THAPAR
UNIVERSITY

COURSE SCHEME

&

SYLLABUS

FOR

B.E.

COMPUTER ENGINEERING

2017

B.E. (COMPUTER ENGINEERING) 2017–COURSE SCHEME (ALL YEARS)

First Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UCB008	APPLIED CHEMISTRY	3	1	2	4.5
2.	UTA007	COMPUTER PROGRAMMING - I	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**	**8**	**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.	UTA009	COMPUTER PROGRAMMING-II	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) (Mangonel)	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.	UCS406	DATA STRUCTURES & ALGORITHMS (4 SELF EFFORT HOURS)	3	0	2	6.0
6.	UCS407	INVENTIONS & INNOVATIONS IN COMPUTING	2	0	0	2.0
7.	UCS303	OPERATING SYSTEMS	3	0	2	4.0
			18	**2**	**12**	**30.0**

Fourth Semester						
S. No.	**Course Number**	**Course Title**	**L**	**T**	**P**	**Cr**
1.	UTA014	ENGINEERING DESIGN PROJECT-II (6 Self-Effort Hours) (Buggy)	1	0	4	6.0
2.	UTA002	MANUFACTURING PROCESSES	2	0	3	3.5
3.	UMA031	OPTIMIZATION TECHNIQUES	3	1	0	3.5
4.	UES010	SOLIDS AND STRUCTURES *	3	1	2	4.5
5.	UES011	THERMO-FLUIDS *	3	1	2	4.5
6.	UCS310	DATABASE MANAGEMENT SYSTEMS	3	0	2	4.0
7.	UCS404	DISCRETE MATHEMATICAL STRUCTURES	3	1	0	3.5
			18	**4**	**13**	**29.5**

:UES010, UES011 Lab to be conducted every alternate week.

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.	UCS793	CAPSTONE PROJECT* (STARTS) SEH-6	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.	UCS793	CAPSTONE PROJECT (CONTINUED) SEH-14	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**

&Output in form of research paper

Eight Semester						
S. No.	Course Number	Course Title	L	T	P	Cr
1.	UCS895	PROJECT SEMESTER/START-UP SEMESTER				20.0
		OR				
2.	USC896	CAPSTONE PROJECT II SEH-20	0	0	4	12.0
3.	UCS806	ETHICAL HACKING	3	0	2	4.0
4.	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

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

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

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.	USE601	SOFTWARE VERIFICATION AND VALIDATION	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

GENERIC ELECTIVE

SR. NO.	CODE	TITLE	L	T	P	CR
1	UHU006	INTRODUCTORY COURSE IN FRENCH	3	0	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

Semester wise Credits for BE (Computer Engineering)	
Semester	**Credits**
First	22
Second	23.5
Third	30
Fourth	29.5
Fifth	27.5
Sixth	24.5
Seventh	24.0
Eight	20
Total Credits	**201**

UCB008: APPLIED CHEMISTRY

L	T	P	Cr
3	1	2	4.5

Course objective: 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: The students will be able to reflect on:

1. concepts of electrodes in electrochemical cells, migration of ions, liquid junction potential and conductometrictitrations.
2. atomic and molecular spectroscopy fundamentals like Beer`s law, flame photometry,atomic absorption spectrophotometry, UV-Vis andIR.
3. water and its treatment methods like lime soda and ionexchange.
4. concept of phase rule, fuel quality parameters and alternativefuels.
5. polymerization, molecular weight determination and applications as biodegradable and conductingpolymers.
6. laboratory techniques like pH metry, potentiometry, colourimetry, conductometry and volumetry.

Text Books

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

Reference Books

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

Evaluation Scheme

MST	EST	Sessional (May include Quizzes/Assignments/Lab Evaluation)
25	40	35

UTA007: COMPUTER PROGRAMMING – I

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.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UEC001: ELECTRONIC ENGINEERING

L	T	P	Cr
3	1	2	4.5

Course objective: 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 outcome (CLO): 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. *M. M. Mano and M.D. Ciletti, Digital Design, Pearson, Prentice Hall,2013.*
2. *Milliman, J. and Halkias, C.C., Electronic Devices and Circuits, Tata McGraw Hill,2007.*
3. *Donald D Givone, Digital Principles and Design, McGraw-Hill,2003.*

Reference Books:

- *John F Wakerly, Digital Design: Principles and Practices, Pearson,(2000).*
- *N Storey, Electronics: A Systems Approach, Pearson, Prentice Hall,(2009).*
- *Boylestad, R.L. and Nashelsky, L., Electronic Devices & Circuit Theory, Perason(2009).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	25
2.	EST	35
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

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

Group assignments: Assignments related to Sanitary landfill systems; e-waste management; Municipal solidwaste management; Biodiversity and biopiracy; Air pollution control systems; Water treatment systems; Wastewater treatment plants; Solar heating systems; Solar power plants; Thermal power plants; Hydroelectric power plants; Biofuels; Environmental status assessments; Energy status assessments, etc.

Course Learning Outcomes:

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

- Correlate major local and regional environmental issues with changes in ecology and human health

- Monitor and document the development and dynamics of ecosystems in experimental or natural microcosms
- Define and document local resource consumption patterns and conservation strategies
- Define opportunities available for energy conservation and for use of renewable energy resources in local and regional entities.

Text Books:

1. *Bharucha, E., Textbook of Environmental Studies, Universities Press (2005).*
2. *Chapman, J.L. and Reiss, M.J., Ecology-Principles and Application, Cambridge University Press (LPE) (1999).*
3. *Joseph, B., Environmental Studies, Tata McGraw-Hill (2006).*
4. *Eastop, T.P. and Croft, D.R. Energy Efficiency for Engineers and Technologists,Longman and Harow (2006)*

Reference Books:

1. *Miller, G.T., Environmental Science- Working with Earth, Thomson (2006).*
2. *Wright, R.T., Environmental Science-Towards a sustainable Future, Prentice Hall (2008) 9th ed.*
3. *O’Callagan, P.W., Energy Management, McGraw Hill Book Co. Ltd. (1993).*

Evaluation Scheme:

S.No	Evaluation Elements	Weightage (%)
1.	MST	30
2.	EST	50
3.	Sessionals (Quizzes/assignments/group presentations)	20

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: Upon completion of this course, the students 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 compute 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, G.B. and Finney, R.L., Calculus and Analytic Geometry, Pearson Education (2007), 9thed.
2. Stewart James, Essential Calculus; Thomson Publishers (2007), 6thed.

Reference Books:

1) Wider David V, Advanced Calculus: Early Transcendentals, Cengage Learning(2007).
2) Apostol Tom M, Calculus, Vol I and II, John Wiley(2003).

Evaluation Scheme:

Sr.No.	Evaluation Elements	Weight age (%)
1.	MST	30
2.	EST	45
3.	Sessionals (May include assignments/quizzes)	25

UES009: MECHANICS

L	T	P	Cr
2	1	0	2.5

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.

Review of Newton's law of motion and vector algebra

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

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

Friction: Sliding, belt, screw and rolling.

Properties of plane surfaces: First moment of area, centroid, second moment of area etc.

