B.Tech CSE

Syllabus

IIIT Nagpur

IIIT Nagpur

Scheme for B.Tech CSE

Year	Semester	Course Code	Course Name	Type	L	T	P	Credits
BS YEAR								
1ST	1ST	MAL 101	Mathematics-I	BS	3	1	0	4
1ST	1ST	BEL 102	Elements of Electrical Engineering	BS	3	0	2	4
1ST	1ST	BSL 101	Applied Sciences	BS	3	0	2	4
1ST	1ST	CSL 101	Computer Programming	DC	3	0	2	4
1ST	1ST	ECL 101	Analog Electronics	DC	3	0	2	4
1ST	1ST	SAP 101	Health, Sports & Safety		0	0	2	0
1ST	1ST	HUL 102	Environmental Studies	HU	2	0	0	2
Subtotal					**17**	**1**	**10**	**22**
1ST	2nd	MAL 102	Mathematics-II	BS	3	1	0	4
1ST	2nd	ECL 102	Digital Electronics	DC	3	0	2	4
1ST	2nd	CSL 102	Data Structures	DC	3	0	2	4
1ST	2nd	CSL 103	Application Programming	DC	3	0	2	4
1ST	2nd	HUL 101	Communication Skills	HU	2	0	2	3
1ST	2nd	BEL 101	Mechanics and Graphics	BS	3	0	2	4
Subtotal					**17**	**1**	**10**	**23**
Total								**45**
SECOND YEAR								
2nd	3rd	MAL 201	Mathematics-III(Statistics & Probability)	BS	3	1	0	4
2nd	3rd	CSL 201	Advanced Data Structures	DC	3	1	2	5
2nd	3rd	CSL 202	Introduction to Object Oriented Programming	DC	3	0	2	4
2nd	3rd	CSL 203	Computer System Organisation	DC	3	1	0	4
2nd	3rd	CSL 204	Discrete Maths and Graph Theory	DC	3	1	0	4
2nd	3rd	CSP 201	IT Workshop-I	DC	1	0	4	3
Subtotal					**16**	**4**	**8**	**24**
2nd	4th	CSL 205	Design and Analysis of Algorithms	DC	3	1	2	5
2nd	4th	CSL 206	Software Engineering	DC	3	1	0	4
2nd	4th	CSL 207	Operating Systems	DC	3	0	2	4
2nd	4th	CSL 208	Design Principles of Programming Languages	DC	3	0	2	4
2nd	4th	HUL 201	Human Values	HU	3	0	0	3
2nd	4th	CSP 202	IT Workshop-II	DC	1	0	4	3
Subtotal					**16**	**2**	**10**	**23**
Total								**47**

Year	Semester	Course Code	Course Name	Type	L	T	P	Credits
THIRD YEAR								
3rd	5th	CSL 301	Database Management Systems	DC	3	0	2	4
3rd	5th	HUL 301	Economics and Business Finance	HU	3	1	0	4
3rd	5th	CSL 302	Computer Networks	DC	3	0	2	4
3rd	5th	CSL 303	Theory of Computation	DC	3	1	0	4
3rd	5th		Elective-I	DE	3	1	0	4
3rd	5th	CSP 301	IT Workshop-III	DC	1	0	4	3
Subtotal					**16**	**3**	**8**	**23**
3rd	6th	CSL 304	Compilers	DC	3	0	2	4
3rd	6th	CSL 305	Cryptography and Network Security	DC	3	1	0	4
3rd	6th	HUL 302	Software Design Documentation	HU	3	0	0	3
3rd	6th		Elective-II	DE	3	0	2	4
3rd	6th		Elective –III	DE	3	0	2	4
		CSP 302	IT Workshop –IV	DC	1	0	4	3
Subtotal					**16**	**1**	**10**	**22**
Total								**45**
FINAL YEAR								
4th	7th	CSL 401	In-house Project	DE	0	0	4	8
4th	7th		Elective-V	DE	3	0	2	4
4th	7th		Elective-VI	DE	3	0	2	4
4th	7th		Elective-VII	DE	3	0	2	4
4th	7th		Elective-VIII	DE	3	0	2	4
4th	7th		Elective-IX	DE	3	0	0	3
Subtotal					**15**	**0**	**12**	**27**
4th	8th		Industry Internship Project	DE	0	0	4	6
Subtotal								**6**
Total								**33**
GRAND TOTAL								**170**

Type	Credits
BS	**24**
HU	**15**
DE	**45**
DC	**86**
TOTAL	**170**

List of Elective Courses

Course Name	Type	L	T	P	Credits
Soft Computing	DE	3	0	2	4
Artificial Intelligence	DE	3	0	2	4
Machine Learning	DE	3	0	2	4
Pattern Recognition	DE	3	0	2	4
Distributed Systems	DE	3	0	2	4
Real Time Systems	DE	3	0	2	4
Mobile Computing	DE	3	0	0	3
Wireless Networks	DE	3	0	0	3
Cloud Computing	DE	3	0	0	3
Software Architecture	DE	3	0	0	3
Software Project Management	DE	3	0	0	3
Software testing and Evaluation	DE	3	0	0	3
Digital Image Processing	DE	3	0	0	3
Signals and Systems	DE	3	0	0	3
Digital Signal Processing	DE	3	0	0	3
Advanced Computer Architecture	DE	3	0	0	3
Embedded System	DE	3	0	2	4
Computer Graphics	DE	3	0	0	3
Human Computer Interaction	DE	3	0	0	3
Natural Language Processing	DE	3	0	0	3
Randomized Algorithms	DE	3	0	2	4
Parallel Algorithms	DE	3	0	2	4
Data Mining and Warehousing	DE	3	0	2	4
Bioinformatics	DE	3	0	2	4
Information Retrieval	DE	3	0	2	4
Business Intelligence	DE	3	0	2	4
Advance Compilers	DE	3	0	0	3
Paradigms Programming Languages	DE	3	0	2	4
Operation Research	DE	3	1	0	4
Introduction to GIS	DE	3	0	0	3
Introduction to Remote Sensing	DE	3	0	0	3
Big Data and Analytics	DE	3	0	2	4

Course Code:	MAL 101	Course Title:	Mathematics-I			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	Nil	Type of Course	Basic Science			

Course Outcomes:

1) To understand importance of calculus infinite series and matrix theory.
2) Applications of calculus infinite series and matrices.
3) Derivation and application of theorems of matrices.

Course Contents:

Differential Calculus: Functions of single variable: Limit, continuity and differentiability. Mean value theorems: Rolle‟s theorem, Lagrange‟s theorem, Cauchy‟s theorem, Taylor‟s theorem with remainders, indeterminate forms, curvature, curve tracing.

Integral Calculus: Fundamental theorem of Integral calculus, mean value theorems, evaluation of definite integrals, Applications in Area, length, volumes and surface of solids of revolutions, Improper integrals: Beta and Gamma functions, differentiation under integral sign.

Infinite series: Sequences, Infinite series of real and complex numbers, Cauchy criterion, tests of convergence, absolute and conditional convergence, improper integrals, improper integrals depending on a parameter, uniform convergence, power series, radius of convergence.

Matrices: Rank of matrix, consistency of a system of equations, linear dependence and independence, linear and orthogonal transformations, Eigen values and eigen vectors, Cayley – Hamilton theorem, reduction to diagonal form, Hermitian and skew Hermitian matrices, Quadratic forms.

Text:

1. Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons
2. Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1, Vol. 2)

Reference:

1. Thomas, G.B. and Finney, R.L, Calculus and Analytic Geometry, Addison Wesley Longman
2. Michael D. Greenberg, Advanced Engineering Mathematics, Pearson Education Pvt. Ltd
3. Jain R.K., Iyengar S.R.K, Advanced Engineering Mathematics, Narosa Publishers

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1)

2)

Course Code:	BSL 101	Course Title:	Applied Sciences			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Nil	Type of Course	Basic Science			

Course Outcomes:

1) To understand the fundamentals of Quantum Mechanics

2) To understand the structure and properties of materials.

3) To know current trends and advances in NEMS and MEMS

Course Contents:

Quantum Mechanics-I: Dual nature of matter, de-Broglie Hypothesis, phase velocity and group velocity, their relations, wave function & its physical significance, probability density, Schrodinger‟s wave equation, eigen values & eigen functions, applications. Electronic conduction in solids: Drude-Lorentz Theory, Drift velocity, relaxation time, mean collision time, mean free path, Electrical conductivity, Quantum free electron theory, density of energy states, Fermi energy, thermionic emission.

Structure of materials, Properties of materials, Transforming materials, Structure and transformation of materials, Electronic properties of materials, Mechanical properties, Engineering applications of materials.

Current trends in Engineering. applications : Quantum information & quantum computing, evolution of quantum theory, quantum computer, nanoscale systems and nanotechnology, nanoscience and technology, composite materials, smart materials and structures, nano and micromechanical systems (NEMS and MEMS).

Text:

1. Resnick, Walker and Halliday, Fundamental of Physics, John Willey and Sons. Inc, 6th Edition, 2005.
2. Streetman B. G., Solid State Electronics, Prentice Hall India (2nd Edition) 1986.
3. Avadhanulu M. N. and P.G. Kshirsagar, A text Book of Engineering Physics, (7th Edition) 2004.
4. Dekkar A.J.; Electrical Engineering Materials; Prentice Hall og India Publication, 1992.
5. Kenneth Krane; Modern Physics; (2 nd Edition); John Wiley Eastern, 1998.
6. Pillai S. O., Solid State Physics, New Age International Publishers, 3 rd edition, 1999.

Reference:

1) John A. Pelesko, David H. Bernstein, “Modeling MEMS and NEMS” CRC Press, 2002

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1. To study the characteristics of Photocell and to determine the work function of the cathode material.

2. To calibrate an electromagnet and to study the dependence of Hall voltage on magnetic field and current through the sample.

3. To study the I/P, O/P and transfer characteristics and to determine „α‟ of transistor in common base mode.

4. To study the forward and reverse characteristics of semiconductor diode.

5. To determine the band-gap in a semiconductor using reverse biased p-n junction diode.

6. To determine e/m for an electron by Thomson‟s method.

7. To calibrate an audio frequency oscillator and to determine the unknown frequency and phase of RC network by using single trace CRO.

8. To determine the radius of curvature of a plano- convex lens using Newton‟s Rings.

9. To determine the wavelength of sodium vapour lamp by plane transmission grating.

<table>
<tr><td>Course Code:</td><td>BEL 101</td><td>Course Title:</td><td colspan="4">Mechanics and Graphics</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Core</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>2</td><td>4</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>Nil</td><td>Type of Course</td><td colspan="4">Basic Science</td></tr>
</table>

Course Outcomes:

1. To enable the students understand the basic concepts of mechanics such as force, equilibrium, moment etc and to analyze simple determinate structures like beam, truss and frame.
2. To impart and include proper understanding of the theory of projection. Improve the visualization skills.
3. To enable the students with various concepts like dimensioning, conventions and standards related to working drawing in order to become professionally efficient.
4. To impart the knowledge on understanding and drawing of simple residential/ office building.

Course Contents:

Use of various drawing instruments , Concept of scales, Representative factor and dimensioning, Orthographic projections of points , lines, plane on principle planes/ Profile plane/ Auxiliary planes. Projection of right regular solids inclined to both the planes. Projection of right regular solids inclined to both the planes. Drawing isometric views from orthographic projection orthographic views.

Principles of Vector representation of force system, Moment of a force about a point and about an axis; couple moment; reduction of a force system to a force – a couple Wrench , Free Body Diagram, Reactions at supports, Equilibrium of Planar (including friction) and Spatial force system,

Internal forces in member: Determination of variation of Axial force (Axial Force Diagram), Shear force (Shear Force Diagram), Bending moment (Bending Moment Diagram) and twisting moment (Torque diagram)

Concept of stress and strain: Normal and shear stress and strain, State of stress at a point, Stress strain curve, Hook‟s law, Modulus of elasticity, Poisson‟s ratio, Modulus of rigidity, Bulk modulus, Transformation of stress.

Text:

1. Singer F.L. and Andrew Pytel, Strength of Material, Harper and Row Publishers, New York.
2. Bhatt N.D. and Panchal V.M., Elementary Engineering Drawing, Charotar Publishing House, 43rd edition.

Reference:

1. Hibbler, Engineering Mechanics, Pearson Education, Asia Pvt Ltd.
2. Beer F.P. and Johnston E.R., Vector Mechanics for Engineers: Statics and Dynamics, Tata McGraw-Hill
3. Irving H. Shames, Engineering Mechanics: Static and Dynamics, Pearson Education, Asia Pvt Ltd.
4. Meriam J.L. and Kraige L.G., Engineering Mechanics, John Wiley and Sons.
5. Stephen Timoshenko, Strength of Materaials, Part -1, CBS Publishers and Distributors, New Delhi.
6. Popov E.P., Mechanics of deformable bodies, Prentice-Hall
7. Beer F.P. and Johnston E.R., Mechanics of materials, McGraw-Hill International

8. Jolhe Dhananjay ,Engineering Drawing with an introduction to AutoCAD, Tata McGraw Hill Publishing Co.Ltd., 1st edition.

