NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

SCHEME OF INSTRUCTION AND SYLLABI

B.Tech. – Computer Science and Engineering

Effective from 2020-21

Contents

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

About the Department of CSE

The Department of Computer Science and Engineering at NIT Andhra Pradesh, offers a B.Tech. Undergraduate programme in Computer Science and Engineering, M.Tech. in Computer Science and Data Analytics, M.S. (by research) and Ph.D. programmes. The department was incepted in the year 2015, the current sanctioned intake is 150 students for the B.Tech. (CSE) programme.

Vision of the Department of CSE:

To strive for excellence in academics, research and technological service to the society, with an intent to nurture the stakeholders and produce Computer Scientists, technologists and Engineers who are globally competent and nationally relevant.

Mission of the Department of CSE:

- **M 1.** To adopt a teaching-learning process that imparts technical skills and state-of-the-art knowledge with a well-blended and balanced mix of theory and practice.
- **M 2.** To create functional centres of excellence that promote research and consultancy in the thrust sub-domains of theoretical computer science, systems and technology.
- **M 3.** To collaborate with industry and higher learning institutes of national/international repute and solve socially relevant problems.

Programme Educational Objectives (PEOs) for the B.Tech. (CSE) Programme:

Within few years after the end of the B.Tech. in Computer Science and Engineering programme, graduates will be able to:

PEO1	Effectively and efficiently function as Software Engineers and Technologists in organizations that deliver Computing or Information Technology based innovations, products and solutions.
PEO2	Enhance academic and research credentials by pursuing specialized postgraduate and / or doctoral education in institutes of higher learning in the niche sub-domains of theoretical Computer Science, Systems, Technology and Applications in the sub-disciplines including but not limited to AI, Data Analytics, Machine Learning, Security and Distributed Computing.
PEO3	Engage in continuous learning and skill enhancement to adapt and contribute to the global and national needs in terms of computing or IT based innovative solutions with a sincere practice of professional ethics and technical quality to meet or surpass the requirements of the stakeholders.

Programme Articulation Matrix (PEO vs. Mission) for the B.Tech. (CSE) Programme:

PEO\Mission	M1	M2	M3
PEO1	S	M	S
PEO2	S	S	M
PEO3	M	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Programme Outcomes (POs) for the B.Tech. (CSE) Programme:

At the end of any B.Tech. program in NIT Andhra Pradesh, graduates will be able to:

PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/Development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12	**Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSO) for the B.Tech. (CSE) Programme:

At the end of the B.Tech. in Computer Science and Engineering programme, graduates will be able to:

PSO1	Design, analyze, implement, verify and validate efficient solutions to complex engineering problems related to the ideation, development, testing and maintenance of computing systems.
PSO2	Construct or leverage contemporary tools, techniques, and frameworks in developing or refactoring a computing system or its component.
PSO3	Apply research-based methods to construct, implement, verify and validate analytical and simulation models for addressing generative, predictive, diagnostic and prescriptive tasks related to Natural Language Processing, Computer Vision, Automation and Security.

Degree Requirements for B.Tech. (CSE) Programme

	Proposed Credits (New regulation)
Basic Science Core (BSC)	**19 (11.72%)**
Engineering Science Core (ESC)	**22 (13.58%)**
Humanities and Social Science Core (HSC)	**06 (3.7%)**
Program Core Courses (PCC)	**63 (38.88%)**
Departmental Elective Courses (DEC)	**15 (9.25%)**
Open Elective Courses (OPC)	**09 (5.55%)**
Program Major Project (PRC)/Skill Development (SD)/Foreign Languages	**22 (13.58%)**
EAA: Games and Sports (MSC)	**2 (1.23%)**
MOOCs (MOE)	**4 (2.46%)**
Total	**162**

Choice Based Credit System: 26.54 %

NOTE: The minimum no. of credits required to award B.Tech. degree is 162 as per the proposed curriculum.

Credit Distribution in Each Semester										
	I	II	III	IV	V	VI	VII	VIII	TOT	REQ
BSC	8	8	3	0	0	0	0	0	**19**	**≥ 19**
ESC	4	10	4	0	4	0	0	0	**22**	**≥ 14**
HSC	3	0	0	0	0	0	3	0	**6**	**≥ 06**
PCC	0	0	13	20	12	11	7	0	**63**	**≥ 62**
DEC	0	0	0	0	0	6	6	3	**15**	**≥ 15**
OPC	0	0	0	0	3	3	0	3	**9**	**≥ 09**
PRC/ SD	5	2	0	2	0	3	4	6	**22**	**≥ 15**
EAA (MSC)	1	1	0	0	0	0	0	0	**2**	**≥ 2**
MOOCS (MOE)	0	0	0	0	2	0	0	2	**4**	**≥ 4**
	21	**21**	**20**	**22**	**21**	**23**	**20**	**14**	**162**	

I Year B.Tech. Course Structure

(Common for all branches)

Physics Cycle							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	MA101/ MA151	Differential and Integral Calculus / Matrices and Differential Equations	3	0	0	**03**	**BSC**
2	HS101	English for Technical Communication	2	0	2	**03**	**HSC**
3	PH101	Engineering Physics	3	0	0	**03**	**BSC**
4	EC101	Basic Electronics Engineering	2	0	0	**02**	**ESC**
5	CE102	Environmental Science and Engineering	2	0	0	**02**	**ESC**
6	CS101	Introduction to Algorithmic Thinking and Programming	3	0	0	**03**	**SD**
7	CS102	Introduction to Algorithmic Thinking and Programming Lab	0	1	2	**02**	**SD**
8	PH102	Engineering Physics Lab	0	1	2	**02**	**BSC**
9	EA101/ EA151	Physical Education/Health Education	0	0	3	**01**	**MSC**
		TOTAL	**15**	**2**	**9**	**21**	

Chemistry Cycle							
S. No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	MA101/ MA151	Differential and Integral Calculus / Matrices and Differential Equations	3	0	0	**03**	**BSC**
2	ME102	Engineering Graphics with Computer Aided Drafting	0	1	2	**02**	**ESC**
3	CY101	Engineering Chemistry	3	0	0	**03**	**BSC**
4	EE101	Elements of Electrical Engineering	2	0	0	**02**	**ESC**
5	BT101	Biology for Engineers	2	0	0	**02**	**ESC**
6	ME101	Basics of Mechanical Engineering	2	0	0	**02**	**ESC**
7	CE101	Engineering Mechanics	2	0	0	**02**	**ESC**
8	ME103	Workshop Practice	0	1	2	**02**	**SD**
9	CY102	Engineering Chemistry Lab	0	1	2	**02**	**BSC**
10	EA101/ EA151	Physical Education/Health Education	0	0	3	**01**	**MSC**
		TOTAL	**14**	**3**	**9**	**21**	

Note:

BSC: Basic Science Core

HSC: Humanities and Social Science Core

DEC: Departmental Elective Courses

Program Major Project (PRC)/Skill Development (SD)/Foreign Languages

ESC: Engineering Science Core

PCC: Program Core Courses

OPC: Open Elective Courses

EAA (MSC): Games and Sports MOOCs (MOE)

II Year B.Tech. CSE Course Structure

Summer Internship – I#

Semester-III							
S. No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	MA204	Probability, Statistics and Stochastic Processes	2	1	0	**03**	**BSC**
2	EC237	Digital Logic Design	3	0	2	**04**	**ESC**
3	CS201	Data Structures and Algorithms	3	0	0	**03**	**PCC**
4	CS202	Operating Systems	3	0	0	**03**	**PCC**
5	CS203	Discrete Mathematics	2	1	0	**03**	**PCC**
6	CS204	Data Structures and Algorithms Lab	0	1	2	**02**	**PCC**
7	CS205	Operating Systems Lab	0	1	2	**02**	**PCC**
		TOTAL	**13**	**4**	**6**	**20**	

Semester-IV							
S. No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	CS251	Object Oriented Programming	2	1	2	**04**	**PCC**
2	CS252	Design and Analysis of Algorithms	3	0	0	**03**	**PCC**
3	CS253	Computer Organization and Architecture	3	0	0	**03**	**PCC**
4	CS254	Theory of Computation	3	0	0	**03**	**PCC**
5	CS255	Database Management Systems	3	0	0	**03**	**PCC**
6	CS256	Unix Tools and Shell Scripting Lab	0	1	2	**02**	**PCC**
7	CS257	DBMS Lab	0	1	2	**02**	**PCC**
8	CS299	Mini Project – I (EPICS based)	0	1	2	**02**	**SD**
		TOTAL	**14**	**4**	**8**	**22**	

Summer Internship – II#

III Year B.Tech. CSE Course Structure

Semester-V							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	CS301	Language Processors	3	0	0	**03**	**PCC**
2	CS302	Theory and Design of Programming Languages	3	0	2	**04**	**PCC**
3	EC337	Microprocessors	3	0	2	**04**	**ESC**
4	CS303	Artificial Intelligence	3	0	0	**03**	**PCC**
5	CS304	Artificial Intelligence Lab	0	1	2	**02**	**PCC**
6		Open Elective – 1/ Foreign language	3	0	0	**03**	**OPC/SD**
7	MCS3YY	MOOCS-1	2	0	0	**02**	**MOE**
		TOTAL	**17**	**1**	**6**	**21**	

Semester-VI							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1		Department Elective –1	3	0	0	**03**	**DEC**
2		Department Elective –2	3	0	0	**03**	**DEC**
3	CS351	Software Engineering	2	0	2	**03**	**PCC**
4	CS352	Computer Networks	3	0	0	**03**	**PCC**
5	CS353	Web Application Development	2	0	2	**03**	**PCC**
6	CS354	Network Programming Lab	0	1	2	**02**	**PCC**
7		Open Elective – 2/ Foreign language	3	0	0	**03**	**OPC/SD**
8	CS399	Mini Project – II	0	0	6	**03**	**SD**
		TOTAL	**16**	**1**	**12**	**23**	

Summer Internship – III#

#: The student can do Summer Internship with duration of minimum 45 days at Institutes / Organizations / Industries and produce the certificate of completion and copy of internship report to the department.

It is optional only, Not Mandatory.

IV Year B.Tech. CSE Course Structure

Semester-VII							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1	SM430	Entrepreneurship for Engineers	3	0	0	**03**	**HSC**
2		Department Elective –3	3	0	0	**03**	**DEC**
3		Department Elective – 4	3	0	0	**03**	**DEC**
4	CS401	Cryptography and Engineering Secure Systems**	3	0	0	**03**	**PCC**
5	CS402	Big Data Engineering	2	1	2	**04**	**PCC**
6	CS449	Project-Work Part - A	0	0	8	**04**	**PRC**
		TOTAL	**14**	**1**	**10**	**20**	

**: The PCC Subject may be offered with the support of Industry.

Semester-VIII							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Cat. Code**
1		Department Elective – 5*	3	0	0	**03**	**DEC**
2		Open Elective – 3*	3	0	0	**03**	**OPC**
3	MCS4YY	MOOCS-2	2	0	0	**02**	**MOE**
4	CS499	Project-Work Part – B (with option of Industrial Training /Internship)	0	0	12	**06**	**PRC**
		TOTAL	**8**	**0**	**12**	**14**	

*If the students are in Industrial training, the electives may be conducted online.

List of Electives

Classification of Electives:

1: Artificial Intelligence

2: Data Engineering

3: Networks & Security

4: Software Engineering

5: Theoretical Computer Science

6: Miscellaneous Computer Science

III Year – II semester

Course Code	Elective name	credits	Elective class
CS361	Optimization Techniques	3-0-0	1
CS362	Design and Analysis of Parallel Algorithms	3-0-0	5
CS363	Graph Algorithms	3-0-0	5
CS364	Advanced Data Structures	3-0-0	5
CS371	Business Intelligence and Data Warehousing	3-0-0	2
CS372	Advanced Data Mining	3-0-0	2
CS373	Applied Machine Learning	3-0-0	1
CS374	Natural Language Processing	3-0-0	1
CS375	Advanced Computational Statistics	3-0-0	1,2
CS381	Agile Methodologies	3-0-0	4
CS382	Software Testing	3-0-0	4
CS383	Design Patterns	3-0-0	4
CS384	Internet of Things and Edge Computing	3-0-0	3

IV Year – I Semester

Course Code	Elective name	credits	Elective class
CS411	Randomized Algorithms	3-0-0	3,5
CS412	Performance Modeling of Computer Systems	3-0-0	3,5
CS413	Foundations of Data Science	3-0-0	1,2
CS421	Distributed Computing	3-0-0	2
CS422	Reinforcement Learning	3-0-0	1
CS423	Soft Computing	3-0-0	1
CS424	Probabilistic Graphical Models	3-0-0	1
CS425	Deep Learning for Vision	3-0-0	1
CS426	Advanced Database Systems	3-0-0	2

CS431	Wireless Technologies	3-0-0	3
CS432	Service Oriented Architecture	3-0-0	4
CS433	Cloud Computing	3-0-0	3
CS434	Blockchains	3-0-0	3
CS435	Network Security	3-0-0	3
CS436	Secure Software Engineering	3-0-0	3,4

IV Year – II Semester

Course Code	**Elective name**	**credits**	**Elective class**
CS461	Deep Learning for NLP	3-0-0	1
CS462	Social Network Analytics	3-0-0	1,2
CS463	Information Retrieval	3-0-0	2
CS471	Security and Privacy	3-0-0	3
CS472	Cyber Laws and IPR	3-0-0	6

Minor in Software Engineering: Course Structure							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Offered Sem**
1	CSM251	Algorithmics and Programming	2	0	2	**03**	**4th**
2	CSM301	Object Oriented Programming	3	0	0	**03**	**5th**
3	CSM302	Object Oriented Programming Lab	0	1	2	**02**	**5th**
4	CSM351	Database Management Systems	2	0	2	**03**	**6th**
5	CSM401	Web Programming	2	0	2	**03**	**7th**
6	CSM402	Software Engineering	1	0	2	**02**	**7th**
		TOTAL	**10**	**1**	**10**	**16**	

Honors in Data Science: Course Structure							
S. No.	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**	**Offered sem**
1	CSH301	Foundations of Data Science	4	0	0	**04**	**5th**
2	CSH302	Advanced Computational Statistics	4	0	0	**04**	**5th**
3	CSH351	Applied Machine Learning	3	0	2	**04**	**6th**
4	CSH352	Advanced Data Mining	4	0	0	**04**	**6th**
5	CSH401	Deep Learning/MOOC*	4	0	0	**04**	**7th**
		TOTAL	**19**	**0**	**2**	**20**	

* Any PG Level course related to Data Science with the approval of DAC PG&R

Note:

1. A student is permitted to do either Minor or Honors only, but not both
2. A student is permitted to do only one Minor/ one Honors.

I Year B.Tech. Course Structure

(Common for All Branches)

MA101	**Differential and Integral Calculus** I B.Tech. I Semester - all sections	BSC	3-0-0	3 Credits

Pre-requisites: None

Differential Calculus of functions of several variable: Review of Limit, continuity (sequential verification) and differentiability, Partial differentiation; Total differentiation; Euler's theorem and generalization; Change of variables- Jacobians; Maxima and minima of functions of several variables (2 and 3 variables); Lagrange's method of multipliers. (14)

Integral Calculus: Convergence of improper integrals; Beta and Gamma integrals; Differentiation under integral sign; Double and Triple integrals - computation of surface areas and volumes; change of variables in double and triple integrals. (14)

Vector Calculus: Scalar and vector fields; vector differentiation; level surfaces; directional derivative; gradient of a scalar field; divergence and curl of a vector field; Laplacian; Line and Surface integrals; Green's theorem in a plane; Stokes' theorem; Gauss Divergence theorem. (14)

Text Reference:

1. Joel R. Hass, Maurice D. Weir, George B. Thomas, Thomas' Calculus, 12th edition, Pearson , 2010.
2. Erwin Kreyszig, *"Advanced Engineering Mathematics"*, Eighth Edition, John Wiley and Sons, 2015
3. B. S. Grewal, *"Higher Engineering Mathematics"*, Khanna Publications, 2015
4. R. K. Jain and S. R. K. Iyengar, *"Advanced Engineering Mathematics"*, Fifth Edition, Narosa Publishing House, 2016.
5. T. M. Apostol, Calculus, Volumes 1 and 2 (2nd Edition), Wiley Eastern, 1980.

MA151	Matrices and Differential Equations I B.Tech. II Semester - all sections	BSC	3-0-0	3 Credits

Pre-requisites: Mathematics-I

Matrix Theory: Linear dependence and independence of vectors; Rank of a matrix; Consistency of the system of linear equations; Eigenvalues and eigenvectors of a matrix; Caley-Hamilton theorem and its applications; Reduction to diagonal form; Reduction of a quadratic form to canonical form - orthogonal transformation; Properties of complex matrices - Hermitian, skew-Hermitian and Unitary matrices. (14)

Ordinary Differential Equations of Higher Order: Higher order linear differential equations with constant coefficients - homogeneous and non-homogeneous; Euler and Cauchy's differential equations; Method of variation of parameters; System of linear differential equations; applications in physical problems - forced oscillations, electric circuits, etc. (14)

Laplace Transforms: Laplace transforms; inverse Laplace transforms; Properties of Laplace transforms; Laplace transforms of unit step function, impulse function, periodic function; Convolution theorem, Solving certain initial value problems, Solving system of linear differential equations, Finding responses of systems to various inputs viz. sinusoidal inputs acting over a time interval, rectangular waves, impulses etc. (14)

Text Reference:

1. E. Kreyszig, Advanced Engineering Mathematics, Eighth Edition, John Wiley and Sons, 2015.
2. B. S. Grewal, Higher Engineering Mathematics, Khanna Publications, 2015.
3. R. K. Jain and S. R. K. Iyengar, Advanced Engineering Mathematics, Fifth Edition, Narosa Publishing House, 2016.
4. G. Strang, Linear Algebra and Its Applications, 4th Edition, Brooks/Cole India, 2006.
5. T. M. Apostol, Calculus, Volume 2 (2nd Edition), Wiley Eastern, 1980.

HS101	ENGLISH FOR TECHNICAL COMMUNICATION	SD	2 – 0 – 2	3 Credits

Pre-requisites: None.

Detailed syllabus

Grammar Principles and Vocabulary Building: -Exposure to basics of grammar-tenses—active and passive voice- their usage-Concord -Error Detection-Idioms and Phrases-Phrasal verbs—their meanings and usage, Synonyms and antonyms

Developing paragraphs using mind mapping- Definition- structure- Types and Composition-unity of theme- coherence- organization patterns-essays and their structure-note-making

Letter Writing: Formal letters-- communicative purpose-strategy- letter format and mechanics- letters of request, complaint and invitation-writing emails
Reading Comprehension –skimming-scanning-intensive and extensive reading-reading to retrieve information ––techniques of comprehension -find clues to locate important points- answering objective type questions–inference, elimination

Delegation- steps involved in delegation-preparing delegation for a program
Preparing Questionnaire-Determine audience and content of each question-response structure-develop wording for each question-establish sequence of questions

Profiling Readers-Audience analysis- Identifying potential audience- Identifying primary, secondary, tertiary readers,and gatekeepers- Identifying the needs, values, and attitude of the readers
Resume Writing-Writing for Professional Networking-Academic writing-research proposals-Interpretation of Graphs.

Technical Report-Writing - kinds of reports-proposals, progress and final reports- their structure- features- process of writing a report-editing.

Language Laboratory
Introduction to basic phonetics: Vowels, Consonants, Diphthongs, phonetic symbols
Listening: Challenges in listening, enhancing listening skills, listening activities
Speaking:JAM using cue cards-role play-Group presentation-presentation with emphasis on body language- public speaking-extempore speech
Group discussion: Dos and don'ts, intensive practice
Mock interview:Interview etiquette, common interview questions

Text Books:

Emden, Joan van. *Effective Communication for Science and Technology*. Macmillan Education UK, 2001.
Mohan, Krishna and Meera Banerji. *Developing Communication Skills*. Macmillan India Limited, 2000.

Murphy, Raymond. *Intermediate English Grammar*. Cambridge University Press, 2014.

Narayanaswami, V. R. *Strengthen Your Writing*. Orient Longman Private Limited, 2005.

Soundaraj, Francis. *Speaking and Writing for Effective Business Communication*. Macmillan Publishers India Limited, 2007.

Ur, Penny. *Discussions that Work*. Cambridge University Press, 1981.

Reference:

Aarts, Bas. *Oxford Modern English Grammar*. Oxford University Press, 2011.

Anderson, Marilyn, Pramod K. Nayar, and Madhucchanda Sen. *Critical Thinking, Academic Writing and Presentation Skills*. Pearson Education, 2008.

Blake,Gary.*The Elements of Technical Writing*. Pearson,2000

Brown, Carla L. *Essential Delegation Skills*. Routledge, 2017.

Busan, Tony. *Mind Map Mastery*. Walkins, 2018.

Carlisle, Joanne and Melinda S. Rice. *Improving Reading Comprehension Research-based Principles and Practices*. York Press, 2002.

Carter, Ronald and Michael McCarthy. *Cambridge Grammar of English: A Comprehensive Guide*. Cambridge University Press, 2006.

Carter, Ronald, Rebecca Hughes, and Michael McCarthy. *Exploring Grammar in Context: Upper-intermediate and Advanced.* Cambridge University Press, 2000.

Eastwood, John. *Oxford Guide to English Grammar*. Oxford University Press, 1994.

Harris, David.F. *Complete Guide to Writing Questionnaires.* I& M Press, 2014.

Hering, Lutz and Heike Hering. *How to Write Technical Reports: Understandable Structure, Good Design, Convincing Presentation*. Springer; 2010.

HuckinN.Thomas and Leslie A.Olsen*Technical Writing and Professional Communication for Non-native Speakers*. McGraw-Hill Education,1991.

Laplante, Phillip A*. Technical Writing: A Practical Guide for Engineers, Scientists, and Nontechnical Professionals.* CRC Press, 2018.

McQuail, Dennis. *Audience Analysis*. Sage, 1997

Ogden, Richard. *Introduction to English Phonetics*. Edinburgh University Press, 2017.

Parker, Glenn M. *Team Players and Teamwork: New Strategies for Developing Successful Collaboration*. Wiley, 2011.

Seely, John. *Oxford Guide to Effective Writing and Speaking: How to Communicate Clearly*. Oxford University Press: 2013.

PH101	Engineering Physics	BSC	3-0-0	3 Credits

Waves and Optics

Interference: Superposition principle, coherence of light, methods to produce coherent light: division of amplitude and wave front division, Young's double slit experiment: concept, working principle, and applications, Newton's ring: concept, working principle, and applications

Diffraction: Fraunhofer's single-slit diffraction, diffraction grating, and resolving power of a grating.

Polarization: Types of optical polarization, various methods to produce polarized light, working and applications of retarder plates, and half-shade polarimeter: construction and working principle.