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 (CLO):

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

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

Text Books

1. *Shames, I. H. Engineering Mechanics: Dynamics, Pearson Education India(2002).*
2. *Beer, Johnston, Clausen and Staab, Vector Mechanics for Engineers, Dynamics, McGraw-Hill Higher Education(2003).*

Reference Books

1) *Hibler, T.A., Engineering Mechanics: Statics and Dynamics, Prentice Hall(2012).*
2) *Timoshenko and Young, Engineering Mechanics, Tata McGraw Hill Education Private Limited(2000).*

Evaluation Scheme

Sr. No.	Evaluation Elements	Weights (%)
1.	MST	30
2.	EST	45
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizzes)	25

UPH004: APPLIED PHYSICS

L	T	P	Cr
3	1	2	4.5

Prerequisite(s): None 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 canesugar solution.
6. Study and proof of Malus' law inpolarization.
7. Determination of beam divergence and beam intensity of a given laser.
8. Determination of displacement and conducting currents through adielectric.
9. Determination of Planck's constant.

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

Course Outcomes:

Upon completion of this course, students will be able to:

1. Understand damped and simple harmonic motion, the role of reverberation in designing a hall and generation and detection of ultrasonic waves.
2. Use Maxwell’s equations to describe propagation of EM waves in a medium.
3. Demonstrate interference, diffraction and polarization of light.
4. Explain the working principle of Lasers.
5. Use the concept of wave function to find probability of a particle confined in a box.

Text Books

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

Reference Books

1 *Wehr, M.R, Richards, J.A., Adair, T.W., Physics of The Atom, Narosa Publishing House (1990) 4th ed.*

2 *Verma, N.K., Physics for Engineers, Prentice Hall of India (2014)1st ed.*

3 *Pedrotti, Frank L., Pedrotti, Leno S., and Pedrotti, Leno M., Introduction to Optics, Pearson Prentice Hall™ (2008) 3rd ed.*

UTA009: COMPUTER PROGRAMMING – II

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++

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UEE001: ELECTRICAL ENGINEERING

L	T	P	Cr.
3	1	2	4.5

Course Objective: To introduce concepts of DC and AC circuits and electromagnetism. To make the students understand the concepts and working of 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 Outcome (CLO):

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

1. Apply networks laws and theorems to solve electric circuits.
2. Analyze transient and steady state response of DC circuits.
3. Signify AC quantities through phasor and compute AC system behaviour during steady state.
4. Explain and analyse the behaviour of transformer.
5. Elucidate the principle and characteristics of DC motor and DC generator**.**

Text Books:

1. *Hughes, E., Smith, I.M., Hiley, J. and Brown, K., Electrical and Electronic Technology, Prentice Hall(2008).*
2. *Nagrath, I.J. and Kothari, D.P., Basic Electrical Engineering, Tata McGraw Hill(2002).*
3. *Naidu, M.S. and Kamashaiah, S., Introduction to Electrical Engineering, Tata McGraw Hill(2007).*

Reference Books:

1. *Chakraborti, A., Basic Electrical Engineering, Tata McGraw□Hill(2008).*
2. *Del Toro, V., Electrical Engineering Fundamentals,* Prentice□Hall of India Private Limited*(2004)*

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	25
2	EST	35
3	Sessional (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	40

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 projection system
2. Isometric Projections
3. Auxiliary Projections
4. Perspective Projections
5. Introduction to Mechanical Drawing
6. Sketching engineering objects
7. Sections, dimensions and tolerances

AutoCAD

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

Micro Projects /Assignments:

1. Completing the views - Identification and drawing of missing lines in the projection of objects
2. Missing views – using two views to draw the projection of the object in the third view, primarily restricting to Elevation, Plan and Profile views
3. Projects related to orthographic and isometric projections
    - Using wax blocks or soap bars to develop three dimensional object from given orthographic projections
    - 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 combination of solids like Prism, Cone, Pyramid, Cylinder, Sphere etc.

5 To draw the detailed and assembly drawings of simple engineering objects/systems with due sectioning (where ever required) along with bill of materials.

e.g. Rivet joints, simple bearing, wooden joints, Two plates connected with nut and bolt etc.

Course Learning Outcomes (CLO):

Upon completion of this module, students will be able to:

1. creatively comprehend geometrical details of common engineering objects
2. draw dimensioned orthographic and isometric projections of simple engineering objects.
3. interpret the meaning and intent of toleranced dimensions and geometric tolerance symbolism;
4. create the engineering drawings for simple engineering objects using AutoCAD
5. manage screen menus and commands using AutoCAD
6. operate data entry modes and define drawings geometrically in terms of Cartesian, polar and relative coordinates in AutoCAD
7. create and edit drawings making selections of objects, discriminating by layering and using entities, object snap modes, editing commands, angles and displacements using AutoCAD

Text Books:

1. *Jolhe, D.A., Engineering Drawing, Tata McGraw Hill,2008*
2. *Davies, B. L., Yarwood, A., Engineering Drawing and Computer Graphics,* Van 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, Dhanpat Rai Publishing Company (P) Ltd, Delhi (2002).*
4. *French, T. E., Vierck, C. J. and Foster, R. J., Fundamental of Engineering Drawing & Graphics Technology, McGraw Hill Book Company, New Delhi(1986).*
5. *Rowan, J. and Sidwell , E. H., Graphics for Engineers, Edward Arnold, London(1968).*

Evaluation Scheme:

Sr. No.	**Evaluation Elements**	**Weightage (%)**
1	Mid semester test (formal written test)	25
2	End semester test (formal written test)	40

3	Sessional: (may include the following) Continuous evaluation of drawing assignments in tutorial/ regular practice of AutoCAD tutorial exercises & Individual independent project work/drawing and AutoCAD assignment	35

UHU003: PROFESSIONAL COMMUNICATION

L	T	P	Cr
2	-	2	3

Course objective: 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.

Detailed Contents:

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 roleplays.
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 outcome (CLO):

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 R.V and Flately M.E., Basic Business Communication Skills for the Empowering the Internet Generation. Tata Mc Graw Hill. New Delhi(2006).*
2. *Raman,M& Sharma, S.,Technical Communication Principles and Practice, Oxford University Press NewDelhi.(2011).*
3. *Mukherjee H.S.,Business Communication-Connecting at Work,Oxford University Press New Delhi, (2013).*

Reference Books:

- *Butterfield, Jeff.,Soft Skills for everyone,Cengage Learning NewDelhi,(2013).*
- *Robbins, S.P., &Hunsaker, P.L.,Training in Interpersonal Skills,PrenticeHallof India New Delhi,(2008).*
- *DiSianza,J.J&Legge,N.J.,Business and PrfofessionalCommunication,Pearson Education India NewDelhi,(2009).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	25
2.	EST	35
3.	Sessionals (Group Discussions; professional presentations;paneldiscussions;publicspeaking;projects,quizzes)	40

UMA004: MATHEMATICS - II

L	T	P	Cr
3	1	0	3.5

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

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

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

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

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

Course Learning Outcomes: Upon completion of this course, the students 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, G.F., Differential Equations (With Applications and Historical Notes), Tata McGraw Hill(2009).
2. Krishnamurthy, V.K., Mainra, V.P. and Arora, J.L., An introduction to Linear Algebra, Affiliated East West Press(1976).

Reference Books:

1. Kreyszig Erwin, Advanced Engineering Mathematics, John Wiley (2006), 8thed.

2. Jain, R.K. and Iyenger, S.R.K , Advanced Engineering Mathematics, Narosa Publishing House(2011), 11thed.

Evaluation Scheme:

Sr.No.	Evaluation Elements	Weight age (%)
1.	MST	30
2.	EST	45
3.	Sessionals (May include assignments/quizzes)	25

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.