9. BIS-SP-46-1988, Handbook BIS SP -46-1988, BIS

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1. Verification of equilibrium equation for coplanar forces.
2. Verification of Lami‟s theorem.
3. Verification of Law of parallelogram of forces.
4. Verification of Law of polygon of forces.
5. Verification of equilibrium equation for spatial forces.
6. Determination of coefficient of friction.
7. Analysis of truss (Analytical / Graphical method).
8. Determination of modulus of elasticity for copper wire.
9. Determination of modus of rigidity of material.
10. Flexural test on beam.

Projection of points and lines
Projections of planes
Projections of lines and planes using Auxiliary planes
Projections of solids
Isometric views

Course Code:	CSL 101	Course Title:	Computer Programming			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Nil	Type of Course	Core Engineering			

Course Outcomes:
1) To understand importance of calculus infinite series and matrix theory.
2) Applications of calculus infinite series and matrices.
3) Derivation and application of theorems of matrices.

Course Contents:

Introduction: Flow charts, data types and storage classes, scope of variables, arithmetic operators, assignment, conditional, arithmetic expressions, enumerated data types, decision making, branching, looping, Switch concept, function and parameter passing, recursive functions, macros.
Basic programming algorithms: Programs to illustrate basic language constructs in C like - Factorial, Sine/cosine and other mathematical series, Fibonacci series, calculating square-root of a number, calculating GCD of 2 integers (Euclid‟s method and otherwise), Calculating LCM of 2 integers and similar such programs.
Arrays and applications: Introduction to one dimensional and 2-D array with examples. Representing a polynomial using 1-D array and polynomial operations, Use of 2-D array to represent a matrix and matrix operations. Character arrays (strings): String related functions (strlen, strcpy, strcat, strcmp, atoi, itoa, reverse, strstr etc) and their function definitions.
Searching and Sorting methods: Selection sort, Bubble sort, Insertion sort, Linear search, merging of 2 sorted arrays.
Structures and Unions: Basic concept, array of structures and its applications.
Pointers: Introduction (declaration and initialization), pointers and arrays, concept of dynamic memory allocation, use of pointers to represent variable-sized 1-D and 2-D arrays, pointers to structures.
File Management in C: Open, close, read and write operations, Sequential and text files.

Text:
1. Kerninghan; Ritchie, “C programming Language”, PHI
2. Balguruswamy, “Programming in ANSI C”, Tata Mcgraw Hill Publishing

Reference:
3. Kakde and Deshpande, “C and data Structure”, Charles River Media Publisher
4. Dromey R G, “How to Solve it by Computer”, PHI
5. Y. Kanetkar, “Let us C ”.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1) Programs based on Conditional statements – eg. Maximum of two numbers, 2nd max, Roots of Quadratic Equations.
2) Program based on loops- GCD, LCM , Sine Series, Finding square root
3) Program based on Arrays – Finding max, Search of an element, Removal of duplicates etc.
4) Programs on sorting – Selection sort, Bubble sort, Insertion sort.
5) Program Files-Text and Binary files, searching in files.

6) Program on Strings – Concatenation, Substring, String Compare

Course Code:	ECL 101	Course Title:	Analog Electronics			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Nil	Type of Course	Basic Science			

Course Outcomes:

1. This course introduces the fundamentals of semiconductor devices, such as diode, BJT, DIAC, LED, UJT etc.
2. To study the V-I characteristics, biasing, small signal analysis, etc. for various electronic devices.
3. The student will be able to apply various devices into electroniccircuits and can compute various parameters.
4. At the end student will be able to study and design various power devices including applications of these devices in to power amplifications.

Course Contents:

P &N Type Semiconductors, Diodes and Power Supplies, Theory of P-N Junction Diode, Junction Capacitance,

Halfwave & Fullwave, Rectifiers, Filters, Ripple-Factor,

Characteristics & Applications of Following Diodes, Zener as Regulators, Schottkey, Photodiode, LED, LCD, Varactor Diode &Tunnel Diode

Junction Transistors Theory of Operation, Static Characteristics , Break Down Voltages, Current Voltage Power Limitations, Biasing of BJT Different Biasing Arrangements, Stability Factor, Thermal Runaway, Power Transistors

Small Signal Analysis & High Frequency Analysis of BJT CE, CB, CC Amplifiers and Comparison
High Frequency Analysis Calculation of Frequency Response, Gain Bandwidth Product

Power Amplifiers Classification A, B, AB, C Classes, Efficiency, Push Pull Configuration, Complimentary Symmetry, Second Harmonic & Cross Over Distortion. Positive and Negative Feedback Amplifiers Classification, Practical Circuits, Applications, Advantages. Oscillators Stability, Barkhausen Criteria, RC, LC & Crystal Oscillators

Field Effect Transistor & MOSFET, Principle of Operation & Characteristic, Biasing Arrangement, Small Signal Analysis of CG, CD & CS, High Frequency

Text:

1)Milman and Halkias, “Integrated Electronics”, Second Edition, 2011, McGraw Hill.
2)Boylestad and Nashelsky, “Electronic Devices & Circuit theory”,2011, Tenth Edition,

Reference:

1) David A. Bell, “Electronic Devices and Circuits”

2)Milman and Halkias, “Electronic Devices and Circuits”, Second Edition, 2011, McGraw Hill.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1)Study of characteristics PN-junction and Zenerdiodes

2) Study of PN-junction diode as full-wave and half wave rectifier

3) Study of Zener Diode as regulator

4) Input and output characteristics of NPN transistor under different configurations.

Course Code:	HUL 101	Course Title:	Communication Skills			
Category:	Core	Credit Assigned	L	T	P	C
			2	0	2	3
Pre-Requisite (if Any)	Nil	Type of Course	Humanities			

Course Outcomes:

1. To impart to the students the skills that they need in their academic, and later in their professional pursuit.
2. To train the students to adopt an innovative approach to English language teaching and learning.

Course Contents:

Importance of Effective Communication; Reading, writing and oral communication skills; Methods/Modes of communication, choice of media; Barriers to communication. Basics of Technical report Writing, Referencing methods, Visual communication and its impact, Hands-on-experiences and Case studies

Text:

1. Orient Longman , A Textbook of English for Engineers and Technologists.

Reference:

1. Quirk R.and Greenbaum S., A University Grammar of English.
2. Krishnaswamy N., English Grammar (Longman Publication) (Macmillan India Ltd)

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1. Presenting a book chapter using powerpoint slides
2. Data Analysis: Maintaining multiple results obtained over time and reporting them using charts and graphs
3. Technical Documentation – Requirement/specification documentation, Design documentation, Test-cases documentation, Use-cases documentation
4. Writing an installation/instruction manual
5. Writing an abstract of a technical article – summarizing an article in 300 words
6. Summarizing 3 papers into a report and its presentation

Course Code:	SAP 101	Course Title:	Health, Sports & Safety			
Category:	Core	Credit Assigned	L	T	P	C
			0	0	2	0
Pre-Requisite (if Any)	Nil	Type of Course	Basic Science			

Course Outcomes:

1. To provide physical fitness and good health.
2. Create awareness among the students about their health status by conducting various tests and measurements and suggest them suitable remedial physical fitness program so that they can improve physical and physiological health status.
3. To improve productivity, foster social harmony, inculcate sense of discipline and dedication in general life, develop the spirit of team work, through various sports activities.

Course Contents:

Development of components of fitness through conditioning exercises:
Strength: (Strength Endurance, Maximum Strength, explosive strength), Endurance: (aerobic endurance, anaerobic endurance, speed endurance and strength endurance), Speed, Co-coordinative ability, Flexibility
Physical Efficiency Test Level 1(Testing and Evaluation of Physical Fitness):
Cooper Test 12 minute run or walk test, Sit and reach test, 100 meter run, one minute sit up test, Push up/Bent knee push up test,
Teaching and development of sports skills: Cognitive, Perceptual, Motor, Perceptual motor. First Aid training:
Intramural phase 1: Identification of sports talent through exposing students to inter-section tournament. Football, Volleyball, throw ball, table tennis & Chess.

Yoga, Meditation and Personal Safety.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)
1) Physical Efficiency Test(Testing and Evaluation of Physical Fitness):1500 meter run, shuttle run, standing broad jump, one minute sit up test, flexibility test
Testing and assessment of selected Physiological parameters through Sports Medicine Research Lab: Total body fat analysis, Harvard step test, BMI, WHR, Back strength, Leg strength, grip strength, resting pulse rate, and resting respiratory rate. Intramural phase 2: Badminton, Basketball, Cricket, Kho-Kho, etc.
Yoga and Meditation.
2)Personal Safety Skill Demonstration

<table>
<tr><td>Course Code:</td><td>MAL 102</td><td>Course Title:</td><td colspan="4">Mathematics-II</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Core</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>4</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>MAL 101</td><td>Type of Course</td><td colspan="4">Basic Science</td></tr>
</table>

Course Outcomes:
To make students understand the basic importance of multi variable calculus (Differential calculus & Integral calculus), Vector calculus and ordinary differential equations in engineering.

Course Contents:

Calculus of Functions of Several Variables: Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, Tangent plane and normal line. Euler‟s theorem on homogeneous functions, Total differentiation, chain rules, Jacobian, Taylor‟s formula, maxima and minima, Lagrange‟s method of undetermined multipliers.
Multiple Integrals: Double and triple integrals, change of order of integration, change of variables, application to area, volumes, Mass, Centre of gravity.
Vector Calculus: Scalar and vector fields, gradient of scalar point function, directional derivatives, divergence and curl of vector point function, solenoidal and irrotational motion. Vector integration: line, surface and volume integrals, Green‟s theorem, Stoke‟s theorem and Gauss divergence theorem (without proof).
Ordinary Differential Equations: First order differential equations: Exact equation, Integrating factors, Reducible to exact differential equations, Linear and Bernoulli‟s form, orthogonal trajectories, Existence and Uniqueness of solutions. Picard‟s theorem, Picard‟s iteration method of
solution (Statements only). Solutions of second and higher order linear equation with constant coefficients, Linear independence and dependence, Method of variation of parameters, Solution of Cauchy‟s equation, simultaneous linear equations

Text:
1. Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons
2. Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1, Vol. 2)
3. Thomas, G.B. and Finney, R.L, Calculus and Analytic Geometry, Addison Wesley Longman.

Reference:
1. Michael D. Greenberg, Advanced Engineering Mathematics, Pearson Education Pvt. Ltd
2. Jain R.K., Iyengar S.R.K, Advanced Engineering Mathematics, Narosa Publishers.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)
1)
2)

Course Code:	BEL 102	Course Title:	Elements of Electrical Engineering

Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Nil	Type of Course	Basic Engineering			

Course Outcomes:

1. To enable the students understand the basic ideas and principles of Electrical Engineering.

2. To impart knowledge for understanding the details of electrical power systems, transformers, generators, motors etc.

Course Contents:

Electrical Circuit: Circuit Elements Resistance, Inductance & Capacitance, Kirchhoff‟s Laws, Voltage Source (Definition, Characteristics of Practical Source, and Equivalent Current Source), and Star-Delta Transformation.

Magnetic Circuit, Flux, MMF, Reluctance, Analogy with Electric Circuits. Simple Calculations for Composite Magnetic Circuits

AC Circuits: Periodic Function, Average & R.M.S., Values, Steady State Behavior With Sinusoidal Excitation, Phasor Representation, Reactance & Impedance, Series & Parallel Circuit, Power Factor, Principle of Generation of Single Phase & Three Phase Voltages, Power in Balanced Three Phase AC System

Electrical Measurements : Definition, Indicating, Integrating & Recording Instruments, Deflecting Controlling & Damping Mechanisms, Ammeter & Voltmeters, P.M.M.C. Type & Moving Iron Type, Electrodynamometer Type Wattmeters, Induction Type Single Phase Energy Meter

Transformers : Introduction, Basic Principles, Construction, Phasor Diagram for Transformer under No Load Condition Transformer On Load, Balance of MMF on Sides, Phasor Diagram, Equivalent Circuit, Open Circuit & Short Circuit Test, Voltage Regulation and Efficiency

Power Systems : Elementary Idea about Power Generation, Transmission and Distribution

Electric Machines :DC Shunt and Series Motor – Construction, Principle of Working, Characteristics, Speed Control and Applications

Induction Motors – Construction, Principle of Working of Single Phase and 3-Phase Motors. Torque Slip Characteristics

Text:

1. Hughes, Electrical Technology, Pearson Publishers

2. Theraja B.L., Electrical Technology, S. Chand Publishers

Reference:

3. Kothari D.P. and Nagrath I.J.,Theory And Problems Of Basic Electrical Engineering, Prentice Hall India

4. Kulshresta D.C., Basic Electrical Engineering, TMH India

5. Mittle and Mittal, Basic Electrical Engineering, TMH, 2005

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1. Study and verification of Kirchhoff‟s laws applied to DC circuits

2. Study of AC series R-L-C circuits

3. Determination of B-H curve of a magnetic material

4. Study of AC parallel R-L-C circuits

5. Study of balanced 3-phase circuits

6. Determination of voltage regulation and efficiency of a single-phase transformer by direct loading

7. Study of speed control of a DC motor by field current control and by armature voltage control

8. Study of reversal of direction of rotation of a 3-phase induction motor

Course Code:	HUL 102	Course Title:	Environmental Studies			
Category:	Core	Credit Assigned	L	T	P	C
			2	0	0	2
Pre-Requisite (if Any)	Nil	Type of Course	Basic Science			

Course Outcomes:

1. Introduce to various natural resources, their importance and status.
2. Introduce to the concepts of ecosystem, their structure and functions.
3. Introduce to the concept of biodiversity conservation.
4. Introduce to possible causes of various forms of environmental pollution and their consequences, methods of prevention.
5. Introduce to various social and climatic changes due to pollution.