Lasers and Optical Communication

LASER: Basic theory of LASER, Einstein's coefficients and their relations, concept of population inversion, components of lasers, modes of laser beam, construction and working principle of various types of lasers: Ruby, Helium-Neon, and semiconductor diode lasers.

Optical Fibre: Optical fibre and its working principle, total internal reflection, numerical aperture, modes of propagation, and classification of optical fibres.

Quantum Physics

Origin of quantum theory and related experiments: Black-Body radiation, photo-electric effect, and Compton effect. Heisenberg's uncertainty principle, de- Broglie's wave concept, phase and group velocities, wave function, and its properties, operators, Schrödinger's time-dependent and time-independent equations, particle in one-dimensional, infinite potential and finite potential wells, and quantum tunneling phenomena and their applications in alpha decay, and scanning tunneling microscopy (STM).

Magnetic, Superconducting and Dielectric Materials

Magnetic Materials: Introduction to Weiss theory of ferromagnetism, concepts of magnetic domains, Curie transition, hard and soft magnetic materials and their applications, magneto-resistance, GMR, and TMR.

Superconducting Materials: Introduction to superconductivity, Meissner effect, Type-I and Type-II superconductors and their applications.

Dielectric Materials: Introduction to dielectrics, dielectric constant, polarizability, frequency and temperature dependent polarization mechanism in dielectrics, dielectric loss, and applications.

Advanced Functional Materials & NDT

Smart Materials: Biomaterials, high-temperature materials and smart materials, applications of functional materials.

Nanomaterials: Introduction, classification, and properties of nanomaterials, various methods of synthesizing nanomaterials: top-down (ball milling) and bottom-up (sol-gel) approaches.

Photovoltaic Materials: Solar spectrum, photovoltaic effect, materials, structure and working principle, I-V characteristics, power conversion efficiency, quantum efficiency, emerging PV technologies, and applications.

NDT: Methods of non-destructive testing

References:

1. A Textbook of Engineering Physics, M. N. Avadhanulu, P. G. Kshirsagar, S. Chand and Company (2015).
2. Concepts of Modern Physics, Beiser A., Mc. Graw Hill Publishers (2003).
3. Optics, Ajoy Ghatak, Tata Mc Graw Hill (2012).
4. Materials Science and Engineering: An Introduction (Tenth edition), William D. Callister, John Wiley & Sons (2018).
5. Introduction to Solid State Physics, Charles Kittel, Wiley Publishers (2011).

EC101	Basic Electronics Engineering	ESC	2 – 0 – 0	2 Credits

Detailed Syllabus:

Introduction to electronics systems, diode circuit models and applications, Zener diode as regulator, photodiode.

Transistor and applications: Introduction to transistors, BJT Characteristics, biasing and applications. FET and MOSFET characteristics and applications.

Feedback in Electronic Systems: open loop and closed loop systems, Negative and positive Feedback, Principles of LC and RC oscillators.

Integrated Circuits: Operational amplifiers Characteristics and applications, linear operations using Op-amps.

Digital Circuits: Number systems and logic gates, Combinational Logic circuits, Sequential Circuits, Analog to Digital and Digital to Analog converters (ADC/DAC).

Laboratory measuring instruments: principles of digital multi-meters, Cathode ray oscilloscopes (CRO).

Reading:

1. Bhargava N. N., D C Kulshreshtha and S C Gupta, Basic Electronics & Linear Circuits, 2nd Edition, Tata McGraw Hill, 2013.
2. S. Sedra and K. C. Smith, Microelectronic Circuits, Oxford University Press , 6th Edition
3. Leach , Malvino, Saha, Digital Principles and Applications, McGraw Hill Education , 8th Edition
4. Boylestad, Robert L., Louis Nashelsky, Electronic Devices and Circuit, Pearson , 11th Edition
5. Helfrick and Cooper, ― Modern Electronic Instrumentation and Measurement Techniques‖ PHI, 2011
6. Neil Storey, Electronics A Systems Approach, 4th Edition, Pearson Education Publishing Company Pvt Ltd.

CE102	ENVIRONMENTAL SCIENCE AND ENGINEERING	ESC	2-0-0	2 Credits

Detailed Syllabus:

Introduction to Environmental Science: Environment and Societal Problems, Major Environmental Issues, Global Climate Change Agreements, Montreal, Kyoto Protocol & Paris Agreement, Basics of Environmental Impact Assessment, Principles of Sustainability, and related indices, Population Dynamics, Urbanization. Identification and Evaluation of Emerging Environmental Issues with Air, Water, Wastewater and Solid Wastes, Introduction to Environmental Forensics.

Water & Wastewater Treatment: Water Sources, constituents, potable water quality requirements (IS 10500), overview of water treatment, sources and types of pollutants, their effects, self-purification capacity of water bodies, principles of wastewater treatment, 5R Concept.

Air & Noise Pollution: Sources, classification and their effects, national ambient air quality standards (NAAQS), air quality index, dispersion of pollutants, control of air pollution, understanding and improving indoor air quality, sources of noise pollution, effects, quantification of noise pollution.

Solid Waste Management: Sources and characteristics of solid waste, effects, 3R concept, sustainable practices in waste management, CPHEEO guidelines for solid waste management, transition to zero waste lifestyle.

Reading:

1. G.B. Masters, Introduction to Environmental Engineering and Science, Pearson Education, 2013.
2. Gerard Kiely, Environmental Engineering, McGraw Hill Education Pvt Ltd, Special Indian Edition, 2007.
3. Benny Joseph, Environmental Science and Engineering, Tata McGraw-Hill, New Delhi, 2006.

References:

1. Peavy, H.S, Rowe, D.R., and G. Tchobanoglous (1985), Environmental Engineering, McGraw Hill Inc., New York
2. W P Cunningham, M A Cunningham, Principles of Environmental Science, Inquiry and Applications, Tata McGraw Hill, Eighth Edition, 2016.

CS101	Introduction to Algorithmic Thinking and Programming	SD	3 – 0 – 0	3 Credits

Pre-requisites: None

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

CO1	Construct algorithms for solving problems that requires solutions involving searching, sorting, selection and / or a numerical method as a sub-routine.
CO2	Analyze the suitability of different algorithmic design paradigms for solving problems with an understanding of the time and space complexities incurred.
CO3	Construct algorithms for solving problems with an understanding of the internals of a computing system and its components like processor, memory and I/O sub-systems.
CO4	Construct efficient modular programs for implementing algorithms by leveraging suitable control structures.
CO5	Construct efficient programs by selecting and using suitable in-built Data Structures and programming language features available.

Course Articulation Matrix:

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L									
CO2	S	M	L									
CO3	S	M	L		L							
CO4	S	M	L		S							
CO5	S	M	L		S							

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Fundamentals of Computers, Historical perspective, Early computers, Modern Computers, Hardware Components of a Computer, Data Representation in Computers, Introduction to Operating Systems, Software and Firmware, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

Problem solving techniques – Algorithmic approach, characteristics of algorithms, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms, Algorithm Analysis.

Basic Syntax in Python, Data Types, Variables, Assignments, immutable variables, Types of Operators, Expressions, Comments, Boolean Logic, Logical Operators in Python.

Conditional statements - If-else, Loops - while, for, Lazy Evaluation

Inbuilt Data Structures and their operations in Python: List, Tuples and Dictionaries.

Fundamental Algorithms: Swapping variables, Problems involving summation of a series, Sine function computation, Base Conversion, generation of sequences like Fibonacci, Reversing the digits of an integer, Character to number conversion.

Factoring Methods: Finding the square root, Finding the smallest divisor of an integer, finding the greatest common divisor using Euclid's algorithm, Computing the prime factors of an integer, generating prime numbers, Raising a number to a large power, Computation of the nth Fibonacci number.

Functions – Modular programming and benefits, user defined functions, library functions, parameter passing, Formal and Actual arguments, named arguments return values, Recursion.

Sorting algorithms: Bubble, Selection and Insertion sorts, Search algorithms: Linear and binary search

String processing: Algorithms for implementing String functions like Strlen, Strcpy, StrRev, Strcmp, Searching for a keyword or pattern in a text.

File and Directory Handling: Reading and Writing to/from a file, Formatted File creation and operations.

Simple 2D Graphics, drawing 2D objects using Turtle Graphics.

Reading List:

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019
2. R.G. Dromey, how to solve it by Computer, Pearson, 2008.

CS102	Introduction to Algorithmic Thinking and Programming Lab	SD	0 – 1 – 2	2 Credits

Pre-requisites: None

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

CO1	Construct, debug, test and run efficient programs by leveraging suitable flow of control constructs and syntactic units of the programming language.
CO2	Construct efficient programs by constructing and translating algorithms for solving problems using sorting, searching, selection and / or arithmetic computations.
CO3	Implement, refactor, test and debug functional programs in a shell-based run time environment.
CO4	Construct efficient programs by demonstrating problem-solving skills and out-of-the-box algorithmic thinking.

Course Articulation Matrix:

PO / CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	S	M	L		S				M			L
CO2	S	M	L		S				M			L
CO3	S	M	L		S				M			L
CO4	S	M	L		S				M			L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Familiarization with Python installation, basic syntax and running scripts in the shell.
2. Programs on conditional control constructs.
3. Programs on iterative constructs. (While, do-while, for).
4. Programs using user defined functions and in-built function calls.
5. Programs related to Recursion.
6. Programs involving in-built data structures like List, Tuples and Dictionaries.

7. Programs related to String processing.
8. Programs related to Files and I/O.
9. Implementation of Factoring methods.
10. Programs that require sorting, searching and selection as sub-routines.
11. Problems involving simple 2D graphics.
12. Implementation of a capstone application to unify the concepts learnt in the course.

Reading List:

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019.
2. R.G. Dromey, how to solve it by Computer, Pearson, 2008.
3. The Python Tutorial, Available at: https://docs.python.org/3/tutorial/.

PH102	Engineering Physics Lab	BSC	0-0-2	2 Credits

List of experiments (any eight of the following):

Exposure to virtual lab (any three of the following):

S. No	Name of the experiment
1	Determination of Planck's constant using light emitting diode.
2	Determination of wavelength of monochromatic light in Newton's ring experiment.
3	Determination of the width of narrow slit by diffraction method.
4	Determination of wavelength of He-Ne laser using diffraction by a metal scale.
5	Determination of capacitance and time constant of a capacitor using R-C circuit.
6	Determination of wavelength of mercury spectrum by normal incidence method (diffraction grating).
7	Determination of specific rotation of an optically active material-using Laurent's half-shade polarimeter.
8	Determination of resonating frequency and bandwidth of an LCR circuit.
9	Determination of dielectric constant of various dielectric materials.
10	Studying B-H curve loop and permeability of magnetic materials.
11	Measuring spatial distribution of magnetic field between a pair of identical coils using Helmholtz coils.
12	Studying current-voltage characteristics of a photovoltaic material using solar cell.
13	Determination of numerical aperture of an optical fibre.
14	Determination of resistivities of various materials using four-probe method.

1. LCR – Series/Parallel
2. B-H Loop tracer
3. Planck's Constant
4. Numerical aperture of Optical Fiber
5. Newton's rings

Micro project:

This can be implemented in the subsequent semesters based on the facilities available. In the case of implementation, three or four experiments from the above listed eight experiments will be replaced with the project (~40 % of the experiments will be relaxed).

References:

1. *Physics Laboratory Manual*, School of Sciences (Physics), National Institute of Technology Andhra Pradesh (2020).

2. *Practical Physics (Electricity, Magnetism, and Electronics),* R. K. Shukla, A Srivastava, New age international publishers (2011).

3. *B.Sc. Practical Physics*, C. L. Arora, S. Chand & Co. Ltd. (2012).

EA101	Physical Education	MSC	0-0-3	1 Credit

Detailed Syllabus:

I. Introduction to Physical Education & EAA = Sports and Games

Meaning & Definition of Physical Education, Aims & Objectives of Physical Education, Importance of Physical Education

II. Physical Fitness & Wellness Lifestyle

Meaning & Importance of Physical Fitness, Components of Physical Fitness (Cardiovascular Endurance, Strength Endurance Muscular Endurance, Flexibility, Body Composition), Components of Motor Fitness (Agility, Balance, Power, Speed, Coordination), Development of Fitness Components

III. Training Methods in Physical Education

Circuit Training (Circuit Training), Continues Training (Endurance), Interval Training (Speed & Endurance), Fartlek Training (Speed Endurance), Weight Training (Maximum Strength), Plyometric Training (Power), Flexibility Training

IV. Test & Measurements

Measurements: Height, Weight, Age, Calculation of BMI, Motor Fitness and Physical Fitness Tests (Pre - Test & Post-Test), Cardiovascular Endurance - 9/12 Minute Run or Walk, Muscular Endurance – Sit Ups for abdominal strength, Strength Endurance – Flexed arm hang for girls / Pull ups for boys, (Speed – 50m Dash or 30mts Fly Start, Strength – Broad Jump, Vertical Jump for Lower Body, Medicine Ball Put for Shoulder Strength, Endurance - 800mts, Flexibility - Bend and Reach, Agility (Coordination)) – Shuttle Run and Box Run

V. Formal Activities

Calisthenics (free hand exercises), Dumbbells, Woops, Wands, Laziums (Rhythmic activities), Aerobic Dance and Marching

VI. Sports / Games

Following sub topics related to any one Game/Sport of choice of student out of: Athletics, Badminton, ball badminton, Kabaddi, Kho-Kho, Table Tennis, Yoga etc., Teaching & Coaching of the Game/Sport, Latest General Rules of the Game/Sport.

Specifications of Play Grounds and Related Sports Equipment

EA151	Health Education	MSC	0-0-3	1 Credit

Health Education & Personal Hygiene

Introduction & Meaning of Health Education, Definition of Health Education, Principles of Health Education, Importance of Health Education, Meaning of Personal Hygiene, Importance of Personal Hygiene, Personal cleanliness (teeth, ears, eyes, nose & throat, nails & fingers, skin, cloths, and hair).

Nutrition

Introduction of Nutrition, Balanced Diet, Daily Energy Requirements, Nutrient Balance, Nutritional Intake, Eating and Competition, Ideal Weight

First Aid & Injury Management

Introduction, Types and Principles of First Aid, Functions of First Aider, Reasons for Sports Injuries,The First Aid and Emergency Treatment in Various cases (drowning, dislocation & fractures, burns, electric shock, animal bite, snake bite, poison, etc.

Human Posture

Introduction, Meaning of Posture, types of Good Posture, causes of Poor Posture, preventive and Remedial Poor Posture, common Postural Deformities, Body Types, Advantages of Good Posture

Yoga

Introduction, Meaning & Importance of Yoga, Elements of Yoga, Introduction - Asanas, Pranayama, Meditation & Yogic Kriyas, Yoga for concentration & related Asanas (standing asanas, sitting asanas, supine and prone postures.), Relaxation Techniques for improving concentration – Yoga – nidra, Pranayama

Sports / Games

Following sub topics related to any one Game/Sport of choice of student out of: Athletics, Badminton, ball badminton, Kabaddi, Kho-Kho, Table Tennis, Yoga etc.,Teaching & Coaching of the Game/Sport., Latest General Rules of the Game/Sport, Specifications of Play Grounds and Related Sports Equipment.

ME102	Engineering Graphics with Computer Aided Drafting	ESC	2-0-0	2 Credits

Note: 50% of the Practice through manual drawing and 50% of the Practice through a Computer Aided Drafting Package.

Detailed Syllabus:

Introduction: Overview of the course, Lines Lettering and Dimensioning: Types of lines, Lettering, Dimensioning, Geometrical Construction of Polygons, Scales. Introduction to Computer Aided Drafting (CAD), DRAW tools, MODIFY tools, TEXT, DIMENSION, PROPERTIES, etc.

Orthographic Projection: Principles, of Orthographic projection, Four Systems of Orthographic Projections.

Projection of Points: Projections of points when they are situated in different quadrants.

Projections of Lines: Projections of a line parallel to one of the reference planes and inclined to the other, line inclined to both the reference planes, Traces.

Projections of Planes: Projections of a plane perpendicular to one of the reference planes and inclined to the other, Oblique planes.

Projections of Solids: Projections of solids whose axis is parallel to one of the reference planes and inclined to the other, axis inclined to both the planes.

Sections of Solids: Sectional planes, Sectional views - Prism, pyramid, cylinder and cone, true shape of the section.

Isometric Views: Isometric axis, Isometric Planes, Isometric View, Isometric projection, Isometric views - simple objects.

Reading:

1. N.D. Bhatt and V.M. Panchal, Engineering Graphics, Charotar Publishers, 2013.

2. Sham Tickoo, AutoCAD 2017 for Engineers & Designers, Dreamtech Press, 23 rd Edition, 2016.

CY101	Engineering Chemistry	BSC	3-0-0	3 Credits

Basic Organic Chemistry

Reaction intermediates: carbocations, carbanions, free radicals and carbenes. Classification of organic reactions, examples and their mechanisms: substitution, addition, elimination and rearrangement reactions. Reimer–Tiemann reaction, Kolbe-Schmidt reaction, Cannizzaro reaction. Pinacol-Pinacolone, Hofmann and Beckmann rearrangements. Diels-Alder reaction.

Spectroscopic Techniques for Chemical Analysis

Introduction of spectroscopy, Quantum aspects of electronic, vibrational and nuclear energy levels. UV-Visible spectroscopy: Principle, Instrumentation, Beer-Lambert's law, Effect of conjugation, Woodward-Fieser empirical rules for acyclic/cyclic dienes. IR spectroscopy: Principle, Factors that affect vibrational frequencies and functional group detection. Proton NMR spectroscopy: Principle, Instrumentation, Chemical equivalency, Chemical shift and spin-spin splitting. Applications of UV-Vis, IR and proton-NMR spectroscopy in determining the structure of small organic molecules.

Coordination Chemistry

Introduction of coordination chemistry, Valence bond (VB) theory and shapes of Inorganic Compounds, Spectrochemical series, Crystal Field theory (CFT): octahedral and tetrahedral complexes, Crystal field splitting energy (CFSE); Molecular Orbital (MO) Theory: Molecular orbital diagrams for octahedral complexes (strong and weak ligand fields).

Electrochemistry

Electrodes, Electrochemical Cells, Electrochemical series and Nernst equation; Conductometry and Potentiometry; Batteries: Types of batteries, Ni-Cd and Lithium (Li)-ion batteries; Fuel Cells: Hydrogen-Oxygen, Methanol-Oxygen fuel cells; Corrosion - Theories of corrosion, Wet corrosion, Types of wet corrosion, Factors affecting the rate of corrosion, Corrosion control methods: Sacrificial anode method and Impressed current method.

Engineering Materials and Applications

Polymers: Introduction, Types of polymerization, Functionality in polymers, Number and Weight average molecular weight, Polydispersity index, Biodegradable polymers; Conductive polymers: classification, examples and applications; Organic light emitting diode (OLED): structure, principle and applications; Optical fibres: principle and Applications.

Reference books:

1. Organic Chemistry, Clayden, Greaves, Warren and Wothers, Oxford University Press, 2014.
2. Organic Spectroscopy, William Kemp, 2nd edition, Macmillan publishers, 2019.
3. Advanced Inorganic Chemistry, F. Albert Cotton, Geoffrey Wilkinson, Carlos A. Murillo and Manfred Bochmann, 6th Edition, 1988.
4. Physical Chemistry, P. Atkins and Julio de Paula, 8th Edition, Freeman & Co. 2017.
5. A Textbook of Engineering Chemistry, Shashi Chawla, 2017.
6. Polymer Science and Technology, Premamoy Ghosh, 3rd edition, McGraw-Hill, 2010.

EE101	Elements of Electrical Engineering	ESC	2-0-0	2 Credits

Detailed Syllabus

Basic Concepts

Electric Charge, Current and Electromotive force, Potential and Potential Difference; Electrical Power and Energy; Ohm's Law, Resistance, Capacitance and Inductance, Series and Parallel Connection of Resistances and Capacitances, Kirchoff's Laws and Their Applications

AC Fundamentals:

Concept of Alternating Voltage and Current, RMS and Average Values, Single Phase and Three Phase Supply; 3-ph Star-Delta connections, Alternating Voltage applied to Pure Resistance, Inductance, Capacitance and their combinations, Concept of Power and Power Factor in AC Circuit.

Measuring Instruments:

Principle and Construction of Instruments used for Measuring Current, Voltage, Power and Energy, Methods and precautions in use of these.

Electromagnetic Induction:

Concept of Magnetic Field, Magnetic Flux, Reluctance, Magneto Motive Force (MMF), Permeability; Self and Mutual Induction, Basic Electromagnetic laws, various losses in magnetic circuits;

Electrical Machines:

Elementary concepts of an electrical machine, Basic principle of a motor and a generator, Classification of Electrical machines; Principles, Construction and Working of a machine; Starters: Need, Construction and Operation; Transformer: Classification, Principles, Construction and Working of a Transformer, Applications of Transformers;

Utilization of Electricity:

Utilization concepts of Electricity for electrolysis process, Electrochemical Cells & Batteries; Application of Electricity, Energy Conversation and Efficiency

Basic Troubleshooting:

Basic Testing and faults diagnosis in electrical systems, various tools and their applications, replacement of different passive components.

Electrical Safety:

Electrical Shock and Precautions against it, Treatment of Electric Shock; Concept of Fuses and Their Classification, Selection and Application; Concept of Earthing.

Reading:

1. Edward Hughes, Electrical & Electronic Technology, Pearson, 12 th Edition, 2016.

2. Vincent Del Toro, Electrical Engineering Fundamentals, Pearson, 2 nd Edition, 2015.

3. V N Mittle and Arvind Mittal, Basic Electrical Engineering, Tata McGraw Hill, 2^{nd} Edition, 2005.

4. E. Openshaw Taylor, Utilization of Electrical Energy, Orient Longman, 2010.

5. B.L.Theraja , Fundamentals of Electrical Engineering and Electronics volume -I, S Chand & Company 2005.

6. Ashfaq Husain, Fundamentals of Electrical Engineering, Dhanpat Rai & Sons 4 th edition, 2010.

7. H.Partab: Art & Science of Utilization of Electric Energy, Dhanpat Rai & Sons, 1998.

8. Fundamentals of Electrical Circuits by Charles k.Alexander, Mattew N.O.Saidiku, Tata McGraw Hill company.

BT101	BIOLOGY FOR ENGINEERS	ESC	2-0-0	2 Credits

Pre-requisites: None

Detailed Syllabus:

Importance of biology to engineers, Molecules of life: Water and Carbon, Evolution and origin of life, Darwins theory, Diversity of life, Chemical basis of life, Nucleic acids, Amino acids and Proteins, Carbohydrates, Lipids and Membranes.

Cell structure and function:

Prokaryotic, Eukaryotic cell and Virus, Sub cellular organelles and their functions, Regulation of cellular metabolism: Cellular respiration and Fermentation, Photosynthesis, Cell division (differences between mitosis and meiosis), Mendel's Law and Patterns of inheritance.

