, Tuple, Dictionary, Set, Stack.

Laboratory work: Implementation of various advanced data structures and algorithms for the problems like MAZE etc. Implementation of various advanced data structures with Graphs and GUI based results to explore the use of formal verification algorithms and verification tools.

Course learning outcomes (CLOs):

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

1. Implement the different tree structures algorithm and analyze in context of asymptotic notation.
2. Identify basic properties of graphs and apply their algorithms to solve real life problems.
3. Demonstrate the usage of algorithms under several categories like string matching, randomized algorithms and genetic algorithms.
4. Implement various advanced data structures using C/Java/Python or related languages.

Text Books:

1. *Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press (2009) 3rd ed.*
2. *Goldberg E. D., Genetic Algorithms, Pearson education 1989 (2009) 1st ed.*

Reference Books:

1. *Sedgewick R. and Wayne K., Algorithms, Addison-Wesley Professional (2011), 4th ed.*

UCS521 ARTIFICIAL INTELLIGENCE

L	T	P	Cr
3	1	0	3.5

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

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 neighbour, 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, existing expert systems like MYCIN, RI, Expert system shells.

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):

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, 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) 3rd ed.*
2. *Luger F. G., Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009) 6th ed.*

Reference Books:

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

UCS507 COMPUTER ARCHTECHTURE AND ORGANIZATION

L	T	P	Cr
3	0	2	4.0

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: Codes, Number System, Logic gates, Flip flops, Registers, Counters, Multiplexer, Demultiplexer, Decoder, Encoder etc.

Register Transfer and Micro operations: Register transfer Language, Register transfer, Bus & memory transfer, Logic micro operations, Shift micro operation.

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

ARM Processor Fundamentals: ARM core data flow model, Architecture, ARM General purpose Register set, Exceptions, Interrupts, Vector Table, ARM processors family.

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

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

Input-Output Organization: Peripheral devices, I/O interface Data transfer schemes, Program control, Interrupt, DMA transfer, I/O processor.

Memory Unit: Memory hierarchy, Processor vs. memory speed, High-speed memories, Cache memory, Associative memory, Interleave, Virtual memory, Memory management.

Introduction to Parallel Processing: Pipelining, Characteristics of multiprocessors, Interconnection structures, Interprocessor arbitration, Interprocessor communication & synchronization.

Laboratory work: Installing software development toolkit for ARM processor-based microcontrollers, Assembly language programming for ARM processors.

Course learning outcomes (CLOs):

After the completion of the course, the student 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. Comprehend the design of basic computer using instruction formats & addressing modes.
3. Identify various memory management techniques and algorithms for performing addition, subtraction and multiplication etc.
4. Acquire the knowledge about pipelining, multiprocessors, and input-output organization.

Text Books:

1. *Mano M. M., Computer System Architecture, Pearson (2007) 3rd ed.*
2. *Hayes, P.J., Computer Architecture and Organization, McGraw Hill (1998) 3rd ed.*

Reference Books:

1. *Hennessy, L. J. and Patterson A. D., Computer Architecture: A Quantitative Approach, Elsevier (2007) 4th ed.*
2. *Leigh E. W. and Ali L. D., System Architecture: software and hardware concepts, South Western Publishing Co. (1988).*

UCS525 PROFESSIONAL PRACTICES

L	T	P	Cr

0 1 2 1.5

Course Objectives: To provide the students with an insight into recent professional and technical practices being followed in industry and academia. The learning requires students to attend lectures delivered by industry experts and academicians and gain an understanding of recent developments happening in the world of computing and technology.

Course Description: The course is directed at tapping the experience and research of resource persons with the objective of expanding the horizons of students' knowledge. The course is offered to fifth semester students, enabling them to use the gained knowledge set in finding solutions to the research problems and projects undertaken as part of the programme curriculum. Students are addressed by experts from the industry and academic institutions which apprise them with the current professional practices, tools, technologies and methodologies being followed in industries.

Course learning outcomes (CLOs):

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

1. Develop and refine skills to understand the problem and identify approach to solve that problem through research and analysis.
2. Gain knowledge about the professional practices adopted in industry.
3. Achieve life-long learning through expert lectures on latest tools and technology.
4. Gain an insight into contemporary issues related to computing technology.

UCS503 SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To apply principles of software development and evolution. To specify, abstract, verify, validate, plan, develop and manage large software and learn emerging trends in software engineering.

Software Engineering and Processes: Introduction to Software Engineering, Software Evolution, Software Characteristics, Software Crisis: Problem and Causes, Software process models (Waterfall, Incremental, and Evolutionary process models and Agile), Software quality concepts, process improvement, software process capability maturity models, Personal Software process and Team Software Process, Overview of Agile Process.

Requirements Engineering: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioralmodeling, documenting Software Requirement Specification (SRS).

Software Design and construction: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, coupling and cohesion, Structured design (top-down functional decomposition), object-oriented design, event driven design, component-level design, test driven design, data-structured centered, aspect oriented design , function oriented, service oriented, Design patterns, 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, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.

Software Project Management: SP Estimation of scope (LOC, FP etc), time (PERT/CPM Networks), and cost (COCOMO models), Quality Management, Plan for software Quality Control and Assurance, Earned Value Analysis.

Advanced Topics: Formal specification, CASE Tools, Software Business Process Reengineering, Configuration Management.

Laboratory work: Implementation of Software Engineering concepts and exposure to CASE tools like Rational Software suit, Turbo Analyst, Silk Suite. Follow entire SDLC depending on project domain.

Course learning outcomes (CLOs):

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

1. Analyze software development process models, including agile models and traditional models like waterfall.
2. Demonstrate the use of software life cycle through requirements gathering, choice of process model and design model.
3. Apply and use various UML models for software analysis, design and testing.
4. Acquire knowledge about the concepts of application of formal specification, case tools and configuration management for software development.
5. Analysis of software estimation techniques for creating project baselines.

Text Books:

1. *Pressman S. R. and Maxim R. B., Software Engineering, A Practitioner's Approach, McGraw Hill International (2015) 8th Edition.*
2. *Sommerville I., Software Engineering, Addison-Wesley Publishing Company (2011) 9th Edition.*

Reference Book:

1. *Foster C. E., Software Engineering: A Methodical Approach, Apress (2014)1st ed.*
2. *Booch G., Rambaugh J., Jacobson I., The Unified Modeling Language User Guide (2005) 2nd Edition.*

UCS701 THEORY OF COMPUTATION

L	T	P	Cr
3	**1**	**2**	**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, Church-Turing Thesis, Variations of Turing Machines, combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem.

Uncomputability: Halting Problem, Turing enumerability, Turing Acceptability and Turing decidabilities, unsolvable problems about Turing machines, Rice's theorem.

Course learning outcomes (CLOs):

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 by understanding the concepts of context-free languages and push-down automata.
3. Apply the concepts of recursive and recursively enumerable languages and design efficient Turing Machines.
4. Solve analytical problems in related areas of theory in computer science

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.*

SIXTH SEMESTER
UCS794 CAPSTONE PROJECT

L	T	P	Cr
0	0	2	-

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):

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.

UCS614 EMBEDDED SYSTEMS DESIGN

L	T	P	Cr
3	0	2	4.0

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

Basics of computer architecture and the binary number system: Basics of computer architecture, Computer languages, RISC and CISC architectures, Number systems, Number format conversions, Computer arithmetic, Units of memory capacity.

Introduction to Embedded systems: Application domain of embedded systems, Desirable features and general characteristics of embedded systems, Model of an Embedded System, Microprocessor vs Micro-controller, Example of a Simple embedded system, Figures of merit for an embedded system, Classification of Scum : 4/8/16/32 Bits, History of embedded systems, Current trends.

Embedded Systems – The hardware point of view: Micro-controller Unit(MCU), A Popular 8-bit MCU, Memory for embedded systems, Low power design, Pull-up and pull-down resistors.

Sensors, ADCs and Actuators: Sensors, Analog to Digital Converters, Actuators.

Examples of Embedded Systems: Mobile Phone, Automotive Electronics, Radio frequency identification (RFID), Wireless sensor networks(WISENET), Robotics, Biomedical Applications, Brain machine interface

Real – time Operating Systems: Real-time tasks, Real-time systems, Types of Real-time tasks, Real-time operating systems, Real- time scheduling algorithms, Rate Monotonic Algorithm, The Earliest deadline first algorithm, Qualitiesof a Good RTOS.

Automated design of Digital IC's: History of integrated circuit (IC) design, Types of Digital IC's, ASIC design, ASIC design: the complete sequence.

Hardware Software Co-design and Embedded Product development lifestyle management: Hardware Software Co-design, Modeling of Systems, Embedded Product Development Lifestyle Management, Lifestyle Models.

Embedded Design-A Systems Perspective: A typical Example, Product Design, The Design Process, Testing, Bulk Manufacturing.

Internet of Things: Sensing and Actuation from Devices, Communication Technologies, Multimedia Technologies, Circuit Switched Networks, Packet Switched Networks.

Laboratory Work: To design and simulate list of combinational andsequential digital circuits using Modelsim& Xilinx –Verilog language. To design and simulate the operations of systems likeverilog using Modelsim& Toggle, Bitwise, Delay andany Control Logic Design in 8051.

Course learning outcomes (CLOs):

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

1. Identify the need and usage of Embedded System.
2. Compare and contrast a Real Time Embedded System from other systems.
3. Describe the kind of memory and processor.
4. Identify and define Bus, Wires and Ports, Basic Protocols of data transfer, Bus arbitration, ISA bus signals, and handshaking, Memory mapped I/O and simple I/O, Parallel I/O and Port Based I/O, examples of interfacing memory to the ports of 8051.
5. Discuss field programmable gate array (FPGA) and its application.
6. Outline the concept of Internet of Things.

Text Books:

1. *Das B. L., Embedded Systems: An Integrated Approach, Pearson (2012) 1st ed.*

Reference Books:

1. *Kamal R., Embedded Systems Architecture, Programming and Design, Tata Mcgraw Hill (2014) 3rd ed.*
2. *Marwedel P., Embedded System Design, Springer International Publishing (2018) 3rd Edition.*

UCS615 IMAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

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):

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.*

UTA012 INNOVATION AND ENTREPRENEURSHIP

L	T	P	Cr
1	0	2	4.5

(5 Self effort hours)

Course Objectives: This course aims to provide the students with a basic understanding in the field of entrepreneurship, entrepreneurial perspectives, concepts and frameworks useful for analysing entrepreneurial opportunities, understanding eco-system stakeholders and comprehending entrepreneurial decision making. It also intends to build competence with respect business model canvas and build understanding with respect to the domain of start-up venture finance.

Introduction to Entrepreneurship: Entrepreneurs; entrepreneurial personality and intentions - characteristics, traits and behavioural; entrepreneurial challenges.

Entrepreneurial Opportunities: Opportunities- discovery/ creation, Pattern identification and recognition for venture creation: prototype and exemplar model, reverse engineering.

Entrepreneurial Process and Decision Making: Entrepreneurial ecosystem, Ideation, development and exploitation of opportunities; Negotiation, decision making process and approaches, - Effectuation and Causation.

Crafting business models and Lean Start-ups: Introduction to business models; Creating value propositions - conventional industry logic, value innovation logic; customer focused innovation; building and analysing business models; Business model canvas , Introduction to lean start-ups, Business Pitching.

Organizing Business and Entrepreneurial Finance: Forms of business organizations; organizational structures; Evolution of organisation, sources and selection of venture finance options and its managerial implications. Policy Initiatives and focus; role of institutions in promoting entrepreneurship.