Course Contents:

Natural resources: Forest resources, Water resources, Mineral resources, Food resources, Energy resources, Land resources.

Ecosystem: Concept of an ecosystem, Structure and functions of an ecosystem, Producers, consumers and decomposers, Ecological succession, Food chain, food webs and pyramids.

Biodiversity and its conservation: Introduction, definitions: genetics, species and diversity, Value of biodiversity, Biodiversity at global, national and local level, India as a mega-diversity nation, Hot-spot of biodiversity, Threat to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Conservation of biodiversity: in-situ and ex-situ conservation.

Environmental pollution: Definition, Causes, effects and control measures of: Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards, Solid waste management: Causes, effects and control measures of urban and industrial wastes. Social issues and environment: Sustainable development, Water conservation, Rain water harvesting, Watershed management, Climate change, Global warming, Acid rain, Ozone layer depletion, Nuclear accident, Holocaust, Environmental rules and regulations.

Human population and environment: Population growth, Environment and human health, Human rights, Value education, Role of information technology in environment and human health.

Text:

1. Rajgopalan R., Environmental Studies.

Reference:

1. Benny Joseph, Environmental Studies, McGraw Hill.
2. Erach Barucha Environmental Studies University press (UGC).

Course	CSL 102	Course Title:	Data Structures

Code:						
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	CSL 101 (Computer Programming)	Type of Course	Computer Science			

Course Outcomes:

- Appreciation and practice of structured programming
- Ability to formulate the problem, devise an algorithm and transform into code
- Understanding different programing techniques and make an informed choice amongst them
- Understanding different sorting algorithms, their advantages and disadvantages,
- Appreciation of concept of dynamic memory allocation and its utilization, dynamic data structures and implementation
- Understanding of concept of Abstract Data Type and implementations.

Course Contents:

Types and operations, Iterative constructs and loop invariants, Quantifiers and loops, Structured programming and modular design, Illustrative examples, Scope rules, parameter passing mechanisms, recursion, program stack and function invocations including recursion, Overview of arrays and array based algorithms - searching and sorting, Mergesort, Quicksort, Binary search, Introduction to Program complexity (Big Oh notation), Sparse matrices.

Structures (Records) and array of structures (records). Database implementation using array of records. Dynamic memory allocation and deallocation. Dynamically allocated single and multi-dimensional arrays.

Concept of an Abstract Data Type (ADT), Lists as dynamic structures, operations on lists, implementation of linked list using arrays and its operations. Introduction to linked list implementation using self-referential-structures/pointers.

Stack, Queues and its operations. Implementation of stacks and queues using both array-based and pointer-based structures. Uses of stacks in simulating recursive procedures/ functions. Applications of stacks and queues.

Lists - Singly-linked lists, doubly linked lists and circular linked lists. List traversal, insertion, deletion at different positions in the linked lists, concatenation, list-reversal etc. Mergesort for linked lists.

Text:

1) Data Structures & Program Design in C: Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.

2) Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

Reference:

1) Aho, Hopcroft and Ullmann, —Data Structures and Algorithms,‖ Addison Wesley, 1983.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1) Implementation of Binary search, Quick Sort, Merge Sort

2) Implementation of linked lists, insertion, deletion, finding an element.

3) Implementation of Sparse matrices, ADT and its Operation.

4) Implementation of Queue and its operations.
5) Implementation of Stacks and its operation.
6) Implementation of Priority Queues and its operations.

Course Code:	CSL 103	Course Title:	Application Programming			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	CSL 101 (Computer Programming)	Type of Course	Computer Science			

Course Outcomes:

- Aware about different tools for Web Programming.
- Background of working on web.
- Construct efficient web pages with CSS and Javascript.
- Demonstrate competency in the use of common HTML code.
- Able to design efficient client as well as server side scripts.

Course Contents:

Internet fundamentals, LAN, WAN, Introduction to common Internet terms, www.

Basics of networking, DNS, URL, firewall, proxy, Web protocols – http and https.

Designing web pages: HTML, forms, DHTML, XML, CSS. Extensible Hypertext Mark up Language (XHTML): XHTML syntax, headings, linking, images, special characters and horizontal rules, lists, tables, forms, internal linking, meta elements.

Introduction to Web Server – Setting up and configuration of Apache Tomcat server, Accessing pages from another machine.

Server Side Programming: Introduction to web programming with PHP.

Client side programming with Javascript

Introduction to Python - Statements and Control Flow, Expressions, Methods, Typing, Libraries and Developmental Environment, Web Programming using Python.

Text:

1) Deitel H.M. and P. J. Deitel, Internet & World Wide Web - How to Program, Prentice-Hall.
2) Goodman D, Morrison M., JavaScript Bible; Wiley India
3) Lutz, Mark, Learning Python (4th ed.). O'Reilly Media

Reference:

1)Garfinkle S., Spafford G; Web Security, Privacy and Commerce; O'Reilly, 2002.
2) Atkinson L., Core PHP Programming, Prentice Hall.
3) N.P.Gopalan, Akilandeswari, Web Technology, Prentice-Hall.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1) Creating an HTML Web page, forms.
2) Creating Home Page using HTML
3) Creating XHTML and CSS and understanding its use in creating Web pages.
4) Setting up and configuration of Apache Tomcat server.

5) Understanding modification of Web.XML
6) Creating Websites using PHP.
7) Understanding Javascript
8) Creating a Web page with back end in PHP and front end in Javascript and hosting it on Apache Tomcat Server.
9) Writing and understanding program in Python.
10) Use Python Libraries like Maths statistics to create programs for Scientific Computations.

Course Code:	ECL 102	Course Title:	Digital Electronics			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Nil	Type of Course	Core Engineering			

Course Outcomes:

1) To understand the fundamentals of digital logic design
2) Applications of combinational and sequentional logic circuits
3) To learn the HDL programming

Course Contents:

NUMBER SYSTEMS: Representations, signed, 1's complement, 2's complement, saturation and overflow in fixed point arithmetic.

BOOLEAN ALGEBRA: Axioms and theorems, DeMorgan‟s law, universal gate, duality, expression manipulation using axioms and theorems.

COMBINATIONAL LOGIC: Introduction to switching algebra, canonical forms, two-level simplification, boolean cube, logic minimization using K-map method, QuineMcCluskey tabular method, minimization for product-of-sum form, minimization for sum-of-product form, multiplexers, demultiplexers, decoders, encoders, hazard free synthesis, Arithmetic circuits, adders, half adder, full adder, BCD adder, ripple carry adder, carry-lookahead adder, combinational multiplier.

SEQUENTIAL LOGIC: Simple circuits with feedback, basic latches, clocks, R-S latch, master-slave latch, J-K flip flop, T flip-flop, D flip-flop, storage registers, shift register, ripple counter, synchronous counters, Finite State Machine (Moore/Mealy Machines), FSM with single/multiple inputs and single/multiple outputs etc.

CONTROLLER DESIGN: Based on minimum number of flip-flops and shift register method. Multiple command responsing register design. Conditional response controller design.

HARDWARE DESCRIPTION LANGUAGE: Programming and simulation, structural specification,
behavioral specification, dataflow modelling, testbench, testing using test vectors, testing using waveforms, design of basic blocks to build larger circuits, case studies, adder, ALU, counters, shift registers, register bank, FSM design example etc.

Text:

1. Digital Design, Morris Mano, Prentice Hall, 2002
2. Digital Fundamentals,10th Ed, Floyd T L, Prentice Hall, 2009.

Reference:

1. Digital Design-Principles and Practices, 4th Ed, J F Wakerly, Prentice Hall, 2006.
2. Fundamentals of Digital Logic with Verilog Design, 2nd Ed, S. Brown and Z. Vrsaniec, McGraw Hill, 2007

Course	CSL 201	Course Title:	Advanced Data Structures

Code:						
Category:	Core	Credit Assigned	L	T	P	C
			3	1	2	5
Pre-Requisite (if Any)	CSL 102 (Data Structures)	Type of Course	Computer Science			

Course Outcomes:

- Appreciation of dynamic data structures, advantages and disadvantages.
- Ability to formulate the problem, devise an algorithm and transform into code.
- Ability to identify problem requirements, constraints to be satisfied and ability to select the best possible data structures to satisfy the constraints.
- Ability to analyze the complexity/efficiency of the algorithm and develop ability to improve the same
- Ability to understand how a newer data structure gets designed as per the requirements and constraints.
- Understanding of advantages and disadvantages of different data structures which may be used to solve the same problem
- Introduction to different algorithmic programing techniques like greedy algorithms, dynamic programming etc. and ability to make an informed choice amongst them
- Ability to communicate about program/algorithm/data-structure efficiency (time and space) and recognize a better solution

Course Contents:

Applications of lists in polynomial representation, multi-precision arithmetic, hash-tables etc. Multi linked structures and an example application like sparse matrices. Implementation of priority queues.

Trees, binary trees, binary trees- basic algorithms and various traversals. Binary Search Trees (BSTs) and insertion, deletion in BSTs. Height-balanced (AVL) trees, insertion/deletion and rotations. Heaps and heapsort. Splay trees.

Multi-way trees and external sorting - B-trees, Red-black trees. Introduction to B+ trees. Tries. Applications of the above mentioned trees.

Generalisation of trees to graphs – their representation & traversals. Dijkstra‟s shortest path algorithm, topological sort, all-pairs shortest paths, minimum spanning trees. Huffman coding. Introduction to network flow problem.

Introduction to Skip lists, data structures for disjoint set representation.

Text:

1) Data Structures & Program Design in C : Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
2) Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

1) List of general assignment based on above syllabus.

Course Code:	MAL 201	Course Title:	Mathematics-III (Statistics and Probability)			
Category:	Basic Science	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	Mathematics-I (101), Mathematics-II (102)	Type of Course	Basic Science			

Course Contents:

Probability

Random Variable & Probability Distributions: Random Variables, Density function, distribution function for continuous and discrete random variables, Joint distributions. Mathematical Expectation: Mathematical Expectation, The variance and Standard deviation , Moment Generating Function, Characteristic Function. Special Probability Distributions: Some special probability distributions like Binomial Poisson, Geometric, Normal, Uniform, Exponential Gamma Beta, Chi-Square, Students ‘t’, F-distribution and Weibull Distribution.

Statistics

Moments, correlation, covariance and regression. Sampling Theory: Population Parameter, Sample Statistics, Sampling distributions, Sample mean,Sampling distribution of means, The Sample variance, The sampling distribution of variance. Estimation Theory: Point estimate and Interval Estimates, Reliability, Confidence interval estimates of population parameters, confidence intervals for means , proportions and variance . Tests of Hypothesis and Significance: Statistical decisions, Tests of hypothesis and significance. Type I and Type II errors. Level of significance, one tailed and two tailed tests.Tests involving small samples and large samples. Fitting theoretical distributions to sample frequency distribution .The chisquare test for goodness of fit.

Text:

1. Paul L. Meyer, Introductory Probability and Statistical Applications,Addison Wesley.
2. Miller and Freund: Probability and Statistics for Engineers Eastern Economy Edition, PHI.
3. E.Parzen: Modern Probability Theory and Its Applications J. Wiley and Sons Inc., New York.

Reference:

1. M.R.Speigal: Probability and Statistics, McGraw-Hill, 1995.
2. V.K. Rohatgi and A.K.M. EhsanesSateh: An Introduction to Probabability and Statistics, John Wiley & Sons.

Course Code:	CSL 202	Course Title:	Introduction to Object Oriented Programming			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Understand the necessity of encapsulation, data hiding, inheritance, and exception handling.
2. Formulate a software application and propose an object oriented design.
3. Write generic programs using the standard template library.
4. Study and use design tools like UML, design patterns etc.

Course Contents:

1. Object Oriented Programming, Features of object oriented programming languages like data encapsulation, inheritance, polymorphism and late binding.
2. Concept of a class, Access control of members of a class, instantiating a class, static and non-static members, overloading a method.
3. Deriving a class from another class, access control of members under derivation, different ways of class derivation, overriding of a method, run time polymorphism.
4. Concept of an abstract class. Concept of an interface. Implementation of an interface.
5. Exception and exception handling mechanisms. Study of exception handling mechanisms in object-oriented languages
6. Introduction to streams, use of stream classes. Serialization and de-serialization of objects.
7. Templates, Implementation of data structures like linked lists, stacks, queues, trees, graphs, hash table etc. using object oriented programming languages.
8. Introduction to concept of refactoring, modeling techniques like UML, Design patterns.