Gene structure and expression

Difference between prokaryotic and eukaryotic gene structure, DNA replication, Transcription, RNA processing and Translation, Control of gene expression (lac operon).

Applications of Biology in Engineering

Genetic engineering (microbe, plant and animal cells for improvement), Industrial Biotechnology (Primary and Secondary metabolites), Environmental engineering, Biopharmaceuticals, Tissue engineering, Biomaterials, Stem cell engineering, Biosensors, Bioinformatics.

Reading:

1. Quillin, Allison Scott Freeman, Kim Quillin and Lizabeth Allison, Biological Science, Pearson Education India, 2016.

2. Reinhard Renneberg, Viola Berkling and Vanya Loroch, Biotechnology for Beginners, Academic Press, 2017.

ME101	Basics of Mechanical Engineering	ESC	2-0-0	2 Credits

Detailed Syllabus:

Evolution of Mechanical Engineering: Introduction, Definition and scope of Mechanical Engineering, relation of Mechanical Engineering with other Engineering Disciplines, Revolutionary Inventions in wheels, tools, windmills, steam engine, CNC machines, Rapid Prototyping, Air-conditioning and Refrigeration, History of Mechanics, Thermodynamics and Heat Transfer, Production and Industrial Engineering, Mechatronics.

Engineering Materials: Introduction to Engineering Materials, Classification and Properties, Alloys. Composites, Micro and Nano Materials.

Manufacturing Processes: Castings - Patterns & Moulding, Metal forming, Hot Working and Cold Working Extrusion, Drawing, Rolling, Forging. Welding - Arc Welding & Gas Welding, Soldering, Brazing. Introduction to Machining processes – Lathe, Milling, Shaping, Drilling, Grinding, Introduction to NC/CNC Machines, 3D Printing.

Power Transmission: Transmission of Power, Belt Drives, Gears and Gear Trains - Simple Problems, **Fasteners and Bearings:** Fasteners - Types and Applications, Bearings - Types and Selection,

Thermodynamics: Introduction to Energy Sources - Thermodynamics - System, State, Properties, Thermodynamic Equilibrium, Process & Cycle, Zeroth law of Thermodynamics, Work & Heat, First law - Cyclic process, Change of State, Cp, Cv, Limitations of First law, Thermal Reservoirs, Heat Engine, Heat Pump/Refrigerator, Efficiency/COP, Second law, PMM2, Carnot Cycle, Entropy - T-S and P-V diagrams.

Introduction to Steam Turbines and I.C. Engines: I.C. Engines: 2-Stroke & 4-Stroke Engines, P-v Diagram; S.I. Engine, C.I. Engine, Differences.

Introduction to Heat Transfer and Refrigeration: Vapor Compression Refrigeration Cycle - Refrigerants, Desirable Properties of Refrigerants. Modes of Heat Transfer, Thermal Resistance Concept, Composite Walls & Cylinders, and Overall Heat Transfer Coefficient – problems.

Reading:

1. Dixit, U.S., Hazarika, M. and Davim, J.P, A Brief History of Mechanical Engineering, Springer, 2017.
2. M.L. Mathur, F.S. Mehta and R.P. Tiwari, R.S. Vaishwnar, Elements of Mechanical Engineering, Jain Brothers, New Delhi, 2008.
3. Praveen Kumar, Basic Mechanical Engineering, Pearson Education, India, 2013.
4. P.N. Gupta, M.P. Poonia, Elements of Mechanical Engineering, Standard Publishers, 2004.
5. C.P. Gupta, Rajendra Prakash, Engineering Heat Transfer, NemChand Brothers, New Delhi, 1994.

6. B.S. Raghuvanshi, Workshop Technology, Vol. 1&2, Dhanpath Rai & Sons, New Delhi, 1989.

CE101	Engineering Mechanics	ESC	2-0-0	2 Credits

Prerequisites: None

Detailed syllabus:

Introduction - Specification of force vector, Formation of Force Vectors, Moment of Force – Cross product – Problems, Resultant of a general force system in space,

Equillibrium of force system- Degrees of freedom - Equilibrium Equations, Degree of Constraints – Free body diagrams.

Coplanar Force Systems - Introduction – Equilibrium equations – All systems, Problems

Coplanar Concurrent force system, Coplanar Parallel force system, Coplanar General force system – Point of action, Method of joints, Method of sections, Method of members.

Friction in rigid bodies- Friction – Coulombs laws of dry friction – Limiting friction, Problems on Wedge friction, Belt Friction-problems.

Centroid & Moment of Inertia - Centroid and M.I – Arial – Radius of Gyration, Parallel axis– Perpendicular axis theorem – Simple Problems.

Dynamics of Particles – Introduction to kinematics- Equations of rectilinear motion, D'Alembert's principle -Simple problems- Introduction to kinetics- Work and Energy.

Reading:

1. J.L.Meriam, L.G. Kraige, Engineering Mechanics, Statics, John Wiley &Sons,7th Edition, 2012.
2. A.K. Tayal, Engineering Mechanics, Umesh Publications, 14th Edition, 2010.
3. S S Bhavikatti and K G Rajashekarappa, Engineering Mechanics, New Age International Publication, 4th Edition.

Reference:

1. Dietmar Gross, Werner Hauger, Jorg Schroder, Wolfgang A. Wall, Nimal Rajapakse, Engineering Mechanics 1, Statics, Springer, 2nd Edition, 2013.

S. Timoshenko, D.H. Young, Pati Sukumar, J V Rao, Engineering Mechanics, Mc-Graw Hill, 5th Edition.

ME103	Workshop Practice	SD	0-1-2	2 Credits

Detailed Syllabus:

Fitting Shop: Preparation of T-Shape Work piece as per the given specifications, Preparation of U-Shape Work piece which contains: Filing, Sawing, Drilling, Grinding, and Practice marking operations.

Machine shop: Study of machine tools in particular Lathe machine (different parts, different operations, study of cutting tools), Demonstration of different operations on Lathe machine, Practice of Facing, Plane Turning, step turning, taper turning, knurling and parting and Study of Quick return mechanism of Shaping operation. Demonstration of the working of CNC and 3D Printing Machines.

Power Tools: Study of different hand operated power tools, uses and their demonstration and Practice of Power tools.

Carpentry: Study of Carpentry Tools, Equipment and different joints, Practice of Cross Half lap joint, half lap Dovetail joint and Mortise Tenon Joint.

Welding: Study of welding tools and welding equipment, Arc Welding Practice (Lap and Butt joint).

CY102	Engineering Chemistry Lab	BSC	0-0-2	2 Credits

List of experiments (any eight of the following):

Exp. No	Name of the experiment
1	Standardization of $KMnO_4$ solution
2	Determination of Iron in Haematite
3	Determination of Hardness of Water
4	Determination of available chlorine in bleaching powder and of iodine in Iodized salt
5	pH-metric titration of an acid vs a base
6	Conductometric titration of an acid vs a base
7	Potentiometric titration of Fe^{2+} against $K_2Cr_2O_7$
8	Colorimetric determination of Potassium Permanganate
9	Determination of rate of Corrosion of mild steel in acidic environment in the absence of presence of an inhibitor
10	Determination of Chlorophyll in Olive oil by using UV and Fluorescence spectroscopic techniques
11	Functional group analysis of organic compounds by using IR spectroscopic technique
12	Organic solvent evaporation by using rotary-evaporation technique

Virtual labs

1. Determination of unknown concentration of analyte by using the Beer-Lambert's law.
2. Identification of unknown components using spectroscopic techniques.
3. Nuclear magnetic resonance spectroscopy and evolution of simple ^{1}H NMR spectra of organic compounds
4. Study of kinetics of a reaction by using spectrophotometric methods.

Reference books:

1. Charles Corwin, Introductory Chemistry laboratory manual: Concepts and Critical Thinking, Pearson Education, 2012.
2. David Collins, Investigating Chemistry: Laboratory Manual, Freeman & Co., 1st Edition, 2006.

II Year B.Tech. (CSE) Courses offered by CSED

CS201	Data Structures and Algorithms	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct Abstract Data Types for modelling entities using appropriate data constructs and methods. (Apply)
CO2	Construct list-based data structures namely Stacks, Queues, Circular Queues and Linked Lists. (Apply)
CO3	Construct non-linear data structures namely Trees & graphs and set-based structures like disjoint sets. (Apply)
CO4	Construct suitable data structures and algorithms to facilitate searching, sorting and selection. (Apply)
CO5	Construct efficient algorithms for performing operations on data structures within a given time and /or space complexity. (Apply)
CO6	Assess the suitability of various data structures for solving a given problem with a comprehension of trade-offs in time and space complexities. (Analyze)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	M
CO2	S	M	L										S	S	M
CO3	S	M	L										S	S	M
CO4	S	M	L										S	S	M
CO5	S	M	L										S	S	M
CO6	S	S	S	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Abstract Data Types (ADT), Arrays and Records, Memory Layout and implementation of Stack, Queue, Linked List (Single, Double and Circular), Applications of Stack, Queue and Linked Lists. Implementation of variants of the Stack, Queue and Linked List such as Two-way Stack, Circular Queue and Doubly Circular Linked Lists. Implementation of Stack and Queue using Linked List.

Memory Layout and Implementation of operations on Trees: Binary Trees, Binary Search Trees, AVL trees, B-Trees, Splay Trees and Tries.

Hashing: Open and Closed addressed tables, Design of good hash functions, Analysis of the efficiency of search operations, Bloom Filters.

Priority Queues: Memory layout, implementation and applications of Heap and its variants namely Skew Heaps, Leftist Heaps, Fibonacci Heaps and Binomial Queues.

Miscellaneous Data Structures: Skip Lists, Disjoint sets and k-d Trees.

Graphs: Memory representations, Depth First and Breadth First Traversals and representative applications like Topological Sorting, Finding Connected Components and Social Network Analysis.

Sorting: Representative Internal Sorting Algorithms like Merge Sort, Quick Sort, Counting Sort, Radix and Bucket Sorts. Analysis of efficiency of sorting, Notion of Stability in Sorting, External Sorting Algorithms.

Amortized Analysis of Data Structures: Accounting and Potential methods.

Reading List:

1. Data structures and Algorithms in C++, Michael T. Goodrich, R. Tamassia, and Mount, John Wiley and Sons, Second Edition, 2011.
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Fourth Edition, 2014.
3. Data structures and algorithms in C++, Fourth Edition, Adam Drozdek, Thomson, Cengage, 2013.
4. Data Structures: A Pseudocode Approach with C++, Richard F. Gilberg, Behrouz A. Forouzan, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

CS202	Operating Systems	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

None

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct process manager with a comprehension of scheduling policies, deadlocks, synchronization and the associated kernel data structures. (Apply)
CO2	Construct memory manager with a comprehension of the issues involved in memory management and management strategies especially for Virtual memory environments. (Apply)
CO3	Design and develop System Call Interface for a given File System. (Apply)
CO4	Design and develop Device Driver for disk access in the xv6. (Apply)
CO5	Analyze the design of Operating System Sub-Components of xv6 (x86 version) with an understanding of the sub-components like Process, Memory, and I/O managers and the interplay among these sub-components. (Analyze)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M		S				M				S	S	M
CO2	S	M	M		S				M				S	S	M
CO3	S	M	M		S				M				S	S	M
CO4	S	M	M		S				M				S	S	M
CO5	S	S	S	M	S				M				S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Processes:

Introduction to operating systems: Process abstraction, Process states, CPU Data Structures for process control. System calls for process management: fork, exec, exit, wait, open, read, write, close. Process Execution, Execution of function call, kernel/user modes, trap instruction, mechanism of context switch.

Scheduling policies: Criteria Considered for CPU Scheduling, scheduling using FIFO, SJF, STCF, RR, MLFQ methods.

Inter Process Communication: IPC, Shared Memory, Signals, Sockets, Pipes and Message Queues. Introduction to XV6.

Process Control in XV6: Process abstraction, Process state transition example. Process management system calls: Fork, Exec, Exit, Wait. Trap handling in XV6, Scheduling and Context switching in XV6, User process creation in XV6.

Memory Management:

Introduction to virtual memory: Goals of memory virtualization, Memory allocation system calls. Mechanism of address translation: Role of OS in translation, Segmentation. Paging: Page table, Multiple page tables. Demand paging: Page faults, Page replacement polices. Memory allocation and free space management algorithms: Variable size allocations and Fixed size allocations.

Virtual Memory and Paging in XV6, Memory management and User processes in XV6.

Concurrency:

Introduction to threads and concurrency: Single threaded process, multiple threaded process, scheduling threads, Race condition and Synchronization. Locks: Building a lock, different types of locking mechanisms. Condition Variables: producer-consumer problem. Semaphores. Concurrency bugs: non-deadlock bugs, deadlock bugs, conditions for deadlock, prevention mechanisms, deadlock avoidance, deadlock detection, recovery from deadlock. Locking in XV6: spin locks, disabling interrupts. Sleep/wakeup functionality in XV6.

I/O and file systems:

Communication with I/O devices: Introduction to input/output devices, Interrupts, Direct Memory Access, Device driver. Files and Directories: File abstraction, Directory tree, Hard links and soft links, mounting a file system, Memory mapping a file. File system implementations: File system, I-node table, File Allocation Table (FAT), Virtual file system. Hard disk internals: Disk scheduling, SSTF, SCAN Algorithm, SPTF.

Device drivers and block I/O in XV6: File system and I/O in XV6, Disk blocks and buffers, Logging layer. File system in XV6: Disk layout, in memory data structure, I-node function, creating a file, system calls: open, link, file read.

Security in Operating System: Goals of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Revocation of Access Rights, Language-Based Protection, Capability-Based Systems, The Security Problem, User Authentication, Program Threats, System Threats, Securing Systems and Facilities, Intrusion Detection, Cryptography, Computer-Security Classifications.

Reading List:

1. Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau Andrea C. Arpaci-Dusseau, Arpaci-Dusseau Books, Amazon Digital Services, 2015.
2. xv6 a simple, Unix-like teaching operating system, Russ Cox Frans Kaashoek Robert Morris, Available From: https://pdos.csail.mit.edu/6.828/2014/xv6/book-rev8.pdf, Accessed on: August 2021.
3. Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Wiley, 10th edition, 2008.

CS203	Discrete Mathematics	PCC	2 – 1 – 0	3 Credits

Pre-requisites:

None.

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct mathematical proofs using mathematical logic and induction. (Apply)
CO2	Apply logical notations to define and reason mathematically about the constructs used in computing. (Apply)
CO3	Apply techniques for counting the occurrences of discrete events including permutations, combinations with or without repetitions. (Apply)
CO4	Solve problems using graph representations. (Apply)
CO5	Formulate and solve recurrence relations. (Apply)
CO6	Identify the algebraic structures and verify their properties. (Analyze)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L									S	M	M
CO2	S	M	M	L									S	M	M
CO3	S	M	M	L									S	M	M
CO4	S	M	M	L									S	M	M
CO5	S	M	M	L									S	M	M
CO6	S	S	M	L									S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Sets and Relations: Sets, Operations on Sets, Venn Diagrams, Multi Sets, Binary Relations, Equivalence Relations, Ordering Relations, Operations on Relations, Partial Orders.

Functions: Definition and Introduction, Composition of Functions, Inverse Functions, Binary and n-ary Operations.

Mathematical Logic and Induction: Statements and Notation, Connectives, Quantified Propositions, Logical Inferences, Methods of Proof of an Implication, First Order Logic and other Methods of Proof, Rules of Inference for Quantified Propositions, Proof by Mathematical Induction.

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion- Exclusion.

Recurrence Relations: Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, The Method of Characteristic Roots, Solutions of Inhomogeneous Recurrence Relations.

Lattices as Partially Ordered Sets: Definition and Examples, Properties of Lattices, Lattices as Algebraic Systems, Sublattices, Direct Product, Homomorphism, Some Special lattices.

Graphs: Basic Definitions Undirected and Directed Graphs, Paths, Representation of Graphs, Reachability, Connected Components, Examples of Special graphs, Graph Isomorphism, Planar Graphs, Euler's Formula, Euler Circuits, Hamiltonian Graphs, Chromatic Number of a Graph, The Four-Color Problem, Graph Traversals, Applications of Graphs.

Algebraic Structures: Groups, Semigroups, Monoids, Rings and Fields.

Reading List:

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, 7th Edition, Tata McGraw Hill Publishers, 2007.
2. Discrete Mathematics for Computer Scientists and Mathematicians, Joe L. Mott, Abraham Kandel, Theodore P. Baker, Second Edition, PHI, 2001.
3. Discrete Mathematics, Norman L Biggs, 2nd Edition, Indian Edition published by Oxford University Press.
4. Discrete Mathematical Structures, Tremblay J. P. and Manohar R., MGH, 1997.

CS204	Data Structures and Algorithms Laboratory	PCC	0 – 1 – 2	2 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Implement Abstract Data Types for modeling entities using appropriate data constructs and methods. (Apply)
CO2	Implement list-based data structures namely Stacks, Queues, Circular Queues and Linked Lists. (Apply)
CO3	Implement non-linear data structures namely Trees & graphs and set-based structures like disjoint sets. (Apply)
CO4	Implement suitable efficient data structures and algorithms to facilitate searching, sorting and selection. (Apply)
CO5	Implement efficient algorithms for performing operations on data structures within a given time and/or space complexity. (Apply)
CO6	Implement solutions after assessing the suitability of different data structures for solving a given problem with a comprehension of trade-offs in time and space complexities by demonstrating problem-solving and programming skills. (Analyze)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M						S	M	L		S	S	M
CO2	S	M	M		S				S	M	L		S	S	M
CO3	S	M	M		S				S	M	L		S	S	M
CO4	S	M	M		S				S	M	L		S	S	M
CO5	S	M	M		S				S	M	L		S	S	M
CO6	S	S	S	M	L				S	M	L		S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Implementation of Stack and Queue using Arrays.
2. Implementation of Stack-based applications like postfix expression evaluation and infix to postfix conversion.
3. Implementation of Queue and Circular Queue.
4. Implementation of Single Linked List, Doubly Linked List and Circular Linked List.
5. Implementation of Stack and Queue using Linked List.
6. Representative problems with solutions involving Stack, Queue and Linked List.
7. Implementation of Binary Search Tree.
8. Implementation of BST traversals in recursive and non-recursive ways.
9. Implementation of AVL Tree.
10. Implementation of Priority Queue.
11. Implementation of Dictionaries using open and closed addressing schemes.
12. Implementation of Trie for fast text matching.
13. Implementation of Quick, Merge, Counting, Radix and Bucket sorts.
14. Implementation of Graphs and Depth First & Breadth First Traversals.
15. Mini project involving design, memory organization, implementation and complexity analysis of data structures and their associated operations.

Reading List:

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia, and Mount, John Wiley and Sons, Second Edition, 2011.
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Fourth Edition, 2014.
3. Data structures and algorithms in C++, Fourth Edition, Adam Drozdek, Thomson, Cengage, 2013.
4. Data Structures: A Pseudocode Approach with C++, Richard F. Gilberg, Behrouz A. Forouzan, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

CS205	Operating Systems Lab	PCC	0– 1 – 2	2 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Implement elementary UNIX system commands. (Apply)
CO2	Develop programs to test synchronization problems. (Apply)
CO3	Design and develop user level thread library. (Apply)
CO4	Design and implement a file system. (Apply)
CO5	Design and implement a memory manager module. (Apply)
CO6	Design and implement a process manager module. (Apply)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M		S			S	S	M			S	S	M
CO2	S	M	M		S			S	S	M			S	S	M
CO3	S	M	M		S			S	S	M			S	S	M
CO4	S	M	M		S			S	S	M			S	S	M
CO5	S	M	M		S			S	S	M			S	S	M
CO6	S	M	M		S			S	S	M			S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarm signals.

3. Write a program to do memory management (allocation/freeing) for fixed chunk sizes using a larger chunk of memory allocated using mmap builtin function. Extend it to manage dynamic memory allocation as done with heap memory.
4. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process. (Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Salesperson problem)
5. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
6. Implement all CPU Scheduling Algorithms using your thread library
7. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
8. A complete file system implementation inside a disk image file.

Reading List:

1. Advanced Programming in the UNIX Environment, Richard Stevens, Stephen Rago, Addison Wesley, Third edition, 2013.
2. Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau Andrea C. Arpaci-Dusseau, Arpaci-Dusseau Books, Amazon Digital Services, 2015.
3. xv6 a simple, Unix-like teaching operating system, Russ Cox Frans Kaashoek Robert Morris, Available From: https://pdos.csail.mit.edu/6.828/2014/xv6/book-rev8.pdf, Accessed on: August 2021.
4. Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Wiley, 10th edition, 2008.

MA204	**Probability, Statistics and Stochastic Processes**	**BSC**	**3 - 0 - 0**	**3 Credits**

Pre-requisites: Differential & Integral Calculus (MA101),
Matrices & Differential Equations (MA151).

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

CO1	Provides a solid foundation about the concept of probability and its features
CO2	Provide the idea of important results used in statistical Inference
CO3	Find the coefficient of correlation and lines of regression
CO4	Understand the concept of stochastic process

Mapping of course outcomes with program outcomes

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4				`								

Random variables and their distributions:

Introduction to Probability, random variables (discrete and continuous), probability density and mass functions, distribution functions, mean and variance, Moment generating and Characteristic function, special distributions (Binomial, Hypergeometric, Poisson, Uniform, Exponential, Normal, Chi-square), Chebyshev's inequality, Statistic estimation of parameters by maximum Likelihood Estimation method parameter. (16)

Testing of Hypothesis:

Testing of Hypothesis, Null and alternative hypothesis, level of significance, one-tailed and two-tailed tests, tests for large samples (tests for single mean, difference of means, single proportion, difference of proportions), tests for small samples (t-test for single mean and difference of means, F-test for comparison of variances), Chi-square test for goodness of fit, analysis of variance (one way classification with the samples of equal and unequal sizes), correlation and regression. Multiple and partial correlation, rank correlation and Karl Pearson coefficient of correlation, lines of regression. (12)

Stochastic Process:

Definition and classification of general stochastic processes. Markov Chains: definition, transition probability matrices, classification of states, limiting properties, Discrete Time Markov Chains: Chapman-Kolmogorov equations, Ergodicity, Reducibility/Irreducibility, Time reversible chains, Applications of Markov chains in Queuing models, Continuous Time Markov Chains: Birth-Death processes, Poisson processes, M/M/1 Queuing Models. (14)

Text Books:

1. R.A. Johnson, Miller and Freund's , Probability and Statistic for Engineers, Pearson Publishers, 9^{th} Edition, 2017.
2. Freund, Modern elementary statistics, PHI, 2006.
3. S.C.Gupta and V.K.Kapoor, Fundamentals of Mathematical Statistics, 2006.