Course learning outcomes (CLOs):

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

1. Define the fundamentals of entrepreneurship.
2. Explain the role of entrepreneurial process and entrepreneurial decision making.
3. Describe various business models and design a business model canvas.
4. Evaluate various forms of enterprises and sources of raising finance for start-up ventures.
5. Articulate the latest developments and challenges in the entrepreneurship.

Text Books:

1. *Ries E., The lean Start-up, Penguin Books Limited (2011).*
2. *Blank S., and Dorf B., The Startup Owner's Manual: The Step by Step Guide for Building a Great Company, K&S Ranch (2013).*
3. *Carter S. and Evans J. D., Enterprise and small business- Principal Practice and Policy, Emerald Group Publishing Limited (2013) 2nd Edition.*

Reference Books:

1. *Byers H. T., Dorf C. R. and Nelson J. A., Technology Ventures: From Idea to Enterprise, McGraw Hill (2014) 4th Edition.*
2. *Osterwalder A. and Pigneur Y., Business Model Generation, Wiley (2010).*
3. *Bagchi S., Go Kiss the World: Life Lessons For the Young Professional, Portfolio Penguin (2008).*
4. *Bagchi S., MBA At 16: A Teenager's Guide to Business, Penguin Books, (2012).*
5. *Bansal R., Stay Hungry Stay Foolish, CIIE IIM Ahmedabad (2008).*
6. *Abrams R., Six-week Start-up, Prentice-Hall of India (2004).*

7. *Verstraete T. and Laffitte E. J., A Business Model of Entrepreneurship Edward Elgar Publishing(2013)*
8. *Johnson S., Where Good Ideas comes from, Penguin Books Limited (2011).*
9. *Michael E. G., Awakening the Entrepreneur Within, Gabor, Primento (2013).*
10. *Guillebeau C., The $100 startup, Macmillan (2012).*
11. *Kelley T., The ten faces of innovation, Currency Doubleday (2005).*

UCS617 MICROPROCESSOR-BASED SYSTEMS DESIGN

L	T	P	Cr
3	0	2	4.0

Course Objectives: To introduce the basics of microprocessors and microcontrollers technology and related applications. Study of the architectural details and programming of 16-bit 8086 microprocessor and its interfacing with various peripheral ICs; Study of architecture and programming of ARM processor.

Introduction to Microprocessors: Need for Flexible Logic and Evolution of Microprocessors, Applications, Generic Architecture of a Microprocessor, Overview of 8085 microprocessor, Architecture, Instruction Set, Interrupts and Programming Examples.

INTEL 8086 Microprocessor: Pin Functions, Architecture, Characteristics and Basic Features of Family, Segmented Memory, Interrupt Structures, INTEL 8086 System Configuration, Description of Instructions, Addressing Modes, Assembly directives. Assembly software programs with algorithms, Loops, Nested loops, Parameter Passing etc.

Interfacing with 8086: Interfacing of RAMs and ROMs along with the explanation of timing diagrams. Interfacing with peripheral ICs like 8255, 8254, 8279, 8259, 8251 etc.

ARM Processor Fundamentals: ARM core data flow model, Architecture, ARM General purpose Register set and GPIO’s, CPSR, Pipeline, Exceptions, Interrupts, Vector Table, ARM processors family, ARM instruction set and Thumb Instruction set.

ARM programming in Assembly: Writing code in assembly, Instruction Scheduling, Register Allocation, Conditional Execution, Looping Constructs, Bit Manipulation, Efficient Switches, Optimized Primitives: Double-Precision Integer Multiplication, Integer Normalization and Count Leading Zeros, Division, Square Roots, Transcendental Functions like log, exp, sin, cos, Endian Reversal and Bit Operations, Saturated and Rounded Arithmetic, Random Number Generation, Exception and Interrupt Handling.

Laboratory Work: Introduction to INTEL kit, Programming examples of 8086 and ARM based processors. Interfacing of LED seven segment display, ADC, DAC, stepper motor etc. Microprocessor based projects.

Projects: ARM based projects to be allocated by concerned faculty.

Course learning outcomes (CLOs):

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

1. Acquire knowledge about the basic concepts of 8085 Microprocessor and its programming.
2. Comprehend the internal architecture of 8086 and its programming using instruction set.
3. Interface different peripheral devices with 8086 microprocessors.
4. Know the internal architecture of ARM processor and its instruction set.
5. Write the programs using ARM processors.

Text Books:

1. *Gaonkar R., Microprocessor Architecture, Programming and Applications with the 8085, Penram International Publishing India Pvt. Ltd. (2013) 6th ed.*
2. *Hall V. D. and Rao S., Microprocessor and Its Interfacing, Tata McGraw Hill Publishing Company (2012) 3rd ed.*
3. *Furber S., ARM System on Chip Architecture, Pearson Education (2000) 2nd Edition.*

SEVENTH SEMESTER

UCS794 CAPSTONE PROJECT

L	T	P	Cr
0	0	2	-

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):

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.

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 anddevelop 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 with examples, Detection of syntactic error in LL and LR parsers, panic mode error recovery and error recovery in YACC tool.

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, Dynamic programming approach for optimal code generation tree, Introduction to retargetable code generation, Code generation for control structures.

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 Prison Bison. Build simple compilers from parsing to intermediate representation to code generation and simple optimization**.**

Course learning outcomes (CLOs):

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

1. In-depth knowledge of working of major phases of compiler.
2. Parser construction using top-down and bottom-up parsing techniques.
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 the course is to understand the interplay between, psychological, ethical and economic principles in governing human behaviour. The course is designed to help the students to understand the basic principles underlying economic behaviour, to acquaint students with the major perspectives in psychology to understand human mind and behaviour and to provide an understanding about the how ethical principles and values serve as a guide to behaviour on a personal level and within professions.

Psychological perspective

Introduction to Psychology: Historical Background, Psychology as a science. Different perspectives in Psychology.

Perception and Learning: Determinants of perception, Learning theories, Behavior Modification.

Motivational and Affective basis of Behavior: Basic Motives and their applications at work. Components of emotions, Cognition and Emotion. Emotional Intelligence.

Group Dynamics and Interpersonal relationships.

Development of self and personality.

Transactional Analysis.

Culture and Mind.

Practicals:

1. Experiments on learning and behaviour modification.
2. Application of Motivation Theories: Need based assessment.
3. Experiments on understanding Emotions and their expressions.
4. Personality Assessment.
5. Exercises on Transactional analysis.
6. Role plays, case studies, simulation tests on human behaviour.

Human values and ethical perspective

Values: Introduction to Values, Allport-Vernon Study of Values, Rokeach Value Survey, Instrumental and Terminal Values.

Value Spectrum for a Good Life: Role of Different Types of Values such as Individual, Societal, Material, Spiritual, Moral, and Psychological in living a good life.

Moral and Ethical Values: Types of Morality, Kant's Principles of Morality, Factors for taking ethical decisions, Kohlberg's Theory of Moral Development.

Analyzing Individual human values such as Creativity, Freedom, Wisdom, Love and Trust.

Professional Ethics and Professional Ethos, Codes of Conduct, Whistle-blowing, Corporate Social Responsibility.

Laboratory Work: Practical application of these concepts by means of Discussions, Role-plays and Presentations, Analysis of Case studies on ethics in business and CSR.

Economic perspective

Basics of Demand and Supply, Production and cost analysis

Market Structure: Perfect and Imperfect Markets.

Investment Decisions: capital Budgeting, Methods of Project Appraisal.

Macroeconomic Issues: Gross domestic product (GDP), Inflation and Financial Markets.

Globalisation: Meaning, General Agreement on Trade and tariffs (GATT), World Trade Organisation (WTO). Global Liberalisation, and its impact on Indian Economy.

Laboratory Work: The practicals will cover numerical on demand, supply, market structures and capital budgeting, Trading games on financial markets, Group discussions and presentations on macroeconomic issues. The practicals will also cover case study analysis on openness and globalisation and the impact of these changes on world and Indian economy.

Micro Project: Global Shifts and the impact of these changes on world and Indian economy.

Course learning outcomes (CLOs):

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

1. Improve the understanding of human behaviour with the help of interplay of professional, psychological and economic activities.
2. Able to apply the knowledge of basic principles of psychology, economics and ethics for the solution of engineering problems.
3. Explain the impact of contemporary issues in psychology, economics and ethical principles on engineering.

Text Books:

1. *Morgan C. T., King R. A., Weisz J. R. &Schopler J., Introduction to Psychology, McGraw Hill Book Co. (2001) 7th Edition..*
2. *Krugman P. and Robin W., Economics, Worth Publisher (2013) 3rdEdition.*
3. *Pindyck S. R., and Rubinfeld L. D., Microeconomic, Pearson Education (2013) 8th Edition.*
4. *Samuelson A. P. and Nordhaus D. W., Economics, McGraw Hill, (2009) 19th Edition.*
5. *Mankiw G. N., Principles of Macroeconomics, South-Western (2014) 7th Edition.*
6. *Gregory R. P. and Stuart C. R., The Global Economy and Its Economic Systems, (2013) South-Western (2014).*

Reference Books:

1. *Berne E., Games People Play – The Basic Hand Book of Transactional Analysis. New York: Ballantine Books (1992).*
2. *Ferrell C. O., Fraedrich J. and Ferrell, Business Ethics: Ethical Decision Making & Cases, Cengage Learning (2017) 11th Edition.*
3. *Schultz P. D. and Schultz E. S., Theories of Personality, Cengage Learning, (2017) 11thEdition.*
4. *Saleem S., Business Environment, Pearson (2015) 3rd Edition.*
5. *Chernilam F., International Buisness-Text and Cases, Prentice Hall (2010) 5th Edition.*
6. *Peterson H. C., Lewis C. W. and Jain K. S., Managerial Economics, Pearson (2005) 4th Edition.*

UCS781 INDEPENDENT STUDY

L	T	P	Cr
0	2	0	1.0

Course Objectives: This course has been designed in order to develop a high level of self-directed learning among the undergraduate students by exposing them to a variety of spectrum involving latest research topics.

Course Description: This learning requires students to form groups of three or four. Each group has to read research papers, study project reports, and to conduct research. This work is a blend of experiential, directed reading or independent research supervised by a faculty member. This course is offered in the seventh semester and during the period they have to find a particular problem to work on, make a poster and write a research paper along with the results analysis by actually applying the techniques on the primary or secondary datasets.

Course learning outcomes (CLOs):

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

1. Apply knowledge, skills and creative ideas to a self-selected topic.
2. Develop research know how to analyze and investigate a problem in a self-directed manner.
3. Understand the problem formulation, its hypothesis and research to draw conclusions.

EIGHTH SEMESTER
UCS895 PROJECT SEMESTER

L	T	P	Cr
			20.0

Course Objectives: The project semester is aimed at developing the undergraduate education programme in engineering to include a practical training in a professional engineering setting (a company, top educational institution, research institute, etc.) hereafter referred to as "host organization" as deemed appropriate. The participating organizations are ones that are either already visiting Thapar Institute of Engineering & Technology for placement or are forming new relationships for mutual benefit. The project semester gives the student an opportunity to translate engineering theory into practice in a professional engineering environment. The technical activity should be related to both the student's engineering studies and to the host organization's. It should involve tasks and methods that are more appropriately completed in a professional engineering environment and should, where possible, make use of human and technology resources provided by the organization. It consolidates the student's prior learning and provides a context for later research studies.

The purpose of the project semester is to further develop the understanding related to the implementation, design and theoretical aspects of the computer science and its application to the practical problems. Many of the subjects that a student had studied in the university have a direct impact on what the student will be doing in the software industry. Student will extend and deepen the knowledge of computer science & engineering while working within the span of project semester.