Text:

1. Bjane Strostrup, “The C++ programming language”, Addison-Wesley
2. Herbert Shildt, “C++: The Complete Reference”, 4th Edition
3. Arnold Ken, Gosling J, “The Java Programming Language”, Addison Wesley
4. Matt Weisfeld, “The Object-Oriented Thought Process”, Pearson
5. Cox Brad, “Object –Oriented Programming: An Evolutionary Approach”, Addison –Wesley

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

- Practicals based on above mentioned syllabus

Course Code:	CSL 203	Course Title:	Computer System Organisation			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Students will learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design.
2. Students will be able to identify where, when and how enhancements of computer performance can be accomplished.
3. Students will learn the sufficient background necessary to read more advance texts as well as journal articles on the field.
4. Student will see how to use concepts of computer organization in real-life settings using various PC performance improvements.
5. Students will also be introduced to more recent applications of computer organization in advanced digital systems.

Course Contents:

1. Addressing methods, their application in implementation of HLL constructs and data
2. structures, instruction formats, expanding opcode method, subroutine linkage in PDP-11 and 68000, zero address machine such as HP3000.
3. Processing unit, bus architecture, execution of a complete instruction, sequencing of control signals, microprogrammed control, microinstruction format, microinstruction sequencing, bit slice concept.
4. Arithmetic, number representations and their operations, design of fast address, signed multiplication, Booth's Algorithm, bit-pair recording, division , floating point numbers and operations, guard bits and rounding.
5. Main memory organization, various technologies used in memory design, higher order memory design, multimodule memories and interleaving, cache memory , concept of cache memory, mapping functions, replacement algorithms. Input-output organization, I/O mapped I/O and memory mapped I/O, Direct Memory Access, interrupts and interrupt handling mechanisms, device identification, vectored interrupts, interrupt nesting, I/O interfaces, synchronous vs. asynchronous data transfer, I/O channels.
6. Computer peripherals, I/O devices such as video terminals, video displays, graphic input devices, printers, magnetic disk, magnetic tape , CDROM systems.
7. RISC philosophy, pipelining, basic concepts in pipelining, delayed branch, branch prediction, data
8. dependency, influence of pipelining on instruction set design, multiple execution units, performance
9. considerations, basic concepts in parallel processing & classification of parallel architectures

Text:

1. Computer Organization , Hamacher, Carl V. et al, McGraw Hill
2. Structured Computer Organization , Tanenbaum A.S, Prentice Hall of India Ltd
3. Computer Organization & Design,The Hardware/ Software Interface, Patterson D. A , Hennessy J. L.

Harcourt Asia, Second Edition.

Course Code:	CSL 204	Course Title:	Discrete Maths and Graph Theory			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Student should be able to use different proof techniques.
2. Students would be able to argue about limits by using PigeonHole principle.
3. Solve problems based on set theory, Permutations and Combinations, as well as Discrete Probability.
4. Students will be able to solve mathematical problems on partial orders, and group theory.
5. Students would be able to model and analyze computational problems in graph theoretical framework.

Course Contents:

1. Set theory, operations on sets – relation and functions, continuity, partial order, equivalence relations, Peano axioms and induction.
2. Mathematical logic, propositions, predicate logic, formal mathematical systems, algebra, homomorphism automorphism.
3. Elements of Theory of some algebras, semigroups, monoids, groups.
4. Rings, fields, lattices, Boolean Algebra
5. Graphs, hypergraphs, transitive closure, trees, spanning trees
6. Combinatorics , generating functions, recurrences, Counting theorem and applications.
7. Eulerian tours, Hamiltonian cycles, Planar Graphs, Connectivity, Colorability, Line Graphs

Text:

1. Kolman,” Discrete Mathematical Structures for Computer Science”, Busby
2. Liu C.L” Combinatorial Mathematics”, McGraw Hill Book Compan

Course Code:	CSP 201	Course Title:	IT Workshop-I			
Category:	Core	Credit Assigned	L	T	P	C
			1	0	4	3
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Effectively use the Unix programming environment - shell, file system, scripts, filters, program development tools.
2. Automate tasks and write simple programs using scripting languages, such as Awk.
3. Develop good programming style, organization, interface, and documentation habits.
4. Use of effective procedures and tools for building, debugging, testing, tuning, and maintaining programs.
5. Use of tools and write programs to assist in developing programs.

Course Contents:

1. Introduction to Linux/Unix OS - ls, wc, chdir, mkdir, chmod, cd, mv, df, du, netstat, ps, more, set, env, setenv, chgrp, man, rm, rmdir, grep, vi, vim, tar, untar, uuencode, find, cat, history, ping, ifconfig, traceroute, whoami
2. Installing Ubuntu Linux (or any variant)
3. Introduction to using different tools for identification of possible errors in C program – gdb, concepts of “core dump”, backtracing using “bt”, using “info” to dump all registers, creating watch-list / watch variables.
4. DDD (Data Display Debugger) – introduction and usage, debugging with ddd (step, step into, step over).
5. IDE for code development
    a. Using DevCpp and/or VisualStudio b. Create a project, using multiple .c and .h files with cross-references c. Setting compiler options and linker options [Concept of optimization] d. Understanding different settings
6. Unix tools - Awk, sed, Emacs
7. Bash scripting – variables, conditionals, loops, finding logged in users
8. Parameter passing to C program from shell (argc / argv)

Assignments :

1. Using/Creating/Modifying/copying Files via C programs. Reading- from / writing-to files
2. Creating a grade card preparation program from individual subject marks stored in files and creating the result.
3. Creating calculator program
4. Translating date / time across different time-zones (with and without daylight saving)
5. Writing shell script for - listing all files above 1000KB, listing all users with more than 10 processes, copying all files ending with .txt extension recursively.
6. Understanding gcc -o1, -o2, -o3 options and comparing the resulting file sizes.

<table>
<tr><td>Course Code:</td><td>CSL 205</td><td>Course Title:</td><td colspan="4">Design and Analysis of Algorithms</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Core</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>1</td><td>2</td><td>5</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>Advanced Data Structures , Mathematics</td><td>Type of Course</td><td colspan="4">Computer Science</td></tr>
</table>

Course Contents:

1. Mathematical foundations, summation of arithmetic and geometric series, □n, □n2, bounding summations using integration, recurrence relations, solutions of recurrence relations using technique of characteristic equation and generating functions.
2. Asymptotic notations of analysis of algorithms, analyzing control structures, worst case and average case analysis, amortized analysis, sorting algorithms such as selection sort, insertion sort, bubble sort, heap sort, lower bound proof, elementary and advanced data structures with operations on them and their time complexity.
3. Divide and conquer basic strategy, binary search, quick sort, merge sort, Fast Fourier Transform etc. Greedy method - basic strategy, application to job sequencing with deadlines problem, minimum cost spanning trees, single source shortest path etc.
4. Dynamic Programming basic strategy, multistage graphs, all pairs shortest path, single source shortest paths, optimal binary search trees, traveling salesman problem.
5. Basic Traversal and Search Techniques, breadth first search and depth first search, connected components. Backtracking basic strategy, 8-Queen’s problem, graph coloring, Hamiltonian cycles etc. NP-hard and NP-complete problems, basic concepts, non deterministic algorithms, NP-hard and NP-complete, Cook’s Theorem, decision and optimization problems, polynomial reduction

Text:

1. Introduction to Algorithms : Cormen T.H. et.al : Prentice Hall of India
2. Computer Algorithms : Horowitz, Sahani, Rajsekharan , Galgotia Publications Pvt.Ltd
3. Fundamentals of Algorithms : Brassard, Bratley , Prentice Hall

Course Code:	CSL 206	Course Title:	Software Engineering			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	Computer Programming(CSL 101)	Type of Course	Computer Science			

Course Outcomes:

1. to look at the large scale software development from a broader perspective, and function in multidisciplinary teams
2. to apply knowledge gained in the course to practical software development situations in methodical way
3. to design software systems to meet desired needs with realistic constraints
4. to communicate effectively in software development activities
5. to get an idea about contemporary issues in Software development and engage in life-long learning, understand professional and ethical responsibility

Course Contents:

1. Software Engineering Process & Management : Generic view, Capability Maturity Model, Process models-waterfall, evolutionary, incremental etc., unified process, agile view, project management, metrices estimation, project scheduling, risk management.
2. Software engineering Principles and Practice : Communication, planning and modeling practices, system engineering and modeling, business process engineering requirement analysis, system analysis- flow oriented and class oriented modeling using data modeling concepts.
3. Software Design Engineering : Design Concepts : Abstraction Architecture, pattern modularity, information hiding, design classes, refacoring etc., Design of web application, architectural design, component level design, user interface design.
4. Software Testing and Quality Management : Testing strategies, testing for object oriented software testing for web applications, validation testing etc. Black box testing, white box testing, Basis path testing. Testing for specialized environments, architectures and application. Quality concepts, quality assurance, software reviews, statistical quality assurance.
5. Software configuration management and advance topics : Elements of configuration management system, process configuration for web engineering, component-based development, clean room software engineering, formal methods, software reengineering, Software Maintenance

Text:

1. Software Engineering by Ian Sommerville ; Pearson Ed
2. Software Engineering: A Practitioner’s Approach by Roger Pressman ; Tata-McGraw Hill

Course Code:	CSL 207	Course Title:	Operating Systems			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Data Structures(CSL 102)	Type of Course	Computer Science			

Course Outcomes:

1. Understand the structure and design issues of operating systems.
2. Learn about and understand theoretical concepts and programming constructs used for the operation of modern operating systems.
3. Understand concepts of OS management domains like process, memory, file systems, storage etc.
4. Familiarity with operating systems like Unix.
5. Gain practical experience with software tools available in modern operating systems such as semaphores, system calls, sockets and threads.

Course Contents:

1. Introduction, basic h/w support necessary for modern operating systems - Services provided by OS, system programs and system calls - brief discussions of evolution of OS - real time and distributed systems : a brief overview of issues.
2. File systems, user interface - disk space management and space allocation strategies - examples from UNIX, DOS, Windows etc - directory structures - disk caching - file system consistency and logs - disk arm scheduling strategies.
3. Processes and 3 levels of scheduling - process control block and context switch - goals of scheduling and different scheduling algorithms - threads : user-level and kernel level.
4. Memory management techniques - contiguous and non-contiguous - paging and segmentation - translation look-aside buffers (TLB) and overheads - virtual memory and demand paging- page faults and instruction restart - problems of large address spaces - page replacement algorithms and working sets - miscellaneous issues.
5. Process cooperation and synchronization - mutual exclusion and implementation - semaphores, conditional critical regions and monitors - classical inter - process communication problems - message passing.
6. Deadlocks and strategies for handling them - protection and security issues - access lists, capabilities, cryptographic techniques - introduction to distributed systems.
7. Linker and Loader - Concept of static and dynamic relocation, external symbols, design of linker, design of object file for different loading schemes.
8. Common Object file format - Structure of object file and executable file, section or segment headers, symbol table, concept of storage class, string various, data types line insert, character, arrays structures.
9. Device Drivers - Device programming, system drivers, non system drivers, virtual drivers, Incorporation of driver routines, Basic device driver operation, character and block drivers.

Text:

1. Tanenbaum A, “Modern Operating Systems”, PHI 2nd Ed
2. Silberchatz & Galvin, “Operating System Concepts”, Addison Wesley

Course Code:	CSL 208	Course Title:	Design Principles of Programming Languages			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. To provide an overview of different programming paradigms
2. Improve the background for choosing appropriate programming languages for certain classes of programming problems
3. Understand the implementation aspects behind different programming constructs
4. Be able in principle to program in an imperative (or procedural), an object-oriented, a functional, and a logical programming language
5. Understand the significance of an implementation of a programming language in a compiler or interpreter
6. Increase the ability to learn new programming languages
7. Increase the capacity to express programming concepts and choose among alternative ways to express things
8. Simulate useful features in languages that lack them
9. Be able in principle to design a new programming language
10. Make good use of debuggers and related tools

Course Contents:

1. Definition of Programming language . Syntax , semantics. High - level languages. Implementation of high-level languages, Compilers and Software interpreters. Data elements, identifiers binding, binding time, binding identifiers to names, binding of attributes, importance of binding time. Concept of r-value and l-value . Effect of environment on a language. Language paradigms.
2. Data type, elementary data type, structured data type, elements of specification and implementation of data type. Implementation of elementary data types : integer, real, character, Boolean and pointer.
3. Implementation of structured data types. Vectors & arrays, records and files. Type checking, type conversion and initialization.
4. Evolution of data type concept. Abstract data type, encapsulation. Design and implementation of new data types through subprograms. Subprogram definition and activation, their implementation, parameter passing, generic subprograms.
5. Sequence control structures used in expressions and their implementation. Sequence control structures used between statements or group of statements and their implementation.