Breakup of lecture details to be taken up by MED:

Lec No.	**Topic**	**Contents**
Lec 1	Introduction	The Mangonel Project. History. Spreadsheet.
Lec 2	PROJECTILE MOTION	no DRAG, Design spread sheet simulator for it.
Lec 3	PROJECTILE MOTION	with DRAG, Design spread sheet simulator for it.
Lec 4	STRUCTURES FAILURE	STATIC LOADS
Lec 5	STRUCTURES FAILURE	DYNAMIC LOADS
Lec 6	REDESIGNING THE MANGONEL	Design constraints and limitations of materials for redesigning the Mangonel for competition as a group.
Lec 7	MANUFACTURING	Manufacturing and assembling the Mangonel.
Lec 8	SIMULATION IN ENGINEERING DESIGN	Simulation as an Analysis Tool in Engineering Design.
Lec 9	ROLE OF MODELLING & PROTOTYPING	The Role of Modelling in Engineering Design.

Breakup of lecture details to be taken up by ECED:

Lec No.	Topic	Contents
Lec 1-5	Digital Electronics	Prototype, Architecture, Using the Integrated Development Environment (IDE) to Prepare an Arduino Sketch, structuring an Arduino Program, Using Simple Primitive Types (Variables), Simple programming examples. Definition of a sensor and actuator.

Tutorial Assignment / Laboratory Work:

Associated Laboratory/Project Program: T- Mechanical Tutorial, L- Electronics Laboratory, W- Mechanical Workshop of “Mangonel” assembly, redesign, operation and reflection.

Title for the weekly work in 15 weeks	Code
Using a spread sheet to develop a simulator	T1
Dynamics of projectile launched by a Mangonel - No Drag	T2
Dynamics of projectile launched by a Mangonel - With Drag	T3
Design against failure under static actions	T4
Design against failure under dynamic actions	T5
Electronics hardware and Arduino controller	L1
Electronics hardware and Arduino controller	L2
Programming the Arduino Controller	L3
Programming the Arduino Controller	L4
Final project of sensors, electronics hardware and programmed Arduino controller based measurement of angular velocity of the “Mangonel” throwing arm.	L5
Assembly of the Mangonel by group	W1
Assembly of the Mangonel by group	W2
Innovative redesign of the Mangonel and its testing by group	W3
Innovative redesign of the Mangonel and its testing by group	W4
Final inter group competition to assess best redesign and understanding of the “Mangonel”.	W5

Project: The Project will facilitate the design, construction and analysis of a “Mangonel”. In addition to some introductory lectures, the content of the students’ work during the semester will consist of:

1. the assembly of a Mangonel from a Bill Of Materials (BOM), detailed engineering drawings of parts, assembly instructions, and few prefabricated parts ;
2. the development of a software tool to allow the trajectory of a “missile” to be studied as a function of various operating parameters in conditions of no-drag and drag due to air;
3. a structural analysis of certain key components of the Mangonel for static and dynamic stresses using values of material properties which will be experimentally determined;

4. the development of a micro-electronic system to allow the angular velocity of the throwing arm to be determined;
5. testing the Mangonel;
6. redesigning the throwing arm of the Mangonel to optimise for distance without compromising its structural integrity;
7. an inter-group competition at the end of the semester with evaluation of the group redesign strategies.

Course Learning Outcomes (CLO):

Upon completion of this module, students will be able to:

1. simulate trajectories of a mass with and without aerodynamic drag using a spreadsheet based software tool to allow trajectories be optimized;
2. perform a test to acquire an engineering material property of strength in bending and analyze the throwing arm of the "Mangonel" under conditions of static and dynamic loading;
3. develop and test software code to process sensor data;
4. design, construct and test an electronic hardware solution to process sensor data;
5. construct and operate a Roman catapult "Mangonel" using tools, materials and assembly instructions, in a group, for a competition;
6. operate and evaluate the innovative redesign of elements of the "Mangonel" for functional and structural performance;

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)*

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	-
2	EST	-
3	Sessional: (may include the following)	
	Mechanical Tutorial Assignments	30
	Electronics Hardware and software Practical work in Laboratory	30
	Assessment of Mechanical contents in Lectures and Tutorials and Electronics contents in Lectures and Practical.	10
	Project (Assembly of the "Mangonel", innovative redesign with reflection, prototype competition, Final Presentation and viva-voce	30

UES012: ENGINEERING MATERIALS

L	T	P	Cr
3	1	2	4.5

Prerequisite(s): None

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, Anelastic 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 eutectic phase diagram, Examples and applications of phase diagrams like Iron - Iron carbide phase diagram.

Electrical and magnetic materials: Conducting and resister 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 Outcomes: 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. W.D. Callister , Materials Science and Engineering; John Wiley & Sons, Singapore, 2002.
2. W.F. Smith, Principles of Materials Science and Engineering: An Introduction; Tata Mc-Graw Hill, 2008.
3. V. Raghavan, Introduction to Materials Science and Engineering; PHI, Delhi, 2005.

Reference Books:

1. S. O. Kasap, Principles of Electronic Engineering Materials; Tata Mc-Graw Hill, 2007.
2. L. H. Van Vlack, Elements of Material Science and Engineering; Thomas Press, India, 1998.
3. K. G. Budinski, Engineering Materials – Properties and selection, Prentince Hall India, 1996

UCS520: COMPUTER NETWORKS

L	T	P	Cr
3	0	2	4

Course objective: 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.

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

1. Conceptualise 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, B.A., Data communication and Networking, McGraw Hill (2006).*
2. *Tanenbaum , A.S., Computer Networks, Prentice Hall (2010).*

Reference Books:

1. *Kurose and Ross, Computer Networking: A Top Down Approach, Addison-Wesley, (2012).*
2. *Stallings, W., Computer Networking with Internet Protocols and Tech, Prentice Hall of India (2010).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/Tutorials/Quizzes/Lab Evaluations)	35

UCS406: DATA STRUCTURES AND ALGORITHMS (4 self effort hours)

L	T	P	Cr
3	0	4	6.0

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):

On completion of this course, the students 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. *Corman, Leiserson&Rivest, Introduction to Algorithms, MIT Press (2009).*
2. *Narasimha Karumanchi, Data Structures and Algorithms Made Easy (2014).*

Reference Books:

1. *Sahni, Sartaj, Data Structures, Algorithms and Applications in C++, Universities Press (2005).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS407: Inventions & Innovations in Computing

L	T	P	Cr
2	0	0	2

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 ligthing and smart traffic management, use of technology and open data, Interpreting Technology Hype, five key phases of a technology's life cycle.

Course Learning Outcome

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

References:

1. Elizabeth Raum, The History of the Computer (Inventions That Changed the World), 2007.
2. Chris Woodford, Communication and Computers (History of Invention), 2004.
3. Ahmad, Ishfaq , Ranka, Sanjay, Handbook of energy - aware and green computing, 2012.
4. Fortino, Giancarlo, Internet of things based on smart objects: technology, middleware and applications Smart City, 2014.
5. Salvi, Dilip M., Inventions that made history, 1990.
6. http://www.hongkiat.com/blog/computer-programming-greatest-inventions/
7. http://www.gartner.com/technology/
8. http://www.forbes.com/technology
9. https://www.top500.org/

Course evaluation scheme (MM: 100):

The breakup for the marks is shown in four activities as under.