Assessment Details: Each student is assigned a faculty mentor from CSED, Thapar Institute of Engineering & Technology, who during the tenure of project semester, visits the workplace of the student twice. Faculty mentor provides 20 marks for goal report and mid-way report both. Industrial mentor evaluates the students on the basis of set parameters and provides marks out of 20. Five marks are for peer review, where a student is judged by his/her peers. Finally, a panel of three faculty members from CSED will evaluate each student during their presentations and viva. Here, the total marks are 55. The final grading is performed on the consolidated marks, i.e. 100.

Course learning outcomes (CLOs):

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

1. Identify, formulate and analyze existing problem in the (non-automated) work flow for performing a specific task.
2. Design and implement automated solutions for the assigned/identified real world problems.
3. Write technical reports.
4. Practice and develop skills in time management and reporting within an industrial or research laboratory setting.
5. Contribute to an ethical and professional work culture and also to learn to work in diverse teams.

UCS896 CAPSTONE PROJECT - II

L	T	P	Cr

0	**0**	**4**	**12.0**

(20 Self effort hours)

Course Objectives: : To facilitate the students learn and apply their earned skill set for the system development life cycle in Computer Engineering. As a part of a team, the students will make a project, which emphasizes hands-on experience, and integrates analytical, design, and development 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: This course is taken by the students who are doing their alternate semester here at CSED Thapar, instead of opting project semester at some software company or research institute. 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, in which the basic sciences, mathematics, and the engineering are applied to convert resources optimally to meet the stated needs. It typically includes both analysis and synthesis performed in an iterative cycle. As part of their design experience, students have an opportunity to define and determine the problem and its scope. The project demonstrates that students have adequate exposure to design, as defined, in engineering contexts. 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 2-3 students, with one team leader. Team lead is having an additional responsibility for maintaining the daily diary. Each Group will work under mentorship of a faculty supervisor as assigned by the department.

Each group must meet the assigned supervisor 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 fulfillment of the attendance as well as the successful completion of the project. The faculty supervisor of the project will continuously judge the development of the workings of the assigned groups.

Course learning outcomes (CLOs):

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

1. Develop skills necessary for time management, reporting and carrying out projects within an organization/industry.
2. Design, develop, debug, document, and deliver automated solutions for real world problems and learn to work in a team environment.
3. Develop technical report writing and verbal communication skills.
4. Experience contemporary computing systems, tools and methodologies and apply experimental and data analysis techniques to the software projects.
5. Apply interdisciplinary fundamentals to the software projects taking into account professional and ethical issues.

UCS806 ETHICAL HACKING

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: Understanding the importance of security, Concept of ethical hacking and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in 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-Aspect of remote password-guessing Role of-eavesdropping, Various methods of password cracking, Keystroke Loggers, 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 802.1I, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

Cryptography: Understand the use of Cryptography over the Internet through PKI, RSA, MD5, Secure Hash Algorithm and Secure Socket Layer.

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):

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

1. Review and summarization of scan, test, hack, and securing own system.
2. Apply in depth knowledge and practical experience in current essential security systems.
3. Analysis of perimeter defences work (no real network is harmed).
4. Evaluation of intruder mechanism and securing a system.
5. Synthesize Intrusion Detection policy, Social Engineering, DDoS attacks, buffer Overflow and Virus Creation.

Text Books:

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

Reference Books:

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

UCS801 SOFTWARE PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course Objectives: Learn and Explore SPM activities through knowledge of software projectmanagement and project planning.

Introduction to Project Management: The characteristics of software projects, Objectives of project management: time, cost and quality, Basics of Project Management, Stakeholders, Stages of Project, The Feasibility Study, Cost-benefit Analysis, Planning, Project Execution, Project and Product Life Cycles, Project Management Knowledge areas, Project Management Tools & Techniques, Project success factors, role of project manager.

Project Management and Planning: System view of project management, Understanding organizations, stakeholder's management, project phases and project's life cycles. Introduction to Agile software, Why planning is necessary, Iterative steps for planning, Project Plan documentation methods, Software Requirement Specification.

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

Project Scope Management: Scope Planning & Scope management plans, Function point calculation, Scope definitions & project scope statement, Work Breakdown Structure (WBS), WBS dictionary, scope verification, scope control.

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

Project Cost management: Basis principles of cost management, Cost estimating, type of cost estimate, cost estimate tools & techniques, COCOMO, Putnam/ SLIM model Estimating by Analogy, cost budgeting, cost control, earned value management, project portfolio management

Project Quality Management: Quality Planning, quality Assurance, Quality control, Tool &techniques for quality control, Pareto Analysis, Six Sigma, CMM, ISO Standards, Juran Methodology

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

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

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

Project procurement management: Procurement management plans, contract statement of work, 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 Management.

Laboratory work: Using Function Point calculation tools for estimation, comparing with COCOMO estimates, Implementation of various exercises using PERT, CPM methods, Preparing schedule, resource allocation etc. using MS Project or Fissure. sim or VENSIM can also be used, Preparing an RMMM Plan for a case study, Preparing Project Plan for a Software Project for Lab Project or case study. Exploring about PMBOK (Project Management Body of Knowledge) and SWEBOK(Software Engineering Body of Knowledge) from related website, Implementation of software project management concepts using related tools and technologies.

Course learning outcomes (CLOs):

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

6. Describe and apply basic concepts related to software project planning, scope and feasibility.

7. Analyze various project estimation techniques.
8. Comprehend the concept of team structure and project communication management.
9. Acquire knowledge about quality assurance, quality control, and risk management.
10. Describe various project management activities such as tracking, project procurement, configuration management, monitoring.

Text Books:

1. *Hughes B. and Cotterell M. and Mall R., Software Project Management, Tata McGraw Hill (2011) 5th Ed.*
2. *Pressman R., A practitioner's Guide to Software Engineering, Tata McGraw Hill (2014) 7th Ed.*

Reference Books:

1. *Stellman A., Greene J., Applied Software Project Management, O'Reilly Media, Inc. (2008).*
2. *Futrell T. R., Shafer F. D. and Shafer I. L., Quality Software Project Management, Prentice Hall (2002).*

ELECTIVE-I

UCS608 PARALLEL AND DISTRIBUTED COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To introduce the fundamentals of parallel and distributed programming and application development in different parallel programming environments.

Parallelism Fundamentals: Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness.

Parallel Architecture: Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA)

Parallel Decomposition and Parallel Performance: Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost-Performance trade-off;

Distributed Computing: Introduction: Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges, A Model of Distributed Computations , A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication.

Communication and Coordination: Shared Memory, Consistency, Atomicity, Message-Passing, Consensus, Conditional Actions, Critical Paths, Scalability, cache coherence in multiprocessor systems, synchronization mechanism.

CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, Developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

Parallel Algorithms design, Analysis, and Programming: Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, Naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms.

Laboratory work: To implement parallel programming using CUDA with emphasis on developing applications for processors with many computation cores, mapping computations to parallel hardware, efficient data structures, paradigms for efficient parallel algorithms.

Course learning outcomes (CLOs):

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

1. Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.
2. Apply parallel algorithms and key technologies.

3. Develop and execute basic parallel and distributed applications using basic programming models and tools.
4. Analyze the performance issues in parallel computing and trade-offs.

Text Books:

1. *Lin, C. and Snyder, L., Principles of Parallel Programming. USA, Addison-Wesley (2009) 1st Edition.*
2. *Grama, A., Gupta, A., Karypis, G. and Kumar. V., Introduction to Parallel Computing, Addison Wesley (2003) 2nd Edition.*

Reference Books:

1. *Gaster, R., B., Howes, L., Kaeli, D., Mistry, P., and Schaa. D., Heterogeneous Computing WithOpencl. Morgan Kaufmann and Elsevier (2012) 2nd Edition.*
2. *Mattson, T., Sanders, B., and Massingill. B., Patterns for Parallel Programming, Addison-Wesley and ACM (2013) 1st Edition.*
3. *Quinn, M., J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill (2004) 1st Edition.*

UCS522 COMPUTER VISION

L	T	P	Cr
3	0	2	4.0

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.

Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, 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, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

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

Shape from X: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

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):

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. 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.*

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 (Classification and Regression Trees, Support vector machines), Unsupervised learning (Clustering), Instance-based learning (K-nearest Neighbor, Locally weighted regression, Radial Basis Function), Reinforcement learning (Learning Task, Q-learning, Value function approximation, Temporal difference learning).

Decision Tree Learning: Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning.

Bayesian Learning: Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm.

Artificial Neural Network: Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, Pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm. Recurrent Networks, Dynamically modifying network structure.

Genetic Algorithms: Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.

Inductive and Analytical Learning: Learning rule sets, Comparison between inductive and analytical learning, Analytical learning with perfect domain theories: Prolog-EBG. Inductive-Analytical approaches to learning, Using prior knowledge to initialize hypothesis (KBANN Algorithm), to alter search objective (Tangent Prop and EBNN Algorithm), to augment search operators (FOCL Algorithm).

Design and Analysis of Machine Learning Experiments: Guidelines for machine learning experiments, Factors, Response, and Strategy of experimentation, Cross-Validation and Resampling methods, measuring classifier performance, Hypothesis testing, Assessing a classification algorithm's performance, Comparing two classification algorithms, Comparing multiple algorithms: Analysis of variance, Comparison over multiple datasets.

Laboratory Work: It is concerned with the design, analysis, implementation, and applications of programs that learn from experience. Learning algorithms can also be used to model aspects of human and animal learning.

Course learning outcomes (CLOs):

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, classification tasks, decision tree learning.
2. Apply decision tree learning, Bayesian learning and artificial neural network in real world problems.
3. Understand the use of genetic algorithms and genetic programming.
4. Apply inductive and analytical learning with related domain theories.
5. Compare different learning models and algorithms and utilize existing machine learning algorithms to design new algorithms.

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.*

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).*

UCS523 COMPUTER AND NETWORK SECURITY

L	T	P	Cr
3	0	2	4.0

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

Basic of Cryptography: Symmetric and asymmetric cryptography, cryptographic hash functions, authentication and key establishment, Message Authentication Codes (MACs), digital signatures, PKI.

Security Vulnerabilities: DoS attacks, Buffer Overflow, Race Conditions, Access Control Problems, Spoofing and Sniffing attacks, ARP Poisoning, Social Engineering and countermeasures.

Internet Security: TCP/IP Security, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS, Secure Shell (SSH), IPsec, Email Security, DNS Security, DNSSEC, Authentication Protocols

Web Security: Phishing attack, SQL Injection, Securing databases and database access, Cross Site Scripting Attacks, Cookies, Session Hijacking, E-commerce security

System Security: Firewalls, Types: Packet filter (stateless, stateful), Application layer proxies, Firewall Location and Configurations, Intruders, Intrusion Detection System, Anomaly and misuse detection.

Wireless Network Security: IEEE 802.11i Wireless LAN Security, Wireless Application Protocol Overview, Wireless Transport Layer Security, WAP End-to-End Security

Laboratory work: Insert malicious shell code into a program file and check its malicious or benign status, create Client Server program to send data across systems as two variants clear text data and encrypted data with different set of encryption algorithms, demonstrate Buffer Overflow and showcase EIP and other register status, perform ARP poisoning, SQL Injection and demonstrate its countermeasure methods, implement stateful firewall using IPTables, showcase different set of security protocol implementation of Wireless LAN,

Course learning outcomes (CLOs):

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

1. Comprehend and implement various cryptographic algorithms to protect the confidential data.
2. Identify network vulnerabilities and apply various security mechanisms to protect networks from security attacks.
3. Apply security tools to locate and fix security leaks in a computer network/software.
4. Secure a web server and web application.
5. Configure firewalls and Intrusion Detection System.