6. Sequence control structures used between subprograms, recursive and non recursive subprogram calls. Data control, referring environment dynamic and static scope, static chain implementation and display implementation.
7. Type definition as mechanism to create new abstract data types, type equivalence, type definitions with parameters. Defining new abstracts data types Storage management issues, like static and dynamic allocation, stack based allocation and management, Heap based allocation and management

Text:

1. Pratt Terence, “Programming Languages, Design and Implementation”, PHI
2. Sethi Ravi, “Programming Languages”, Addison Wesley

Course Code:	CSP 202	Course Title:	IT Workshop-II			
Category:	Core	Credit Assigned	L	T	P	C
			1	0	4	3
Pre-Requisite (if Any)	IT Workshop-I	Type of Course	Computer Science			

Course Outcomes:

1. Understand the installation of operating systems, web server and databases and their configuration.
2. Understand advanced commands of UNIX and automate tasks using scripts. The students should get understanding of the advanced Linux operating system. Get to know the necessary tools for Linux administration and script development and learn about special features offered by Linux.
3. Understand programming languages like Python.
4. Learn to use to write large software and debug them.
5. Build a mini project using the overall concepts learned in the lab.

Course Contents:

1. Introduction to advanced unix/linux commands and tools – redirection, piping, mount, unmount, lynx, ftp, telnet, su, rlogin, rcp, cron, cksum, cmp, ln, , top, uptime
2. Makefile – writing Makefile, compilation via make
3. Python – Basic ideas, variables, conditionals, loops, if, for, while
4. Installing Linux (or any variant / Ubuntu) on virtual machine
5. Basic assignments in Apache Tomcat web server / Mysql
6. Installing and using Apache Web server to develop a website, Creating a simple website – using Linux, Apache, and Python)
7. Mini projects like (1) Implementation of garbage collector (2) Software interrupt handler

Course Code:	CSL 301	Course Title:	Database Management Systems			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Advanced Data Structures (CSL 202)	Type of Course	Computer Science			

Course Outcomes:

1. To obtain sound knowledge in the theory, principles and applications of database management system.
2. Design and develop data model given their specifications and within performance and cost constraints.
3. Acquire and understand new knowledge, use them to develop data centric application and to understand the importance of lifelong learning.
4. Perform experiments in different disciplines of database management system.

Course Contents:

1. Database system concepts and Architecture - concept of relational database, Relational data model , Relational algebra, SQL-the relational database standard, ER and EER model.
2. Database design theory - Functional dependencies and normalization, relational database design algorithms, practical database design and demoralization, Relational constants, programmatic ways for implementing constraints, triggers, Chase algorithm.
3. Physical database design - Concept of physical and logical hierarchy, storage structures like cluster, index organized table, partitions, various table storage parameters and block storage parameters, concept of index, B-trees, hash index, function index, bitmap index.
4. Process and memory management in database - Various types of tasks in database, database buffer management, log buffer management code reuse, concept of two tier and N-tier architecture, data dictionary and catalog information database recovery technique. Arier Algorithm for recovery.
5. Query optimization and performance tuning - Various techniques for query optimization, strong and weak equivalence, cost base optimization, Use of different storage structures in query optimization.
6. Transaction Processing - Transaction and system concepts, Desirable properties of transaction, Schedules and recoverability, serializability of schedules, concurrency control, lock base protocols and time stamp based protocols, read consistency.

Text:

1. Fundamentals of Database Systems : Elmasiri and Navathe, Addisson Wesley, 2000
2. Principles of Database Systems : Ullman , Golgotia Publications 1988

Course Code:	CSL 302	Course Title:	Computer Networks			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Student will be able to
2. Defining, using and implementing Computer Networks and the basic components of a Network system, explain the importance of data communications, how communication works in data networks and the internet, recognize the different internetworking devices and their functions.
3. Explain the role of protocols in networking, Analyze the role and services and features of the various layers of data networks, analyze the features and operations of various application layer protocols such as Http, DNS, Telnet, FTP and SMTP.
4. Knowing and Applying pieces of hardware and software to make networks more efficient, faster, more secure, easier to use, able to transmit several simultaneous messages, and able to interconnect with other networks, differentiating the various types of network configurations and applying them to meet the changing and challenging networking needs of organizations, defining and analyzing the
5. circuits available for voice and data networks, their transmission speeds (bandwidth), and how they are packaged for commercial use.
6. Defining the different protocols, software, and network architectures, defining the concept of local area networks, their topologies, protocols and applications , analyzing why networks need security and control, what errors might occur, and how to control network errors.

Course Contents:

1. Computer Networks, evolution of Computer Networks, application of Computer Networks.
2. Layered Network Architecture: requirement for layered approach, basic concept of layering in the network model, define entities, protocols, interface in networking context, ISO’s OSI Reference Model, functions of the seven layers of OSI Model , TCP/IP model, difference between OSI and TCP/IP model
3. Data and Signal: Define data, signal. Time domain and frequency domain representation of signal, bandwidth of a signal and medium, Sources of impairment, Attenuation, distortion, noise, data rate Limits and Nyquist bit rate, between Bit Rate and Baud Rate, Sources of noise. FDM and TDM, synchronous and asynchronous TDM
4. Transmission Media: Various Transmission Media - guided and unguided media, characteristics of the popular guided transmission media: Twisted-pair, Coaxial cable, Optical fiber, Sources of transmission impairment, Shannon Capacity
5. Network Topology- what is network topology, characteristics of the following topologies: Mesh,
6. Bus, Star, Ring, Tree, Unconstrained
7. Medium Access Control (MAC) Techniques - goals and requirements of Medium Access Control (MAC) techniques, key issues related to MAC techniques, Classify various contention based techniques such as ALHOA, CSMA, CSMA/CD and CSMA/CA. MAC techniques: Polling, Token passing. FDMA, TDMA, CDMA.

8. IEEE 802 LANs - basic characteristics of LANs, operation of IEEE 802 LANs , 802.3 - CSMA/CD-based (Ethernet), 802.4 – Token bus-based, 802.5 – Token ring-based, Compare performance of the three LANs
9. Introduction of High Speed LANs, Fast Ethernet and Gigabit Ethernet, wireless LANs
10. Need for wireless LAN, limitations and challenges of wireless LAN IEEE 802.11 WLAN - Transmission media, Topology, Medium Access Control, Security
11. Interfacing to the media and synchronization: modes of communication, Asynchronous and Synchronous modes of communication. Error Detection and Correction: need for error detection and

12. correction, simple parity check, 2-D parity check, checksum, cyclic redundancy check., Hamming's code
13. Flow Control and Error Control : need for flow and error control, Stop-and-wait flow control, Sliding-window flow control, Stop-and-wait ARQ, Go-back-N ARQ, Selective-repeat ARQ, Selective-repeat ARQ.
14. HDLC: how HDLC works, piggybacking in HDLC, data transparency in HDLC

15. Switching Techniques: Circuit Switching - need for circuit switching , how circuit switching takes place, space-division and time-division switching, Packet Switching - need for packet switching, how packet switching takes place, difference between virtual-circuit and datagram type packet switching, Message switching, Compare circuit switching, packet switching, message switching.
16. Need for internetworking, Introduction of internetworking devices- Hubs, Switches, Bridges, Router, Gateways
17. Internet Protocol (IP): different classes of IP addresses, concept of subnet masking, sub-netting super-netting, network address translation table, ARP/RARP protocol, fragmentation and reassembly, ICMP protocols, key features of IPv6
18. Transport layer: Connection establishment and release – timer management - multiplexing - flow control working of TCP and UDP. QoS parameters,
19. ATM network, ATM signaling, PNNI routing I ATM
20. Application Layer Protocols: DNS, Telnet, ICMP, RPC, SMTP, FTP, SNMP
21. Routing Algorithms: Adaptive routing, Non-adaptive routing, Dijkstra's SP algo, flooding, flow based, distance vector routing, linked state routing, RIP- routing information protocol, OSPF - (Open shortest path first), BGP - Border gateway protocol: operation of the BGP protocol
22. Congestion Control: causes for congestion, effects of congestion , various open-loop and close-loop congestion control techniques: The leaky bucket algorithm , The token bucket algorithm, Choke packets, Load shedding, jitter control, distinguish between flow and congestion control

Text:

1. Tanenbaum A. S, "Computer Networks", PHI 4th Edition
2. James F. Kurose and Keith W. Ross : Computer Networking A Top-Down Approach Featuring the Internet, 3rd Edition.
3. Peterson, Davie, "Computers Networks", Elsevier 3rd Edition
4. William Stallings, "Data and Computer Communications", PHI 6th Edition
5. Reference:

Reference:

1. Simon Haykin, “Communication Systems”, John Wiley 4th Edititon
2. Douglas Comer, “Computer Networks and Internets”, Addison Wesley 2nd Edittion
3. Peterson, Simon, “Computer Networks: A Systems Approach”, Pearson Education, Asia
4. Behrouz A Forouzan : Data Communication and Networking, 4th Edition.

List of Lab Assignments / Experiments OR List of Tools on which the lab assignment should be based (If Any)

CN Lab :

Programs based on

1. Using TCP sockets or Network socket programming
2. Client-server application for chat
3. PC to PC file transfer using serial port
4. Implementation of Shortest path routing
5. Implementation of Sliding Window Protocol
6. Implementation of Address Resolution Protocol
7. Implementation of Open Shortest Path First Protocol
8. Using n/w simulators like: NS2, DLC/DLL simulator
9. Implementation of multi thread client server application.
10. Implementation of TCP/IP Echo
11. Using simple UDP
12. Using RPC / RMI

Course Code:	CSL 303	Course Title:	Theory of Computation			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. Given a language the student can find the appropriate machine for recognition of that language.
2. The student is able to convert machine to grammar and vice versa.
3. The student gains knowledge about hierarchy of languages.
4. The student develops analytical thinking and intuition for problem solving.
5. The student is able to show whether a given problem is solvable or unsolvable.

Course Contents:

1. Preliminaries - Sets, operations, relations, transitive closure, countability and diagonalisation, induction and proof methods- pigeon-hole principle and simple applications - concept of language - grammars and production rules - Chomsky hierarchy.
2. Regular grammars, deterministic finite automata - non determinism, conversion to deterministic automata- e-closures, regular expressions, finite automata, regular sets.
3. Pump lemma for regular sets- closure properties of regular sets, decision properties for regular sets, minimization of automata.
4. Context - free languages, parse trees and ambiguity, reduction of CFGS, Chomsky and Griebach normal forms, push - down Automata (PDA), non determinism, acceptance by two methods and their equivalence, CFLs and PDAs – Pumping lemma for context free languages, Closure and decision properties of CFLs.
5. Timing machines – variants, recursively enumerable (r.e.) sets, recursive sets, TM as computer of function, decidability and solvability, Halting Problem, reductions, Post correspondence Problem (PCP) and unsolvability of ambiguity problem of CFGs.
6. Introduction to recursive function theory - primitive recursive and partial recursive functions Church -Turing thesis - convergence of view points of what “computibility” is : Semi formal treatment.

Text:

1. Martin John, “Introduction to languages and the theory of computation”, TMH
2. Motwani Hopcroft, Ullman, “Introduction to Automata Theory, Languages and computation”, Pearson Education

Course Code:	CSP 301	Course Title:	IT Workshop-III			
Category:	Core	Credit Assigned	L	T	P	C
			1	0	4	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. To inform students about various elements that make the advanced web work, and the way they relate to open-source platforms.
2. To substantially strengthen students' programming ability by requiring them to understand engineering tools that help in programming productivity.
3. To provide exposure to a broad range of open-source technologies including encryption and mixed use of languages.

Course Contents:

Main theme - Use of open source tools

1. Advanced use of Apache Web server: Understaing web.xml, creating a web-site having multiple pages and forms.
2. Java Native Interface (JNI) – Calling C / C++ code from Java and vice versa.
3. Automatic testing tools – Junit, NUnit,
4. Introduction to Android Platform and APIs / libraries provided. A sample standalone game / application on Android. An application on Android that requires backend – like an Android application for news updates.
5. Introduction to SSL. Use digital certificates to encrypt / decrypt data in transfers

Notes - Keytool in Java allows to create / store / manipulate certificates. Also, refer www.thawte.com for free download/creation of a certificate

- Learning software engineering tools
- Design tools - Rational Rose / visio
- Memory leaks - Purify /
- Code Coverage tools - Clover
- Testing tools – Loadrunner, Winrunner
- Version Control Tools – CVS
- Bug Tracking Tools - Bugzilla

Course Code:	CSL 304	Course Title:	Compilers			
Category:	Core	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	None	Type of Course	Computer Science			

Course Outcomes:

1. This course introduces students how a compiler translates the higher level language into machine language.
2. The student should be able to analyze issues associated with the implementation of higher-level programming languages.
3. To inform students about different parsing techniques, techniques to generate intermediate code and different optimization techniques. Understanding of compiler optimization techniques would enable students to write reasonably efficient programs.
4. The students will also appreciate the need of understandable error reports, accurate and reliable object code, and efficient use of in-memory data structures.