1. Two Quizzes : 20 Marks (10 + 10)
2. Case studies as assignment : 40 Marks (10 + 10 + 10 +10)
3. Presentation: 20 Marks
4. Poster presentation: 20 Marks

UCS303: OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course objective: 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, reentrancy).

Scheduling and Dispatch: Dispatching and context switching, Preemptive and non-preemptive 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, nonsequential, 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):
On completion of this course, the students will be able to

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

2. Distinguish 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. Examine and categorise various memory management techniques like caching, paging, segmentation, virtual memory, and thrashing.
5. Appraise high-level 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, P.B. and Gagne, G., Operating System Concepts, John Wiley (2013).*
2. *Stallings, Willam, Operating Systems Internals and Design Principles, Prentice Hall (2014).*

Reference Books:

1. *Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Ed., O'Reilly Media, November(2005).*
2. *Michael Kifer, Scott Smolka, Introduction to Operating System Design and Implementation: The OSP 2 Approach, Springer(2007).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals(Assignments/Projects/Tutorials/Quizzes/Lab Evaluations)	35

UTA014: ENGINEERING DESIGN PROJECT-II

(Includes project with 6 self 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).*

Evaluation Scheme:

SNo.	Evaluation Elements	Weightage (%)
1.	Mid Semester evaluation 1	20
2.	Mid Semester evaluation 2	20
3.	Mid Semester evaluation 3	20
4.	End Semester Evaluation	40

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 Dye 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 Outcomes:

The students will be able to

1. Identify and understand the basic manufacturing processes like single and multipoint machining, forming, welding, casting etc.
2. Acquire basic operational skills in different manufacturing processes like machining, forming, welding, casting, sheet metal operations, pattern making etc.

Text Books

1. *Degarmo, E. P., Kohser, R. A. and Black, J. T., Materials and Processes in Manufacturing, Prentice Hall of India (2002).*
2. *Kalpakjian, S. and Schmid, S. R., Manufacturing Processes for Engineering Materials, Pearson Education Asia (2000).*

Reference Books

1. *Chapman, W. A. J. , Workshop Technology, Vol.1 & II, Arnold Publishers (2001).*
2. *Zimmer E. W. and Groover, M. P., Computer Aided Designing and Manufacturing, Prentice Hall of India (2008).*
3. *Pandey, P. C. and Shan, H. S., Modern Machining Processes, Tata McGraw Hill (2004).*
4. *Mishra, P. K., Non Conventional Machining, Narosa Publications (2006).*
5. *Campbell, J. S., Principles of Manufacturing, Materials and Processes, Tata McGraw Hill Company (1995).*
6. *Lindberg, A. R., Process and Materials of Manufacture, Prentice Hall of India (1998).*

UMA031: OPTIMIZATION TECHNIQUES

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

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

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

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

Integer Programming: Branch and bound technique.

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 outcome: Upon Completion of this course, the students would 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) to solve two person zero-sum games

Text Books:

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

Recommended Books:

1) Pant J. C., Introduction to optimization: Operations Research, Jain Brothers (2004)
2) Bazaarra Mokhtar S., Jarvis John J. and Shirali Hanif D., Linear Programming and Network flows, John Wiley and Sons (1990)
3) Swarup, K., Gupta, P. K., Mammohan, Operations Research, Sultan Chand & Sons, (2010).

Evaluation Scheme:

Sr.No.	Evaluation Elements	Weight age (%)
1.	MST	30
2.	EST	45
3.	Sessionals (May include assignments/quizzes)	25

UES011: THERMO-FLUIDS

L	T	P	Cr
3	1	2	4.5

Course Objective

To understand basic concepts of fluid flow and thermodynamics and their applications in solving engineering problems

Fluid Mechanics

- **Introduction**: Definition of a fluid and its properties
- **Hydrostatics**: Measurement of pressure, thrust on submerged surfaces
- **Principles of Fluid Motion**: Description of fluid flow; continuity equation; Euler and Bernoulli equations; Pitot total head and static tubes, venturi-meter, orifice-meter, rotameter; Momentum equation and its applications
- **Pipe Flow**: Fully developed flow; laminar pipe flow; turbulent pipe flow, major and minor losses; Hydraulic gradient line (HGL) and total energy line (TEL)
- **Boundary Layer**: Boundary layer profile; displacement, momentum and energy thickness

Thermodynamics

- **Introduction**: Properties of matter, the state postulate, energy, processes and thermodynamic systems;
- **Properties of Pure Substances**: property tables, property diagrams, phase change, equations of state (ideal gas);
- **Energy**: Energy transfer by heat, work and mass;
- **First Law of Thermodynamics**: Closed system, open system, steady-flow engineering devices;
- **Second Law of Thermodynamics**: Statements of the Second Law, heat engines, refrigeration devices, reversible versus irreversible processes, the Carnot cycle.

Laboratory/Project programme

List of Experiments

1. Verification of Bernoulli's theorem
2. Determination of hydrostatic force and its location on a vertically immersed surface
3. Determination of friction factor for pipes of different materials
4. Determination of loss coefficients for various pipe fittings
5. Verification of momentum equation
6. Visualization of laminar and turbulent flow, and rotameter
7. Calibration of a venturi-meter
8. Boundary layer over a flat plate

Sample List of Micro-Projects

Students in a group of 4/5 members will be assigned a micro project.

1. Design a physical system to demonstrate the applicability of Bernoulli's equation
2. Determine the pressure distribution around the airfoil body with the help of wind tunnel
3. Demonstrate the first law of thermodynamics for an open system, for example: a ordinary hair dryer
4. Develop a computer program for solving pipe flow network.

Course Learning Outcomes (CLO):

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

1. analyze and solve problems of simple fluid based engineering systems including pressures and forces on submerged surfaces
2. analyze fluid flow problems with the application of the mass, momentum and energy equations
3. evaluate practical problems associated with pipe flow systems
4. conceptualize and describe practical flow systems such as boundary layers and their importance in engineering analysis
5. estimate fluid properties and solve basic problems using property tables, property diagrams and equations of state
6. analyze and solve problems related to closed systems and steady-flow devices by applying the conservation of energy principle
7. analyze the second law of thermodynamics for various systems and to evaluate the performance of heat engines, refrigerators and heat pumps.

Textbooks

1. *Kumar, D. S, Fluid Mechanics and Fluid Power Engineering, S. K. Kataria (2009)*
2. *Cengel and Boles, Thermodynamics: an Engineering Approach, McGraw-Hill (2011)*

Reference Books

1. *Jain, A. K. , Fluid Mechanics: including Hydraulic Machines, Khanna Publishers (2003)*
2. *Rao, Y.V. C, An Introduction to Thermodynamics, Universities Press (2004*

Evaluation Scheme:

S. No.	Evaluation Elements	Weightage (%)
1	MST	25
2	EST	40
3	Sessional (may be tutorials/ quizzes/ assignments/lab/ project)	35

UCS310: DATABASE MANAGEMENT SYSTEM

L	T	P	Cr
3	0	2	4.0

Course objective: 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, 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):

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

1. Analyze the Information Systems as socio-technical systems, its need and advantages as compared to traditional file based systems.
2. Comprehend architecture of DBMS, conceptual data modelling, logical database design and physical database design.
3. Analyze Database design using E-R data model by identifying entities, attributes, relationships, generalization and specialization along with relational algebra.
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 with usage of DDL aspect of SQL, DML aspect of SQL, aggregate functions, group by clause, sub query, joins, co-related sub query and indexes, cursor, stored function and procedure, triggers etc.