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.*

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.*

UCS524 ENGINEERING SOFTWARE AS A SERVICE

L	T	P	Cr
3	0	2	4.0

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

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

SaaS Framework: 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

SaaS Framework: 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):

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.*

Mathematical Modeling and Simulation

L	T	P	Cr
3	0	2	4.0

Objectives: The primary goal is to provide students with 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, and to implement and test a variety of simulation and data analysis libraries and programs. MATLAB is the software environment used for implementation and application of simulations. Case studies in industry and engineering applications are used to illustrate the techniques and modelling concepts. Examples of simulation and analysis methods will be related to the linear and non-linear, deterministic and non-deterministic systems.

Contents:

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, diffusion and air pollution models, statistical models, Poisson process, stochastic models, computer data communications, 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.

Textbooks/Reference 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, 2002.
4. J. Nathan Kutz, Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data, Oxford University Press, 2013.

Course Learning outcomes:

At the end of the course, the student will be able to:

1. formulate various mathmatical 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.
5. build tools to view and control simulations and their results.

ELECTIVE-II

UCS631 GPU COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To study architecture and capabilities of modern GPUs and learn programming techniques for the GPU such as CUDA programming model.

Introduction: Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding Up Real Applications, Parallel Programming Languages and Models.

History of GPU Computing: Evolution of Graphics Pipelines, The Era of Fixed-Function Graphics Pipelines, Evolution of Programmable Real-Time Graphics, Unified Graphics and Computing Processors, GPGPU, Scalable GPUs, Recent Developments, Future Trends.

Introduction to Data Parallelism and CUDA C: Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading.

Data-Parallel Execution Model: CUDA Thread Organization, Mapping Threads to Multidimensional Data, Matrix-Matrix Multiplication—A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Thread Scheduling and Latency Tolerance.

CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Tiled Matrix – À Matrix Multiplication Kernel, Memory as a Limiting Factor to Parallelism.

An Introduction to OpenCL: Data Parallelism Model, Device Architecture, Kernel Functions, Device Management and Kernel Launch, Electrostatic Potential Map in OpenCL.

Parallel Programming with OpenACC: OpenACC Versus CUDA C, Execution Model, Memory Model, Basic OpenACC Programs, Parallel Construct, Loop Construct, Kernels Construct, Data Management, Asynchronous Computation and Data Transfer.

Laboratory work: Practice programs using CUDA, OpenCL and OpenACC.

Course learning outcomes (CLOs):

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

1. Comprehend commonly used terms in parallel computing.
2. Understand common GPU architectures and Programming Models.
3. Implement algorithms efficiently for common application kernels.
4. Develop efficient parallel algorithms to solve given problems.

Text Books:

1. *Sanders, J. and Kandrot, E., CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional (2012) 4th Edition.*
2. *Kirk, D. and Hwu, M., W., Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann (2016) 3rd Edition.*

Reference Books:

1. *Hwu, M., W., A GPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann (2011) 1st Edition.*

UCS632 3D MODELLING AND ANIMATION

L	T	P	Cr
3	0	2	4.0

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.

Introduction: Definition of Computer-based Animation, Basic Types of Animation: Real Time,Non-real-time, Definition of Modelling, Creation of 3D objects. Exploring the Max Interface, Controlling & Configuring the Viewports, Customizing the Max Interface & Setting Preferences, Working with Files, Importing & Exporting, Selecting Objects & Setting Object Properties, Duplicating Objects, Creating & Editing Standard Primitive & extended Primitives objects, Transforming objects, Pivoting, aligning etc.

2D Splines & Shapes& compound object: Understanding 2D Splines & shape, Extrude & Bevel 2D object to 3D, Understanding Loft & terrain, Modeling simple objects with splines, Understanding morph, scatter, conform, connect compound objects, blobmesh, Boolean,Proboolean&procutter compound object.

3D Modelling: Modeling with Polygons, using the graphite, working with XRefs, Building simple scenes, Building complex scenes with XRefs, using assets tracking, deforming surfaces & using the mesh modifiers, modeling with patches & NURBS.

Keyframe Animation: Creating Keyframes, Auto Keyframes, Move & Scale Keyframe on the timeline, Animating with constraints &simple controllers, animation Modifiers & complex controllers, function curves in the track view, motion mixer etc..

Simulation & Effects: Bind to Space Warp object, Gravity, wind, displace force object, deflectors, FFD space warp, wave, ripple, bomb, Creating particle system through PArray, understanding particle flow user interface, how to particle flow works, hair& fur modifier, cloth & garment maker modifiers etc.

Lighting & Camera: Configuring & Aiming Cameras, camera motion blur, camera depth of field, camera tracking, using basic lights & lighting Techniques, working with advanced lighting, Light Tracing, Radiosity, video post, mental ray lighting etc.

Texturing with Max: Using the material editor & the material explorer, creating& applying standard materials, adding material details with maps, creating compound materials &material modifiers, unwrapping UVs & mapping texture, using atmospheric & render effects etc.

Rendering with V-Ray: V-ray light setup, V-ray rendering settings, HDRI Illumination, Fine-tuning shadows, Final render setting etc.

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):

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

1. Describe Computer-based animation using 3D modeling tool (Blender/ Max).
2. Develop the practical skills in 2D Splines, Shapes & compound objects.
3. Illustrate the theoretical and practical aspects of 3D Modeling, Keyframe Animation, Simulation and Effects.
4. Demonstrate different types of animation and its effects in the real world.
5. 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, CRCPress (2017) 1st Edition.*
2. *Chopine, A., 3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation, Focal Press (2011) 1stEdition.*
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.*

UCS633 DATA ANALYTICS AND VISUALIZATON

L	T	P	Cr
3	0	2	4.0

Course Objectives: To learn the analysis of various types of data and its visualization using visualization tools.

Data Representation: Data Objects and Attribute Types: Nominal, Binary, Ordinal, Numeric, Discrete and Continuous, Types of data: Record, Temporal, Spatial Temporal, Graph, Unstructured and Semi structured data, Basic Statistical Descriptions of Data.

Introduction to Data Analysis: Probability and Random Variables, Correlation, Regression.

Data Analysis Pipeline: Data pre-processing- Attribute values, Attribute transformation, Sampling, Dimensionality reduction: PCA, Eigen faces, Multidimensional Scaling, Non-linear Methods, Graph-based Semi-supervised Learning, Representation Learning Feature subset selection, Distance and Similarity calculation.

Data Mining Techniques for Analysis: Classification: Decision tree induction, Bayes classification, Rule-based classification, Support Vector Machines, Classification Using Frequent Patterns, k-Nearest-Neighbor, Fuzzy-set approach Classifier, Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering

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.

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

Course learning outcomes (CLOs):

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

1. Analyze and extract features of complex datasets.
2. Evaluate and visualize inter-dependencies among variables in dataset.
3. Apply techniques for classification and clustering in datasets.
4. Develop and validate models for real life datasets.

Text Books:

1. *Han, J., Kamber, M. and Pei, J., Data Mining Concepts and Techniques, Morgan Kaufmann (2011) 3rd Edition*
2. *Peng, D., R., R Programming for Data Science, Lulu.com (2012).*

Reference Books:

1. *Hastie, T., Tibshirani, R. and Friedman, J., The Elements of Statistical Learning, Springer (2009) 2nd Edition.*
2. *Simon, P., The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions, John Wiley & Sons (2014).*

UCS634 SECURE CODING

L	T	P	Cr

3 0 2 4.0

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.

Decipher journey starting from FQDN to html page getting served to browser, Authoritative reply, revisit layer 2 and layer 3 of TCP/IP, DNS poisoning, ARP poisoning, C language obfuscation. ARP poisoning and its countermeasures. Buffer Overrun- Stack overrun, Heap Overrun, Array Indexing Errors, Format String Bugs, PE Code injection.

Malware Terminology: Rootkits, Trapdoors, Botnets, Key loggers, Honeypots. Active and Passive Security Attacks. IP Spoofing, Tear drop, DoS, DDoS, XSS, SQL injection, Smurf, Man in middle, Format String attack.

Types of Security Vulnerabilities: buffer overflows, Invalidated input, race conditions, access-control problems, weaknesses in authentication, authorization, or cryptographic practices. Access Control Problems.

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. Defence in Depth and Principle of Least Privilege.

Secure Coding Techniques: Protection against DoS attacks, Application Failure Attacks, CPU Starvation Attack.

Database and Web-specific issues: SQL Injection Techniques and Remedies, Race conditions, Time of Check Versus Time of Use and its protection mechanisms. Validating Input and Interprocess Communication, Securing Signal Handlers and File Operations. XSS scripting attack and its types – Persistent and Non-persistent attack XSS Countermeasures and Bypassing the XSS Filters.

Laboratory Work: In this Lab, the student will be able to practically understand how all the security attacks does has happened, as well as 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 that consists of using network monitoring tools, implementing different types of attacks and some protection schemes. Evaluation will be mainly based on projects and assignments.

Course learning outcomes (CLOs):

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

1. Implement ARP poisoning attack and demonstrate countermeasure against these for different operating environments.
2. Implement DNS poisoning attack and demonstrate authoritative reply in this context.
3. Implement PE Code injection and demonstrate control hijacking via EIP manipulation
4. Demonstrate skills needed to deal with common programming errors and develop secure applications.
5. Demonstrate client-side attacks and identify nature of threats to software and incorporate secure coding practices throughout the planning and development of software product.
6. Demonstrate SQL, XSS attack and suggest countermeasures for the same.

Text Books:

1. *Howard, M. and LeBlanc, D., Writing Secure Code, Howard, Microsoft Press (2002) 2nd Edition.*
2. *Deckard, J., Buffer Overflow Attacks: Detect, Exploit, Syngress (2005) 1st Edition.*

Reference Books:

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

USE401 SOFTWARE METRICS AND QUALITY MANAGEMENT

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: This course aims to equip students with the knowledge and techniques of professional practices in software processes and activities. It prepares students to manage the development of high quality software using proven techniques and established standards in software quality management. It will also inculcate knowledge of different metrics associated with Software Development and evaluation.

Software Metrics: Measurement in software engineering, software metrics, Metrics data collection and analysis.

Measuring internal product attributes: Aspects of software size, length, functionality and complexity, measuring structure, types of structural measures, control-flow structure, and modularity and information flow attributes, data structures.

Measuring external product attributes: Modeling software quality, software reliability, software reliability problem, parametric reliability growth models, predictive accuracy, recalibration of software-reliability growth predictions, importance of operational environment, and wider aspects of software reliability.

Metrics for object-oriented systems and component-based system: object-oriented metrics and its characteristics various object-oriented, MOOD metrics; component-based metrics and its characteristics and various component-based suites.

Dynamic Metrics: Runtime Software Metrics, Extent of Class Usage, Dynamic Coupling, Dynamic Cohesion, and Data Structure Metrics.

Software Quality: Concepts of software quality, software quality control and software quality assurance, evolution of SQA, major SQA activities and issues, zero defect software.

Software Quality Assurance: SQA techniques; Management review process, technical review process, walkthrough, software inspection process, configuration audits, and document verification.

Error Reporting, Trend Analysis and Corrective Action: Identification, Analysis and Correction of defect, implementation of correction, regression testing; Categorization of defect w.r.t development phases; Error quantity, error frequency, program unit complexity, compilation frequency; Corrective action and documenting the corrective action, periodic review of actions taken.

Case Studies: CASE tools, Quality management standards, Quality standards with emphasis on ISO approach, Capability Maturity Models-CMM and CMMI, TQM Models, Bootstrap methodology, The SPICE project, ISO/IEC 15504, Six Sigma Concept for Software Quality.

Laboratory Work: To Work on small projects, build metrics and analyze, check the quality of the projects and do a comparative study with other projects.