Course Contents:

1. Introduction to compilers, compilers and translators, phase structure of a typical compiler, Number of passes, ideas about lexical analysis, syntax analysis, code optimization and code generation, design of lexical analyzer.
2. Syntax specification of programming languages**,** Design of top-down parser, bottom up parsing technique, LR parsing algorithm, Design of SLR, LALR,LR parsers. Dealing with ambiguity of the grammar.
3. Study of syntax directed definitions and syntax directed translation schemes as notational frame work to specify the translations. Using syntax directed translation schemes for translation of expressions, controls structures, declarations , procedure calls.

4. Storage allocation and run time storage administration, symbol table management, Error detection and recovery, error recovery in LR parsing, error recovery in LL parsing, Automatic error recovery in YACC.
5. Introduction to Important code optimization techniques, loop optimization, control flow analysis, data flow analysis, setting up data flow equations to compute reaching definitions, available expressions, Live variables. Problems in code generation , simple code generator code generation from DAG, Peephole optimization
6. Assembler, Macroprocessor - Concept of assembler, design of single pass and two pass assembler, forward reference, design of output file of assembler, concept of macro, macro call within macro, macro definition within macro, recursive macro calls, design of macro processor.
7. Lexical Analysis - Role of lexical analyzer, recognition of tokens, tool for study of lex.

Text:

1. Principles and practice of compiler writing : Aho, Sethi , Ullman , Addison Wesley
2. Compiler Design in C : Alan Holub , PHI
3. Crafting a compiler : Fischer and LeBlanc , Addison Wesley
4. Principles of Compiler Design : Aho A. V., Ullman J.D , Narosa Publishing House.

Course Code:	CSL 305	Course Title:	Cryptography and Network Security			
Category:	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Pre-Requisite (if Any)	Computer Networks	Type of Course	Computer Science			

Course Outcomes:

1. Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.
2. Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.
3. Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
4. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

Course Contents:

1. Classical Ciphers: Affine, Playfair , Hill Cipher;
2. Modern Block and Stream Ciphers: DES, AES, RC4, A5/1; Block Modes of Operation: ECB, CBC, CFB, OFB, CTR
3. Asymmetric Key Cryptosystems: RSA; Digital Signatures: DSS; Hash and MAC: SHA-512
4. Key Management: Digital Certificates, PKI; Authentication: One-Way Authentication, Mutual Authentication, Dictionary Attacks, Centralized Authentication, The Needham-Schroeder Protocol, Kerberos
5. Network Layer Security: IPSec; Transport Layer Security: SSL/TLS
6. Non-cryptographic Protocol Vulnerabilities: DoS and DDoS, Session Highjacking and Spoofing, ARP Spoofing and Attacks on DNS
7. Software Vulnerabilities: Phishing, Buffer overflow, cross site scripting and SQL injection
8. Viruses, Worms, and other Malware: Virus and Worm Features, Internet Scanning Worms, Mobile Malware and Botnets
9. Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control, SELinux and Recent Trends
10. RFIDs and E-Passports
11. Electronic payment

Text:

1. Forouzan, Cryptography and Network Security, TMH
2. Bernard Menezes, Network Security and Cryptography, Cengage
3. Radia Perlman Network Security: Private Communication in a Public World, Prentice Hall 2002

Reference:

1. Bruce Schneier Applied Cryptography , 2nd Edition John Wiley & Sons 1996
2. Douglas Stinson Cryptography Theory and Practice CRC Press 1995
3. Alfred Menezes, Paul van Oorschot, Scott Vanstone Handbook of Applied Cryptography CRC Press 1997
4. Pfleeger and Pfleeger, Security in Computing, Pearson

Course Code:	CSP 302	Course Title:	IT Workshop –IV			
Category:	Core	Credit Assigned	L	T	P	C
			1	0	4	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Install and use OpenStack/ CloudStack open source cloud platform
2. Create an free account and use commercial cloud offering like Amazon/ BlueMix/ SalesForce/ Google/ Azure/ Rackspace
3. Examine the design of task and data parallel distributed algorithms for Clouds and use them to construct Cloud applications.
4. Install Hadoop and understand map-reduce. Create an application that uses map-reduce.
5. Install and use no-sql database like mongodb/cassendra/hbase/neo4j
6. Understand and describe the standards and technologies of modern web services implementations.
7. Effectively use market-leading development tools to create and use web services.
8. Identify and select the appropriate framework components in the creation of web service solutions.
9. Students will be able to coordinate the use of information security tools within an organization.
10. Securing web services.

Course Contents:

1. Cloud systems, Cloud technologies, hands-on project on Cloud infrastructure
2. Hadoop usage and understanding of map-reduce paradigm
3. No-SQL database storage, and query
4. XML- XML Namespace, XML Schema, XML Schema Design Pattern, XPath, XSLT, XSLT Advanced Usage,
5. Service oriented architecture, and web services
6. Web services styles:
7. SOAP-standard web services that use XML-formatted SOAP messages
8. RESTful web services, which leverage HTTP-standard methods and use XML
9. Working with JavaScript and web services
10. Create and use SOAP-standard web services
11. Security and authentication
12. Study and write software that interoperates with a well-known existing public web services application programming interface
13. Survey of web services implementations on other platforms (e.g. PHP, Java)

Course Code:	CSL 306	Course Title:	Soft Computing			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Advanced data Structures	Type of Course	Computer Science			

Course Outcomes:

1. Appreciation of the unified and exact mathematical basis as well as the general principles of various soft computing techniques.
2. Appreciation of basic knowledge about the theory and key algorithms that form the foundation for artificial neural network and practical knowledge of learning algorithms and methods
3. Appreciation of basic knowledge about the theory and key algorithms that form the foundation for fuzzy logic and become aware of the use of fuzzy inference systems in the design of intelligent or humanistic systems.
4. Ability to understand the principles, advantages, limitations and possible applications of learning.
5. Ability to identify and apply the appropriate learning technique to classification, pattern recognition, optimization and decision problems.

Course Contents:

1. Neural Networks: History, overview of biological neuro-system, mathematical models of neurons, ANN architecture, Learning rules, Learning Paradigms-Supervised, Unsupervised and reinforcement Learning, Learning Tasks, ANN training Algorithms-Single layer perceptron, multi-layer perceptron, Self-organizing Map, Applications of Artificial Neural Networks.

2. Introduction to fuzzy set, Operations on fuzzy sets, Fuzzy relation, Fuzzy implication, approximate reasoning, Fuzzy rule-based systems, Fuzzy reasoning schemes, Fuzzy logic controller.

3. Implementing fuzzy IF-THEN rules by trainable neural nets. Fuzzy neurons, Hybrid neural networks, Neuro-fuzzy classifiers.

Text:

1. Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence; Roger Jang, Tsai Sun, Eiji Mizutani, PHI.
2. Soft Computing and Its Applications : R.A. Aliev, R.R. Aliev
3. Neural Network: A Comprehensive Foundation; Simon Haykin, PHI.
4. Elements of artificial Neural Networks; Kishan Mehtrotra, S. Ranka, Penram International Publishing (India).
5. Fuzzy Logic with Engineering Applications; Timothy Ross, McGraw-Hill.
6. Neural Networks and Fuzzy Systems: Bar Kosko , PHI.

Course Code:	CSL 402	Course Title:	Artificial Intelligence			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Formulate problems so that exploratory search can be applied.
2. Implement optimal, heuristic and memory bounded search techniques.
3. Represent knowledge using formal logic and design algorithms to work in a semi-observable environment using logical reasoning.
4. Design and develop practical algorithms for solving real-life planning problems.
5. Implement probabilistic reasoning techniques to work in uncertain environments.

Course Contents:

1. Introduction: What is AI? , History, Overview, Intelligent Agents, Performance Measure, Rationality, Structure of Agents, Problem-solving agents, Problem Formulation, Uninformed Search Strategies
2. Informed (Heuristic) Search and Exploration, Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions, inventing admissible heuristic functions, Local Search algorithms, Hill-climbing, Simulated Annealing, Genetic Algorithms, Online search
3. Constraint Satisfaction Problems, Backtracking Search, variable and value ordering, constraint propagation, intelligent backtracking, local search for CSPs, Adversarial Search, Games, The minimax algorithm, Alpha-Beta pruning, Imperfect Real-Time Decisions, Games that include an Element of Chance.
4. Knowledge Based Agents, Logic, Propositional Logic, Inference, Equivalence, Validity and Satisfiability, Resolution, Forward and Backward Chaining, DPLL algorithm, Local search algorithms, First Order Logic, Models for first order logic, Symbols and Interpretations, Terms, Atomic sentences, complex sentences, Quantifiers, Inference in FOL, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution
5. Planning, Language of planning problems, planning with state-space search, forward and backward state-space search, Heuristics for state-space search, partial order planning, planning graphs, planning with propositional logic
6. Uncertainty, Handing uncertain knowledge, rational decisions, basics of probability, axioms of probability, inference using full joint distributions, independence, Baye’s Rule and conditional independence, Bayesian networks, Semantics of Bayesian networks, Exact and Approximate inference in Bayesian Networks

Text:

Artificial Intelligence a Modern Approach : Russel and Norvig , Pearson Education, 2nd

□ Artificial Intelligence – A Practical Approach : Patterson , Tata McGraw Hill, 3rd

Course Code:	CSL 403	Course Title:	Machine Learning			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

Students will be able to:

1. Recognize the characteristics of machine learning that make it useful to real-world problems.
2. Understand the basic underlying concepts for supervised discriminative and generative learning.
3. Understand the concepts of cross-validation and regularization; be able to use them for estimation of algorithm parameters.
4. Characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
5. Effectively use machine learning toolboxes.
6. Be able to use support vector machines.
7. Be able to use regularized regression algorithms.
8. Understand the concept behind neural networks for learning non-linear functions.
9. Understand and apply unsupervised algorithms for clustering.
10. Understand the foundation of generative models.
11. Understand the inference and learning algorithms for the hidden Markov model.
12. Understand the learning algorithm for hidden Markov model with latent variables.
13. Understand algorithms for learning Bayesian networks.
14. Understand reinforcement learning algorithms.

Course Contents:

Intro, Perceptron, Linear Regression

1. Logistic Regression, SVM
2. SVM, Multiclass & Ordinal Classification
3. Kernel Methods
4. Decision Trees
5. Neural Networks
6. Clustering, Generative Models, Mixture Models
7. Hidden Markov Model
8. Bayesian networks

Text:

Mitchell Tom. Machine Learning. McGraw Hill, 1997.

1. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, Wiley, New York, 2001.
2. Ethem Alpaydin, Introduction to Machine Learning, PHI
3. Chris Bishop, Pattern Recognition and Machine Learning

<table>
<tr><td>Course Code:</td><td>CSL 404</td><td>Course Title:</td><td colspan="4">Pattern Recognition</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Elective</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>2</td><td>4</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td></td><td>Type of Course</td><td colspan="4">Computer Science</td></tr>
</table>

Course Outcomes:

1. To obtain sound knowledge in the theory, principles and applications of pattern recognition
2. Apply knowledge of mathematics, science, and engineering in the design and development of recognition system.
3. Perform experiments on different software packages either obtain from external parties or developed by themselves and analyse the experimental results.
4. Design and develop software for character recognition, image recognition given their specifications and within performance and cost constraints.
5. Ability to understand the computing needs of inter-disciplinary scientific and engineering disciplines and design and develop algorithms and techniques for achieving these.

Course Contents:

1. Applications of pattern recognition, statistical decision theory, probability of events, Random variables, Estimation of parameters, Minimum Risk Estimators.
2. Bay’s Theorem, conditionally independent features. Decision boundries, Estimation of error rates, characteristics curves.
3. Histograms, Kernel and window parameters, Nearest Neighbour classification techniques, Adaptive Decision boundries, clustering.
4. Artificial Neural Networks, Nets without hidden layers and with hidden layers. The back propagation Algorithm, Hopfield Nets.
5. Gray level scaling transformations. Equalization smoothing transformations. Edge detection Logarithmic Gray scale level scaling.
6. Scene segmentation and labelling , counting objects, Hough Transforms, Eigenvector line fitting , Fourier transforms.

Text:

1. Pattern Classification : Richard O.Duda, Peter E.Hart, David G.Shork, John Wiley & Sons 200, 2nd Edittion
2. Pattern Recognition and Image Analysis , Earl Gose, Richard Johnsonbough , Steve Jost . Prentice Hall
3. of India
4. Pattern Recognition and Image Processing : Sing Tze bow; Marcel Dekker

Course Code:	CSL 405	Course Title:	Distributed Systems			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Identify the advantages and challenges in designing distributed algorithms for different primitives like mutual exclusion, deadlock detection, agreement, etc.
2. Design and develop distributed programs using sockets and RPC/RMI.
3. Differentiate between different types of faults and fault handling techniques in order to implement fault tolerant systems.
4. Analyze different algorithms and techniques for the design and development of distributed systems subject to specific design and performance constraints.