Reference Books:

1. Kantiswarup, P.K.Gupta and Manmohan Singh, Operations Research, S.Chand & Co, 2010.
2. B. Prabhakara Rao (Author), T.S.R. Murthy, Probability Theory and Stochastic Processes Kindle Edition , 2019.
3. U.N. Bhat and Gregory K. Miller, Elements of Applied Stochastic Processes, John Wiley and Sons, 2002.
4. K.S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, Prentice Hall of India, 2008

CS251	Object Oriented Programming	PCC	2 – 1 – 2	4 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)
iii. Data Structures and Algorithms (CS201)
iv. Data Structures and Algorithms Lab (CS204)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct programs using Object Oriented Design principles like encapsulation, abstraction, polymorphism, inheritance and typing. (Apply)
CO2	Develop applications with handlers for user-defined exceptions, according to the given requirements. (Apply)
CO3	Construct efficient multi-threaded programs with synchronization constructs. (Apply)
CO4	Develop interactive GUI applications with event handling that provide rich user experience. (Apply)
CO5	Construct programs using the suitable Collection classes and interfaces for efficient modelling of the objects and entities of the program. (Apply)
CO6	Develop applications that use file input and output. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M		S				M	M			S	S	S
CO2	S	M	M		S				M	M			S	S	S
CO3	S	M	M		S				M	M			S	S	S
CO4	S	M	M		S				M	M			S	S	S
CO5	S	M	M		S				M	M			S	S	S
CO6	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Overview of Object-Oriented Programming and its need, Java Programming Elements: Classes and Objects, Data types, Constructors, Input-Output Handling, Control structures, Method overloading and overriding, Abstraction and Inheritance, Interfaces, understanding final and static: classes, blocks and methods, Packages.

Exception Handling: Types of Exceptions, Exception classes, try, catch, throw, throws and finally, Exception Handling with Method Overriding, Custom Exceptions, finalize.

Multithreaded Programming: Introduction to multitasking through processes and threads, creating threads, thread life cycle, thread scheduling, thread priorities, daemon thread, synchronization.

Garbage Collection, Runtime class and Memory management in Java.

String handling: String, String Buffer, StringBuilder and tokenization.

Generics, Lambda expressions.

The Collections framework: List, Set and Map interfaces and the corresponding implementation classes, Enumerator and iterators.

Event handling: Event, Listeners and adapter classes, anonymous inner classes.

Abstract Windowing Toolkit (AWT): Button, Label, Checkbox, Checkbox Group, Text Field, Text Area, Choice, List, Menu, Panel, Scrollbar and Swing components, Layout managers, Complex Components.

MVC and introduction to creational, structural design patterns and behavioural patterns.

File I/O: Streams, Readers and Writers, Pipes and Filters, Random Access, Scanner.

List of Experiments:

1. Develop programs to familiarize with Object Oriented Design concepts.
2. Implement programs to illustrate overloading and overriding.
3. Implement programs to familiarize with the Testing and Debugging facilities.
4. Implement abstract classes and Interfaces.
5. Implement programs using Static classes, blocks and methods.
6. Implement programs to compare shallow and deep copy.
7. Develop programs using Exception Handling.
8. Develop programs using multi-threading and synchronization.
9. Implement programs using Generics.
10. Implement programs using Linked data structures, Heaps, priority queues, binary search trees and the interfaces List, Set and Map.
11. Implement programs using String processing facilities offered by String, String Buffer and StringBuilder classes.

12. Develop Event-driven programs for GUI applications that are interactive, responding to events originating from keyboard and mouse.
13. Implement programs using Streams and File I/O for reading and writing the contents in sequential and random order.
14. Given two words of equal length that are in a dictionary, implement an efficient program to transform one word into another word by changing only one letter at a time.
15. Numbers are randomly generated and passed to a method, implement an efficient program to find and maintain the median value as new values are generated.
16. Given an N x N matrix of positive and negative integers, implement an efficient program to find the sub-matrix with the largest possible sum.

Reading List:

1. Java: The Complete Reference, Herbert Schildt, 11th edition, Mc Graw Hill, 2019.
2. Head First Java, Kathy Sierra & Bert Bates, 3rd edition, O'Reilly, 2005.
3. Clean Code, Robert C Martin, Pearson, 2012.
4. Timothy Budd, Object Oriented Programming with Java, Pearson Education, 2009.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: https://cse.iitkgp.ac.in/~dsamanta/java/index.htm, Accessed on: August 2021.
6. Design Patterns, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley, 1994.

CS252	Design and Analysis of Algorithms	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)
iii. Data Structures and Algorithms (CS201)
iv. Data Structures and Algorithms Lab (CS204)
v. Discrete Mathematics (CS203)

Course Outcomes: At the end of this course, students will be able to:

CO1	Infer which algorithm design paradigm is suitable for solving a given problem. (Analyze)
CO2	Construct algorithms for a given complex engineering problem using an appropriate design paradigm such as divide and conquer, dynamic programming, greedy, backtracking and branch-and-bound. (Apply)
CO3	Construct correctness proofs for algorithms. (Apply)
CO4	Infer the asymptotic time and space complexities of algorithms. (Apply)
CO5	Assess the impact of the choice of data structure on the performance of algorithms. (Apply)
CO6	Construct algorithms for graph-based problems by modeling entities as graphs and reducing the solution to standard graph theoretic algorithms. (Apply)
CO7	Construct reductions between problems with a comprehension of the complexity classes namely P, NP, NP-Complete & NP-Hard and their relationships. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	S	S
CO2	S	M	L	L									S	S	S
CO3	S	M	L	L									S	S	S
CO4	S	M	L	L									S	S	S
CO5	S	M	L	L									S	S	S
CO6	S	M	L	L									S	S	S
CO7	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to Algorithm Design and Analysis, Asymptotics, Correctness Proofs for Algorithms.

Divide and Conquer Algorithms: Solving Recurrences, Recurrence Trees, Substitution method, Master's Theorem, Merge Sort, Quick Sort and Randomized Quicksort with their analyses, Maximum sub-array problem, Linear time Selection, Strassen's Matrix Multiplication, Multiplication of Large integers.

Greedy Algorithms: Design scenarios calling for a greedy design, Making Change problem, Activity Selection Problem, Fractional Knapsack problem, Prim's and Kruskal's algorithms for Minimum Spanning Tree construction, Dijkstra's algorithm for finding single source shortest paths.

Dynamic Programming: Design scenarios calling for a dynamic programming design, Chained Matrix Multiplication problem, 0/1 Knapsack problem, Travelling Salesperson Problem, Optimal Binary Search Tree, Floyd Warshall's Algorithm, Vertex Cover of a Tree.

Backtracking: N-Queen's problem, Enumeration of Independent Sets of a problem, Graph Coloring problem, Robotic path finding problem in an unknown terrain.

Branch and Bound: TSP. Set Cover and 0/1 Knapsack problem.

String algorithms: KMP Algorithm, Boyer Moore Algorithm and Rabin-Karp Algorithm.

Flow-based Algorithms: Max-flow Min-cut theorem, augmenting paths and finding min-cuts.

Complexity classes: P, NP, NP-Complete and NP-Hard, Standard reductions.

Reading List:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.
2. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, First Edition, PHI, 2000.
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, First Edition, McGraw Hill Education, 2006.
6. Sara Baase, Computer algorithms: Introduction to Design and Analysis, Second Edition, Addison Wesley publication, 1998.

CS253	Computer Organization and Architecture	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Digital Logic Design (EC237)

Course Outcomes: At the end of the course, students will be able to:

CO1	Construct assembly language programs to perform the specified task with a comprehension of addressing modes. (Apply)
CO2	Analyze the speed and cost of memory access based on availability of data in cache memory, main memory and secondary memory. (Apply)
CO3	Assess the performance of data transfer to and from the peripherals and identify the suitable mode of transfer. (Apply)
CO4	Design Arithmetic Logic Unit components and microprogrammed control units satisfying the specified constraints. (Apply)
CO5	Design pipelines for high performance considering possible hazards, data paths and control paths. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L						L			S	S	S
CO2	S	M	L	L						L			S	S	S
CO3	S	M	M	L						L			S	S	S
CO4	S	M	M	L						L			S	S	S
CO5	S	M	M	L						L			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Basic Structures of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multi-computers, Historical Perspective.

Machine instructions and Programs: Numbers, Arithmetic Operations and Characters, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Example Programs, Encoding of Machine Instructions.

Registers and addressing, IA-32 Instructions, IA-32 Assembly Language, Program Flow Control, Logic and Shift/Rotate Instructions, I/O Operations, Subroutines, Other Instructions, Program Examples.

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

The Memory System: Some Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed Size and Cost, Cache Memories: mapping functions, cache performance and replacement algorithms, levels and locality, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication, Fast Multiplication, Integer Division, Floating Point Numbers and Operations, Implementing Floating Point Operations.

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Microprogrammed Control.

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets, Data Path and Control Considerations, Super Scalar Operation, UltraSPARC 2 Example, Performance Consideration.

Introduction to superscalar architectures, VLIW machines.

Reading List:

1. Computer Organization, Carl Hamacher, 5th Edition, McGraw Hill Publishers, 2002.
2. Computer Organization and Architecture Designing for Performance, William Stallings, 8th Edition, Pearson Education, 2010.
3. Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 5th Edition, Morgan Kauffmann Publishers, 2012.
4. Computer System Architecture, Morris Mano, 3rd Edition, PHI, 2017.
5. Computer Architecture: A Quantitative Approach, Hennessy, J. L., and D. A. Patterson. 3rd ed. San Mateo, CA: Morgan Kaufman, 2002.

CS254	Theory of Computation	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

None

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct, compare and analyze computational machine models for solving problems and recognizing languages. (Analyze)
CO2	Infer the properties of languages, automata and grammars with the help of rigorous proofs. (Analyze)
CO3	Infer the limitations of the computational models and prove the existence of such limitations. (Analyze)
CO4	Construct algorithms for solving problems and show the correctness of algorithms using different computational machine models. (Analyze)
CO5	Identify the dependency of application domains like Compiler Design on the computational models. (Analyze)

Course Articulation Matrix:

PO/ PSO CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	S	S	M										S	M	M
CO2	S	S	M										S	M	M
CO3	S	S	M										S	M	M
CO4	S	S	M										S	M	M
CO5	S	S	M										S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Finite State Automata (FSA): Alphabets, Strings, Languages, Deterministic FSA (DFSA), non-deterministic FSA (NFSA) with Epsilon-Transitions, Uses of ∊ Transitions. Regular expressions, Finite Automata and Regular Expressions, Conversion among Regular Expressions, DFSA and NFSA, Applications of Regular Expressions, Applications of Regular Expressions in UNIX and Lexical Analysis, Algebraic Laws for Regular Expressions, Laws Involving Closures, Subset Construction Algorithm, FSA

Minimization, Regular Languages and properties, Pumping Lemma, Myhill-Nerode Theorem.

Context Free Grammars - Derivations Using a Grammar, Leftmost and Rightmost Derivations, The Language of a Grammar, Sentential Forms, Parse Trees, eliminating useless symbols, Applications of Context-Free Grammars, Parsers, The YACC Parser Generator, Ambiguity in Grammars and Languages, Ambiguous Grammars, Removing Ambiguity from Grammars. Normal Forms: Chomsky Normal Form and Greibach Normal Form, Chomsky Hierarchy, Context Free Languages (CFLs), properties, Pumping Lemma for CFLs, CYK Algorithm.

Push Down Automata (PDA): Definition, Instantaneous Descriptions of a PDA, The Languages of a PDA, Acceptance by Final State, Acceptance by Empty Stack, Equivalence of PDAs and CFGs, Deterministic Context Free Languages.

Turing Machines (TMs): Definition, Instantaneous Descriptions for the Turing Machines, Transition Diagrams for Turing Machines, The Language of a Turing Machine, Turing Machines and Halting, Configurations: TM as a computing device and as a recognition device. Programming Techniques for Turing Machines: Storage in the State, Multiple Tracks, Shifting Over, Subroutines, Extensions to the Basic Turing-Machines, Multiple Turing Machines, Computable Functions. Multi-Tape Turing Machines, Linear Bounded Automata.

Recursive and recursively enumerable languages. Undecidability of Halting Problem. Reductions, Theory of NP Completeness.

Reading List:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Introduction to Automata Theory, Languages and Computation, Second Edition, Pearson, 2001.
2. Michael Sipser, Introduction to Theory of Computation, Third Edition, Course Technology, 2012.
3. Automata and Computability, Dexter C. Kozen, First Edition, Springer Publishers, 2007.
4. Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, First Edition, Prentice Hall Publishers, 1981.
5. J. Martin, Introduction to Languages and the Theory of computation, Third Edition, Tata Mc Graw Hill, 2007.
6. Peter Linz, An Introduction to Formal Languages and Automata, Sixth Edition, Jones & Bartlett Learning, 2017.

CS255	Database Management Systems	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Data Structures and Algorithms (CS201)
ii. Data Structures and Algorithms Lab (CS204)
iii. Operating Systems (CS202)
iv. Operating Systems Lab (CS205)
v. Discrete Mathematics (CS203)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct database schema using Database models at conceptual level identifying entities and relationships among entities using E-R and relational models. (Apply)
CO2	Construct database and implement queries using SQL constructs for a given requirement specification. (Apply)
CO3	Construct database design using Normalization and Functional Dependencies to store information without redundancy. (Analyze)
CO4	Design Data Storage Structures and Indexing at physical level to build fast and reliable storage systems and enable quick access to data. (Analyze)
CO5	Apply concurrency control mechanisms with the help of locking, stamping and optimistic methods, and recovery mechanisms to maintain database integrity. (Apply)
CO6	Apply strategies for transaction management with a comprehension of the ACID properties to be maintained. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	L	S				M	L			S	S	L
CO2	S	M	L	L	S				M	L			S	S	L
CO3	S	M	S	L	S				M	L			S	S	M
CO4	S	M	S	L	S				M	L			S	S	M
CO5	S	M	S	L	S				M	L			S	S	M
CO6	S	M	S	L	S				M	L			S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction: Basic concepts of database systems, Database System Applications, Purpose of Database Systems, Data View, Database Languages, Characteristics of a database approach, Advantages of DBMS over file system, Database Design, Database Engine, Database and Application Architecture, Database Users and Administrators, History of Database systems

Database models: ER-model: Features of the E-R model, Database design using the E-R model, Relational model: Relational Database Schemas, Keys, Integrity constraints over relations, E-R model to relational schema, relational algebra, relational calculus.

SQL: Form of a basic SQL Query, Insert, Delete and Update database, Set Operations, Nested queries, Aggregate Functions, Null Values, Integrity Constraints, Join Expressions – Natural join, Equi-join, Inner join, Outer join, Views, Triggers.

Database design: Functional-Dependency, Soundness and completeness of Armstrong's axioms, Normal Forms, Loss-less decomposition, Dependency preservation, Decomposition Using Functional Dependencies, Multivalued Dependencies and 4NF, Join dependencies and 5NF, Decomposition Algorithms.

Data Storage Structure and Indexing: Memory hierarchy, Redundant Arrays of Independent Disks, File organization and indexing, Index Data Structures: Hash-based indexing and Tree-based Indexing (B+ trees and B-trees), Introduction to Query optimization and execution.

Transactions, concurrency control and recovery: ACID properties, Transactions and Schedules, Concurrent execution of transactions, Lock-Based Protocols, Deadlock Handling, Concurrency control without locking, Introduction to crash recovery, ARIES, Write-Ahead log protocol, checkpointing, recovery from a system crash, security, authorization and access control.

Reading List:

1. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 7th edition, McGraw-Hill, 2010.
2. R Elmasri, S Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016.
3. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd edition, McGraw-Hill, 2003.
4. C J Date, An Introduction to Database Systems, Pearson/Addison Wesley, 2003.

CS256	Unix Tools and Shell Scripting Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites:

i. Operating Systems (CS202)
ii. Operating Systems Lab (CS205)

Course Outcomes: At the end of this course, students will be able to:

CO1	Develop text data processing applications using Unix commands and filters. (Apply)
CO2	Design and develop text-based user interface components. (Apply)
CO3	Perform user management, network management and backup. (Apply)
CO4	Apply CVS/git utilities for Software version management. (Apply)
CO5	Apply SSH for distributing tasks. (Apply)
CO6	Construct complex applications for a given requirement, using the tools and interfaces offered by the UNIX Shell. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	S	M	L		S			S	S	M			S	M	
CO2	S	M	L		S			S	S	M			S	M	
CO3	S	M	L		S			S	S	M			S	M	
CO4	S	M	L		S			S	S	M			S	M	
CO5	S	M	L		S			S	S	M			S	M	
CO6	S	M	L	L	S			S	S	M			S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Explore different options for the following Unix Commands using man: at, banner, batch, bc, cal, cat, cd, cmp, comm, chmod, chown, chgrp, cp, cron, cut, date, dd, diff, echo, finger, find, ftp, head, kill, lock, ln, ls, lp, lpstat, man, mesg, mkdir, more, mv, nl, nice, passwd, pr, paste, ping, ps, pwd, rcp, rlogin, rmdir, rm, rsh, split, sort, tail, talk, tar, telnet, touch, tput, tr, tty, uname, uniq, wc, who, write.

2. Write shell script for Pattern matching using wild cards, escaping, quoting, iterations, and if then else constructs.
3. Write shell script for searching for various patterns using grep, pr, head, tail, cut, paste, sort, uniq, and tr.
4. Write shell script for pattern matching using awk, and sed utilities.
5. Write shell script for backup using tar and cpio.
6. Write a make script with dependencies for compiling a static link library/dynamic link library and building an executable to link to such libraries.
7. Automate make script creation for bigger projects.
8. Create a project under CVS/git repository and record multiple versions/branches and practice merging of branches.
9. Debug a C program using gdb to understand core dump of a failed execution.
10. Install requisite packages using rpm/deb/apt.
11. Write scripts for Simple Distribution of tasks using ssh.
12. Write scripts for installation of software in multiple machines.
13. Write scripts for obtaining usage stats memory usage and CPU usage for multiple machines.
14. A capstone mini project to unify the comprehension of the tools and utilities learnt in this course.

Reading List:

1. Sumitabha Das, Unix Concepts and Applications, TMH, 4th edition, 2008.
2. John R Levine, Tony Mason, Doug Brown, Lex and Yacc, Orielly, 2nd edition, 2009.

CS257	Database Management Systems Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites:

i. Data Structures and Algorithms (CS201)
ii. Data Structures and Algorithms Lab (CS204)
iii. Operating Systems (CS202)
iv. Operating Systems Lab (CS205)
v. Discrete Mathematics (CS203)

Course Outcomes: At the end of this course, students will be able to:

CO1	Design database schema and Implement SQL queries for a given requirement specification. (Apply)
CO2	Assess the impact of different query execution plans and access paths on query performance. (Analyze)
CO3	Implement database maintenance and control using authorization access control, transaction and concurrency management and recover constructs. (Apply)
CO4	Design and develop an application using stored procedures for a given requirement specification. (Apply)
CO5	Design and develop an application with database connectivity for a given requirement specification. (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L	S			S	S	M			S	S	L
CO2	S	S	S	L	S			S	S	M			S	S	L
CO3	S	M	M	L	S			S	S	M			S	S	L
CO4	S	M	M	L	S			S	S	M			S	S	L
CO5	S	M	M	L	S	S	S	S	S	M	L	L	S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Familiarization with installation of any database management system.

2. Design a database schema and create a database through implementation of Data Definition Language commands - create, alter, drop, rename and truncate.
3. Querying and modifying the database using Data Manipulation Language commands -select, insert, update, delete.
4. Implementation of Aggregate Functions – sum, avg, min, max, count. Use group-by and having clause.
5. Perform join operations – natural join, equi-join, outer join, left outer join, right outer join, inner join and assess the impact of query plans on the performance of join heavy queries.
6. Perform set operations - union, intersection, set difference.
7. Implementation of corelated sub-queries and nested queries.
8. Creating and querying views and materialized views.
9. Implementation of Data Control Language commands – grant and revoke.
10. Implementation of Transaction Control Language commands - commit, save point, and rollback.
11. Implementation of PL/SQL block using variables, operators, data types, control structures.
12. Implementation of PL/SQL stored procedures.
13. Implementation of PL/SQL stored functions.
14. Implementation of PL/SQL cursors structures.
15. Implementation of PL/SQL exception handlers.
16. Implementation of triggers.
17. Implementation of an application with database connectivity for storing and retrieving information.

Reading List:

1. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 7th edition, McGraw-Hill, 2010.

2. R Elmasri, S Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016.

3. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd edition, McGraw-Hill, 2003.

4. C J Date, An Introduction to Database Systems, 3rd edition, Pearson/Addison Wesley, 2003.

5. PLSQL accessed through: https://www.oracle.com/in/database/technologies/appdev/plsql.html, Accessed on August 2021.

6. MySQL, accessed through: https://www.mysql.com/, Accessed on August 2021.

7. PostgreSQL, accessed through: https://www.postgresql.org/, Accessed on August 2021.

8. JDBC,accessed throughgh: https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html, Accessed on August 2021.

III Year B.Tech. (CSE) Courses offered by CSED

CS301	Language Processors	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Theory of Computation (CS254)

Course Outcomes: At the end of the course the student will be able to:

CO1	Build a lexical analyser for the given language. (Apply)
CO2	Construct suitable parsers for a given language construct with a comprehension of different parsers and their applicability to different constructs. (Apply)
CO3	Construct syntax directed translation schemes subsequent to the identification of appropriate synthesized and inherited attributes. (Apply)
CO4	Construct intermediate code generators to generate intermediate code in the form of three address code representations. (Apply)
CO5	Construct a code generator and code optimizer with a comprehension of the runtime environment. (Apply)
CO6	Construct system software like assemblers, linkers and loaders with a comprehension of the underlying runtime architecture. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S		
CO2	S	M	L										S		
CO3	S	M	L										S		
CO4	S	M	L										S		
CO5	S	S	M										S		
CO6	S	M	L										S		

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Introduction to System Software: Macro processors, One pass and two pass assemblers, Linkers, Loaders, Types of Loaders, Interpreters, Debuggers, Compilers.

Phases of Compilers - Compiler Construction Tools - Bootstrapping

Lexical analyzer - The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzers.

Parsing - The Role of the Parser, Context-Free Grammars, Top-Down Parsing, Bottom-Up Parsing, Operator-Precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators.

Syntax-Directed Translation- Syntax-Directed Definitions, Construction of Syntax Trees, Bottom- Up Evaluation of S-Attributed Definitions, L-Attributed Definitions, Top-Down Translation, Bottom-Up Evaluation of Inherited Attributes, Recursive Evaluators, Space for Attribute Values at Compile Time, Assigning Spaces at Compiler-Construction Time, Analysis of Syntax-Directed Definitions.

Type Checking- Type Systems, Specification of a Simple Type Checker, Equivalence of Type Expressions, Type Conversions, Overloading of Functions and Operators, Polymorphic Functions, An algorithm for Unification.