Text Books:

1. *H. F. Korth & Silverschatz, A., Database System Concepts, Tata McGraw Hill (2010).*
2. *Elmasri & Navathe, Fundamentals of Database Systems, Addison-Wesley (2011)..*

Reference Books:

1. *Hoffer, Prescott, Mcfadden, Modern Database Management, Paperback International (2012).*
2. *Martin Gruber, Understanding SQL, BPB Publication (1994).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1.	MST	20
2.	EST	45
3.	Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations)	35

UCS404: DISCRETE MATHEMATICAL STRUCTURES

L	T	P	Cr
3	1	0	3.5

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

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

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

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

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

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

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

Course learning outcomes (CLOs):

On completion of this course, the students 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, K.H., Discrete Mathematics and its Applications, McGraw Hill (2011).*
2. *Tremblay, J.P. and Manohar R., Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill (2007).*

Reference Books:

1. *Haggard G., Schlipf J. and Whitesides, Sue, Discrete Mathematics for Computer Science, Cengage Learning, (2008).*
2. *Johnsonbaugh R., Discrete Mathematics, Pearson Education, (2007).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	30
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	25

UCS616: ADVANCED DATA STRUCTURES AND ALGORITHMS

L	T	P	Cr
3	0	2	4.0

Course Objective: 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 (CLO):

On completion of this course, the students 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. Thomas Coremen, Introduction to Algorithms, PHI (2009).

2. David E. Goldberg, Genetic Algorithm, Pearson education (2005).

3. Roger Sedgewick and Kevin Wayne , Algorithms, Addison-Wesley Professional(2011) .

Reference Books:

1. *Sahni, Sartaj, Data Structures, Algorithms and Applications in C++, MIT Press (2005).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS521: ARTIFICIAL INTELLIGENCE

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

Course objective: 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):

On completion of this course, the students 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. Design of programs in AI language(s).
5. Acquire knowledge about the architecture of an expert system and design new expert systems for real life applications.

Text Books:

1. *Rich E., Artificial Intelligence, Tata McGraw Hills (2009).*
2. *George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia (2009).*

Reference Books:

1. *Patterson D.W, Introduction to AI and Expert Systems, Mc GrawHill (1998).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS507: COMPUTER ARCHITECTURE AND ORGANIZATION

L	T	P	Cr
3	0	2	4.0

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

1. Illustrate various elementary concepts of computer architecture including, syntax of register transfer language, micro operations, instruction cycle, and control unit.
2. Describe the design of basic computer with instruction formats & addressing modes

3. Explore various memory management techniques and algorithms for performing addition, subtraction and division etc.
4. Interpret the concepts of pipelining, multiprocessors, and inter processor communication.

Text Books:

1. *Mano, Morris M., Computer System Architectue, Prentice Hall (1992).*
2. *Hayes, J.P., Computer Architecture and Organization, McGraw Hill (1998).*

Reference Books:

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

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS503: SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course objective: 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 behavioural modeling, 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):
On completion of this course, the students 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 R., Software Engineering, A Practitioner's Approach, McGraw Hill International (2014).*
2. *Sommerville I., Software Engineering, Addison-Wesley Publishing Company (2010).*

Reference Books:

1. *Jalote P., An integrated Approach to Software Engineering, Narosa(2005).*
2. *Booch G.,RambaughJ.,JacobsonI.,The Unified Modeling Language User Guide (2005).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS701: THEORY OF COMPUTATION

L	T	P	Cr
3	1	0	3.5

Course objective: 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):

On completion of this course, the students 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 mastering 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 J. E., Ullman J. D. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education (2006).*

2. *John C. Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011).*

Reference Books:

1. *Daniel A. Cohen, Introduction to Computer Theory, John Wiley and Sons (1996).*
2. *Michael Sipser, Introduction to the Theory of Computation, Thomson (2007).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	30
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	25

UCS614: EMBEDDED SYSTEMS DESIGN

L	T	P	Cr
3	0	2	4.0

Course Objective: 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, Ad Cs 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, Qualities of 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 and sequential digital circuits using Modelsim & Xilinx – Verilog language. To design and simulate the operations of systems like verilog using Modelsim & Toggle, Bitwise, Delay and any Control Logic Design in 8051.

Course Learning Outcomes (CLO):

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

1. To comprehend the need and usage of Embedded System.
2. To distinguish a Real Time Embedded System from other systems
3. To explain the kind of memory and processor.
4. To learn 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 , Example of interfacing memory to the ports of 8051
5. To define what is a field programmable gate array (FPGA) and the use of it.
6. To understand the Internet of Things.

Text Book: *Lyla B. Das, Embedded Systems: An Integrated Approach , Pearson*

Reference Book: *Raj Kamal, Embedded Systems Architecture, Programming and Design, Tata Mcgraw Hill, New Delhi.*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS615: IMAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objective: 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.

Course Learning Outcomes (CLO):

On completion of this course, the students 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 model.
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 interactive image processing applications.

Text Books:

1. Rafeal C.Gonzalez, Richard E.Woods, Digital Image Processing, Third Edition, Pearson Education/PHI.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, Second Edition, Thomson Learning.

Reference Books:

1. Alasdair McAndrew, Introduction to Digital Image Processing with Matlab, Thomson Course Technology.
2. Adrian Low, Computer Vision and Image Processing, Second Edition, B.S.Publications.
3. *Rafeal C.Gonzalez, Richard E.Woods, Steven L. Eddins, Digital Image Proc*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS617: MICROPROCESSOR-BASED SYSTEMS DESIGN

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

Course objective: 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):

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

1. Acquired knowledge about Microprocessors and its need.

2. Foster ability to write the programming using 8086 microprocessor
3. Foster ability to understand the internal architecture and interfacing of different peripheral devices with 8086 microprocessor.
4. Foster ability to write the programming using ARM processors.
5. Foster ability to understand the internal architecture and interfacing of different peripheral devices with 8086 and ARM processors.

Text Books

1. *Gaonkar, Ramesh., Microprocessor Architecture, Programming and Applications with the 8085, Penram International Publishing India PVT.LTD. (2005).*
2. *Hall, D.V., Microprocessor and Interfacing, Tata McGraw Hill Publishing Company (2006).*
3. *Steve Furber, ARM System on chip Architecture Addison Wesley (2000).*

Reference Books

1. *Gibson, Glenn A., Liu, Yu-Cheng., Microcomputer Systems: The 8086/8088 Family Architecture Programming And Design, Pearson (2001).*
2. *Andrew N. Sloss, ARM System Developer's Guide, Morgan Kaufmann publications (2004).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS793: CAPSTONE PROJECT

L	T	P	Cr
0	**0**	**2**	**8.0**

Course objective: The objective of the capstone project is to give a student the opportunity toweave together the interdisciplinary elements of their curricula into an integrated project.

Course learning outcomes (CLOs):

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

1. Develop skills necessary for structuring, managing, and carrying out projects within an organization/industry.
2. Design, develop, debug, document, and deliver a software 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 the extent of their interdisciplinary work over the course of their academic career considering issues of professionalism and ethics.

UCS802: COMPILER CONSTRUCTION

L	T	P	Cr
3	0	2	4.0

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

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

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

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

Error Analysis: Introduction to error analysis, detection, reporting and recovery from compilation errors, Classification of error-lexical, syntactic and semantic 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 colouring in optimization, Live ranges of run time values

Laboratory work: Construct a lexical analyzer using Flex. Construct a parser using PrisonBison. Build simple compilers from parsing to intermediate representation to code generation and simple optimization.