Course learning outcomes (CLOs):

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

1. Acquire basic knowledge of Software quality models.
2. Exemplify Quality measurement and metrics, Quality plan and implementation
3. Articulate Quality control and reliability of quality process and Quality management system models
4. Articulate Complexity metrics and Customer Satisfaction and International quality standards.
5. Control and manage the project and processes, apply configuration management on the basis of collected metrics.

Text Books:

3. *Horch, W., J., Practical Guide to Software Quality Management (Artech House Computing Library), Artech House Publishers (2003) 2nd Edition.*
4. *Weinberg, M., G., Quality Software Management, Volume 1: Systems Thinking, Dorset House Publishing (2011).*

Reference Books:

3. *Kan, H., S., Metrics and Models in Software Quality Engineering, Addison-Wesley Professional (2002) 2nd Edition.*
4. *Jones, C., Applied Software Measurement, McGraw Hill Production (2008) 3rd Edition.*

Text Books:

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

Reference Books:

1. *Mathur P. A., Foundations of Software Testing, Pearson (2013), 2nd ed.*
2. *Beizer B., Software Testing Techniques, Van Nostrand Reinhold (1983), 1sted.*
3. *Rakitin R. S., Software Verification and Validation for Practitioners and Managers, Artech House (2001), 2nd ed.*

MATRIX COMPUTATION

L	T	P	Cr
3	0	2	4.0

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 eigen values 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, linear dependence and independence, bases and dimensions, linear transformations, orthogonal and orthonormal bases, orthogonal projections and least squares, Gram Schimdt process, diagonalizable matrices, Householder triangularizations, QR factorization.

Conditioning and stability: Error propagation in computer arithmetic, conditioning and condition number, stability of a numerical algorithms, vector and matrix norms, convergent matrices, condition number of a matrix, sensitivity analysis, closeness to singularity, stability of Householder triangularization, conditioning and stability of least square algorithms.

Iterative techniques for Eigenvalues: Eigenvalue problems, reduction to Hessenberg or tridiagonal form, iterative techniques for finding eigenvalues and vectors, power method, Rayleigh quotient, inverse power iteration, QR algorithm, Arnoldi iteration, Lanczos iteration, biorthogonal methods.

Unsymmetric Eigenvalue Problems: Properties and Decompositions, Perturbation Theory, Real Schur forms, Invariant subspace computations, Generalized eigenvalue problem, Hamiltonian and Product egenvalue problems, Pseudospectra, Large Sparse Eigenvalue Problems.

Singular value decomposition: SVD and their applications, Perturbation theorem for singular values, Outer product expansion of a matrix, Least square solutions, Rank-Deficient least squares problems. square and underdetermined systems, subspace computations with the SVD.

Real life applications of eigenvalues and singular values: Discussion of real life problems based on eigenvalues and SVDs. The problems based on eigenvalues can be taken from the field of computer science (google search engine) and from fluid mechanics. Applications of SVDs in inverse problems.

For the implementation of above taught methods, Matlab programs will be taught during the Lab hours.

Textbooks/Reference Books:

1. Gene H. Golub and Charles F. Van Loan, Matrix Computations, Johns Hopkins University Press, fourth edition, 2012.
2. L. N. Trefethen and David Bau, Computational Linear Algebra, SIAM, 1997.
3. William Ford, Numerical Linear Algebra with Applications, Academic Press, 2014.
4. Roger A. Horn and Charles R. Johnson, Matrix Analysis, second Edition, Cambridge University Press, 2012.
5. Gilbert Strang, Linear algebra and its Applications, fourth edition, CENGAGE, 2007.

Course learning outcomes (CLOs):

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

1. Explain and fluently apply fundamental linear algebra concepts such as matrix norms, eigen- and singular values and vectors and linear transformations.
2. Transform matrices into triangular, Hessenberg, tri-diagonal form.
3. Compute eigenvalues using direct and iterative techniques.
4. Compute and analyze singular values using direct and iterative techniques.
5. Implement linear algebra algorithms using Matlab.
6. Apply linear algebra concepts on real life applications

ELECTIVE-III

UCS641 CLOUD COMPUTING

L	T	P	Cr
3	0	2	4.0

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: Overview of Existing Hosting Platforms, Cluster Computing, Grid Computing, Utility Computing, Autonomic Computing, Green Computing, Cloud Computing, history and evolution, practical applications of cloud computing for various industries, IoT, economics and benefits of cloud computing, spot markets, pricing models, Supercomputing-on-demand.

Cloud Issues and Challenges: Cloud computing issues and challenges like Security, Elasticity, Resource management and Scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management, Big Data, Pre-reservation and Cloud bursting.

Data Center: Classic Data Center, Virtualized Data Center (Compute, Storage, Networking and Application), Business Continuity in VDC.

Cloud Computing Architecture: Cloud Architecture model, Types of Clouds: Public Private & Hybrid Clouds, Cloud based services: Iaas, PaaS and SaaS.

Classification of Cloud Implementations: Amazon Web Services, The Elastic Compute Cloud (EC2), The Simple Storage Service (S3), The Simple Queuing Services (SQS), Google AppEngine - PaaS, Windows Azure, Aneka, Hadoop, A Comparison of Cloud Computing Platforms.

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, VMware vCloud – IaaS

Cloud based Data Storage: Introduction No-SQL databases, Map- Reduce framework for Simplified data processing on Large clusters using Hadoop, Design of data applications based on Map Reduce in Apache Hadoop, Task Partitioning, Data partitioning, Data Synchronization, Distributed File system, Data Replication, Shared access to weakly consistent to data stores.

Laboratory work: To implement Cloud, Apache and Hadoop framework and related services. To understand various concepts practically about virtualization, data storage. To implement few algorithms with the help of MapReduce and some high-level language**.**

Course learning outcomes (CLOs):

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

1. Explain the basic concepts along with evolution and features of cloud computing.
2. Demonstrate the concept of existing cloud paradigms and platforms.
3. Classify the issues of cloud computing in various cloud models.
4. Apply the knowledge of virtualization through different virtualization technologies.
5. Apply the concept of Map reduce framework using 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. *Miller M., Cloud Computing, Que Publishing (2008) 1st ed.*
3. *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.*

UCS642 AUGMENTED AND VIRTUAL REALITY

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: To understand the basic concepts of Augmented and Virtual Reality. The student must be able to apply the various concepts of Augmented and Virtual Reality in other application areas.

Introduction of Virtual Reality: Fundamental concept and components of Virtual Reality, primary features and present development on Virtual Reality

Multiple Modals of Input and Output Interface in Virtual Reality: Input -- Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3D Menus & 3DScanner etc. Output -- Visual /Auditory / Haptic Devices

Visual Computation in Virtual Reality: Fundamentals of computer graphics, software and hardware technology on stereoscopic display, advanced techniques in CG: Management of large scale environments & real time rendering

Environment Modeling in Virtual Reality: Geometric Modeling, Behavior Simulation, Physically Based Simulation.

Interactive Techniques in Virtual Reality: Body Track, Hand Gesture, 3D Menus, Object Grasp.

Introduction of Augmented Reality (AR): System structure of Augmented Reality, key technology in AR.

Development Tools and Frameworks in Virtual Reality: Frameworks of software development tools in VR, X3D Standard, Vega, MultiGen, Virtools etc.

Application of VR in Digital Entertainment: VR technology in film & TV production, VR technology in physical exercises and games, demonstration of digital entertainment by VR.

Laboratory Work: To implement various techniques studied during course.

Course learning outcomes (CLOs):

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

1. Analyze the components of AR and VR systems, its current and upcoming trends, types, platforms, and devices.
2. Assess and compare technologies in the context of AR and VR systems design.
3. Implement various techniques and algorithms used to solve complex computing problems in AR and VR systems.
4. Develop interactive augmented reality applications for PC and Mobile based devices using a variety of input devices.
5. Demonstrate the knowledge of the research literature in augmented reality for both compositing and interactive applications.

Text Books:

1. *Doug A. B., Kruijff E., LaViola J. J. and Poupyrev I. , 3D User Interfaces: Theory and Practice , Addison-Wesley (2005,2011p) 2nd ed.*
2. *Parisi T., Learning Virtual Reality, O'Reilly (2016) 1st ed.*
3. *Schmalstieg D. and Hollerer T., AugmentedAnd Virtual Reality, Addison-Wesley (2016).*

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).*

UML602 NATURAL LANGUAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

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; Morphological Diversity of Indian Languages; Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Named Entities.

Phrase structure and constituency models: phrase structure grammar; dependency grammar; formal language theory.

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; Indian Language Word Nets and Multilingual Dictionaries; 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.

Course learning outcomes (CLOs):

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), 2nded.*
2. *Manning D. C. and Schütze H., Foundations of Statistical Natural Language Processing MIT Press (1999) 1sted.*

Reference Books:

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

UCS643 CYBER FORENSICS

L	T	P	Cr
3	**0**	**2**	**4.0**

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, Deleted File Recovery Tools, Recovering Deleted Partitions, Deleted Partition Recovery Tools, 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, Data Recovery Software and Documentation, Decrypting Encrypted Data, Documenting Evidence, Evidence Tagging and Marking, Evidence Logs, Documenting the Chain of Custody, Computer Forensic Resources, Computer Forensic Training and Certification, Computer Forensic Equipment and Software, Computer Forensic Services, 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

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):

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 GuillossouF.,Cyber Forensics: From Data to Digital Evidence, Wiley (2012).*

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.*

UCS644 SOFTWARE VERIFICATION AND VALIDATION

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course makes students understand the concepts and theory related to software testing. Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage. Understand how software developers can integrate a testing framework into code development in order to incrementally develop and test code.

Introduction: Terminology, evolving nature of area, Errors, Faults and Failures, Correctness and reliability, Testing and debugging, Static and dynamic testing, Exhaustive testing: Theoretical foundations: impracticality of testing all data, impracticality of testing all paths, no absolute proof of correctness.

Software Verification and Validation Approaches and their Applicability: Software technical reviews; Software testing: levels of testing - module, integration, system, regression; Testing techniques and their applicability-functional testing and analysis, structural testing and analysis, error-oriented testing and analysis, hybrid approaches, integration strategies, transaction flow analysis, stress analysis, failure analysis, concurrency analysis, performance analysis; Proof of correctness; simulation and prototyping; Requirement tracing.

Test Generation: Test generations from requirements, Test generation pats, Data flow analysis, Finite State Machines models for flow analysis, Regular expressions-based testing, Test Selection, Minimizations and Prioritization, Regression Testing.

Program Mutation Testing: Introduction, Mutation and mutants, Mutation operators, Equivalent mutants, Fault detection using mutants, Types of mutants, Mutation operators for C and Java.

Laboratory Work: To Use various verification and validation testing tools and to apply these tools on few examples and case studies.

Course learning outcomes (CLOs):

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

1. Comprehend the theoretical foundations of testing.
2. Comprehend software testing levels, testing techniques and their applicability.
3. Generate test cases from software requirements, data flows and finite state machines.
4. Perform fault detection using mutants for operators of C and Java language.

Text Books:

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

Reference Books:

1. *Mathur P. A., Foundations of Software Testing, Pearson (2013), 2nd ed.*
2. *Beizer B., Software Testing Techniques, Van Nostrand Reinhold (1983), 1st ed.*
3. *Rakitin R. S., Software Verification and Validation for Practitioners and Managers, Artech House (2001), 2nd ed.*

Financial Mathematics

L	T	P	Cr
3	0	2	4.0

Course Objective: 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: Portfolio optimization: introduction, risk, return, two-assets portfolio, Markowitz curve, efficient frontier, Multi-assets all risky portfolio, mean-variance Markowitz model, two fund theorem

Portfolio with one risk free asset, one fund theorem, CAPM, market line, beta, systematic and unsystematic risks, factor models, other risk measures, stochastic dominance and portfolio optimization

Risk neutral probability measure (RNPM) (discrete case), existence of RNPM, Binomail lattice model, Binomial formula for pricing both European style and American style options, dividend and non-divided cases

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

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.