Run-Time Environments - Source Language Issues, Storage Organization, Storage-Allocation Strategies, Access to Nonlocal Names, Parameter Passing, Symbol Tables, Language Facilities for Dynamic Storage Allocation, Dynamic Storage Allocation Techniques, Storage Allocation in Fortran.

Intermediate Code Generation - Intermediate Languages, Declarations, Assignment Statements, Boolean Expressions, Case Statements, Backpatching, Procedure Calls.

Code Generation - Issues in the Design of a Code Generator, The Target Machine, Run-Time Storage Management, Basic Blocks and Flow Graphs, Next-Use Information, A Simple Code Generator, Register Allocation and Assignment, The Dag Representation of Basic Blocks, Peephole Optimization, Generating Code from DAGs, Dynamic Programming Code-Generation Algorithm, Code-Generator Generators.

Reading List:

1. Alfred V. Aho, Monical S.Lam, Ravi Sethi, and Jeffrey D. Ullman Compilers - Principles, Techniques and Tools, 2nd Edition, Pearson, 2007.
2. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kauffmann, 2001.

CS302	Theory and design of Programming Languages	PCC	3 – 0 – 2	4 Credits

This course aims to establish a broad framework in which to formulate and analyze the design aspects of programming languages. Students of this course will be exposed to multiple programming language design paradigms, their underlying principles, representative programming languages, their constructs and programming techniques.

Pre-requisites:

None.

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

CO1	Demonstrate a comprehension of the concepts found across different programming languages using a precise single math framework. (Apply)
CO2	Prove the properties of programming languages using structural induction with a comprehension of abstract syntax, binding, and scope of identifiers. (Analyze)
CO3	Construct precise math formalisms for specifying the type-system and the runtime behaviour of a program. (Analyze)
CO4	Prove that a language is safe with respect to types. (Analyze)
CO5	Construct programs across languages from multiple design paradigms with a comprehension of their underlying design principles, constructs, and a sensible integration of multiple features of the language. (Analyze)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	M	
CO2	S	S	M										S	M	
CO3	S	S	M										S	M	
CO4	S	S	M										S	M	
CO5	S	S	M		S			S	S	S		S	S	S	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Syntactic Objects: Abstract Syntax Trees, Abstract Binding Trees.

Inductive definitions, Hypothetical and general judgements: Derivability, admissibility.

Church's Lambda calculus.

Static and dynamic semantics, type safety, Binding.

Higher-order total computation: Gödel's T, Higher-order programming in T.

Generic programming: inductive and co-inductive types.

Polymorphism, genericity, Existential types, Data Abstraction.

Parametricity: verification of ADTs, Higher-order partial computation: Plotkin's PCF.

General recursive types: FPC, Dynamic Languages: DPCF, Dynamic dispatch, calculus and DPCF.

Exceptions and control data. Programming with continuations.

Imperative programming: Modernized Algol.

Dynamic Classification, Lazy evaluation.

Parallelism: Work & span, Brent's theorem.

Haskell and ML: Constructs and programming techniques.

Concurrency: calculus.

Logic paradigm: Prolog, constructs, and programming techniques

List of Experiments:

1. Familiarization with the syntactic constructs of Modernized Algol.
2. Exercises involving imperative programming languages like Modernized Algol.
3. Familiarization with Concurrent programming constructs.
4. Case Studies in Concurrent programming.
5. Familiarization with the syntactic constructs of Haskell.
6. Case Studies in Haskell.
7. Familiarization with ML constructs.
8. Case Studies in ML.
9. Familiarization with Prolog constructs.
10. Case Studies in Prolog.

Reading List:

1. Robert Harper, Practical Foundations for Programming Languages, Cambridge University Press, Second Edition, 2016.

2. Michael L. Scott, "Programming Language Pragmatics", Fourth Edition, Morgan Kaufmann, 2016.

CS303	Artificial Intelligence	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Discrete Mathematics (CS203)
ii. Data Structures and Algorithms (CS201)
iii. Probability, Statistics and Stochastic Processes (MA239)
iv. Design and Analysis of Algorithms (CS252)

Course Outcomes: At the end of the course, the students will be able to:

CO1	Solve single agent and multi agent problems by applying the appropriate search techniques. (apply)
CO2	Create logical agents to do inference using first-order logic. (apply)
CO3	Construct an agent that can reason under uncertainty and performs inference in a Bayesian Network. (apply)
CO4	Build a ML model from the data and evaluate its performance. (apply)
CO5	Construct an agent that can learn with complete data and learn with hidden variables. (apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	S	S	L	M								S	L	M
CO2	S	S	S	M	M								S	L	L
CO3	S	S	S	M	M								S	L	L
CO4	S	S	S	M	M								S	L	M
CO5	S	S	S	M	M								S	L	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction and Scope, Agents and Environments, Concept of Rationality.

Solving problems by Search: Uninformed Search Strategies, Breadth-first search, Depth-first search, Searching with Partial Information, Informed (Heuristic) Search Strategies, Greedy best-first search.

A* Search: Minimizing the total estimated solution cost, Heuristic Functions, Local Search Algorithms, and Optimization Problems, Online Search Agents, and Unknown Environments.

Adversarial Search: Games, the minimax algorithm, Optimal decisions in multiplayer games, Alpha-Beta Pruning, Evaluation functions, cutting off search, Games that Include an Element of Chance.

Constraint Satisfaction Problem: Definition, Constraint Propagation, Search for CSPs.

Inference In First Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution. The Wumpus World.

Planning: Classical Planning, algorithms for planning as state space search, planning graphs.

Uncertainty: Inference Using Full Joint Distributions, Independence, Bayes Rule, and its Use, The Wumpus World Revisited.

Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distribution, Exact Inference in Bayesian Networks, Approximate Inference in Bayesian Networks.

Learning from Examples: Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, The Theory of Learning, Regression and Classification with Linear Models, ANNs, Non-Parametric Models, SVMs, Ensemble Learning.

Unsupervised Learning: Clustering, dimensionality reduction, association analysis.

Statistical Learning Methods: Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: The EM Algorithm.

Introduction to Reinforcement Learning.

Reading List:

1. Stuart Russell, Peter Norvig, “Artificial Intelligence -A Modern Approach”, 3/e, Pearson, 2003.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, “The Elements of Statistical Learning: Data Mining, Inference, And Prediction”, 2nd Edition, Springer, 2017.
3. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2010.
4. https://github.com/emredjan/ISL-python

CS304	Artificial Intelligence lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming Lab (CS102)
ii. Data Structures and Algorithms (CS201)

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

CO1	Implement an efficient search technique for the given problem (Apply)
CO2	Implement an optimized multiplayer game (Apply)
CO3	Implement inference in probabilistic models (Apply)
CO4	Implement supervised ML models (Apply)
CO5	Implement unsupervised ML models (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	S	S	L			M	L	L		S	M	M
CO2	S	S	S	S	S	L			M	L	L		S	M	M
CO3	S	S	S	S	S	L			M	L	L		S	M	M
CO4	S	S	S	S	S	L			M	L	L		S	M	M
CO5	S	S	S	S	S	L			M	L	L		S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Tutorial and Laboratory Exercises:

1. Implement uninformed search techniques such as DFS, BFS, and IDS.
2. Implement informed search technique A* with suitable heuristic functions.
3. Implement local search techniques such as Hill Climbing, GA, and SA.
4. Implement multiplayer games using minimax, $\propto$-β pruning, and expecti minimax. (e.g: tic tac toe, Nim, a problem involving dice).
5. Solve Constraint Satisfaction Problems (e.g: Map coloring).
6. Implement first-order logic inference.
7. Implement Inference in Bayesian Networks.
8. Implement a Decision Tree.
9. Implement Regression (Linear, Logistic)

10. Implement kNN for classification and Regression.
11. Implement SVM for classification and Regression.
12. Implement ANN for classification and Regression.
13. Implement clustering using the EM algorithm.
14. Use the EM algorithm for learning with hidden variables.

Reading List:

1. Stuart Russell, Peter Norvig, “Artificial Intelligence -A Modern Approach”, 3/e, Pearson, 2003.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, “The Elements of Statistical Learning: Data Mining, Inference, And Prediction”, 2nd Edition, Springer, 2017.
3. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2010.
4. Introduction to Statistical Learning in Python: https://github.com/emredjan/ISL-python
5. Representations and Inference for Logic: http://aima.cs.berkeley.edu/python/logic.html

CS351	Software Engineering	PCC	2-0-2	3 Credits

Pre-Requisites:

None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze various software engineering models for software project design and choose the most appropriate one for the problem. (Analyze)
CO2	Design and validate SRS documents to develop a quality project. (Create)
CO3	Evaluate the effectiveness of an organization's software development/ implementation practices, suggest improvements, and define a process improvement strategy. (Apply)
CO4	Draw UML diagrams in order to forward and reverse engineer the complex software projects. (Create)
CO5	Apply manual and automated testing methods to verify and validate the software. (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	L	M		L	M	L		M	M	M		S	S	
CO2	M	L	M			M	M		M	M	L		S	S	
CO3	M	L	M			M	M			L	M		S	S	
CO4	M	L	M		M	M	M		M	M	L		S	S	
CO5	M	L	M		M	M	M		M	M	L		S	S	

Detailed Syllabus:

Introduction to Software Engineering, Software Process, Software life cycle models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Spiral Model, Introduction to Agility, Agile process, Extreme programming, XP Process. Storyboarding.

Software Requirements: Functional and Non-Functional, User requirements, System requirements, Software Requirements Document, Attributes of bad SRS documents, Requirement Engineering Process: Feasibility Studies, Requirements elicitation and analysis, requirements validation, requirements management, Classical analysis: Structured system Analysis, Petri Nets, Data Dictionary.

Design process: Design Concepts, Design Model, Design Heuristic, Architectural Design, Architectural styles, Architectural Design, Architectural Mapping using Data Flow.

User Interface Design: Interface analysis, Interface Design, Component level Design: Designing Class based components, traditional Components.

Software Design: Approaches to software design: function-oriented design, object-oriented design. Object Modelling Using UML: Basic object orientation Concepts, Use case Model, Class diagram, Interaction diagrams, Activity Diagram, state chart Diagram, Component and Deployment diagrams. System design examples.

Software testing: White box testing, black box testing, Regression Testing, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging. Test driven development, Junit, Testing applications: conventional, object-oriented, and, web. Software configuration management: Product metrics, Assessment: Team Analysis in Metrics Calculation.

Software Implementation Techniques: Coding practices, Refactoring, Maintenance and Reengineering, BPR model, Reengineering process model, Reverse and Forward Engineering. Version control, CI/CD.

Software Project Management: LOC Estimation, FP Based Estimation, COCOMO I & II Model. Project Scheduling, Earned Value Analysis Planning, Project Plan, Planning Process, RFP Risk Management, Identification, Projection, Risk Management, Risk Identification, Risk Mitigation, RMMM Plan, CASE TOOLS.

List of Experiments:

Choose a real-life application and incrementally build the application using the SE practices as a case study. You are advised to simulate the distributed environment for the product development. Students are encouraged to use pair programming for the development of the product.

1. Identify the requirements from the problem statement. Conduct feasibility study and validate the requirements.
2. Prepare the SRS document.
3. Estimate the project cost using project estimation techniques and complexity metrics.

4. Model use case diagrams and by capturing the use case scenarios from the problem statement. Model Class diagrams and Sequence diagrams. Apply forward Engineering.

5. Estimate the Cyclomatic complexity. Write Junit test cases. Perform test driven development for every function.

6. Perform automatic testing of the application.

Reading List:

1. Roger S. Pressman, Software Engineering, A Practitioner's approach, McGraw Hill, 8^{th} edition, 2014.
2. Ian Sommerville, "Software Engineering", 7th edition, Pearson education.7/e, 2005.
3. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Publication, 2009.
4. Timothy C Lethbridge, "Object-Oriented Software Engineering, Practical software development using UML and Java", McGraw-Hill Education, 2^{nd} edition, 2004.
5. Jeevan Kumar, "Grokking the Object-Oriented Design Interview" https://akshay-iyangar.github.io/system-design/
6. Kent Beck, Cynthia Andres, "Extreme Programming Explained: Embrace Change", 2^{nd} edition, Addison Wesley, 2004.

CS352	Computer Networks	PCC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Operating Systems (CS202)

Course Outcomes: At the end of the course the student will be able to:

CO1	Assess the functionalities of OSI and TCP/IP models (Analyze)
CO2	Analyze MAC layer protocols and LAN technologies (Analyze)
CO3	Design applications using internet protocols (Apply)
CO4	Construct routing and congestion control algorithms (Apply)
CO5	Construct application layer protocols (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	M
CO2	S	M	L										S	S	M
CO3	S	M	L										S	S	M
CO4	S	M	L										S	S	M
CO5	S	S	S	M									S	S	S

Detailed syllabus:

Introduction – network architecture - protocol implementation issues - network design. Reference models - The OSI Reference Model - the TCP/IP Model - A Comparison of the OSI and TCP/IP Models

Datalink Layer- framing and error detection and correction - sliding window protocol - media access control - ethernet - CSMA - token ring - switching - spanning tree protocol - LANs.

Network layer – network layer design issues - internet protocol (IP) – addressing and forwarding – ARP protocols – network address translation (NAT) - routing algorithms - congestion control algorithms – Internetworking.

Transport layer – user datagram protocol (UDP) - transmission control protocol (TCP) – TCP connection management - TCP congestion control – TCP versions – sliding window implantation in TCP.

Application layer - domain name service (DNS) – simple mail transfer protocol – file transfer protocol - world wide web - hypertext transfer protocol – peer to peer to networking.

Reading List:

1. Larry L Peterson, Bruce S Davis, Computer Networks, 5th Edition, Elsevier, 2012.
2. Andrew S. Tanenbaum, David J Wetherall, Computer Networks, 5th Edition, Pearson Edu.

CS353	Web Application Development	PCC	2 – 0 – 2	3 Credits

Pre-requisites:

i. Database Management Systems (CS255)

Course Outcomes: At the end of this course, students will be able to:

CO1	Analyze the basic web components and development environment. (Analyze)
CO2	Analyze a standard way to organize the data using XML/JSON. (Analyze)
CO3	Construct a simple interactive web application using both client side and server-side scripting. (Apply)
CO4	Build a simple web application using RESTful web service. (Apply)
CO5	Examine the need of security in developing web applications. (Analyze)

Course Articulation Matrix:

PO/ PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L	M								M	L	L
CO2	S	M	M	L	M				M				S	M	L
CO3	S	M	S	L	S				S		M		S	M	L
CO4	S	M	S	M	S				S		M		S	M	L
CO5	S	M	S	M	S				S		M		S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Deconstructing the web from URL to website - Different types of web applications, HTML5 – Static vs Dynamic sites, HTML5: headers, Linking, Images, Image map, meta elements, frameset, HTML forms, cascading style sheet, DHTML: object model, Event model, two tier and Three tier architecture, tools to analyze HTTP requests.

UI design for web applications using CSS3.

Developer Tools: Introduction - Git, IDE/ Editors, Deployment, Console

Frameworks: Mean / Mern / Django

XML Basics- Structure- Elements and attributes- Namespaces- Working with DTD Schema- Grouping elements- writing and Parsing XML Document. JSON - Introduction.

Client-side scripting – Javascript (ES6+) / JQuery, DOM manipulation through XQuery, Responsive Web Design, Front end framework- Angular JS/ React JS.

Different kinds of servers - web servers – Apache & nginx, File servers – time servers- DB servers. Introduction to NodeJS – asynchronous nature of nodejs – simple apps Integrating application with DB – DB drivers- Integrate NodeJS with NOSQL, Integrate NodeJS with SQL.

Web Architecture & Web services: MVC introduction- thin clients Vs Thick clients. Web services – Introduction- SOAP, REST – writing a RESTful service (nodejs + express). SOAP Vs REST.

Web security model - HTTPS: Goals and Pitfalls - Injection flaws: Cross site scripting, SQL Injection, Command Injection, Cross site request forgery and HTTP header injection - Cookie Flaws and Server Misconfiguration - Session management and user authentication.

List of Experiments:

Choose a real-life application and incrementally build the full stack application as a case study.

1. Construct a static website using basic HTML5 and CSS3
2. Design XML document for formatting, styling, and schema
3. Construct a RESTful service by utilizing Back-end application
4. Develop a full stack application and deploy using Github/Heroku
5. Simulate SQL injection

Reading List:

1. Jeffrey C. Jackson, “Web Technologies – A Computer Science Perspective”, Pearson Education, Fourth Edition, 2012.
2. Paul Deitel, Harvey Deitel, Abbey Deitel, “Internet and World Wide Web How to Program”, Pearson, Fifth Edition, 2011.
3. Leonard Richardson & Sam Ruby, “RESTful Web Services”, O’Reilly, 2007
4. Anthony, Accomazzo, Murray Nathaniel, Lerner Ari, “Fullstack React: The Complete Guide to React JS and Friends”, Fullstack.io, 2017.
5. Brown, Ethan, “Web Development with Node and Express: Leveraging the JavaScript Stack”, O'Reilly Media, 2019.
6. Dayley B., “Node.js, MongoDB, and AngularJS Web Development”, Addison-Wesley Professional, 2014.
7. Laura Lernay, Rafe Colburn, Jennifer Kyrnir, “Mastering HTML, CSS &Javascript Web”, BPB Publications, 2016.

CS354	Network Programming Lab	PCC	0 – 1 – 2	2 Credits

Pre-requisites:

i. Operating Systems Lab (CS205)

Course Outcomes: At the end of the course the student will be able to:

CO1	Construct error detecting and correcting codes. (Apply)
CO2	Construct and analyze Routing Algorithms and Internetworking. (Analyze)
CO3	Assess packet sniffing and analyze packets in network traffic. (Analyze)
CO4	Construct programs for client-server applications. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	M
CO2	S	M	L										S	S	M
CO3	S	S	M	L									S	S	M
CO4	S	M	L										S	S	M

Detailed Syllabus:

Assignment-1: Programs to implement error detection and correction

Task: Propose a few approaches for error correction and implement the same.

Assignment-2: Client-Server applications using inter process communication mechanisms

a) FIFO b) Message queues c) Shared memory

Assignment-3: Implementation of routing algorithms

Task 1: Implement Distance Vector Routing and Link State Routing algorithms.

Task 2: Find MST (minimum spanning tree) and SPT (shortest path tree) for the following graph.
G (V1, V2, W) where G is a connected undirected nonnegative weighted graph with V as vertices, E as edges, Was Weight.

Graph

(a,c,6)
(a,b,6)
(a,d,6)
(b,d,2)
(c,d,2)

Assignment 4. Generate a forwarding table with following details

a. Number of subnets

b. Subnet mask

c. Broadcast address (Target)

d. Number of hosts per subnet

e. First Host ID

f. Last Host ID

g. Subnet address

Assignment-5: Implement ARP and RARP

Task: Assume a network of n processes, where each process is a node in the network. Let one process maintain a table of MAC id and IP addresses of each node. Each node knows its MAC id. Now simulate ARP through broadcast and targeted messages. Use IPC for the communication (which one do you prefer and why?). Which Data Structure will you use and why? Include simulation of assigning IP addresses, ARP query, and ARP response packets. Use the standard packet structure as far as possible.

Assignment-6: Installation of Wireshark.

Task 1: Perform packet sniffing that includes basic packet capturing, filtering and analyzing.

Task 2: Prepare a report on "observation of DHCP invocation through Wireshark"

Assignment-7: Implementation of application layer protocols

Task 1: Implement a relay-based Peer-to-Peer System using Client-Server socket programming. In this application, you require implementing three programs, namely Peer Client, and Relay Server and Peer Nodes, and they communicate with each other based on TCP sockets.

Task 2: Write a client-server programming: Client has a list of words (at least 20 words) and server has a dictionary (at least 15 entries). Client picks up a random word from its list and sends it to server. Server responds with its meaning, if found else adds this entry to its dictionary with its meaning.

Assignment-8: Connectionless Client-Server applications

Task: Write a client server socket programming: write a client-server hamming code check program steps to follow:

(a) Client sends a hamming code to the server.

(b) Server checks the data for any error and accepts it.

(c) Server replies with 'good data' / 'bad data' depending upon there is no error on with error.

Assignment-9: Implementation of chat servers and mail servers

Task 1: Write a client server socket programming, server stores a TXT file consisting of State Name and Capital city. Client will send 'State Name' as request, and server should respond with matching 'Capital City' and vice-versa, if found else adds this entry to the TXT file. Connection should terminate only if client terminates it.

Task 2: Write a client-server programming to download a file: client will specify commands like. /a. out < server ip >< port no >< get >< file name >. (Use different directories for client and server)

Assignment-10: Design and implementation of a text-based conferencing system

You are asked to design and implement a server-based text-based conferencing system with the following features.

- A server will maintain the accounts of the various users, through a simple user registration process.
- Users can log onto the system whenever they want.
- A designated user will be authorized to initiate a conference, view the users presently logged on, and invite them selectively to the conference.
- During the conference, any user can type some text at his/her terminal which can be seen by all other users participating in the conference.

Just the outline of the requirements has been specified. You are also required to first chalk out the detailed specifications of your design before starting to implement the same.

Assignment 11:

- Write a TCP client-server system to allow client programs to get the system date and time from the server. When a client connects to the server, the server gets the local time on the machine and sends it to the client. The client displays the date and time on the screen, and terminates.
- Write a simple UDP iterative server and client to convert a given DNS name (for example, www.google.com) into its IP address(es). The client will read the DNS name as a string from the user and send it to the server. The server will convert it to one or more IP addresses and return it back to the client. The client will then print ALL the addresses returned, and exit.

Reading List:

1. W. Richard Stevens, UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998.

2. W. Richard Stevens, UNIX Network Programming, Volume 2, Second Edition: Interprocess Communications, Prentice Hall, 1999
3. W. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Pearson Education, Second Edition.

IV Year B.Tech. (CSE) Courses offered by CSED

CS401	Cryptography and Engineering Secure Systems	PCC	3 - 0 - 0	3 Credits

Pre-requisites:

i. Operating Systems (CS202)
ii. Computer Networks (CS352)

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

CO1	Assess and exploit vulnerabilities in software and hardware platforms. (Analyze)
CO2	Assess the systems for certain form of vulnerabilities. (Analyze)
CO3	Construct software and hardware for security evaluation of well-known attacks. (Apply)
CO4	Construct and apply software validation and verification techniques to test security vulnerabilities. (Apply)
CO5	Construct case studies to think like an attacker in order to expose security vulnerabilities in software systems. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	-	-	-	-	-	-	-	-	S	S	S
CO2	S	S	S	M	-	-	-	-	-	-	-	-	S	S	S
CO3	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M
CO4	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M
CO5	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M

Detailed syllabus:

Cryptography: Introduction to Security - Classifications of Attacks; Security Services; Security Mechanisms - Mathematics of Cryptography – Stream and Block Ciphers – Public and Private Key Cryptosystems.