Course Contents:

1. Introduction and motivation to Distributed Systems,Characteristics, Applications, Challenges, Architecture types, Fundamental models. (2 weeks)
2. Inter-process and inter-node communication using Sockets – connection oriented and connection-less, Remote Procedure Calls, Remote Method Invocation (1 week)
3. Distributed File System Design and Implementation, Case Studies of NFS, Andrew File Systems, HDFS, Distributed Resource Management. (1 week)
4. Clock Synchronization Techniques, Network Time Protocol, Logical Clocks, Vector Clocks. (1 week)
5. Causally Ordered Broadcast and Unicast, Termination Detection – Ring based and Dijkstra Scholten algorithms, Leader Election – Ring based, Franklin’s algorithm and Bully Algorithm (2 weeks)
6. Distributed Mutual Exclusion – Token based algorithms – Lamport’s, Ricart-Agarwala, Maekawa’s algorithms, Non Token based Algorithms – Suzuki Kasami, Raymond’s algorithms, comparison of different algorithms. (2 weeks)
7. Distributed Deadlock Detection, Resource and Communication Deadlocks – Centralized technique, Distributed technique - edge chasing and path pushing algorithms, Hierarchical technique, Recovery from Deadlocks. (1 week)
8. Fault Tolerance, Handling Crash faults –Two phase commit protocol, Non-blocking three phase commit protocol, Birman-Joseph Atomic Broadcast Protocol, Voting techniques for fault tolerance. (2 weeks)
9. Recovery – forward and backward recovery, undo-redo logs, Coordinated and Uncoordinated Checkpointing and Recovery algos (1 week)
10. Agreement protocols – LSP Oral Messages, Agreement using Signed Messages (1 week)

Text:

1. Singhal and Shivratri, “Advanced concepts in Operating Systems”, McGraw Hill
2. Coulouris, “Distributed Systems”, AWL Press. Pearson Education
3. Tanenbaum, “Modern Operating Systems”, PHI

Course Code:	CSL 406	Course Title:	Real Time Systems			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)	Operating Systems	Type of Course	Computer Science			

Course Outcomes:

1. Enumerate the need and the challenges in the design of hard and soft real time systems.
2. Compare different scheduling algorithms and the schedulability criteria.
3. Determine schedulability of a set of periodic tasks given a scheduling algorithm.
4. Develop algorithms to decide the admission criterion of sporadic jobs and the schedule of aperiodic jobs.
5. Integrate resource access mechanisms with the scheduling techniques and develop integrated schedulibility criteria.

Course Contents:

1. Introduction to RTS, WCET notion, Types of RTS, Task Types, Jobs – Periodic, Sporadic, Aperiodic, Applications of RTS, Predictability, Reference Model, Types of schedulers, Cyclic and Priority based Schedulers. (1 week)

2. Cyclic, priority based schedulers – static/dynamic – RM, EDF, LST, Optimality of EDF, Non-optimality of EDF, Scheduling with precedence constraints, Multiprocessor scheduling – static and dynamic systems, Problems of Predictability in multi-processor systems, Predictability of preemptive priority based scheduling in uniprocessor systems, Performance Measure of validation techniques. (2 weeks)

3. Cyclic scheduling, frame size constraints, Job Slicing, Aperiodic job scheduling using Slack stealing, Sporadic job scheduling, Practical considerations, Disadv of cyclic scheduling. (2 weeks)

4. Priority Based Sched, Static-Dynamic Systems, Fixed, Variable Priorities, Schedulable Utilization, Schedulable Utilization of EDF, Schedulability Test of EDF, Unpredictability of Dynamic Priority in Overload, Liu-Layland Theorem, Optimality of RM in Simply-Periodic Systems, Concept of Critical Instants, Time Demand Analysis, Practical factors - Non-preemption, self-suspension, context switch time, Limited priority levels, Mapping techniques, Impact on schedulability, Tick Scheduling. (3 weeks)

5. Aperiodic jobs in Priority based systems, Polling Server, Combining with background server,Polling Server Parameters, Deferrable Server (DS), Combining with Background Server, Deferrable Server parameters, Disadv of DS, Simple Sporadic Server Rules, Combining Background time, Proof of Simple Sporadic Server as a periodic task, Constant Utilization Server for Deadline Driven systems, Total Bandwidth Server, Starvation free CU/Background Server, Preemptive Weighted Fair Queuing Server, Scheduling of Sporadic Jobs in Fixed Priority and Dynamic Priority Systems. (3 weeks)

6. Resource Control, Model, Priority Inversion, Uncontrolled Priority Inversion, Anomalies, NPCS, Blocking Time, Disadvantages of NPCS, Priority Inheritance Protocol, Deadlocks due to Priority Inheritance Protocol, Priority Ceiling Protocol, Deadlock

Avoidance,Analysis of Priority Ceiling Protocol, Blocking time, context switches, Stack Sharing Priority Ceiling Protocol, example, Priority Ceiling Protocol in Dynamic Priority Systems, Preemption Levels, Fixed Preemption Level Systems like EDF, Basic Preemption Ceiling Protocol, Multiple units of resources, Priority ceiling, Preemption ceiling and stack based preemption ceiling protocols for multiple unit resources. (3 weeks)

Text:

1. Real-Time Systems :Jane W.S. Liu, Pearson Education
2. Real Time Systems : C.M. Krishna & Kang G. Shin : McGraw Hill

Course Code:	CSL 407	Course Title:	Cloud Computing			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Understanding of different layers of cloud computing, infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Introduction of practical IaaS, PasS, SaaS, including Amazon ECS, GAE, Force.com, Microsoft Azure, etc.

2. Gain knowledge of cloud storage system design issues, including directory management, data placement, and consistency issues. Practical cloud storage system solutions including GFS, Big Table, HDFS, etc.

3. Student should have good knowledge of authentication, authorization and secure access in the cloud. Introduction to cloud security. Secure computation in the cloud.

4. Insights into the virtualization technologies: Hypervisor, emulation, and application VM. Platform virtualization, storage virtualization, and network virtualization.

Course Contents:

1. Cloud system architectures, Cloud programming frameworks and what is "infrastructure-as-a-service", "platform-as-a-service" and "software-as-a-service" , Cloud computing delivery models – public, private and hybrid clouds, cloud-in-a-box

2. "Big data" concepts, storage and management, Security, scalability, privacy, lock-in, and other risks (and mitigations) for individuals and companies

3. Virtualization, clustering and resource management, HPC in cloud computing

4. Cloud applications considerations for updates, backups, disaster recovery and fault tolerance, Data center networks and Energy use in data centers

5. Introduction to cloud enabling technologies: Introduction to Hadoop, Map-reduce, NoSQL, MongoDB, Cassandra, Web Servers, Encryption techniques, SSL

6. Case Study: Design of a cloud system

Text:

1. Resse G., Cloud Application Architectures: Building Applications and Infrastructure in the Cloud,O' Reilly.

Reference:

2. Buyya R., Broberg J., Goscinski A. M., Cloud Computing – Principles and Paradigms,Wiley.

Course Code:	CSL 408	Course Title:	Software Architecture			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

On completion of this course students will be able to:

1. Design and understand software architecture for large scale software systems.
2. Recognize major software architectural styles, design patterns, and frameworks
3. Describe a software architecture using various documentation approaches and architectural description languages
4. Develop architectural alternatives for a problem and select among them
5. Use well-understood paradigms for designing new systems

Course Contents:

3. Software process and the role of modeling and analysis, software architecture and software design. (1 week)
4. Software Modeling and Analysis: Analysis modeling and best practices, traditional best practice diagrams such as DFDs and ERDs (2 weeks)
5. Software Architecture: architectural styles, architectural patterns, analysis of architectures, formal descriptions of software architectures , Architectural description languages and tools (3 weeks)
6. Software Design: design best practices, design patterns, design case studies, component technology, object oriented frameworks, distributed objects, interoperability standards, case studies., software quality (3 weeks)
7. UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns, documenting software architecture, reconstructing software architecture. (2 weeks)
8. Middleware components, programming models, implementation, systems qualities Moving from qualities to architecture and views Components and COTS, Economics- Driven Architecture, Software product line, Software architecture future. (2 weeks)
9. Issues in Software Architecture: Scalability and interoperability issues, web application architectures, case studies. (2 weeks)

Text:

1. M. Shaw, “Software Architecture Perspectives on an Emerging Discipline”, PHI
2. Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice”, Pearson Education Asia
3. R. Taylor, N. Medvidovic, E. Dashofy, “Software Architecture – Foundations, Theory, and Practice”, Wiley India
4. Jan Bosch, “Design and Use of Software Architectures”, Addision-Wesley-Pearson Education

5. Christine Hofmeister, Robert Nord, Dilip Soni, “Aoolied Software Architecture”, Addision-Wesley Pearson Education
6. Dikel, D.Met Al, “Software Architecture: Organizational Principles and Pattern”, Prentice Hall

Course Code:	CSL 409	Course Title:	Software Project Management			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Match orgnizational needs to the most effective software development model
2. Understanding the basic concepts and issues of software project management
3. Effectively planning the Software projects and employ mechanisms for tracking the software projects
4. Implementing project plans through managing people, commnication and change
5. Developing skills for tracking and controlling software deliverables and address real-world management challenges

Course Contents:

1. Overview of Software Project Management, The Project Life Cycle, Software Development Life Cycle Models, Life Cycles and Metrics, Process Maturity: SEI CMM
2. Estimation Techniques of IT, Project Scoping, Project Planning, Project Control, Project Phase-Out, Risk Management, Configuration Management
3. People Management, Team Dynamics, Net Present Value, Project Portfolio Management, Software Quality Assurance, Project Leadership

Text:

1. R.K. Wysocki et al. : Effective Project Management: Traditional, Agile, Extreme, 5th Edition, Wiley India, 2011.
2. C. Jones : Applied Software Measurement, Assuring Productivity and Quality, McGraw Hill

Reference:

1. D. I. Cleland : Project Management, Strategic Design and Implementation, 3rd edition, McGraw-Hill.

Course Code:	CSL 307	Course Title:	Signals and Systems			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

2. This Course will introduce you to the fundamental ideas of signals and system analysis. Applications of these ideas include audio and image processing, communications,control, machine learning, and finance.
3. The topics well cover in the course include basic properties of signals and systems, the processing of signals by linear systems, Fourier series and transforms, sampling,discrete-time processing of continuous-time signals.
4. This course will serve as a central building block for students interested in further studying information processing in any form.

Course Contents:

1. Elements of Signal Space Theory: Different types of signals, Linearity, Time invariance and causality, Impulse sequence, Impulse functions and other singularity functions.
2. Convolution: Convolution sum, Convolution integral and their evaluation, Time domain representation and analysis, of LTI systems based on convolution and differential equations.
3. Multi Input-Output Discrete and Continuous Systems: State model representation, Solution of state, equations, State transition matrix.
4. Transform Domain Considerations: Laplace transforms and Z-transforms, Application of transforms to discrete and continuous systems analysis, Transfer function, Block diagram representation, DFT.
5. Fourier series and Fourier Transform: Sampling theorem, Discrete Fourier transform (DFT), Estimating Fourier Transform using (DFT).

Text:

1. Signals and Systems - Continuous and Discrete:;,"Ziemer, R.F., Tranter,W.H. and Fannin, JD.R",Prentice Hall ; 1998,4th Edition
2. Signals and Systems,"Oppenheim, A.V., Willsky, A.S. and Young, I.T",Prentice Hall ; 1983 Edition
3. Signals and Systems,"Roberts, M.J",Tata McGraw-Hill ; 2003Edition

Course Code:	CSL 411	Course Title:	Digital Signal Processing			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course				

Course Contents:

1. Discrete time signals; Sequences; representation of signals on orthogonal basis; Sampling and reconstruction of signals.
2. Discrete time systems; attributes, Z- Transform, Analysis of LSI systems, frequency analysis, Inverse Systems.
3. Signal flow graph representation, DF1, DF2, parallel and cascade form. Finite word-length effects in Digital Filters
4. Discrete Fourier Transform (DFT), Fast Fourier Transform algorithms.
5. Design of FIR Digital Filters: Window method, Park-McClellan's Method.
6. Design of IIR Digital Filters: Butterworth, Chebyschev approximations. Lowpass, Bandpass Bandstop and Highpass filters. Bilinear, impulse invariant frequency transformations.