Suggested texts

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).

ELECTIVE-IV

UCS741 SIMULATION AND MODELLING

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives:

Introduction to Modeling and Simulation: Basic concept of Simulation, Advantages, Disadvantages, Applications of simulation, limitation of simulation, Model and types of models, modeling and simulation, Continuous and discrete simulation, analog and digital simulation, System environment, components of a system, steps in a simulation study, Simulation of Queuing and Inventory System.

Random Numbers generation: Pseudo-random generators, Testing of Pseudo-random number generators, Generation of non-uniformly distributed random numbers

Parallel process modeling: Using Petri nets and finite automata in simulation, Cellular automata and simulation.

Simulation Experiments: Run length of Static and Dynamic Stochastic Simulation Experiments, Minimizing variability in simulators without increasing Number of simulation Runs.

Design of Simulators: Design of Application Simulators for Multi-server Queuing System, PERT, Optimizing Inventory Policy and Cost in Business environment.

Input Modeling: Data collection, Identification and distribution with data, parameter estimation, Goodness of fit tests, Selection of input models without data, Multivariate and time series analysis. Verification and Validation of Model: Model Building, Verification, Calibration and Validation of Models.

Output Analysis: Types of Simulations with Respect to Output Analysis, Stochastic Nature of output data, Measures of Performance and their estimation, Output analysis of terminating simulation, Output analysis of steady state simulations. Simulation Software's: Selection of Simulation Software, Simulation packages, Trend in Simulation Software.

Laboratory Work: To carry out work on any simulation tools, Implementation of various techniques to generate random numbers. Apply any simulation model in real life applications**.**

Course learning outcomes (CLOs):

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

1. Describe the role of various elements of discrete event simulation and modeling paradigm.
2. Conceptualize real world situations related to systems development decisions, originating from source requirements and goals.
3. Generate and test random number variates and apply them to develop simulation models.
4. Interpret the model and apply the results to resolve critical issues in a real-world environment.
5. Classify various simulation models and their usage in real-life applications.

Text Books:

1. *Payne A. J., Introduction to Simulation: Programming Techniques and Methods of Analysis, McGraw Hill (1982).*
2. *Gorden G., System Simulation, Prentice Hall publication (1978), 2nd ed.*

Reference Books:

1. *Narsingh D., Systems Simulation with Digital Computer, PHI Publication (EEE) (2004), 3rd ed.*
2. *Banks J., Carson J. S., Nelson L. B., Nicol M. D, Discrete Event system Simulation, Pearson Education, Asia (2010), 5th ed.*

UCG731 GAME DESIGN AND DEVELOPMENT

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: Familiarizing with the various components involved in game development and exposure to the Window-based game programming.

Introduction: History of Video Games, Impact of Games on Society, Game Design, Game types, Game genres, Game Writing, UI Layout, Asset Management, game state, gamer services and Interactive Storytelling Understanding Hardware, Input Devices, Output Devices, Network

Requirements, Managing Game Performance, CPU vs.GPU, and Graphics Networking Performance.

Game Design and Development Concepts: Mathematical concepts, Collision Detection and resolution, Real-time game Physics, Graphics, Character Animation, Animate basic characters, Transform objects, Artificial Intelligence Agents, Architecture, and Techniques, Overview of Path finding, Audio Programming, Networking and Multiplayer.

Audio Visual Design and Production: Visual Design, 3D Modeling using 3D Studio Max, 3D Environments, 2D Textures and Texture mapping, Special Effects, Lighting, Animation, Cinematography, Audio design and production using Autodesk Maya Software.

Game Programming: Programming Fundamentals, Game Architecture, Memory and I/O system, Debugging Games, Introducing Object Oriented Programming concepts using C++ details, Number Systems, Programming: Basic Windows Programming, GDI and Menus, Dialogs and Controls, Sprite Animation, AI Techniques implementation.

Working with Unity and Scripting: Unity Demos, Courses Wiki, Lesson Files, Managing Project, Interface and Assets, Unity Interfaces, Prototyping and Scripting Basics, Collection, Inventory and HUD, Building Unity Game, Terrain, Unity Terrain Assets, Camera, Layer, GUI, Curves, Surfaces, Visible Surface Identification, 2D Games, UVs Animation, Movie and Audio, Scene Modeling, Unity Optimization Application and Techniques, Unity Deployment methods, character scripting.

Laboratory Work: 3D game development walkthrough on Unity 4.3 software, Maya, Audio Listeners and Sources on Unity 4.3 software, Learning C++ with SDL library and developing gaming programs and modules with C++.

Course learning outcomes (CLOs):

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

1. Illustrate the basic concepts, requirements, and processes of Game design and development.
2. Understand the physics and mathematics behind the game engine.
3. Discuss the elements contributing to the design of an advanced 3D game (AI and Networking based game).
4. Develop Windows and Android based 3D games using C#.
5. Implement some advanced real-world components relevant to games using AR and VR.

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.*

UCS742 DEEP LEARNING

L	T	P	Cr
3	0	2	4.0

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

Machine Learning Basics: Learning, Underfitting, Overfitting, Estimators, Bias, Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning, Unsupervised Learning and Stochastic Gradient Decent.

Deep Feedforward Network: Feed-forward Networks, Gradient-based Learning, Hidden Units, Architecture Design, Computational Graphs, Back-Propagation, Regularization, Parameter Penalties, Data Augmentation, Multi-task Learning, Bagging, Dropout and Adversarial Training and Optimization.

Convolution Networks: Convolution Operation, Pooling, Basic Convolution Function, Convolution Algorithm, Unsupervised Features and Neuroscientific for convolution Network.

Sequence Modelling: Recurrent Neural Networks (RNNs), Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Network, Recursive Neural Networks and Echo State networks.

Deep Generative Models: Boltzmann Machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, Sigmoid Belief Networks, Directed Generative Net, Drawing Samples from Auto –encoders.

Laboratory Work: To implement models 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. Comprehend the advancements in learning techniques.
2. Compare and explain various deep learning architectures and algorithms.
3. Demonstrate the applications of deep learning in various fields.
4. Apply deep learning specific open source libraries for solving real life problems.

Text Books:

1. *Goodfellow L., Bengio Y. and Courville A., Deep Learning, MIT Press (2016).*
2. *Patterson J. and Gibson A., Deep Learning: A Practitioner's Approach, O'Reilly (2017), 1st ed.*

Reference Books:

1. *Haykin S., Neural Network and Machine Learning, Prentice Hall Pearson (2009), 3rd ed.*
2. *Geron A., Hands-on Machine Learning with Sci-kit and TensorFlow, O'Reilly Media (2017).*

UCS742 DEEP LEARNING

L	T	P	Cr
3	0	2	4.0

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

Machine Learning Basics: Learning, Underfitting, Overfitting, Estimators, Bias, Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning, Unsupervised Learning and Stochastic Gradient Decent.

Deep Feedforward Network: Feed-forward Networks, Gradient-based Learning, Hidden Units, Architecture Design, Computational Graphs, Back-Propagation, Regularization, Parameter Penalties, Data Augmentation, Multi-task Learning, Bagging, Dropout and Adversarial Training and Optimization.

Convolution Networks: Convolution Operation, Pooling, Basic Convolution Function, Convolution Algorithm, Unsupervised Features and Neuroscientific for convolution Network.

Sequence Modelling: Recurrent Neural Networks (RNNs), Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Network, Recursive Neural Networks and Echo State networks.

Deep Generative Models: Boltzmann Machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, Sigmoid Belief Networks, Directed Generative Net, Drawing Samples from Auto –encoders.

Laboratory Work: To implement models 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. Comprehend the advancements in learning techniques.
2. Compare and explain various deep learning architectures and algorithms.
3. Demonstrate the applications of deep learning in various fields.
4. Apply deep learning specific open source libraries for solving real life problems.

Text Books:

1. *Goodfellow L., Bengio Y. and Courville A., Deep Learning, MIT Press (2016).*
2. *Patterson J. and Gibson A., Deep Learning: A Practitioner's Approach, O'Reilly (2017), 1st ed.*

Reference Books:

1. *Haykin S., Neural Network and Machine Learning, Prentice Hall Pearson (2009), 3rd ed.*
2. *Geron A., Hands-on Machine Learning with Sci-kit and TensorFlow, O'Reilly Media (2017).*

UCS7709 ADVANCED TOPICS IN SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To apply advance topics in software engineering. To specify, abstract, verify and validate solutions to large-size problems, to plan, develop and manage large software using state-of-the-art methodologies and learn emerging trends in software engineering.
Formal Methods: Basic concepts, mathematical preliminaries, Applying mathematical notations for formal specification, formal specification languages, using Z to represent an example software component, the ten commandments of formal methods, formal methods- the road ahead.
Cleanroom Software Engineering: approach, functional specification, design and testing.
Component-Based Software Engineering: CBSE process, domain engineering, component-based development, classifying and retrieving components, and economics of CBSE.
Computer-Aided Software Engineering: Building blocks for CASE, taxonomy of CASE tools, integrated CASE environments, integration architecture, CASE repository, case Study of tools like TCS Robot.
Reengineering: Business process reengineering, software reengineering, reverse reengineering, restructuring, forward reengineering, Economics of reengineering.
Web Engineering: Attributes of web-based applications, the WebE process, a framework for Web Engineering, formulating, analyzing web-based systems, design and testing for web-based applications, Management issues.
Mobile Development Process: Model View Controller, Presentation Abstraction Control, UML based development, Use cases, Testing: Mobile infrastructure, Validating use cases, Effect of dimensions of mobility on testing, Case study: IT company, Requirements, Detailed design, Implementation.
Software Engineering Issues in Embedded Systems: Characteristics of embedded systems I/O, Embedded systems/real time systems. Embedded software architecture, control loop, interrupts control system, co-operating multitasking, pre-emptive multitasking, Domain analysis, Software element analysis, requirement analysis, Specification, Software architecture, Software analysis design, implementation, testing, validation, verification and debugging of embedded systems.
Laboratory Work: To implement the advance concepts in the lab using related tools and to develop the project using related technologies.
Course learning outcomes (CLOs):
After the completion of the course, the student will be able to:
1. Comprehend concepts of formal methods and apply mathematical notations for formal specification.
2. Evaluate various approaches for software engineering, including cleanroom software engineering and component-based software engineering.
3. Demonstrate the use of various tools like CASE and TCS robot.
4. Comprehend web engineering and create web-based application and apply re-engineering concepts on traditional applications.
5. Apply software engineering for Mobile Development Process and Embedded Systems

Text Books:

1. *Pressman S. R., and Maxim R. b., Software Engineering a Practitioners Approach, McGraw-Hill Education (2015). 8th ed.*
2. *Sommerville L., Software Engineering, Pearson Education (2016) 10th ed.*

Reference Books:

1. *Bowen J., Formal Specification and Documentation using Z - A Case Study Approach, International Thomson Computer Press (2003) 2nd ed.*
2. *Oshana R. and Kraeling M., Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications, Newnes Publisher (2013) 1st ed.*

Computational Number Theory

L	T	P	Cr
3	0	2	4.0

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.

Course Content

Divisibility and Primes- Euclid's algorithm for the greatest common divisor, linear Diophantine equations, Prime numbers, Fundamental Theorem of arithmetic, 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- Classical Ciphers, Block and stream ciphers, 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

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. I. Niven, H.S. Zukermann, H.L. Montgomery, An introduction to theory of numbers, Willey, 1998
3. D. Burton, Elementary Number Theory, McGraw-Hill, 2005

Reference Books:

1. Behrouz A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, McGraw Hill, 2010.
2. J. Pipher, J. Hoffstein and J.H. Silverman, An introduction to Mathematical Cryptography, Springer-verlag 2008.