Hardware Security: Hardware Trojans and Detection; PUFs - Power Analysis Attacks and Countermeasures -Fault Attacks - Implementation Aspects of Crypto Algorithms (AES and ECC). Micro Architectural Security Timing attacks and Covert Channels - RAM based attacks - Cold boot.

Operating System Security: Stack Smashing Attacks - Dynamic Memory Allocation Attacks - Format String Vulnerabilities - ROP attacks - Side Channel Attacks in Operating Systems; Countermeasures - Non-executable stacks - Capability based Systems - Canaries - Malware Analysis Techniques.

Reading List:

1. A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRCPress, 2004.
2. Swarup Bhunia and Mark Tehranipoor, Hardware Security: A Hands-on Learning Approach, Elsevier, 2019.
3. William Stallings and Lawrie Brown, Computer Security: Principles and Practice, Pearson, 2017.
4. Trent Jaeger, Operating System Security, Morgan & Claypool Publishers, 2009.

CS402	Big Data Engineering	PCC	2 – 1 – 2	4 Credits

Pre-requisites:

i. Data Structures and Algorithms (CS201)
ii. Database Management Systems (CS255)
iii. Probability, Statistics and Stochastic Processes (MA239)

Course Outcomes: At the end of the course, the students will be able to:

CO1	Analyze Map-Reduce programming model for data-intensive applications (Analyze)
CO2	Build a Big Data computing platform with distributed file system. (Apply)
CO3	Implement optimized join operations for a distributed environment. (Apply)
CO4	Perform exploratory data analysis on a distributed environment. (Apply)
CO5	Construct and evaluate ML models for Big Data. (Apply)
CO6	Implement stream processing applications. (Apply)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	S	S				M	M		L	S	S	M
CO2	S	S	S	S	S				M	M		L	S	S	M
CO3	S	S	S	S	S				M	M		L	S	S	M
CO4	S	S	S	S	S				M	M		L	S	S	M
CO5	S	S	S	S	S				M	M		L	S	S	M
CO6	S	S	S	S	S				M	M		L	S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Big Data – An Introduction: Big Data - Definition, an overview of Big Data; HPC & Big Data, Big Data Characteristics, Issues, and challenges of Big Data, Big data Technological approaches and Potential use cases for Big Data.

MapReduce and the New Software Stack: Hadoop Framework, Hadoop echo system, Distributed File Systems (HDFS and YARN), MapReduce, Algorithms Using MapReduce: Matrix-vector multiplication, Matrix multiplication, Relational Algebra operations, Join operations: optimization for join operations, theta join, skew aware join, Extensions to MapReduce, The Communication-Cost Model, Complexity Theory for MapReduce.

Exploratory Data Analysis: Visual data analysis techniques, interaction, techniques-Systems and applications.

In memory processing: Apache Spark, RDD programming, Spark SQL, Datasets and Dataframes, SparkML.

Machine Learning for Big Data: Selecting, Extracting and Transforming features, Basic statistics, ML pipelines, Supervised and Unsupervised models, Model selection and tuning.

NoSQL: CAP Theorem, Birth of NoSQL, Key-value databases, Document databases, Graph Databases, Column Databases, Distributed Data bases: Mango DB, Hive, Cassandra.

Stream processing: Introduction to Streams Concepts, Stream Data Model and Architecture, Sampling Data in a Stream, Filtering Streams, Algorithms for streams such as Counting Distinct Elements in a Stream, Estimating Moments, Counting Ones in a Window, Decaying Window.

List of Experiments:

1. Install Hadoop as single node and multi node cluster.
2. Implement Map Reduce algorithm Word Count to be executed on the cluster with voluminous input data.
3. Implement Page Rank on a large web graph. (Map Reduce or Spark)
4. Implement optimized join operations on Spark (multi way and skew-aware joins)
5. Perform exploratory data analysis and visualization using Spark and MLlib.
6. Build supervised models using Spark and MLlib.
7. Build unsupervised models using Spark and MLlib.
8. Count triangles in a given graph using Neo4j.
9. Find a minimum spanning tree of a given graph using Neo4j.
10. Estimate moments of a data stream using Apache Kafka and Spark streaming.
11. Count distinct elements of a data stream Apache Kafka and Spark streaming.

Reading List:

1. Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, “Mining of Massive Datasets”, Cambridge Universities Press, 3rd ed, 2020.
2. David Loshin, “Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph”, Elsevier, August 23, 2013.

3. Tom White, “Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale”, Orielly, 4th Edition, 2015.
4. Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills, “Advanced Analytics with Spark: Patterns for Learning from Data at Scale”, Orielly, 2015.
5. Dan Sullivan, “NoSQL for Mere Mortals”, Addison-Wesley, 2015
6. https://spark.apache.org/docs/latest/api/python/pyspark.html
7. https://github.com/adamjshook/mapreducepatterns
8. https://spark.apache.org/mllib/

SM430	Entrepreneurship for Engineers	HSC	3 – 0 – 0	3 Credits

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

CO1	Acquaint themselves with starting new ventures and introducing new products and service ideas
CO2	Explore the processes of establishing a start-up
CO3	Develop strategies and methods to mobilize resources
CO4	Create venture capitalists, consultants to new firms or new business development units of larger corporates

Detailed syllabus:

The Early Career Dilemmas of an Entrepreneur: Discover ourselves- The Entrepreneur's Role, Task and Personality- A Typology of Entrepreneurs: Defining Survival and Success, Entrepreneurship as a Style of Management- The Entrepreneurial Venture and the Entrepreneurial Organisation- Identify Problems Worth Solving-Customer Identification-Choosing a Direction Opportunity recognition and entry strategies- Business model identification- validation- New product Franchising-Partial Momentum- Sponsorship and Acquisition- The Strategic Window of Opportunity- Scanning-Positioning and Analysing, Intellectual Property-Creation and Protection

Gaining Commitment- Gathering the Resources- The Business Plan as an Entrepreneurial Tool- Financial Projections: how to do them the right way- Debt- Venture Capital and other forms of Financing-Sources of External Support-Developing Entrepreneurial Marketing-Competencies- Networks and Frameworks- ustaining Competitiveness- Maintaining Competitive Advantage- The Changing Role of the Entrepreneur- Mid- Career Dilemmas-Harvesting Strategies versus Go for Growth

Characteristics and special needs- Business/project planning- Business Plan preparation- Implementation Process- Planning support systems (enterprise operation)- Legal Issues (licensing, patents, contracts etc.)-Effectuation and Causation-Art of negotiation-Conflict
Management

References:

B.D.Singh. *Managing Conflict and Resolution.* Excel Books.2008
B. R. Barringer and D. Ireland, *Entrepreneurship*, Prentice Hall, 2009.
G. Kawasaki, L. Filby, *The Art of the Start 2.0: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything* , Penguin, 2015.
R. Bansal, *Connect the Dots*, Westland, 2011.
Ries, Eric *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.
S. S. Khanka, *Entrepreneurial Development*, S.Chand &Co. 2006.
S. Blank and B. Dorf, *Startup Owner's Manual: The Step-By- Step Guide for Building a Great Company*, K&S Ranch Publishing, 2012.

Elective Courses offered by CSED

CS361	Optimization Techniques	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will learn how to model problems as linear/quadratic constrained optimization objectives and to solve the problems using efficient algorithms. Students will also learn notions of convexity and its impact in realizing efficient solutions. Further students shall be learning the concepts of solving constrained non-linear optimization objectives.

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)

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

CO1	Formulate a given optimization scenario as a suitable linear/non-linear constrained objective. (Analyze)
CO2	Construct efficient algorithms for solving linear programming for engineering application scenarios. (Apply)
CO3	Construct efficient algorithms for solving quadratic optimization objectives (Apply)
CO4	Analyze the properties of convexity and duality gap of optimization objectives. (Analyze)
CO5	Construct efficient algorithms for constrained non-linear objectives (Analyze)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	M									S	M	L
CO2	S	M	L	L									S	L	L
CO3	S	S	M	M									S	M	L
CO4	S	S	M	L									S	L	L
CO5	S	S	M	M									S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction: Linear and Integer Programs: Formulating real world problems as linear and integer linear programs, formulating combinatorial optimization problems as integer linear programs, recap of important concepts in linear algebra. Geometry of Polyhedra: Feasible region of LPs and polyhedra

Solving linear programs: Possible outcomes (infeasibility, unboundedness, and optimality) and their certificates, bases and canonical forms, the Simplex method and its geometric interpretation, the ellipsoid method and separation oracles.

Convex sets and functions: convex sets, examples and properties, convex functions, strict and strong convexity, examples, and convexity preserving operations Equivalent definitions of convexity under differentiability assumptions.

Unconstrained Optimization: Maxima, minima, stationary point, saddle point, local and global maximum/minimum. First order and second order conditions for optimality. Linear, quadratic and convex optimization problems, examples.

Constrained Optimization: Constrained Optimization problem, feasible set. Lagrangian, KKT conditions for Linear and quadratic Optimization

Duality for Convex Optimization theorem of alternatives, Farka's lemma.

Algorithms for optimization: Gradient descent with fixed step size, line search and Armijo-Goldstein rule, Newton method and variations, Conjugate gradient and Quasi-newton methods,

Algorithms for constrained Optimization: Projected gradient descent, dual ascent, alternating direction method of multipliers.

Reading List:

1. Boyd, Stephen, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.
2. Luenberger, David G., and Yinyu Ye. Linear and nonlinear programming. 4th edition. Springer, 2015.
3. Cook, Cunningham, Pulleyblank and Schrijver. Combinatorial Optimization. Wiley-Interscience, 1998.
4. Nocedal and Wright. Numerical Optimization. Springer, 2006.
5. Bertsekas, Dimitri P. Nonlinear programming. Belmont: Athena scientific, 1999.
6. Research articles as chosen by the instructor.

CS362	Design and Analysis of Parallel Algorithms	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will learn design and analysis principles for constructing scalable parallel algorithms across various Flynn's taxonomy of parallel computers. The course deals with problems from both numerical and non-numerical natures such as linear algebraic problems, graph theoretic problems, discrete optimization search related problems, and dynamic programming algorithms.

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)

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

CO1	Analyze the resource requirements of parallel algorithms with a comprehension of the parallel computer features and the underlying type of interconnection network. (Analyze)
CO2	Infer the most suitable parallel algorithm among a set of candidate algorithms to solve a given problem based on metrics such as cost, speed-up, efficiency and scalability. (Analyze)
CO3	Construct parallel algorithms for solving numerical problems from domains like Linear Algebra having representative problems such as matrix multiplication, transpose, and direct and iterative methods for solving systems of linear equations. (Apply)
CO4	Construct parallel algorithms for solving problems from non-numerical domains like graph theory, search in discrete optimization, and monadic and polyadic dynamic programming problems. (Apply)
CO5	Construct implementations of parallel algorithms on top of message-passing based parallel programming frameworks like Message Passing Interface. (Apply).

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	S	M										S	M	L
CO3	S	M	L										S	M	L
CO4	S	M	L										S	M	L
CO5	S	M	L										S	S	L

S: Strong correlation, M: Medium correlation, L: Low correlatio

Detailed Syllabus:

Introduction: The need for parallel computers, Models of computation (SISD, MISD, SIMD, MIMD), Analyzing algorithms, Expressing Algorithms.

Selection and Merging: Introduction – Selection and Merging, the problem and a lower bound, A Sequential algorithm, Desirable properties for parallel algorithms, broadcasting a datum, Computing All Sums, An algorithm for parallel selection, A network for merging, Merging on CREW model, Merging on the EREW model.

Sorting: Introduction, A network for sorting, Sorting on a linear array, Sorting on the CRCW model, Sorting on CREW model, Sorting on the EREW model.

Searching: Introduction, searching a sorted sequence, EREW, CREW, CRCW searching, searching a random sequence, Searching on SM SIMD computers, searching on a Tree, Searching on a Mesh.

Performance Analysis: Amdahl's Law, Gustafson Barsis's law- Karp Flatt metric- Iso efficiency metric.

Numerical problems and implementation: Matrix operations, Transposition, Mesh Transpose, Shuffle Transpose, EREW Transpose, Matrix by Matrix multiplication, Mesh multiplication, Cube multiplication, CRCW multiplication, Matrix by Vector multiplication, Linear Array multiplication, Tree multiplication, Convolution, solving systems of linear equations (SIMD/MIMD), Finding roots of non-linear equations.

Graph Theory: connectivity matrix computation, finding connected components, computing all-pair shortest paths, minimum spanning trees, Traversals: Parallelizing breadth first and depth first searches.

Optimization problems: Job sequencing with deadlines, Knapsack problem.

MPI primitives with case-studies on implementation of parallel algorithms.

Reading List:

1. S.G. Akl, "The design and analysis of parallel algorithms", Prentice Hall of India, 1989.
2. Michael Jay Quinn, "Parallel programming in C with MPI and OpenMP", McGraw-Hill Higher Education, 2004.
3. S. Lakshmivarahan and S.K. Dhall, "Analysis and design of parallel algorithms – Arithmetic and Matrix problems", McGraw Hill, 1990.

CS363	Graph Algorithms	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will learn how to model problems as graph-based problems and to solve the problem. Students will also learn standard graph algorithms and notions in this course across a spectrum of problems from the conventional spanning trees, network flows to the contemporary social network analytics.

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)

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

CO1	Formulate a given problem as a suitable graph-theoretic problem. (Analyze)
CO2	Construct efficient algorithms for standard graph-theoretic problems with an analysis of their time and space requirements. (Apply)
CO3	Construct an algorithm to solve a given graph-theoretic problem by applying a standard graph theoretic algorithm in isolation or by combining multiple standard algorithms in a pipeline. (Analyze)
CO4	Analyze the properties of a given social network using standard graph-theoretic notions and algorithms. (Analyze)
CO5	Construct and analyze algorithms for solving network-flow-based problems with a comprehension of the relationship between cuts and flows. (Analyze)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Basic definitions and notions in Graph theory, Representation of graphs – a review, Motivational examples for graphs: Web search engines and page rank algorithm, Web crawlers and social networks.

Strongly Connected Components, Biconnected components, Tarjan's algorithm, Invasion percolation with a case study on simulating river networks, Spanning Trees, Prim's algorithm, Kruskal's algorithm – a review, Prim-Dijkstra-Jarnik algorithm, Boruvka's algorithm.

Review: DFS-BFS algorithm, Applications of DFS-BFS, Directed Acyclic Graphs, Topological ordering, Algorithms to find single source shortest paths, All pair shortest paths.

Johnson's algorithm, Suurballe's Algorithm, Case studies on Map path planning, Landmark-based distance estimation, Widest path problems.

Eulerian graphs and standard results relating to characterization. Hamiltonian graphs, Dirac theorem, Chavathal theorem, closure of graph. Non-Hamiltonian graph with maximum number of edges. Self-centered graphs and related theorems.

Graph Coloring, Chromatic number, Greedy heuristic for Graph coloring, Interval graphs, Perfect graphs, Chordal graphs.

Planar graphs, Euler's formula, maximum number of edges in a planar graph. Five colour theorem. Graph Isomorphism.

Travelling Salesperson Problem, MST-doubling heuristic for finding TSP, Christofide's algorithm, Dynamic programming formulation for TSP – a review.

Matching, Bipartite graphs, Hopcroft-Karp algorithm, Finding Vertex covers and Independent Sets using matching, Stable matching, Gale-Shapley algorithm for Stable matching.

Max flow-Min Cut theorem, Ford-Fulkerson's algorithm, Case Study: Bipartite maximum matching as a flow problem.

Social network analytics: Erdős numbers, the Oracle of Bacon, and the Milgram small-world experiment, Properties – sparsity and power law, centrality, degeneracy and k-cores, Network models - Erdős–Rényi model, ERGM, random graphs with fixed degrees, Barabási-Albert model, Kleinberg's model, Network Cliques, Bron-Kerbosch algorithm.

Case-studies from competitive programming venues.

Reading List:

1. Douglas B. West, Introduction to Graph Theory, Second Edition, Pearson, 2000.
2. M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, First Edition, Academic Press, 1997.
3. R. J. Wilson and J. J. Watkins, Graphs: An Introductory Approach, First Edition, Wiley, 1990.

CS364	Advanced Data Structures	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Data Structures and Algorithms (CS201)

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

CO1	Construct Double Ended Heap structures and Heaps of optimal Complexity. (Apply)
CO2	Construct a suitable hashing method to facilitate efficient searching. (Apply)
CO3	Construct a suitable data structure for computational Geometry problems. (Apply)
CO4	Construct a suitable data structure for representing sets of Intervals namely Segment Trees, Orthogonal Range Trees. (Apply)
CO5	Construct suitable data structures for representing strings for efficient search and retrieval operations. (Apply)
CO6	Construct suitable persistent and dynamic data structures for solving a given complex engineering problem. (Apply)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	
CO2	S	M	L										S	S	
CO3	S	M	L										S	S	
CO4	S	M	L										S	S	
CO5	S	M	L										S	S	
CO6	S	M	L										S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Heaps: Heaps of optimal complexity, Double Ended Heap structures and Multidimensional Heaps, Heap related structures with Constant-Time updates.

Hashing: Perfect Hashing, Universal Hashing, Cuckoo Hashing and Random Graphs

Computational Geometry: One Dimensional Range Searching, Two-Dimensional Range Searching, constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quadtrees.

Tree Structures for Sets of Intervals: Interval Graphs, Interval Trees, Segment Trees-Trees for union of intervals, Trees for sums of weighted intervals, Trees for interval restricted Maximum sum queries, Orthogonal Range Trees-Higher Dimensional Segment Trees, Other systems of building blocks, Range counting and the semi group model.

Data Structure Transformations: Making structure Dynamic, Making structures Persistent.

Data Structure for Strings: Suffix Trees, Suffix Arrays, Linear Time construction of Suffix Trees, Aho-Corasick algorithm for pattern searching.

Probabilistic Data Structures: Skip Lists, Bloom Filters, Random Binary Trees, Cuckoo Filters, Treaps, Rapidly-exploring Random Trees.

Reading List:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, second Edition, Pearson, 2004.
2. Michael T. Goodrich, Roberto Tamassia, Algorithm Design, First Edition, Wiley, 2006.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, PHI, 2009.
4. Peter Brass, Advanced Data Structures,First Edition, Cambridge University Press, 2008.
5. Reinhard Kutzelnigg, Random Graphs and Cuckoo Hashing, First Edition, Sudwestdeutscher Verlag Fur Hochschulschriften AG, 2009.

CS371	Business Intelligence and Data Warehousing	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)
ii. Database Management Systems (CS255)
iii. Database Management Systems Lab (CS257)

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

CO1	Apply computerized support for managerial decision making
CO2	Design decision support system to support for decision making
CO3	Design Data warehouse
CO4	Develop business reporting, data/information visualization techniques for business performance management
CO5	Analyze how analytics are powering consumer applications and creating a new opportunity for entrepreneurship

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	L							S	S	S
CO2	S	S	S	M	M	L							S	S	S
CO3	S	S	S	M	M	L							S	S	S
CO4	S	S	S	M	M	L							S	S	S
CO5	S	S	S	M	M	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction:

Business Intelligence, Analytics, and Decision support: Business Environments and Computerized Decision Support, Managerial Decision Making, Information Systems Support for Decision Making, Framework for Computerized Decision Support, Decision Support Systems, Framework for Business Intelligence, Business Analytics

Foundations and technologies for Decision Making: Decision Making, Phases of the Decision-Making Process, Decision Making: Phases, Support for Decision Making, Decision Support Systems: Capabilities, Classifications, Components.

Data Warehousing: Definitions and Concepts, Process Overview, Architectures, Data Integration and the Extraction, Transformation, and Load (ETL) Processes, Data Warehouse Development, Data Warehousing Implementation, Real-Time Data Warehousing, Data Warehouse Administration, Security Issues, and Future Trends

Business Reporting, Visual Analytics, and Business Performance Management: Business Reporting Definitions and Concepts, Data and Information Visualization, Different Types of Charts and Graphs, Visual Analytics, High-Powered Visual Analytics Environments, Performance Dashboards, Business Performance Management, Performance Measurement, Balanced Scorecards, Six Sigma as a Performance Measurement System.

Business Analytics: Emerging trends and Future Impacts

Location-Based Analytics for Organizations, Analytics Applications for Consumers, Recommendation Engines, Web 2.0 and Online Social Networking, Cloud Computing and BI, Impacts of Analytics in Organizations, Issues of Legality, Privacy, and Ethics, Analytics Ecosystem

Reading List:

1. Ramesh Sharda, Dursun Delen, Efraim Turban, Business Intelligence and Analytics Systems for Decision Support, 10th Edition, Pearson Education, 2014
2. Howson, C, Successful Business Intelligence: Secrets to Making BI" a Killer app. McGraw - Hill, 2007.
3. David Loshin, Business Intelligence - The Savy Manager's Guide Getting Onboard with Emerging IT, Morgan Kaufmann Publishers, 2009

CS372	Advanced Data Mining	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)
ii. Database Management Systems (CS255)
iii. Database Management Systems Lab (CS257)

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

CO1	Identify and characterize sequence families. (Analyze)
CO2	Design models for analyzing stream data. (Analyze)
CO3	Summarize effectively which patterns of the graph stand out. (Apply)
CO4	Analyze the large-scale data that is derived from social networks (Analyze)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	L							S	S	S
CO2	S	S	S	M	M	L							S	S	S
CO3	S	S	S	M	M	L							S	S	S
CO4	S	S	S	M	M	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Sequential Pattern Mining concepts: Frequent and Closed Sequence Patterns: Sequential Patterns, GSP: An Apriori-like Method, PrefixSpan: A Pattern-growth, Depth-first Search Method, Mining Sequential Patterns with Constraints, Mining Closed Sequential Patterns;

Classification, Clustering, Features and Distances of Sequence Data: Three Tasks on Sequence Classification/Clustering, Sequence Features, Distance Functions over Sequences, Classification of Sequence Data, Clustering Sequence Data.

Mining Data Streams: The Stream Data Model, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Counting Ones in a Window, Decaying Windows

Graph Mining: Patterns in Static Graphs: Heavy-tailed Degree Distribution, Eigenvalue Power Law (EPL), Small Diameter, Triangle Power Laws (TPL) **Patterns in Evolving Graphs:** Shrinking Diameters, Densification Power Law (DPL), Diameter-plot and Gelling Point, Oscillating Non-Largest Connected Components Sizes, Principal Eigenvalue Ova-Time **Patterns in Weighted Graphs:** Snapshot Power Laws (SPL) – Fortification, Weight Power Law (WPL), Weighted Principal Eigenvalue Over Time.