Text:

1. Oppenheim & Schafer, “Discrete Time Signal Processing”, PHI Ltd
2. ▪ Proakis John and Manolakis D.G, “Digital Signal Processing: Principles Algorithms and Applications”, Prentice Hall 1992.
3. ▪ Cavicchi Thomas J, “Digital Signal Processing”, Wiley 2002
4. ▪ Mitra S.K , “Digital Signal Processing A Computer -Based Approach”, Tata McGraw-Hill

Course Code:	CSL 412	Course Title:	Advanced Computer Architecture			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Make the students aware about various trends in computer design, architecture of advanced processors.
2. Realization about issues related to instruction level, thread level, data level parallelism in multi/many core systems, memory organization & optimization techniques.
3. Understand the design issues with shared/distributed memory systems, multi / many core / GPGPU architecture

Course Contents:

1. Classes of computers, Trends in technology, power and costs, dependability, quantitative principles of computer design, Introduction to computing models.
2. Principles of scalable performance, performance metrics and measures, speedup performance laws, advanced processor technology, superscalar and VLIW processors, Verified memory, cache memory organizations, shared memory organizations. Memory hierarchy, cache performance, protection and examples of virtual memory, cache coherence.
3. Pipeline and superscalar techniques, linear pipeline processors, reservation and latency analysis, collision free scheduling, pipeline schedule optimization, instruction pipeline design, arithmetic pipeline design, super scalar and super pipeline design.
4. Multiprocessors and multi-computers, Brief overview of SIMD, MIMD, vector architectures and multi-core architectures.
5. Elementary theory about dependence analysis, techniques for extraction of parallelism, branch prediction, dynamic scheduling, multiple issue and speculation, limits on instruction level parallelism, Thread level parallelism

Text:

1. Computer Architecture : A Quantitative Approach : Hennessy and Patterson : Morgan Kaufmann: 4th
2. Advanced Computer Architecture, Kai Hwang , McGraw Hill
3. Advanced Computer Architectures : A design space approach, Sima D, Fountain T. and Kacsuk P, Pearson Education

<table>
<tr><td>Course Code:</td><td>CSL 413</td><td>Course Title:</td><td colspan="4">Embedded System</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Elective</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>2</td><td>4</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td></td><td>Type of Course</td><td colspan="4"></td></tr>
</table>

Course Outcomes:

Student will be able to

1. Describe the differences between the general computing system and the embedded system, also recognize the classification of embedded systems, become aware of the architecture of the ARM processor and its programming aspects (assembly Level)
2. Become aware of interrupts, hyper threading and software optimization, design real time embedded systems using the concepts of RTOS, analyze various examples of embedded systems based on ARM processor
3. To work on design and development of protocols related to real-time communication, to develop real-time algorithm for task scheduling , the students will be able to design, simulate , built and debug complex combinational and sequential circuits based on an abstract functional specification.

Course Contents:

1. Embedded systems overview, Hardware / Software co-design, Examples of embedded systems, Components of Embedded systems – sensors, actuators, micro-controller processor Technology, IC technology, Issues in Design Technology
2. Scheduling Paradigms for real time systems - Static Priorities, Static Schedules, Dynamic Scheduling, Pre-emptive, Non-pre-emptive, Rate Monotonic, EDF
3. Real World Issues like - Task Assignment, CPU utilization, Blocking,Unpredictability, performance Measures
4. Hard & Soft Real Time Systems, Real Time Operating System – concepts, RTOS services, capabilities, Resource Management
5. Programming Languages for Embedded Systems - Desired Language Characteristics,
6. Tools for building Embedded systems, Embedded Software Development Methodology
7. Issues in real time databases, real time communications, Fault Tolerant Techniques –
8. Fault Types, Detection, Recovery

Text:

1. Frank Vahid, “Embedded System Design- A Unified Hardware / Software Introduction”, John Wiley & Sons
2. Krishna C.M. , Kang G. Shin, “RTS : Real Time Systems”, McGraw Hill
3. David Simon, “An Embedded Software Primer”, Addison Wesley, -2000

Course Code:	CSL 308	Course Title:	Computer Graphics			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Students will demonstrate an understanding of contemporary graphics hardware
2. Students will create interactive graphics applications in C++ using one or more graphics application programming interfaces, write program functions to implement graphics primitives.
3. Students will write programs that demonstrate geometrical transformations
4. Understand contemporary graphics principles and graphics hardware, demonstrate geometrical transformations.
5. Understand and demonstrate 2D image processing techniques, demonstrate 3D image processing techniques.
6. Understand and demonstrate computer graphics animation, Create interactive graphics applications in C++ using one or more graphics application programming interfaces.

Course Contents:

1. Basic fundamentals of random scan, raster-scan devices, LCD displays - point and line drawing techniques and algorithms - input/output devices and interactive techniques.
2. Polygon filling methods: Seed fill, edge flag algorithm etc. - scan conversion techniques - anti aliasing techniques - clipping algorithms, Polygon clipping, Viewing transformation, Windowing transformation.
3. Linear transformation: rotation, scaling, translation in 3D -homogeneous coordinates - normalized device coordinates - windowing and viewporting, Cartesian Coordinates, Word view etc.
4. Curve generation - cubic splines, Beziers, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function.
5. Review of 3D vector algebra - parallel and perspective projections and transformation - hidden line/ surface elimination - shading and rendering - ray tracing techniques.
6. Graphics software packages - segmentation and display files - graphics standards - graphics and computer networks - basic principles of X windows, X terminals, Functions for segmenting display files.

Text:

1. Procedural Elements of Computer Graphics : Rogers : McGraw Hill.
2. Principles of Interactive Computer Graphics : Newman, Sproull,
3. McGraw Hill
4. Computer Graphics : Hearn, Baker, PHI, India
5. Introduction to Computer graphics: Foley, Vanpam, Hughes, Philips,
6. Foley, Vanpam, Hughes, Philips

<table>
<tr><td>Course Code:</td><td>CSL 414</td><td>Course Title:</td><td colspan="4">Data Mining and Warehousing</td></tr>
<tr><td rowspan="2">Category:</td><td rowspan="2">Elective</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>2</td><td>4</td></tr>
<tr><td>Pre-Requisite (if Any)</td><td></td><td>Type of Course</td><td colspan="4">Computer Science</td></tr>
</table>

Course Outcomes:

1. Identify the scope and necessity of Data Mining & Warehousing for the society.
2. Describe the designing of Data Warehousing so that it can be able to solve the root problems.
3. To understand various tools of Data Mining and their techniques to solve the real time problems.
4. To develop ability to design various algorithms based on data mining tools.
5. To develop further interest in research and design of new Data Mining techniques.

Course Contents:

1. Introduction to Data warehousing **-** Application of Data warehousing and mining, Datawarehouse development life cycle, Data warehouse analysis, CUBE, ROLL UP and STAR queries.
2. Data Warehouse Design **-** Massive denormalisation, STAR schema design ,Data ware house Architecture, OLAP, ROLAP and MOLAP , concepts of Fact and dimension table
3. Space Management in Data warehouse - Schemas for storing data in warehouse using different storage structures, B-tree index, hash index, clusters, Bitmap index functional index, domain index, Data partitions.
4. Performance and Tuning **-** Query optimization, memory management, process management. I/o management for Data warehouse.
5. Data Mining Tools –Association rules, a priori algorithm, Fp-trees algorithm, constraints and solution.
6. Cluster analysis- paradigms, DBSCAN, cluster algorithms.
7. Mining tools- decision trees and applications.

Text:

1. Jiawei Han, Micheline Kamber, “Data mining- Concepts & Techniques”, Morgan Kaufmann
2. Michale Corey, Michale Abbey; Oracle 8i Data Warehousing; Tata McGraw Hill.
3. Navathe and Elmasry ; Fundamentals of Database Systems; Addison Wesley, 2000
4. Arun Pujari; Data Mining; Orient Longman, 2003

Course Code:	CSL 415	Course Title:	Bioinformatics			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. This course introduces students to the basic string based computational methods and algorithms that can be used to understand the cell and biological systems.
2. Students will know algorithms and programming techniques like dynamic programming, hashing, and suffix trees.
3. The course focuses on computational approaches to: genetic and physical mapping; genome sequencing, assembly, and annotation.
4. This course will help students develop multidisciplinary approach to the systematic analysis and modeling of complex biological phenomena.
5. Students are also made aware of the currently emerging research areas in the fields of computational and systems biology.

Course Contents:

1. Basics of biology
2. Sequences: Problem statement, Edit distance and substitution matrices, Global and local alignments, Spliced alignment, Space-efficient sequence alignment, Multiple alignment,
3. Structures: Protein alignment, Protein structure prediction
4. Phylogenetic trees: Large parsimony and small parsimony problems, Probabilistic approaches, Grammar-based approaches
5. Overview of Gene Control, Working of Genetic Switches, Introductory Systems Biology, The biochemical paradigm, genetic paradigm and the systems paradigm
6. Building an Organism Starting From a Single Cell -Quorum Sensing – Programmed Population Control by Cell-Cell Communication and Regulated Killing; Gene regulation at a single cell level- Transcription Networks -basic concepts -coherent Feed Forward Loop (FFL) and delay gate -The incoherent FFL -Temporal order, Signaling networks and neuron circuits -Aspects of multi-stability in gene networks.
7. Modeling biological systems, Hidden Markov models
8. Miscellaneous topics: Pathways and networks, Microarrays, Biomedical images, Genetic Algorithms and applications,

Text:

1. "An Introduction to Bioinformatics Algorithms" by Jones, Pevzner. MIT Press.
2. "Algorithms on Strings, Trees and Sequences" by Gusfield. Cambridge University Press.
3. “An Introduction to Systems Biology: Design Principles of Biological Circuits” by Alon. Chapman & Hall/CRC Press.

Course Code:	CSL 416	Course Title:	Information Retrieval			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. Understanding the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system
2. Understanding the data structures like Inverted Indices used in Information retrieval systems
3. Understanding the basics of web search
4. Understanding the different techniques for compression of an index including the dictionary and its posting list
5. Understanding the different components of an Information retrieval system
6. Developing the ability of develop a complete IR system from scratch

Course Contents:

1. Boolean retrieval
2. the term vocabulary and postings lists,
3. Dictionaries and tolerant retrieval,
4. Introduction to index-construction and index-compression
5. Scoring, term weighting and the vector space model
6. Computing scores in a complete search system, Evaluation in information retrieval, Introduction to Relevance feedback and query expansion.
7. Probabilistic information retrieval, review of basic probability theory, the probability ranking principle, the binary independence model
8. Language models for information retrieval, Language modeling versus other approaches to IR, Text classification and Naive Bayes, Bayesian Network approaches to IR.
9. Vector space classification, Support vector machines and machine learning on documents, Fl at clustering, Hierarchical clustering, Matrix decomposition and latent semantic indexing.
10. Introduction to Web search basics, Web crawling and indexes, Link analysis
11. Typical Assignments : Based on techniques studied, implementation of those techniques, study of research papers.

Text:

1. An Introduction to Information Retrieval: Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Cambridge University Press, Cambridge, England, 2009
2. Information Retrieval: Implementing and evaluating search engines: Stefan Büttcher, Charles L. A. Clarke, Gordon V. Cormack, MIT Press, 2010

Reference:

1. Information Retrieval: Algorithms and Heuristics : David A. Grossman, Ophir Frieder, Springer.
2. Information Retrieval: Data Structures and Algorithms by Frakes, Pearson.

Course Code:	CSL 417	Course Title:	Business Intelligence			
Category:	Elective	Credit Assigned	L	T	P	C
			3	0	2	4
Pre-Requisite (if Any)		Type of Course	Computer Science			

Course Outcomes:

1. To obtain sound knowledge in the theory, principles and applications of Business Intelligent System.
2. Design and develop multidimensional data model given their specifications and within performance and cost constraints.
3. Acquire and understand mathematical concepts to develop data centric decision models.
4. Perform experiments in different disciplines of Business Intelligent System.

Course Contents:

1. Introduction to Business Intelligence
   What is Business Intelligence, Why do we need Business Intelligence, EIS, MIS, DSS & BI , Information Pyramid – Data, Information, Knowledge & Intelligence. Basis for Operational, Tactical & Strategic Decision Making, OLTP Vs. OLAP, Requirements Gathering in BI through Business Questions, BI in various Domains and Functional Area
2. Principles of Dimensional Modelling
   Foundation for Fact based decision making, The STAR and SNOWFLAKE schema, Pros & Cons of the STAR/SNOWFLAKE Schema Dimensional Model, Slowly Changing Dimension tables, Fact-less Fact Tables, Aggregation Strategy, Time Dimension
3. Business Intelligence System Architecture
   Need for Enterprise Class Business Intelligence Infrastructure, The BI Ecosystem, Building Blocks of a N-Tier BI System – Servers & Communication Protocols, The Central Repository – Metadata, Information Consumption User Interfaces – Desktop Vs. Web Vs. Mobile, Open Architecture, Scalability, Performance in BI – In Memory Analytics
4. BI Project Lifecycle
   Typical BI Project Lifecycle, Requirements Gathering & Analysis – Functional & Non-Functional Requirements, Reports & Dashboards Design – Mock-up and Storyboarding, Testing in a BI Project, BI Project Deployment, Post Production Support **Introduction to**
5. Enterprise Class BI Tool
   First Level of Abstraction of the Data Warehouse in MicroStrategy, Building the Schema Objects – Attributes, Facts, Transformation & Hierarchies, Building Reusable Application Objects – Metrics, Filters, Prompts, Five Styles of BI, Building Reports – Grids & Graphs, Report Manipulation over the Web – Pivoting, Sorting, Drilling, Exporting etc., Setting up Report Distribution, Report Project

Text:

1. Turban E., Sharda R., Delen D., King D., Business Intelligence, Pearson Education.

Reference:

- Sabherwal R. and Becerra-Fernandez I., Business Intelligence, Wiley.
- Kimball R., Ross M., The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Wiley and Sons (2010).