GENERIC ELECTIVE

UPH063 NANOSCIENCE AND NANOMATERIALS

	L	T	P	Cr
Prerequisite(s): None	**3**	**1**	**0**	**3.5**

Course Objectives: To introduce the basic concept of Nanoscience and advanced applications of nanotechnology,

Fundamental of Nanoscience: Features of Nanosystem, Free electron theory and its features, Idea of band structures, Density of states in bands, Variation of density of state and band gap with size of crystal,
Quantum Size Effect: Concepts of quantum effects, Schrodinger time independent and time dependent equation, Electron confinement in one-dimensional well and three-dimensional infinite square well, Idea of quantum well structure, Quantum dots and quantum wires,
Nano Materials: Classification of Nano Materials their properties, Basic concept relevant to application, Fullerenes, Nanotubes and nano-wires, Thin films chemical sensors, Gas sensors, Vapour sensors and Bio sensors,
Synthesis and processing: Sol-gel process, Cluster beam evaporation, Ion beam deposition, Chemical bath deposition with capping techniques and ball milling, Cluster assembly and mechanical attrition, Sputtering method, Thermal evaporation, Laser method,
Characterization: Determination of particle size, XRD technique, Photo luminescence, Electron microscopy, Raman spectroscopy, STEM, AFM,
Applications: Photonic crystals, Smart materials, Fuel and solar cells, Opto-electronic devices

Course Learning Outcomes: Students will be able to:

1. discriminate between bulk and nano materials,
2. establish the size and shape dependence of Materials' properties,
3. correlate 'quantum confinement' and 'quantum size effect' with physical and chemical properties of nanomaterials,
4. uses top-down and bottom-up methods to synthesize nanoparticles and control their size and shape
5. characterize nanomaterials with various physico-chemical characterization tools and use them in development of modern technologies

Recommended Books:

1. *Booker, R., Boysen, E., Nanotechnology, Wiley India Pvt, Ltd, (2008)*
2. *Rogers, B., Pennathur, S., Adams, J., Nanotechnology, CRS Press (2007)*
3. *Bandyopadhyay, A,K., Nano Materials, New Age Int,, (2007)*
4. *Niemeyer, C. N., and Mirkin, C, A., Nanobiotechnology: Concepts, Applications and Perspectives, Wiley VCH, Weinhein, Germany (2007)*

UEN004 TECHNOLOGIES FOR SUSTAINABLE DEVELOPMENT

L	T	P	Cr
3	0	0	3.0

Course Objectives: To provide acquaintance with modern cleaner production processes and emerging energy technologies; and to facilitate understanding the need and application of green and renewable technologies for sustainable development of the Industry/society

Course Contents:

Concepts of Sustainability and Industrial Processes: Industrialization and sustainable development; Cleaner production (CP) in achieving sustainability; Source reduction techniques - Raw material substitution; Process modification and equipment optimization; Product design or modification; Reuse and recycling strategies; Resources and by-product recovery from wastes; Treatment and disposal; CDM and Pollution prevention programs; Good housekeeping; CP audits, **Green Design: G**reen buildings - benefits and challenges; public policies and market-driven initiatives; Effective green specifications; Energy efficient design; Passive solar design; Green power; Green materials and Leadership in Energy and Environmental Design (LEED)

Renewable and Emerging Energy Technologies: Introduction to renewable energy technologies- Solar; wind; tidal; biomass; hydropower; geothermal energy technologies; Emerging concepts; Biomolecules and energy; Fuel cells; Fourth generation energy systems.

Course Learning Outcomes (CLOs):

The students will be able to:

1. comprehend basic concepts in source reduction, waste treatment and management
2. Identify and plan cleaner production flow charts/processes for specific industrial sectors
3. examine and evaluate present and future advancements in emerging and renewable energy technologies

Recommended Books

1. *Kirkwood, R,C, and Longley, A,J, (Eds,), Clean Technology and the Environment, Chapman & Hall, London (1995),*
2. *World Bank Group; Pollution Prevention and Abatement Handbook – Towards Cleaner Production, World Bank and UNEP; Washington DC (1998),*
3. *Modak, P,, Visvanathan, C, and Parasnis, M,, Cleaner Production Audit, Course Material on Cleaner Production and Waste Minimization; United Nations Industrial Development Organization (UNIDP) (1995),*
4. *Rao, S, and Parulekar, B,B,, Energy Technology: Non-conventional; Renewable and Conventional; Khanna Pub,(2005) 3rd Ed.*

UHU009 INTRODUCTION TO COGNITIVE SCIENCE

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

Course Objectives: This course provides an introduction to the study of intelligence, mind and brain from an interdisciplinary perspective, It encompasses the contemporary views of how the mind works, the nature of reason, and how thought processes are reflected in the language we use, Central to the course is the modern computational theory of mind and it specifies the underlying mechanisms through which the brain processes language, thinks thoughts, and develops consciousness,

Course Contents

Overview of Cognitive Science: Newell's big question, Constituent disciplines, Interdisciplinary approach, Unity and diversity of cognitive science,

Philosophy: Philosophy of Mind, Cartesian dualism Nativism vs, empiricism, Mind-body problem, Functionalism, Turing Test, Modularity of mind, Consciousness, Phineas Gage, Physicalism.

Psychology: Behaviorism vs, cognitive psychology, The cognitive revolution in psychology, Hardware/software distinction , Perception and psychophysics, Visual cognition, Temporal dynamics of visual perception, Pattern recognition, David Marr's computational theory of vision, Learning and memory, Theories of learning, Multiple memory systems, Working Memory and Executive Control, Memory span, Dissociations of short- and long-term memory, Baddeley's working memory model.

Linguistics: Components of a grammar, Chomsky, Phrases and constituents, Productivity, Generative grammars, Compositional syntax, Productivity by recursion, Surface- and deep structures, Referential theory of meaning, Compositional semantics, Semantics, Language acquisition, Language and thought.

Neuroscience: Brain anatomy, Hierarchical functional organization, Decorticate animals, Neuroimaging, Neurophysiology, Neuron doctrine, Ion channels, Action potentials, Synaptic transmission, Synaptic plasticity, Biological basis of learning, Brain damage, Amnesia, Aphasia, Agnosia, Parallel Distributed Processing(PDP), Computational cognitive neuroscience, The appeal of the PDP approach, Biological Basis of Learning, Cajal's synaptic plasticity hypothesis, Long-term potentiation (LTP) and depotentiation (LTD), NMDA receptors and their role in LTP, Synaptic consolidation, Vertical integration, The Problem of representation, Shannon's information theory.

Artificial Intelligence: Turing machines, Physical symbol systems, Symbols and Search Connectionism, Machine Learning,**,** Weak versus strong AI, Subfields, applications, and recent trends in AI, Turing Test revisited, SHRDLU, Heuristic search, General Problem Solver (GPS), Means-ends analysis.

Cognitive architectures: Tripartite architecture, Integration, ACT-R Architecture Modularity.

Recommended Books

1. *Bermúdez, J.L., Cognitive Science: An Introduction to the Science of the Mind (2nd Ed,), Cambridge, UK: Cambridge (2014).*
2. *Friedenberg ,J,D, and Silverman,G, Cognitive Science: An Introduction To The Study Of Mind, Sage Publications:, London* (2014)
3. *Thagard, P., Mind: An introduction to Cognitive Science, MIT Press, (2005)*
4. *Thagard, P., (1998) Mind Readings: Introductory Selections on Cognitive Science, MIT Press, Cambridge, Mass,*

UHU008 INTRODUCTION TO CORPORATE FINANCE

L	T	P	Cr
3	0	0	3.0

Course Objective: This course aims to provide the students with the fundamental concepts, principles and approaches of corporate finance, enable the students to apply relevant principles and approaches in solving problems of corporate finance and help the students improve their overall capacities.

Course Content:

Introduction to corporate finance: Finance and corporate finance. Forms of business organizations, basic types of financial management decisions, the goal of financial management, the agency problem; the role of the financial manager; basic types of financial management decisions.
Financial statements analysis: Balance sheet, income statement, cash flow, fund flow financial statement analysis Computing and interpreting financial ratios; conducting trend analysis and Du Pont analysis.
The time value of money: Time value of money, future value and compounding, present value and discounting, uneven cash flow and annuity, discounted cash flow valuation.
Risk and return: Introduction to systematic and unsystematic risks, computation of risk and return, security market line, capital asset pricing model.
Long-term financial planning & Financial Decisions: Various sources of long term financing, the elements and role of financial planning, financial planning model, percentage of sales approach, external financing needed. Cost of capital, financial leverage, operating leverage. Capital structure, theories of capital structure net income, net operating income & M&M proposition I and II.
Short-term financial planning and management: Working capital, operating cycle, cash cycle, cash budget, short-term financial policy, cash management, inventory management, credit management.
Capital budgeting : Concepts and procedures of capital budgeting, investment criteria (net present value, payback, discounted payback, average accounting return, internal rate of return, profitability index), incremental cash flows, scenario analysis, sensitivity analysis, break-even analysis,
Dividend policy: Dividend, dividend policy, Various models of dividend policy (Residual approach, Walter model, Gordon Model, M&M, Determinants of dividend policy.
Security valuation: Bond features, bond valuation, bond yields, bond risks, stock features, common stock valuation, and dividend discount & dividend growth models. Common stock yields, preferred stock valuation.

Recommended Books:

1. *Brealey, R. A., Myers. S.C., Allen, F., Principles of Corporate Finance (9th edition), The McGraw-Hill, London, (2006).*
2. *Ehrhardt, M.C., Brigham, E.F., Financial Management: Theory and Practice (10th edition) South Western-Cengage, New York (2011)*
3. *Van Horne, J.C., Wachowicz, J.M., Kuhlemeyer, G.A., 2005, Fundamentals of Financial Management, Pearson, Vancouver (2010)*
4. *Pandey, I. M., Financial management, Vikas Publishing House Pvt. Ltd., Noida (2011)*
5. *Elton, E.J. and Gruber, M.J., Modern Portfolio Theory and Investment Analysis, (7th Edition), John Wiley and Sons, New York (2007)*

UMA062 GRAPH THEORY AND APPLICATIONS

L	T	P	Cr
3	1	0	3.5

Course Objectives: The objective of the course is to introduce students with the fundamental concepts in graph Theory, with a sense of some its modern applications. They will be able to use these methods in subsequent courses in the computer, electrical and other engineering,

Introduction: Graph, Finite and infinite graph, incidence and degree, Isolated vertex, Pendent vertex and null graph, Isomorphism, Sub graph, Walks, Paths and circuits, Euler circuit and path, Hamilton path and circuit, Euler formula, Homeomorphic graph, Bipartite graph, Edge connectivity, Computer representation of graph, Digraph.

Tree and Fundamental Circuits: Tree, Distance and center in a tree, Binary tree, Spanning tree, Finding all spanning tree of a graph, Minimum spanning tree.

Graph and Tree Algorithms: Shortest path algorithms, Shortest path between all pairs of vertices, Depth first search and breadth first of a graph, Huffman coding, Cuts set and cut vertices, Warshall's algorithm, topological sorting.

Planar and Dual Graph: Planner graph, Kuratowski's theorem, Representation of planar graph, five-color theorem, Geometric dual.

Coloring of Graphs: Chromatic number, Vertex coloring, Edge coloring, Chromatic partitioning, Chromatic polynomial, covering.