Course learning outcomes (CLOs):

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

1. Design and construction of compilers and knowledge of working of major phases of compilation.
2. Construct parsers.
3. Implement a simple compiler for a language chosen.
4. Classify various parameters passing scheme, explain memory management of a programming languages and perform code optimization.

Text Books:

1. *Aho A.V., Ullman J. D., Sethi R., Compilers Principles, Techniques and Tools, Pearson Education (2005).*
2. *John Levine, Tony Mason, Doug Brown, Lex and Yacc, O'Reilly (1992).*

Reference Books:

1. *Kenneth C. Louden, Compiler Construction and Practices, Thomson Publication (1997).*
2. *Dhamdhere, Compiler Construction, Macmillan Publication (2008).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UHU005: HUMANITIES FOR ENGINEERS

L	T	P	Cr
2	0	2	3

Course Objectives: The objective of the course is to understand the interplay between, psychological, ethical and economic principles in governing human behavior. The course is designed to help the students to understand the basic principles underlying economic behavior, to acquaint students with the major perspectives in psychology to understand human mind and behavior and to provide an understanding about the how ethical principles and values serve as a guide to behavior on a personal level and within professions.

UNIT I: 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 behavior 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 behavior.

UNIT II: 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.

UNIT III: 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 (CLO):

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

1. Improve the understanding of human behavior 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. (International Student (1986).

*2. A. N. Tripathi****,*** *Human Values****,*** *New Age International (P) Ltd (2009).*

3. Krugman, Paul and Wells Robin, Economics, W.H. Freeman & Co Ltd. Fourth Edition (2015).

4. RubinfeldPindyck. Microeconomic Theory and application, Pearson Education New Delhi (2012).

5. Samuelson, Paul, A. and Nordhaus, William, D. Economics, McGraw Hill, (2009).

6. Mankiw, Gregory N. Principles of Macroeconomics, South-Western College Pub., (2014).

7. Gregory, Paul R. and Stuart, Robert C. The Global Economy and Its Economic Systems, 2013 South-Western College Pub (2013).

Reference Books:

1. *Atkinson, R.L., Atkinson, R.C., Smith, E.E., Bem, D.J. and Nolen-Hoeksema, S. (2000). Hilgard's Introduction to Psychology, New York: Harcourt College Publishers.*

2. *Berne, Eric (1964). Games People Play – The Basic Hand Book of Transactional Analysis. New York: Ballantine Books.*

3. *Ferrell, O. C and Ferrell, John Fraedrich Business Ethics: Ethical Decision Making & Cases, Cengage Learning (2014).*

2 Duane P. Schultz and Sydney Ellen Schultz, Theories of Personality, Cengage Learning, (2008).

3 Saleem Shaikh. Business Environment, Pearson (2007).

4 Chernilam, Francis International Buisness-Text and Cases, Prentice Hall (2013).

5 Salvatore, Dominick, Srivastav, Rakesh., Managerial Economics: Principles with Worldwide Applications, Oxford, 2012.

6 Peterson H. Craig. and. Lewis, W. Cris. Managerial Economics, Macmillan Pub Co; (1990).

UCS781: INDEPENDENT STUDY

Course Learning Outcomes:

1. Develop and refine skills in analysis, research, organization and communication.
2. Delve deeply into a subject of intense personal interest.
3. Prepare for professional accreditation by some external body or enhance a carrier you're already in.

UCS896: CAPSTONE PROJECT-II

L	T	P	Cr
0	**0**	**6**	**16.0**

Course objective: The objective of the capstone project is to give a student the prospect to intertwine together the interdisciplinary fundamentals of their curricula into an integrated project.

Course learning outcomes (CLOs):

On completion of this course, the students 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

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, Session Hijacking Tools.

Hacking Wireless Networks: Introduction to 802.1I,Role of WE?, 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, IPsec can, SuperScan etc. and *Hacking Tools likes Trinoo, TFN2K,* Zombic,Zapper

Recommended Books

I. Network Security and Ethical Hacking Rajat Aare, Luniver Press. 30-Nor-2006

2. Network intrusion alert cm ethical hacking guide to intrusion detection, Ankit Podia,

Menu Zacharia, Thomson Course Technology PTR, 12-Jun-2007

3.Ethical !lacking, Thomas Mathew ,0571 Publisher, 28-Nor-2003

4. Hacking Exposed: Network Security Secrets & Solutions, Stuart McClure, Joel SeatnbraV and George Kurtz, McGraw-llill, 2005

Course learning outcomes (CLOs):

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

1. Apply knowledge into an interactive environment where they are shown how to scan, test, hack and secure their own systems.
2. Remember in-depth knowledge and practical experience with the current essential security systems.
3. Understand how perimeter defenses work and then be led into scanning and attacking their own networks, no real network is harmed.
4. Evaluate how intruders escalate privileges and what steps can be taken to secure a system.
5. Analyze Intrusion Detection, Policy Creation, Social Engineering, DDoS Attacks, Buffer Overflows and Virus Creation.

UCS801: SOFTWARE PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

Course objective: Learn and Explore SPM activities through knowledge of software project management 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 & Planning: System view of project management, Understanding organizations, stakeholder's management, project phases & project 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 organisational charts, responsibility assignment metrics, acquiring project team, resource assignment, resource loading, resource levelling, Different team structures developing project team

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):

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

1. Describe and apply basic concepts related to software project planning, scope and feasibility.
2. Analyze various project estimation techniques, especially size estimation (FP), effort estimation (COCOMO models), schedule estimation (GANTT charts), and cost estimation.
3. Illustrate the concept of team structure and project communication management.
4. Acquire knowledge about quality assurance, quality control, and risk management.
5. Describe various project management activities such as tracking, project procurement, configuration management, monitoring.

Text Books:

1. Bob Hughes and Mike Cotterell, Software Project Management, Tata McGraw Hill (2009).

2. Roger Pressman, A practitioner's Guide to Software Engineering, Tata McGraw Hill(2014)

.

Reference Books:

2. *Andrew Stellman; Jennifer Greene,Applied Software Project Management,O'Reilly Media, Inc. (2005).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS895: PROJECT SEMESTER

L	T	P	Cr
-	-	-	20.0

Course objective: The objective of the project semester is to make the students solve real world problems using automated solutions, while developing management and writing skills amongst them.

Course learning outcomes (CLOs):

On completion of this course, the students 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.

UCS608: PARALLEL AND DISTRIBUTED COMPUTING

L	T	P	Cr
3	0	2	4.0

Course objective: 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):
On completion of this course, the students 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. *C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley (2008).*
2. *A Grama, A Gupra, G Karypis, V Kumar. Introduction to Parallel Computing, Addison Wesley (2003).*

Reference Books:

1. *B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing With Opencl. Morgan Kaufmann and Elsevier (2011).*
2. *T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley (2004).*
3. *Quinn, M. J.,Parallel Programming in C with MPI and OpenMP, McGraw-Hill (2004).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS523: COMPUTER AND NETWORK SECURITY

L	T	P	Cr
3	0	2	4.0

Course objective: 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):
On completion of this course, the students 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 IDS

Text Books:

1. *Network Security Essentials, William Stallings, Prentice Hall (2013).*

Reference Books:

1. *Firewalls and Internet Security, William R. Cheswick and Steven M. Bellovin, Addison-Wesley Professional (2003).*
2. *Cryptography and Network Security, W. Stallings, Prentice Hall (2010).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS522: COMPUTER VISION

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

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

1. Understand the fundamental problems of computer vision.
2. Analyze techniques, mathematical concepts and algorithms used in computer vision to facilitate further study in this area.
3. Implement different concepts and techniques covered in the course.
4. Utilize programming and scientific tools for relevant software implementation.