Mining Social-Network Graphs: Social Networks as Graphs, Clustering of Social-Network Graphs, Direct Discovery of Communities, Partitioning of Graphs, Finding Overlapping Communities, Simrank, Counting Triangles, Neighbourhood Properties of Graphs

Reading List:

1. [1] G Dong and J Pei, Sequence Data Mining, Springer, 2007
2. Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman “Mining of Massive Datasets”, Cambridge University Press, 2014
3. Dunham, Margaret H., Data Mining: Introductory and Advanced Topics, Prentice Hall PTR, USA, 2002.
4. Deepayan Chakrabarti, Christos Faloutsos, Graph mining laws tools and case studies, Morgan and Claypool, 2012.

CS373	Applied Machine Learning	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Artificial Intelligence (CS303)

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

CO1	Constructing learning objectives and associated algorithms for Regression and Classification using Probabilistic Approaches (Apply)
CO2	Constructing learning objectives and algorithms for regression and classification using the maximum margin principle (Apply)
CO3	Constructing algorithms for mixture models using Expectation Maximization (Apply)
CO4	Designing models for Sequence Labelling problems (Apply)
CO5	Applying feature reduction and clustering strategies for different feature categories (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	M								L	S	M	S
CO2	S	M	L	M								L	S	L	S
CO3	S	S	M	M								L	S	M	S
CO4	S	S	M	M								L	S	M	S
CO5	S	S	M	M								L	S	M	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Review: Basics of Linear Algebra, Probability Theory and Optimization: Vectors, Inner product, Outer product, Inverse of a matrix, Eigenanalysis, Singular value decomposition, Probability distributions – Discrete distributions and Continuous distributions; Independence of events, Conditional probability distribution and Joint probability distribution, Bayes theorem, Unconstrained optimization, Constrained optimization – Lagrangian multiplier method.

Dimensionality Reduction Techniques: Principal component analysis, Fisher discriminant analysis, Multiple discriminant analysis, Probabilistic PCA, Feature Reduction: case study

Pattern Clustering: Criterion functions for clustering, Techniques for clustering -- K-means clustering, Hierarchical clustering, Density based clustering and Spectral clustering; Cluster validation, Spectral Clustering for transfer learning

Methods for Function Approximation: Linear models for regression, Parameter estimation methods - Maximum likelihood method and Maximum a posteriori method; Regularization, Ridge regression, Lasso, Bias-Variance decomposition, Bayesian linear regression.

Probabilistic Models for Classification: Bayesian decision theory, Bayes classifier, Minimum error-rate classification, Normal (Gaussian) density – Discriminant functions, Decision surfaces, Maximum-Likelihood estimation, Maximum a posteriori estimation; Naive Bayes classifier, Non-parametric techniques for density estimation -- Parzen-window method, K-nearest neighbors method, Softmax(Maximum Entropy) Classification, efficient parameter estimation for large-scale softmax based classifier

Maximum Margin Approaches: Support Vector Regression: Primal and Dual forms, Support Vector Classification: Primal and Dual forms, Linear vs Non-Linear Kernels, SVM for Non-Linear Decision boundary, multi-Class SVM

Mixture Models: Gaussian mixture models -- Expectation-Maximization method for parameter estimation; Mixture of Multinomials, EM algorithm for semi-supervised learning,

Sequence Labelling Models: Hidden Markov models (HMMs) for sequential pattern classification: forward backward algorithm, Viterbi algorithm, Conditional Random Fields (CRF), Parameter estimation for CRF, POS Tagging/Named Entity Recognition using HMM/CRF

Reading List:

1. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006
2. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001
3. S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, 2009
4. E. Alpaydin, Introduction to Machine Learning, Prentice-Hall of India, 2010
5. G. James, D. Witten, T. Hastie and R. Tibshirani, Introduction to Statistical Learning, Springer, 2013.
6. Charles Sutton and Andrew McCallum. 2012. An Introduction to Conditional Random Fields. Found. Trends Mach. Learn. 4, 4 (April 2012), 267–373.
7. L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989, doi: 10.1109/5.18626.

CS374	Natural Language Processing	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Data Structures and Algorithms (CS201)

ii. Probability, Statistics and Stochastic Processes (MA239)

Course Outcomes: At the end of the course, the students will be able to:

CO1	Perform language modelling with n-grams (apply)
CO2	Build a text classification model and evaluate its performance (apply)
CO3	Use word vectors and neural models to build text-based applications (apply)
CO4	Apply sequence labelling to perform POS and NER tagging (apply)
CO5	Apply grammars and parsers to build NLP applications (apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	L		L					S	M	M
CO2	S	S	S	M	M	L		L					S	M	M
CO3	S	S	S	M	M	L		L					S	M	M
CO4	S	S	S	M	M	L		L					S	M	M
CO5	S	S	S	M	M	L		L					S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction, Regular Expressions, Text Normalization, and Edit Distance.

N-Gram Language Models, Evaluating Language Models, Generalization and Zeros, Smoothing, Perplexity's Relation to Entropy.

Spelling Correction and the Noisy Channel.

Naive Bayes and Sentiment Classification: Naive Bayes Classifiers: training, example, Optimizing for Sentiment Analysis, Naive Bayes for other text classification tasks, Naive Bayes as a Language Model, Evaluation: Precision, Recall, F-measure, Statistical Significance Testing.

Vector Semantics and Embeddings, Lexical Semantics, Vector Semantics, Words, and **Vectors**: Cosine similarity, TF-IDF, Pointwise Mutual Information (PMI), Applications, Word2vec, Semantic properties of embeddings, Bias and Embeddings, Evaluating Vector Models.

Neural Networks and Neural Language Models: XOR problem solution with NN, Feed Forward NN for NLP: Classification, Feed-Forward neural language model, Training the neural language model.

Sequence Labelling for Parts of Speech and Named Entities, HMM POS Tagging, Conditional Random Fields, Evaluation of Named Entity Recognition.

Constituency Grammars: Context-Free Grammars, Some Grammar Rules for English, Treebanks, Lexicalized grammars.

Constituency Parsing: Ambiguity, CKY Parsing: A Dynamic Programming Approach, Span-Based Neural Constituency Parsing, CCG Parsing.

Dependency Parsing: Dependency Relations, Dependency Formalisms, Transition-Based Dependency Parsing, Graph-Based Dependency Parsing.

Applications: Information Extraction, Text Summarization, Sentiment Analysis, and Opinion Mining.

Reading List:

1. Daniel Jurafsky and James H. Martin, “Speech and Language Processing”, 3rd ed, 2021, https://web.stanford.edu/~jurafsky/slp3/
2. James Allen “Natural Language Understanding”, Pearson Education, 2003
3. Christopher D.Manning and Hinrich Schutze, “ Foundations of Statistical Natural Language Processing “, MIT Press, 1999.
4. Daniel Jurafsky and James H. Martin, “Speech and Language Processing”, Pearson, 2008.
5. Charniack, Eugene, “Statistical Language Learning”, MIT Press, 1993.

CS375	Advanced Computational Statistics	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will gain a broad comprehension of the importance of computation in statistics and machine learning. Focus of this course will be on the mathematical and statistical underpinnings of why and how seminal modern statistical methods and inference works.

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)

ii. Differential and Integral Calculus (MA101)

iii. Matrices and Differential Equations (MA151)

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

CO1	Apply theorems and probability axioms in constructing statistical models and inferring their parameters. (Apply)
CO2	Apply CDF estimation techniques, hypothesis testing, and Bayesian inference in performing inference with respect to parametric and non-parametric models. (Apply)
CO3	Construct statistical models using stochastic processes with a comprehension of their underlying representational ability in modeling the uncertainty in the problem domain. (Apply)
CO4	Apply stochastic optimization techniques and Bayesian modeling / inference techniques with a comprehension of how randomness and uncertainty in the domain is modeled. (Apply)
CO5	Illustrate Monte Carlo methods with an understanding of the role of various random sampling strategies in solving a given problem. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	S
CO2	S	M	L										S	S	S
CO3	S	M	L										S	S	S
CO4	S	M	L										S	S	S
CO5	S	M	L										S	S	S

CO6	M	M	L										M	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Probability:

Review: Probability (Sample spaces, Conditional probability, independent events, Bayes' theorem).

Review: Random variables (Distribution and probability functions, Discrete and continuous random variables, Bivariate, Marginal and conditional distributions, Multivariate distributions and IID samples, Expectation, Variance, Covariance, Conditional expectation, Moment generating functions).

Probability Inequalities, Convergence types of Random variables, Law of large numbers, Central limit theorem, The Delta method, L1 Convergence.

Statistical Inference:

Parametric and non-parametric models, Point estimation, Confidence sets, Hypothesis testing a review.

CDF estimation, Statistical functionals, Bootstrap: Simulation, Variance estimation, Confidence intervals, Percentile intervals.

Parametric inference: Method of moments, Maximum likelihood estimators: Properties, Consistency and Equivariance, Asymptotic normality, Optimality, Multiparameter models, Parametric bootstrap, Sufficiency, Exponential families and Conditional maximum likelihood estimators. Case study on Noise Contrastive Estimation.

Hypothesis testing and p-values: Wald test, Chi-square distribution, Pearson's Chi-square test for multinomial data, Permutation test, Likelihood ratio test, Multiple testing, Goodness-of-fit tests.

Bayesian Inference: Functions of parameters, Simulation, Large sample properties of Bayes' procedures, Flat priors, improper priors and non-informative priors, Multiparameter problems, Bayesian testing, Strengths and weaknesses of Bayesian inference.

Statistical Processes and Methods:

Stochastic Processes: Markov processes, Poisson processes.

Stochastic optimization: Robbins-Monro and Kiefer-Wolfowitz algorithms, simulated annealing, stochastic gradient methods.

Simulation methods:

Monte Carlo methods: Rejection sampling, importance sampling, variance reduction methods (Rao-Blackwellization, stratified sampling).

Reading List:

1. Larry Wasserman, All of Statistics, First Edition, Springer-Verlag, 2004.

2. Geof H. Givens and Jennifer A. Hoeting, Computational Statistics, Second Edition,Wiley, 2005.
3. Christian P. Robert, George Casella, Monte Carlo Statistical Methods, Springer, First Edition, 2004.

CS381	Agile Methodologies	DEC	3-0-0	3 Credits

Pre-requisites:

i. Software Engineering (CS351)

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply agile methodology and agile process to create high quality software. (Apply)
CO2	Use Agile methodology for knowledge management. (Apply)
CO3	Apply Agile development and testing techniques to manage risks. (Apply)
CO4	Analyze the pros and cons of working in Agile Team. (Analyze)
CO5	Apply Feature Driven Development on large size projects. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	M	M	L	L	L	M	M	M	L	S	M	
CO2	M	M	M	S	M	L	L	L	M	M	M	L	S	M	
CO3	M	M	M	S	M	L	L	L	M	M	M	L	S	M	
CO4	M	M	L	S	S	L	L	L	M	M	M	L	M	M	
CO5	M	M	L	M	L	L	L	L	M	M	M	L	S	M	

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Agile Methodology: Theories for Agile Management, Agile Software Development – Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Project Management, Agile Team Interactions, Ethics in Agile Teams, Agility in Design, Testing, Agile Documentations, Agile Drivers, Capabilities and Values.

Agile Process: Lean Production - SCRUM, Crystal, Feature Driven Development- Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.

Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM).

Agility and Requirements Engineering: Impact of Agile Processes in RE, Current Agile Practices, Variance, Overview of RE Using Agile, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Abstraction Model, Requirements Management in Agile

Environment, Agile Requirements Prioritization, Agile Requirements Modelling, Generation – Concurrency in Agile Requirements Generation.

Agility and Quality Assurance: Agile Product Development, Agile Metrics, Feature Driven Development (FDD), Financial and Production Metrics in FDD, Agile Approach to Quality Assurance, Test Driven Development, Agile Approach in Global Software Development.

Reading List:

1. David J. Anderson and Eli Schragenheim, Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Pearson, 2003.
2. Hazza and Dubinsky, Agile Software Engineering, Springer, 2009th edition, 2008.
3. Craig Larman, —Agile and Iterative Development, Addison-Wesley, First Edition, 2003.

CS382	Software Testing	DEC	3-0-0	3 Credits

Pre-requisites:

i. Software Engineering (CS351)

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze various test processes and continuous quality improvement. (Analyze)
CO2	Classify and analyze types of bugs, errors and faults based on severity levels. (Analyze)
CO3	Model the behaviour of the software application using FSM. (Apply)
CO4	Apply software testing techniques suitable to the environment used for building the software. (Apply)
CO5	Identify the suitable testing tools for an application to generate reports using that tool. (Analyze)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	M	L	L		L	M	M		L	M	M	
CO2	M	M	M	S	M	L		L	L	L		L	M	M	
CO3	S	L	M	S	M	L		L	M	L		L	M	M	
CO4	M	M	L	S	M	L		L	M	L		L	M	M	
CO5	M	M	L	M	M	L		L	L	M		L	M	M	

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction: Testing as an Engineering Activity, Principles of Testing, Vmodel concepts, Tester's Role in a Software Development Organization, Bugs, Origins of Defects, Cost of defects, Defect Classes, The Defect Repository and Test Developing a Defect Repository, Testing strategies and techniques: Design, Developer/Tester Support of Defect Prevention strategies.

Testing Strategies and Techniques: Unit Testing, integration testing, system and acceptance testing, performance testing, regression testing, internationalization testing, ad hoc testing, object-oriented testing, Usability and Accessibility Testing.

Black Box Approach to Test Case Design: Random Testing, Requirements based testing, Equivalence partitioning, boundary value analysis, cause effect graphing, state-based testing, domain testing.

White Box Approach to Test design: testing, Test Adequacy Criteria, static testing vs. structural adequacy criteria based on control flow, principles of mutation testing, equivalent mutants, fault detection using mutation, Test assessment using mutation.

Test Management: People and organizational issues in testing, Organization structures for testing teams, testing services, Test Planning, Test Plan Components, Test Plan Attachments, Locating Test Items, test management, test process, Reporting Test Results, Introducing the test specialist Test Automation Skills needed by a test specialist, Building a Testing Group.

Software Test Automation: Software test automation, skill needed for automation, scope of automation, design and architecture for automation, requirements for a test tool, challenges in automation, Test metrics and measurements, project, progress and productivity metrics.

Reading List:

1. Aditya P. Mathur “Foundations of Software Testing”, Second Edition, Pearson 2014.
2. Srinivasan Desikan, Gopalaswamy Ramesh, “Software testing Education, principles and practices”, First Edition, Pearson Education, 2009
3. Naresh Chauhan, Software Testing, Principles and practices, Oxford Press, Second Edition, 2016.

CS383	Design Patterns	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Object Oriented Programming (CS251)

Course Outcomes: At the end of this course, students will be able to:

CO1	Analyze common design patterns in the context of incremental/iterative development (Analyze)
CO2	Evaluate software source code using patterns (Evaluate)
CO3	Analyze software design practices through design patterns (Analyze)
CO4	Implement the design patterns in an object-oriented language (Apply)
CO5	Apply various design pattern strategies to design an appropriate solution for the problem (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	S	M	S						L		S	M	M
CO2	S	M	S	M	S						L		S	M	M
CO3	S	M	S	M	S						L		S	M	M
CO4	S	M	S	M	S				M		L		S	M	M
CO5	S	M	S	M	S				M		L		S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Design Pattern, Architectural Design Patterns, Describing Design Patterns, The Catalog of Design patterns, selecting a Design Pattern, Using Design Patterns for Design problems

Architectural Patterns -- Layered architecture, Pipers and Filters, Blackboard, Broker, MVC, MVVM, Micro-Kernel, Master-Slave, PAC, others.

Creational Patterns: Singleton, Abstract Factory, Builder, Factory Method, Prototype, Implementation in various languages like Python, Java.

Structural Pattern Part – I: Adaptor, Bridge, and Composite.

Structural Pattern Part – II: Decorator, Arcade, Flyweight, Proxy.

Behaviour Patterns Part – I: Chain of Responsibility, Command, Interpreter, and Iterator.

Behaviour Patterns Part – II: Mediator, Memento, Observer.

Behaviour Patterns Part – III State, Strategy, Template Method, Visitor.

Compound patterns, Case study: using design patterns to solve an industry level problem.

Reading List:

1. Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates “Head First Design Patterns”, O'Reilly, 2020.
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides “Design Patterns: Elements of Reusable Objectoriented Software” Addison-Wesley.
3. Mark Grand, “Patterns in JAVA Vol-I”, Wiley Dream Tech.
4. Mark Grand, “Patterns in JAVA Vol – II”, Wiley Dream Tech.
5. Mark Grand, “JAVA Enterprise Design Patterns Vol – III”, Wiley Dream Tech.
6. Rahul Verma, Chetan Giridhar, “Design patterns in Python”,Testing Perspective, 2011.
7. Martin Fowler, with Dave Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and Randy Stafford, “Patterns of Enterprise Application Architecture”, 1st Edition, Addison-Wesley

CS384	Internet of Things and Edge Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Computer Networks (CS352)
ii. Introduction to algorithmic thinking and programming (CS101)

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze the protocol Stack for Internet of Things to address the heterogeneity in devices and networks. (Analyze)
CO2	Construct smart IoT Applications using smart sensor devices and cloud systems. (Apply)
CO3	Construct smart mobile apps for societal applications. (Apply)
CO4	Implement secure protocols for IoT systems. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	S
CO2	S	M	L										S	S	M
CO3	S	M	L										S	S	M
CO4	S	M	L										S	S	M

Detailed syllabus:

Introduction to IoT - Sensing, Actuation, Basics of Networking - Communication Protocols Sensor Networks - Machine-to-Machine Communications - Interoperability in IoT, Introduction to Arduino Programming - Integration of Sensors and Actuators with Arduino -Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi - Introduction to SDN, SDN for IoT - Data Handling and Analytics - Security of IoT; Cloud Computing - Cloud databases and services - Introduction to Edge Computing - Edge Architectures and Deployment Modes – Edge Computing Applications.

Reading List:

1. Arshdeeep Bahga, Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press, 2015

2. Raj Kamal, Internet of Things: Architecture and Design Principles, McGraw Hill Education private limited, 2017
3. Kai Hwang, Min Chen, Big Data Analytics for Cloud, IoT and Cognitive Computing, Wiley, 2018.

CS411	Randomized Algorithms	DEC	3 – 0 – 0	3 Credits

This course introduces the need for randomized algorithms and the different design paradigms for constructing randomized algorithms. The Course will also emphasize on analyzing the expected time complexity and the nature and bounds on the error probabilities of the randomized algorithms.

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)

ii. Discrete Mathematics (CS203)

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

CO1	Construct and Analyze algorithms using a probabilistic framework. (Analyze)
CO2	Assess the computational hardness of a problem with a comprehension of the randomized complexity classes. (Analyze)
CO3	Construct Las Vegas algorithms for a given problem and compute the expected running time. (Analyze)
CO4	Construct Monte-Carlo algorithms for a given problem and compute the probability of getting an incorrect output. (Analyze)
CO5	Design and analyze solutions for complex problems using randomization design paradigms like Foiling the Adversary, Abundance of Witnesses, Fingerprinting, Random Sampling, Amplification and Random Rounding. (Analyze)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	S	M										S	M	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to Randomized Algorithms: Review on Algebra, Number theory, Combinatorics and Probability theory, Probabilistic analysis of Algorithms, Randomized Complexity classes and hardness of problems, Randomness as a source of efficiency-designing a communication protocol, Models of Randomized Algorithms, Classification-Las Vegas, Monte-Carlo (one-sided error, bounded-error and unbounded-error), Classification of Randomized Algorithms for Optimization problems.

Design Paradigms: Foiling the Adversary, Abundance of Witnesses, Fingerprinting, Random Sampling, Amplification, Random Rounding.

Representative Algorithms: Foiling the Adversary – Universal Hashing, Fingerprinting Communication protocols, Verification of Matrix Multiplication, Equivalence of Two polynomials, Success Amplification and Random Sampling – Min-Cut, Satisfiability and repeated random sampling, Abundance of Witnesses and Optimization & random rounding – Primality Testing, Max-SAT review, hybrid sampling & rounding, Derandomization Techniques.

Reading List:

1. Juraj Hromkovic– Design and Analysis of Randomized Algorithms, First edition, Springer, 2005.
2. Randomized Algorithms – Rajeev Motwani, Prabhakar Raghavan, Cambridge University Press, 1995.
3. Introduction to Algorithms –Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest and Clifford Stein, third edition, PHI, 2010.

CS412	Performance Modeling of Computer systems	DEC	3 – 0 – 0	3 Credits

This course introduces the techniques of analytical modeling such as Markov chains, and queuing models, experimental designs, and discrete event simulations for modeling a computing system or its component and to analyze its performance with respect to well-defined metrics.

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)

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

CO1	Formulate performance models for a given computer and a communication system by applying modeling techniques like Markov Chains, Queuing theory and Queue networks. (Apply)
CO2	Design experiments to measure the performance of a computer system with an understanding of the appropriate performance metrics to be used. (Apply)
CO3	Model, characterize and reproduce workloads to a computer system. (Apply)
CO4	Construct discrete event simulation models for a computer system and analyze the simulation results. (Analyze)
CO5	Analyze, present, and interpret the experimental results to evaluate alternative system implementations. (Analyze)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M										S	M	
CO2	S	M	M										S	M	
CO3	S	M	M										S	M	
CO4	S	S	M										S	M	
CO5	S	S	M										S	M	

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Overview of Performance Modeling: introduction, Avoiding common mistakes, Selection of techniques and metrics.

Modeling Techniques: Random variables and probability distributions, Stochastic processes, Markov Chain, Continuous time Markov Chain, Queuing Theory, Queuing Networks.

Workloads: Types, Selection, Characterization Techniques-Averaging, Histograms, Clustering, Principal Component Analysis, Markov Models.

Data Presentation: Guidelines, Common mistakes, Pictorial Games, Gantt Charts, Kiviat Graphs-Shapes & Applications, Schumacher Charts.

Experimental Design and Analysis: 2^k factorial designs, $2^k r$ Factorial Designs with replications, $2^k p$ Fractional Factorial designs.

Simulation: Analysis of simulation results – Model Verification Techniques, Validation, Stopping Criteria and Variance reduction, Discrete Event Simulations-Programming aspects.

Queuing Models: Queuing Theory, Analysis of a single queue, M/M/1 queue, M/M/m queue, M/M/m/B queue, Queuing Network models of computer systems, Operational laws, Analysis of queuing systems.

Reading List:

1. Raj Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", Wiley, 2015.
2. Mor Harchol-Balter, Performance Modeling and Design of Computer Systems: Queueing Theory in Action, Cambridge University Press, 2013.

CS413	Foundations of Data Science	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will learn how to solve standard distributed computing problems. Students will also learn consensus algorithms. Further students shall also learn designing widely used algorithms for map-reduce/spark paradigms of distributed computing.