Application of Graphs and Trees: Konigsberg bridge problem, Utilities problem, Electrical network problem, Seating problem, Chinese postman problem, Shortest path problem, Job sequence problem, Travelling salesman problem, Ranking the participant in a tournament, Graph in switching and coding theory, Time table and exam scheduling, Applications of tree and graph in computer science.

Course learning outcomes: Upon completion of the course, the students will be able to:

1) understand the basic concepts of graphs, directed graphs, and weighted graphs and able to present a graph by matrices.
2) understand the properties of trees and able to find a minimal spanning tree for a given weighted graph.
3) understand Eulerian and Hamiltonian graphs.
4) apply shortest path algorithm to solve Chinese Postman Problem .
5) apply the knowledge of graphs to solve the real life problem.

Recommended Books

1. *Deo, N., Graph Theory with Application to Engineering with Computer Science, PHI, New Delhi (2007)*
2. *West, D. B., Introduction to Graph Theory, Pearson Education, London (2008)*
3. *Bondy, J. A. and Murty, U.S.R., Graph Theory with Applications, North Holland Publication, London (2000)*
4. *Rosen, K. H., Discrete Mathematics and its Applications, Tata-McGraw Hill, New Delhi (2007)*

UMA061 ADVANCED NUMERICAL METHODS

L	T	P	Cr
3	1	0	3.5

Course Objective: The main objective of this course is to motivate the students to understand and learn various advanced numerical techniques to solve mathematical problems governing various engineering and physical problems.

Non-Linear Equations: Methods for multiple roots, Muller's, Iteration and Newton-Raphson method for non-linear system of equations and Newton-Raphson method for complex roots.

Polynomial Equations: Descartes' rule of sign, Birge-vieta, Giraffe's methods.

System of Linear Equations: Cholesky and Partition methods, SOR method with optimal relaxation parameters.

Eigen-Values and Eigen-Vectors: Similarity transformations, Gerschgorin's bound(s) on eigenvalues, Given's and Rutishauser methods.

Interpolation and Approximation: Cubic and B – Spline and bivariate interpolation, Least squares approximations, Gram-Schmidt orthogonalisation process and approximation by orthogonal polynomial, Legendre and Chebyshev polynomials and approximation.

Differentiation and Integration: Differentiation and integration using cubic splines, Romberg integration and multiple integrals.

Ordinary differential Equations: Milne's, Adams-Moulton and Adam's Bashforth methods with their convergence and stability, Shooting and finite difference methods for second order boundary value problems.

Course learning outcomes: Upon completion of this course, the students will be able to**:**

1. find multiple roots of equation and apply Newton -Raphson's method to obtain complex roots as well solution of system of non - linear equations.
2. learn how to obtain numerical solution of polynomial equations using Birge - Vitae and Giraffe's methods.
3. apply Cholesky, Partition and SOR methods to solve system of linear equations.
4. understand how to approximate the functions using Spline, B- Spline, least square approximations
5. learn how to solve definite integrals by using cubic spline, Romberg and initial value problems and boundary value problems numerically.

Recommended Books

1. *Gerald, C.F. and Wheatley, P.O., Applied Numerical Analysis, Pearson Education (2008) 7th ed.*
2. *Gupta, S.R., Elements of Numerical Analysis, MacMillan India (2009).*
3. *Atkinson, K.E., An introduction to Numerical Analysis, John Wiley (2004) 2nd ed.*
4. *S.D. Conte, S.D. and Carl D. Boor, Elementary Numerical Analysis: An Algorithmic Approach, Tata McGraw Hill (2005).*
5. *Jain M. K., Iyengar. S.R.K. and Jain, R.K. Numerical Methods for Scientific and Engineering Computation, New Age International (2008) 5th ed.*

UHU006 INTRODUCTORY COURSE IN FRENCH

L	T	P	Cr
2	2	0	3.0

Course objective: The objective of the course is to introduce to the students:

1. The basics of French language to the students. It assumes that the students have minimal or no prior knowledge of the language.
2. To help them acquire skills in writing and speaking in French, comprehending written and spoken French.
3. The students are trained in order to introduce themselves and others, to carry out short conversation, to ask for simple information, to understand and write short and simple messages, to interact in a basic way.
4. The main focus of the students will be on real life language use, integration of French and francophone culture, & basic phrases aimed at the satisfaction of needs of concrete type.
5. During class time the students are expected to engage in group & pair work.

Course Contents:

Communicative skills: Greetings and Its Usage, Asking for and giving personal information, How to ask and answer questions, How to talk over the phone, Exchange simple information on preference, feelings etc. Invite, accept, or refuse invitation, Fix an appointment, Describe the weather, Ask for/give explanations, Describe a person, an object, an event, a place.
Grammar: Pronouns: Pronom sujets (Je/Tu/Il/Elle/Nous/Vous/Ils/Elles), Nouns: Genders, Articles: Definite article and Indefinite articles, Verbs: Regular verbs (-er, -ir ending) Irregular verbs (-re ending), Auxiliary verbs (avoir, être, aller). Adjective: Description, Adjective possessive, Simple Negation, Tense: Present, Future, Questions, Singular & plural.
Vocabulary: Countries and Nationalities, Professions, Numbers (ordinal, cardinal), Colours, Food and drinks, Days of the week, Months, Family, Places.
Phonetics: The course develops the ability, to pronounce words, say sentences, questions and give orders using the right accent and intonation. To express surprise, doubt, fear, and all positive or negative feelings using the right intonation. To distinguish voiced and unvoiced consonants. To distinguish between vowel sounds.

Course Learning Outcomes:

Upon the completion of the course:

1. The students begin to communicate in simple everyday situations acquiring basic grammatical structure and vocabulary.
2. The course develops oral and reading comprehension skills as well as speaking and writing.
3. Students can demonstrate understanding of simple information in a variety of authentic materials such as posters, advertisement, signs etc.
4. Discuss different professions, courses and areas of specialisation.
5. Write simple messages, letters, composition and dialogues. Complete simple forms and documents.
6. Express feelings, preferences, wishes and opinions and display basic awareness of francophone studies.
7. Units on pronunciation and spelling expose students to the different sounds in the French language and how they are transcribed.

Recommended Books:

1. *Alter ego-1 : Méthode de français by Annie Berthet, Catherine Hugot, Véronique M. Kizirion, Beatrix Sampsonis, Monique Waendendries, Editions Hachette français langue étrangère.*
2. *Connexions-1 : Méthode de français by Régine Mérieux, Yves Loiseau, Editions Didier*
3. *Version Originale-1: Méthode de français by Monique Denyer, Agustin Garmendia.*
4. *Marie-Laure Lions-Olivieri, Editions Maison des Langues, Paris 2009*

5. *Latitudes-1 : Méthode de français by Régine Mérieux, Yves Loiseau, Editions Didier*
6. *Campus-1 : Méthode de français by Jacky Girardet, Jacques Pécheur, Editions CLE International.*
7. *Echo-1 : Méthode de français by J. Girardet, J. Pécheur, Editions CLE International.*

UBTXXX BIOLOGY FOR ENGINEERS

L	T	P	Cr
2	1	0	2.5

Course Objective: To learn about living world and basic functioning of biological systems. The course encompasses understanding of origin of life, its evolution and some of its central characteristics. It also aims to familiarize engineering students to some of the intricate biological phenomena and mechanisms.

Detailed Contents:

Characteristics of life: Living versus non-living organisms, origin of life, theory of evolution, diversity of life, classification of life into animals, plants, fungi, protists, archea and bacteria. Phylogenetics and its relationship with evolution.

Introduction to biological systems: Cell as basic unit of life, cellular organelles and their functions, important biomacromolecules (carbohydrates, lipids, proteins and nucleic acids) and their properties.

Cell membrane: Membrane structure, selective permeability, transport across cell membrane, active and passive transport, membrane proteins, type of transport proteins, channels and pumps, examples of membrane transport in cell physiology.

Classical and molecular genetics: Heredity and laws of genetics, genetic material and genetic information, Structure and properties of DNA, central dogma, replication of genetic information, universal codon system, encoding of genetic information via transcription and translation.

Course Learning Outcomes (CLOs): After completion of this course the students will be able to:

1. Describe living-systems and differentiate them from non-living systems
2. Explain the theory of evolution and apply it non-living world
3. Apply properties of nucleic acids in molecular recognition based diagnostics
4. Familiarized with various transport mechanisms across cell membranes
5. Explain how genetic information is stored, replicated and encoded in living organisms.

Recommended Books:

1. *Nelson, D.L., Cox, M.M., Lehninger: Principles of Biochemistry, WH Freeman (2008) 5th ed.*
2. *Dhami, P.S., Srivastava, H.N. Chopra, G., A Textbook of Biology, Pradeep Publications (2008).*
3. *Das, H.K., Textbook of Biotechnology, John Wiley & Sons (2004) 3rd Edition.*
4. *Gardner, E.J., Simmons, M., Peter, S.D., Principles of Genetics, John Wiley & Sons (2008)*
5. *Albert, B., Essential Cell Biology, Taylor & Francis, London (2009)*

UCS001: INTRODUCTION TO CYBER SECURITY

L	T	P	Cr
3	0	0	3.0

Course Objective: In this course, the student will learn about the essential building blocks and basic concepts around cyber security such as Confidentiality, Integrity, Availability, Authentication, Authorization, Vulnerability, Threat & Risk and so on.

Course Content

Introduction: Introduction to Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control, and Cryptography, Authentication, Access Control, Cryptography
Programs and Programming: Unintentional (Non-malicious) Programming Oversights, Malicious Code—Malware, Countermeasures
Web Security: User Side, Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks
Operating Systems Security: Security in Operating Systems, Security in the Design of Operating Systems, Rootkit
Network Security: Network Concepts, Threats to Network Communications, Wireless Network Security, Denial of Service, Distributed Denial-of-Service Strategic Defenses: Security Countermeasures, Cryptography in Network Security, Firewalls, Intrusion Detection and Prevention Systems, Network Management
Cloud Computing and Security: Cloud Computing Concepts, Moving to the Cloud, Cloud Security Tools and Techniques, Cloud Identity Management, Securing IaaS
Privacy: Privacy Concepts, Privacy Principles and Policies, Authentication and Privacy, Data Mining, Privacy on the Web, Email Security, Privacy Impacts of Emerging Technologies, Where the Field Is Headed
Management and Incidents: Security Planning, Business Continuity Planning, Handling Incidents, Risk Analysis, Dealing with Disaster
Legal Issues and Ethics: Protecting Programs and Data, Information and the Law, Rights of Employees and Employers, Redress for Software Failures, Computer Crime, Ethical Issues in Computer Security, Incident Analysis with Ethics
Emerging Topics: The Internet of Things, Economics, Computerized Elections, Cyber Warfare.
Course Learning Outcomes: After completion of this course, the students will be able to:

1. Understand the broad set of technical, social & political aspects of Cyber Security and security management methods to maintain security protection
2. Appreciate the vulnerabilities and threats posed by criminals, terrorist and nation states to national infrastructure
3. Understand the nature of secure software development and operating systems

4. Recognize the role security management plays in cyber security defense and legal and social issues at play in developing solutions.

Text Books:

1. *Pfleeger, C.P., Security in Computing, Prentice Hall, 2010, 5th edition.*
2. *Schneier, Bruce. Applied Cryptography, Second Edition, John Wiley & Sons, 1996.*

Reference Books:

1. *Rhodes-Ousley, Mark. Information Security: The Complete Reference, Second Edition, Information Security Management: Concepts and Practice. New York, McGraw-Hill, 2013.*
2. *Whitman, Michael E. and Herbert J. Mattord. Roadmap to Information Security for IT and Infosec Managers. Boston, MA: Course Technology, 2011.*