Text Books:

1. *Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.*
2. *Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.*

Reference Books:

1. *Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.*
2. *K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.*
3. *R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UML501: MACHINE LEARNING

L	T	P	Cr
3	0	2	4.0

Course objective: 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):

On completion of this course, the students 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 perfect domain theories.
5. Critically evaluate and compare different learning models and learning algorithms and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.

Text Books:

1. *Mitchell T.M., Machine Learning, McGraw Hill (1997).*
2. *Alpaydin E., Introduction to Machine Learning, MIT Press (2010).*

Reference Books:

1. *Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006).*
2. *Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS524: ENGINEERING SOFTWARE AS A SERVICE

L	T	P	Cr
3	0	2	4.0

Course Objective

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

Course Learning Outcomes (CLO):

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

1. Explain the Agile Software Development concepts, Software as a Cloud Service and SaaS architecture
2. Synthesis a SaaS Application using model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages using Ruby on Rails
3. Design SaaS Client Framework using Java Script
3. Demonstrate the use of Behavior Driven Design (BDD) and User Stories for analyzing the requirements and designing the solution of Web Service
4. Use of Test Driven Design (TDD) approach for testing the service from Plan and Development agile perspective

Text Book

1. "Engineering Software as a Service: An Agile Approach Using Cloud Computing" by David Patterson, Armando Fox

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS643: CYBER FORENSICS

L	T	P	Cr
3	0	2	4.0

Course objective: 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

Lab 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):

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

1 Familiarize with Cyber Crime & Forensics Ontology
2 Analyse & Demonstrate the Crime Scene and Criminology.
3 Redesign the crime scene using Digital Investigation Process
4 Preparing and documenting the evidence for Judicial proceedings.

Text Books:

1 *Scene of the Cybercrime, Debra Littlejohn Shinder, Michael Cross, Syngress, 2nd Edition, 2008*
2 *Cyber Forensics: from Data to Digital Evidence, Albert J. Marcella Jr., Wiley,1stEdition,2012*

Reference Books:

1. *Computer Forensics: Investigating Network Intrusions and Cyber Crime, EC Council Press Series, Cengage Learning, 2010.*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS642: AUGMENTED AND VIRTUAL REALITY

L	T	P	Cr
3	0	2	4.0

Course Objective: 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, Virtoolsetc

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 (CLO):

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

1. Understand the processes involved in the creation of 3D animation and how to balance the interaction of vision, budget, and time constraints within productions.

2. Evaluate diverse methods available for achieving similar results and the decision making processes involved at various stages of project development.

3. Analyze the differences among various animation tools.

Text Books:

1. *Bowman, Doug A.; Kruijff, Ernst; LaViola Jr., Joseph J.; Poupyrev, Ivan* ***, 3D User Interfaces: Theory and Practice*** *, Addison-Wesley , 2005 , ISBN:0201758679.*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UML602: NATURAL LANGUAGE PROCESSING

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

Course Objective: 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 (CLO):

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

1. Comprehend the concept of natural language processing, its challenges and applications.
2. Comprehend the concepts word forms of the language by considering the concept of morphology analysis.
3. Ability to perform syntax and semantics in natural language processing.
4. Ability to design and analyze various NLP algorithms.
5. Acquire knowledge of machine learning techniques used in NLP, including hidden markov models, N-Grams and probabilistic context-free grammar.

Text Books:

1. *Jurafsky, D. and Martin J. H., Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Upper Saddle River, NJ: Prentice-Hall, 2000. ISBN: 0130950696.*

2. *Manning, Christopher D., and HinrichSchütze. Foundations of Statistical Natural Language Processing, Cambridge, MA: MIT Press, 1999. ISBN: 0262133601.*

Reference Books:

1. Dale R., Moisl H. and Somers H., Handbook of Natural Language Processing, MARCEL DEKKER, Inc., 2009.
2. Bird, S.K. Ewan and Loper E., Natural Language Processing with Python, Oreilly Publication, 2009.

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (May include Assignments/Projects/Tutorials/Quizes/Lab Evaluations)	35

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.

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

Recommended Books

2. Practical Guide to Software Quality Management (Artech House Computing Library) 2nd edition, 2003

3. Quality Software Management, Volume 1: Systems Thinking, 2011, Dorset House Publishing
4. Metrics and Models in Software Quality Engineering 2 Edition, Pearson, 2003.
5. Applied Software Measurement by Capers Jones, Tata McGraw Hill, 2008 3rded

Course Learning Outcomes (CLO):

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

1 Comprehend the basic knowledge of Software quality models
2 Classify the quality measurement aspects and metrics.
3 Analyse the control, reliability and management of quality process.
4 Analyse Complexity metrics, Customer Satisfaction and International quality standards like ISO, CMM
5 Evaluate the project and processes, configuration management on the basis of collected metrics.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS631: GPU COMPUTING

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

Course Objective: To study architecture and capabilities of modern GPUs ans 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 (CLO):

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

1. Define terminology commonly used in parallel computing, such as ***efficiency*** and ***speedup***.
2. Describe common GPU architectures and programming models.
3. Implement efficient algorithms for common application kernels, such as matrix multiplication.
4. Given a problem, develop an efficient parallel algorithm to solve it.
5. Given a problem, implement an efficient and correct code to solve it, analyze its performance, and give convincing written and oral presentations explaining your achievements.

Text Books:

1. J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General- Purpose GPU Programming, nvidia, (2011).

2. David B. Kirk, Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, (2010).

Reference Books:

1. ***Wen-mei W. Hwu****, AGPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann, (2011).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/Tutorials/Quizzes/Lab Evaluations)	35

UCS632: 3D MODELLING & ANIMATION

L	T	P	Cr
3	0	2	4.0

Course Objective: To develop the skill & knowledge in 3D Modeling & Animation. Students will understand the knowhow 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, HDR IIllumination, Fine-tuning shadows, Final render setting etc.

Course learning outcome (CLO):

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

1. Define Computer-based Animation & Getting Started with Max.
2. 2D Splines, Shapes & compound object.
3. 3D Modeling, Keyframe Animation, Simulation & 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. *NewRiders, "3dsmax7 Fundamentals", BPB, 2005.*
2. *Isaac Kerlow, "The Art of 3D Computer Animation and Effects", 4th edition, Wiley, 2009.*

Reference Books:

1. *Lance Flavell., "Beginning Blender: Open Source 3D Modelling, Animation, and Game Design", 1st edition, Apress, 2010.*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS633: DATA ANALYTICS AND VISUALIZATION

L	T	P	Cr
3	0	2	4.0

Course Objective: 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 techniquessuch as classification clustering on real world problems using R.

Course Learning Outcomes (CLO):

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

1. To analyse the need and usage of analyticsand visualization techniques.
2. To implement how tomanage, manipulate, cleanse and analyse data.
3. To Implement various data clustering and classification approaches in R.
4. To demonstrate the use of R on real life problem.

Text Books:

1.Jiawei Han, Micheline Kamber, Jian Pei , Data Mining Concepts and Techniques, (3rd Ed.),Morgan Kaufmann

2.Roger D. Peng,RProgrammingforDataScience

Reference Books:

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

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS634: SECURE CODING

L T P Cr

3 0 2 4.0

Course Objective: 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, Authoritatite reply, revisit layer 2 and layer 3 of TCP/IP, DNS poisioning, ARP poisioning, C language obsfuscation. ARP poisioning 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.

Recommended Reading

1. *Writing Secure Code, Michael Howard and David LeBlanc, Microsoft Press*
2. *Buffer Overflow Attacks: Detect, Exploit, Prevent by Jason Deckard, Syngress*
3. *Threat Modeling, Frank Swiderski and Window Snyder,Microsoft Professional*

COURSE OUTCOMES (CLOs)

1. To implement ARP posioning attack and demonstrate countermeasure against these for different operating environments.
2. To implement DNS posioning attack and demonstrate authoritative reply in this context.
3. To implement PE Code injection and demonstrate control hijacking via EIP manipulation
4. To demonstrate skills needed to deal with common programming errors and develop secure applications.
5. To 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. To demonstrate SQL, XSS attack and suggest countermeasures for the same.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

USE601: 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

Recommended Books

1. Software Verification and Validation: An Engineering and Scientific Approach, Marcus S. Fisher, Springer, 2007
2. Foundations of Software Testing, Aditya P. Mathur, Pearson Education, 2008
3. Software Testing: Principles and Practices, Srinivasan Desikan, GopalaswamyRamesh, Pearson Education India,2006

COURSE OUTCOMES (COs)

1 Capable to comprehend the concepts related to theoretical foundations of testing and debugging.

2 Competent to know and demonstrate software verification and validation approaches and their applicability.

3 Proficient to formulate and generate test cases from specifications

4 Able to exemplify program mutation testing strategies using programming language.

5 Proficient to formulate and generate test cases from finite state machine model etc.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)

1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS641: CLOUD COMPUTING

L	T	P	Cr
3	0	2	4.0

Course Objective: 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):

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

1. To explain the basic concepts along with evolution and features of cloud computing.
2. To demonstrate the concept of existing cloud paradigms and platforms.
3. To explore the issues of cloud computing in addition with various cloud models.
4. To attain the knowledge of virtualization through virtualization technologies.

5. To interpret the concept of Map reduce framework using SQL and NO SQL databases.

Text Books:

1. *Raj Kumar Buyya, James Broberg, AndrezeiM.Goscinski, Cloud Computing: Principles and paradigms, MIT Press (2011).*
2. *Michael Miller, Cloud Computing, Que Publishing (2008).*

Reference Books:

1. *Cloud Computing: A practical Approach Anthony Velte, Toby Velte and Robert Elsenpeter by Tata McGrawHill (2009).*
2. *Judith Hurwitz, Robin Bllor, Marcia Kaufman, F Halper, Cloud Computing for dummies (2009).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCG731: GAME DESIGN & DEVELOPMENT

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

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

1. Discuss in detail, the basic concepts, requirements and processes of game development.
2. Discuss and explain the concepts, tools and techniques for development of multi-player games
3. Describe the audio-video development and production process associated with games.
4. Develop a simple demo/game using C++ and/or Unity 3D.

5. Implement some advanced real-world components relevant to games.

Text Books:

1. *Steve Rabin, Introduction to Game Development, Cengage Technology (2010).*
2. *Michael Dawson, Beginning C++ Through Game Programming, Cengage Learning(2010).*

Reference Books:

1. *Kelly C., Programming 2D Games, A K Peters/CRC Press(2012).*
2. *A. Thorn, Learn Unity for 2D Game Development, Apress, (2013).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS709: ADVANCED TOPICS IN SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course Objective: 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 (CLO):

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

1. Comprehend concepts of formal methods and apply mathematical notations for formal specification.

2. Recognize 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. *Roger S. Pressman, Software Engineering a Practitioners Approach, McGraw-Hill (2014).*

Reference Books:

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

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS741: SIMULATION AND MODELLING

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

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.

Lab 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.

Text Books

1. Payne James A. , Introduction to Simulation : Programming Techniques and Methods of Analysis, McGraw Hill International Editions, Computer Science services, New York.
2. Geoffrey G., System Simulation, Prentice Hall publication, 2nd Edition, 1978.

Reference Books

1. Narsingh D., Systems Simulation with Digital Computer, PHI Publication (EEE), 3rd Edition, 2004.
2. Banks J., CarsonJ. S., NelsonB. L., NicolD. M, Discrete Event system Simulation, Pearson Education, Asia, 4th Edition, 2007.

Course Outcomes: After the successful completion of the course, the students 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 how to uses these model in real-life applications.

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35

UCS743: ADVANCED COMPUTER NETWORKS

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

Course Objective: This course aims to provide advanced background on relevant computer networking topics to have a comprehensive and deep knowledge in computer networks.

Review of Computer Networks, Devices and the Internet: Internet, Network edge, Network core, Access Networks and Physical media, ISPs and Internet Backbones, Delay and Loss in Packet-Switched Networks, Networking and Internet - Foundation of Networking Protocols: 5-layer TCP/IP Model, 7-Layer OSI Model, Internet Protocols and Addressing. Multiplexers, Modems and Internet Access Devices, Switching and Routing Devices, Router Structure. The Link Layer and Local Area Networks-Link Layer, Introduction and Services, Error- Detection and Error-Correction techniques, Multiple Access Protocols, Link Layer Addressing, Ethernet, Interconnections: Hubs and Switches, PPP: The Point-to-Point Protocol, Link Virtualization

Data-link protocols: Ethernet, Token Ring and Wireless (802.11). Wireless Networks and Mobile IP :Infrastructure of Wireless Networks, Wireless LAN Technologies, IEEE 802.11 Wireless Standard, Cellular Networks, Mobile IP, Wireless Mesh Networks (WMNs), Multiple access schemes

Routing and Internetworking: Network–Layer Routing, Least-Cost-Path algorithms, Non-Least-Cost-Path algorithms, Intra-domain Routing Protocols, Inter-domain Routing Protocols, Congestion Control at Network Layer. Logical Addressing: IPv4 Addresses, IPv6 Addresses - Internet Protocol: Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6 – Multicasting Techniques and Protocols: Basic Definitions and Techniques, Intra-domain Multicast Protocols, Inter-domain Multicast Protocols, Node-Level Multicast algorithms

Transport and Application Layer Protocols: Client-Server and Peer-To-Peer Application Communication, Protocols on the transport layer, reliable communication. Routing packets through a LAN and WAN. Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, TCP Congestion Control. Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name System (DNS), P2P File Sharing, Socket Programming with TCP and UDP, Building a Simple Web Server

Laboratory Work: consists of creating simulated networks and passing packets through them using different routing techniques. It has different Lab Practical related to advanced computer networks.

Recommended Books

1. Computer Networking: A Top-Down Approach, James F. Kuros and Keith W. Ross, Pearson, 6th Edition,2012
2. A Practical Guide to Advanced Networking**,** Jeffrey S. Beasley and Piyasat Nilkaew, Pearson, 3rd Edition,2012
3. Computer Networks , Andrew S. Tanenbaum, David J. Wetherall, Prentice, 5th Edition,2010

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20

2	EST	45
3	Sessionals (Assignments/Projects/ Tutorials/Quizzes/Lab Evaluations)	35