Pre-requisites:

i. Artificial Intelligence (CS303)

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

CO1	Construction, analysis and evaluation of learning objectives for Regression (Analyze)
CO2	Analyze efficiency large scale matrix factorizations for different data science scenarios (Analyze)
CO3	Apply basic machine learning algorithms (Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes) for predictive modeling (Apply)
CO4	Designing Recommender System for different applications. (Analyze)
CO5	Designing feature reduction objectives using PCA (Apply)
CO6	Analyze various clustering objectives K-Means-EM-SpectralClustering (Analyze)

Course Articulation Matrix:

CO \ PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Basics of Optimization in Data Science: Linear Programming Objectives, Primal-Dual Methods, Quadratic Programming, Convex Optimization, Gradient Descent, Adaptive Learning Rate, Quasi Newton's method, Constrained Optimization, KKT Conditions for Optimization Problems

Probability theory including random variables, conditional probability, Bayes law, concentration of measure, linear algebra including eigenvalues, norms, elementary spectral graph theory.

Regression: Least Squares Objective for Multiple Linear Regression, Gauss-Markov Theorem, Statistical Tests, Weighted Least Squares, Box-Cox Transformation, Polynomial & Spline Regression, Ridge Regression, Bias-Variance Tradeoff, Subset Selection, LASSO, Adaptive LASSO, Elastic Net, Dantzig Selector, SLOPE and sorted penalties.

Matrix Factorizations used in Data Science:

Cholesky Decomposition and QR Factorization: Concepts, Applications

Eigen Vector Decomposition, Solving Large Scale Value Problems using Lanczos Method, Arnoldi's Iteration

SVD: Geometric Interpretation, Best Rank-K Approximation using SVD, Power Method for SVD, Efficient methods for SVD, Applications of SVD

PCA: PCA learning objective: Application of PCA for dimensionality reduction

Recommendation Systems: Collaborative Filtering using gradient Descent and Alternating Least Squares for recommender systems.

Classification: GLM methods for classification, SVM, Naïve Bayes, Evaluation of classification methods – Confusion matrix, Students T-tests and ROC curves, Feature Selection for Classification

Clustering: Choosing distance metrics – Different clustering approaches – hierarchical agglomerative clustering, k-means (Lloyd's algorithm), EM Algorithm for clustering, Spectral Clustering: Graph Laplacian, Properties of Graph Laplacian, Application of Spectral Clustering for Transfer-learning.

Link Analysis: Random Walks, Markov Chains, Stationary Distribution, Metro Polis Hastings Algorithm, Gibbs Sampling, Convergence of Random Walks on Undirected Graphs, Page Rank, HITS, combating spam, personalized page rank

High Dimensional Space High dimensional sphere, volumes of high dimensional solids, gaussians in high dimension, high dimensional point sets, Johnson-Lindenstrauss theorem.

Reading List:

1. Avrim Blum, John Hopcroft, and Ravindran Kannan: Foundations of Data Science Cambridge University Press, 2020
2. Jianqing Fan, Runze Li, Cun-Hui Zhang, Hui Zou Statistical Foundations of Data Science, CRC Press 2020
3. Jure Leskovec, Anand Rajaraman, Jeff Ullman: Mining of Massive Datasets, Cambridge University Press, 2016
4. G. Strang. Introduction to Linear Algebra, Wellesley-Cambridge Press, Fifth edition, USA, 2016
5. Relevant Research articles shared by the instructor

CS421	Distributed Computing	DEC	3 – 0 – 0	3 Credits

Students of this course upon completion will learn how to solve standard distributed computing problems. Students will also learn consensus algorithms. Further algorithms for global knowledge, recording distributed snapshots, distributed mutual exclusion with an emphasis to reduce communication cost are also studied.

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)

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

CO1	Construct efficient distributed algorithms for standard graph-theoretic problems with an analysis of their time, space and communication requirements (Apply)
CO2	Constructing efficient distributed algorithms for achieving stable properties (Apply)
CO3	Analyze the properties of various algorithms for fault tolerance/ consenses (Analyze)
CO4	Construct and analyze algorithms for solving distributed mutual exclusion. (Apply)
CO5	Analyze the properties of distributed transaction processing/ concurrency algorithms (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Taxonomy of distributed systems, Failure Detection, Clocks and Time, Logical Time, Distributed Consistency Models,

Leader election in rings. Basic computational tasks in general synchronous networks: leader election. Breadth-first search. Shortest Path Algorithms, Floyd-Warshal's Algorithm, Minimal Spanning Trees, Maximal Independent Sets, Stable property detection. Distributed termination. MutiCast Models, Distributed snapshot recording. Deadlock detection. Asynchronous shared-memory systems. The mutual exclusion problem, Distributed Mutual exclusion algorithms.

Fault-tolerant consensus, Link failures: the two generals' problem. Process failures (stopping, Byzantine). Algorithms for agreement with stopping and Byzantine failures. PBFT, Paxos, zookeeper, RAFT, A case study of Bitcoin consensus

Distributed Transaction Processing and Concurrency Control, 2PC, 3PC vs View Stamped Replication, 2PL, Distributed Deadlocks

Distributed Hash Tables, Key-Value Stores (Cassandra)

Reading List:

1. A. Kshemkalyani and M. Singhal, Distributed Computing:Principles, Algorithms, and Systems Cambridge University Press 2008 ISBN: 978-0-521-87634-6
2. Lynch, Nancy. Distributed Algorithms. Burlington, MA: Morgan Kaufmann, 1996. ISBN:9781558603486.
3. Michel Raynal Distributed Algorithms for Message-Passing Systems ISBN 978-3-642-38122-5 © Springer-Verlag Berlin Heidelberg 2013
4. A Guided Tour on the Theory and Practice of State Machine Replication, Alysson Neves Bessani and Eduardo Alchieri Book Chapter
5. Relevant Research articles shared by the instructor.

CS422	Reinforcement Learning	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)
ii. Artificial Intelligence (CS303)

Course Outcomes: At the end of the course, the students will be able to:

CO1	Formulate a given problem as a reinforcement learning problem (Apply)
CO2	Evaluate feedback of RL in a non-associative setting (Apply)
CO3	Apply basic RL algorithms for simple sequential decision-making problems in uncertain conditions (Apply)
CO4	Construct RL systems for Planning through the use of Dynamic Programming and Monte Carlo (Apply)
CO5	Construct RL agents using TD learning and Function approximation. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO2
CO1	S	S	S	M	M	L		L					S	M	M
CO2	S	S	S	M	M	L		L					S	M	M
CO3	S	S	S	M	M	L		L					S	M	M
CO4	S	S	S	M	M	L		L					S	M	M
CO5	S	S	S	M	M	L		L					S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction and Scope. Elements of Reinforcement Learning, RL comparison to other Machine Learning approaches, Examples of modeling through RL. Review of Probability.

Multi-armed Bandits. A k -armed Bandit Problem, Action-value Methods, the 10-armed Testbed, Incremental Implementation, Tracking a Nonstationary Problem, Optimistic Initial Values, Upper-Confidence-Bound Action Selection, Gradient Bandit Algorithms, Associative Search (Contextual Bandits).

Markov Decision Process. The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Unified Notation for Episodic and Continuing Tasks, Policies and Value Functions, Optimal Policies and Optimal Value Functions, Optimality and Approximation.

Prediction and Control by Dynamic Programing. Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration, Efficiency of Dynamic Programming.

Monte Carlo Methods. Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-policy Monte Carlo Control.

Temporal-Difference Learning. TD Prediction, Advantages of TD Prediction Methods, Optimality of TD (0), Sarsa: On-policy TD control, Q-Learning and their variants. Expected Sarsa, Maximization Bias, and Double Learning.

Function Approximation Methods. Value-function approximation, The prediction objective, Stochastic-gradient and semi-gradient methods, Linear methods, Nonlinear function approximation: ANN, Least-squares TD, Memory-based and kernel-based function approximation.

On-policy Control with Approximation. Episodic Semi-gradient Control, Semi-gradient n-step Sarsa, Average Reward: A New Problem Setting for Continuing Tasks, Deprecating the Discounted Setting, Differential Semi-gradient n-step Sarsa.

Introduction to Policy Gradients.

Reading List:

1. Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", 2nd Edition, MIT Press, 2020, http://incompleteideas.net/book/RLbook2020.pdf
2. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012, https://probml.github.io/pml-book/book0.html

CS423	Soft computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Artificial Intelligence (CS303)
ii. Artificial Intelligence Lab (CS304)

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

CO1	Develop a model for the system using fuzzy modeling and analyze few paradigms for making decisions within a fuzzy environment. (Analyze)
CO2	Identifying underlying structure in data using Fuzzy Classification and Fuzzy Pattern Recognition (Apply)
CO3	Apply the mechanics of the binary and continuous Genetic Algorithm to optimization problems. (Apply)
CO4	Apply Natural Optimization like simulated annealing, ant colony optimization, and evolutionary strategies to optimization problems. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S										S	M	S
CO2	S	S	S										S	M	S
CO3	S	S	S										S	M	S
CO4	S	S	S										S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Fuzzy systems: Introduction

Automated Methods for Fuzzy Systems:

Definitions, Batch Least Squares Algorithm, Recursive Least Squares Algorithm, Gradient Method, Clustering Method, Learning from Examples

Decision Making with Fuzzy Information: Fuzzy Synthetic Evaluation, Fuzzy Ordering, Nontransitive Ranking, Preference and Consensus, Multi-objective Decision Making, Fuzzy Bayesian Decision Method, Decision Making Under Fuzzy States and Fuzzy Actions

Fuzzy Classification: Classification by Equivalence Relations, Cluster Analysis, Cluster Validity, c-Means Clustering, Hard c-Means, Fuzzy c-Means, Classification Metric, Hardening the Fuzzy c-Partition, Similarity Relations from Clustering

Fuzzy Pattern Recognition: Feature Analysis, Partitions of the Feature Space, Single-Sample Identification, Multi-feature Pattern Recognition, Image Processing

Introduction to Optimization: Finding the Best Solution, Minimum-Seeking Algorithms, Natural Optimization Methods, Biological Optimization: Natural Selection, The Genetic Algorithm, Binary Genetic Algorithm, Continuous Genetic Algorithm

Natural Optimization Algorithms: Simulated Annealing, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Programming (GP), Cultural Algorithms, Evolutionary Strategies

Reading List:

1. Timothy J. Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, Willey, 2010.
2. S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Pearson Education, 2009.
3. Practical Genetic Algorithms, Randy L. Haupt and sue Ellen Haupt, John Willey & Sons, 2002.

CS424	Probabilistic Graphical Models	DEC	3 – 0 – 0	3 Credits

Probabilistic graphical models (PGMs) are helpful in characterizing complex relationships among many random variates by fusing graph theory and probability theory and providing a sound framework. PGMs are helpful in reasoning under uncertainty especially in application domains like Natural Language Processing, Computer Vision, and Computational Biology.

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)

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

CO1	Construct probabilistic graphical models to characterize uncertainty inherent in problem domains such as Natural Language Processing and Computer Vision. (Apply)
CO2	Construct fundamental algorithms for probabilistic inference in both directed (Bayesian) and Undirected (Markovian) probabilistic graphical models. (Apply)
CO3	Construct algorithms for learning the structure and parameters of probabilistic graphical models. (Apply)
CO4	Apply Bayesian principles to model domain knowledge under uncertainty. (Apply)
CO5	Construct algorithms to perform inference in statistical and causal models. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L									S	M	S
CO2	S	M	M	L									S	M	S
CO3	S	M	M	L									S	M	S
CO4	S	M	M	L									S	M	S
CO5	S	M	M	L									S	M	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction and Motivation: Probabilistic Graphical Models, Representation, Inference and Learning.

Representation:

Bayesian Networks: independence properties, D-separation Algorithm, I-equivalence, Minimal I-maps, Perfect maps.

Undirected graphical models: Parameterization, Gibbs distribution and Markov networks, Independencies in Markov networks, Conversion to/from Bayesian Networks to Markov Networks, partially directed models, Conditional Random Fields.

Inference:

Exact inference-variable elimination, Clique Trees, Message passing, Belief update, Clique tree construction.

Inference as Optimization: propagation-based approximation.

Particle-based approximate inference: forward sampling, likelihood weighting and importance sampling, Markov Chain Monte Carlo methods, Gibbs Sampling algorithm.

MAP Inference, Inference in hybrid networks and temporal models.

Learning:

Learning graphical models: goals, learning as optimization

Parameter estimation: Maximum Likelihood Estimation (MLE), MLE for Bayesian networks.

Structure Learning in Bayesian Networks: constraint-based approaches, structure search, scoring structures,

Reading List:

1. Daphne Koller and Nir Friedman, Probabilistic Graphical Models, First Edition, MIT Press 2009.
2. Chris Bishop, Pattern Recognition and Machine Learning, First Edition, Springer 2006.

CS425	Deep Learning for Vision	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Applied Machine Learning (CS373)

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

CO1	Formulate basic knowledge, theories and methods in image processing and computer vision (Apply)
CO2	Apply appropriate image processing methods for image filtering, image restoration, image reconstruction, segmentation, classification and representation (Apply)
CO3	Assess Recurrent Neural Networks working for solving problems in image processing. (Evaluate)
CO4	Analyze existing practical Attention models (Analyze)
CO5	Develop Deep Generative Models solve real-world problems (Create)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	L	L
CO2	S	M	M	L									S	M	L
CO3	S	S	M	L									S	M	L
CO4	S	S	M	L									S	M	L
CO5	S	S	S	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction and Overview:

Introduction to Image Formation, Capture and Representation; Linear Filtering, Correlation, Convolution, Visual Features and Representations: Edge, Blobs, Corner Detection, Scale Space and Scale Selection; SIFT, SURF, HoG, LBP, etc. Visual Matching: Bag-of-words, VLAD. (Introductory level)

Deep Learning:

Introduction of Deep Learning, Multi-layer Perceptron, Backpropagation

Convolutional Neural Networks (CNNs):

Introduction to CNNs; Evolution of CNN Architectures: Alex Net, ZF Net, VGG, Inception Net, Res Net, Dense Net.

Visualization and Understanding CNNs:

Visualization of Kernels; Backprop-to-image/Deconvolution Methods; Deep Dream, Hallucination, Neural Style Transfer; CAM, Grad-CAM, Grad-CAM++; CNNs for Recognition and Verification: Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss; CNNs for Detection: Background of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, Retina Net; CNNs for Segmentation: FCN, Seg Net, U-Net, Mask-RCNN

Recurrent Neural Networks (RNNs):

Introduction to RNNs; CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition

Attention Models:

Introduction to Attention Models in Vision; Vision and Language: Image Captioning, Visual QA, Visual Dialog; Spatial Transformers; Transformer Networks

Deep Generative Models:

Popular Deep Generative Models: GANs, VAEs; Other Generative Models: Pixel RNNs, NADE, Normalizing Flows.

Reading List:

1. Goodfellow I, Bengio Y, and Courville A, Deep Learning, MIT Press, 2016
2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.
3. Michael Nielsen, Neural Networks and Deep Learning, 2016
4. Simon Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012.

CS426	Advanced Database Systems	DEC	3 – 0 – 0	3 Credits

Database Technology has evolved over past four decades bringing immense value among practitioners. The objective of this course is familiarizing the students with internal details of Database System implementation to make them realize the importance of design choices in the evolution of the architectures of DBMS. Further this course aims at understanding storage formats suitable for big-data/In-Memory database platforms for hybrid query workloads.

Pre-requisites:

i. Database Management Systems (CS255)
ii. Operating Systems (CS202)

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

CO1	Analyse the tradeoffs in the design of Database Management Systems (Apply)
CO2	Designing efficient indexing schemes for In-Memory DBMS(Apply)
CO3	Analysing storage models for varying data processing workloads (Apply)
CO4	Designing Efficient Query Optimization strategies for large-scale parallel joins (Apply)
CO5	Analysing Multi-Version Concurrency in large-scale in-memory database systems (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M										S	M	S
CO2	S	M	M										S	M	S
CO3	S	M	M										S	M	S
CO4	S	M	M										S	M	S
CO5	S	M	M										S	M	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction and History of Databases,
In-Memory Databases: Concurrency in large parallel databases, Multi-Version Concurrency Control: Design Tradeoffs, Garbage Collection, Design of Protocols, Scheduling
Indexing Structures in DBMS: Concurrency in B+ Tree, Bw-Tree, Radix Trees, Hash Tables, Trie, Multi-Dimensional Indexing.

Storage Models: Data Layout, Optimal Column layout for hybrid workloads, Row-Storage vs Columnar Storage, Compression, Integration of compression and execution in columnar dbms.

Recovery: ARIES, Constant Time Recovery, Write-behind Logging

Optimization: Access path selection in Main-Memory Optimized Data Systems, Materialization Strategies, Vectorization for large In-Memory DBMS, Compiled Vs Vectorized Queries, Equi-Join Strategies in large scale databases (In-Memory), Sort vs Hash Join in Multi-Core DBMS, Query Optimization in traditional DBMS, Harnessing best of many plans, Top-Down vs Bottom-Up

Reading List:

1. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke McGraw Hill Publishers 3rd Edition
2. Daphne Koller and Nir Friedman, Probabilistic Graphical Models, First Edition, MIT Press 2009.
3. Chris Bishop, Pattern Recognition and Machine Learning, First Edition, Springer 2006.
4. M. Stonebraker, et al., What Goes Around Comes Around, in Readings in Database Systems, 4th Edition, 2006
5. A. Pavlo, et al., What's New with NewSQL?, in SIGMOD Record (vol. 45, iss. 2), 2016
6. X. Yu, et al., Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores, in VLDB, 2014
7. Y. Wu, et al., An Empirical Evaluation of In-Memory Multi-Version Concurrency Control, in VLDB, 2017
8. T. Neumann, et al., Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems, in SIGMOD, 2015
9. J. Böttcher, et al., Scalable Garbage Collection for In-Memory MVCC Systems, in VLDB, 2019
10. Z. Wang, et al., Building A Bw-Tree Takes More Than Just Buzz Words, in SIGMOD, 2018
11. V. Alvarez, et al., A Comparison of Adaptive Radix Trees and Hash Tables, in ICDE, 2015
12. R. Binna, et al., HOT: A Height Optimized Trie Index for Main-Memory Database Systems, in SIGMOD, 2018
13. M. Athanassoulis, et al., Optimal Column Layout for Hybrid Workloads, in VLDB, 2019
14. D. Abadi, et al., Integrating Compression and Execution in Column-Oriented Database Systems, in SIGMOD, 2006
15. P. Antonopoulos, et al., Constant Time Recovery in Azure SQL Database, in VLDB, 2019

16. V. Leis, et al., Morsel-Driven Parallelism: A NUMA-Aware Query Evaluation Framework for the Many-Core Age, in SIGMOD, 2014

17. M. Kester, et al., Access Path Selection in Main-Memory Optimized Data Systems: Should I Scan or Should I Probe?, in SIGMOD, 2017

18. T. Neumann, Efficiently Compiling Efficient Query Plans for Modern Hardware, in VLDB, 2011

19. O. Polychroniou, et al., Rethinking SIMD Vectorization for In-Memory Databases, in SIGMOD, 2015

20. T. Kersten, et al., Everything You Always Wanted to Know About Compiled and Vectorized Queries But Were Afraid to Ask, in VLDB, 2018

21. S. Schuh, et al., An Experimental Comparison of Thirteen Relational Equi-Joins in Main Memory, in SIGMOD, 2016

22. S. Chaudhuri, An Overview of Query Optimization in Relational Systems, in PODS, 1998

23. E. Begoli, et al., Apache Calcite: A Foundational Framework for Optimized Query Processing Over Heterogeneous Data Sources

24. V. Leis, et al., How Good are Query Optimizers, Really?, in VLDB, 2015

25. Ding, et al., Plan Stitch: Harnessing the Best of Many Plans, in VLDB, 2018

26. J. Arulraj, et al., Write-Behind Logging, in VLDB, 2016

CS431	Wireless Technologies	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Computer Networks (CS352)

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

CO1	Differentiate Wired and Wireless Technologies in selection of suitable technology for given requirements. (Analyze)
CO2	Implement suitable Wireless Local loop Technology. (Apply)
CO3	Apply suitable techniques in the selection of CDMA or TDMA. (Apply)
CO4	Apply suitable techniques in selection of suitable satellite parameters and configurations. (Apply)
CO5	Apply suitable IEEE standard in selection of Wireless LAN technology. (Apply)
CO6	Analyze different IEEE Medium Access Control protocols and choose a suitable one for given requirements. (Analyze)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M		S				M				S	S	M
CO2	S	M	M		S				M				S	S	M
CO3	S	M	M		S				M				S	S	M
CO4	S	M	M		S				M				S	S	M
CO5	S	S	S	M	S				M				S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Technical Review- Fundamentals: Transmission Fundamentals, Signals for Conveying Information, Analog and Digital Data Transmission, Channel Capacity, Transmission Media.

Multiplexing Communication Networks: LANs, MANs, and WANs, Switching Techniques, Circuit Switching, Packet Switching.

Asynchronous Transfer Mode Protocols and the TCP/IP Suit: The Need for a Protocol Architecture, The TCP/IP Protocol Architecture, The OSI Model, Internetworking.

Wireless Networking- Satellite Communications: Satellite Parameters and Configurations.

Capacity Allocation: Frequency Division, Time Division.

Cellular Wireless Networks: Principles of Cellular Networks, First-Generation Analog, Second-Generation TDMA, Second-Generation CDMA, Third-Generation Systems.

Cordless Systems and Wireless Local Loop: Cordless Systems, Wireless Local Loop, WiMAX and IEEE 802.16 Broadband Wireless Access Standards.

Mobile IP and Wireless Access Protocol: Mobile IP, Wireless Application Protocol

Wireless LANs- Wireless Lan Technology: Overview, Infrared LANs, Spread Spectrum LANs, Narrowband Microwave LANs Wi-Fi and the IEEE 802.11

Wireless Lan Standard: IEEE 802 Protocol Architecture, IEEE 802.11 Architecture and Services, IEEE 802.11 Medium Access Control, IEEE 802.11 Physical Layer, Other IEEE 802.11 Standards.

Bluetooth and IEEE 802.15: Overview, Radio Specification, Baseband Specification, Link Manager Protocol, Logical Link Control and Adaptation Protocol, IEEE 802.15

Reading List:

1. W. Stallings, "Wireless Communications and Networks", Second Edition,Pearson Education, 2002.
2. T S Rappaport, "Wireless Communications: Principles & Practice", Second Edition, Pearson Education, 2002.
3. J Schiller, "Mobile Communications", Second Edition,Addison Wesley, 2000.
4. V K Garg, "IS-95 CDMA and CDMA2000", First Edition,Prentice Hall PTR, 2007.

CS432	Service-Oriented Architecture	DEC	3-0-0	3 credits

Pre-requisites:

i. Object Oriented Programming (CS251)

Course Outcomes: At the end of the course the student will be able to:

CO1	Design software-oriented architectures. (Apply)
CO2	Design medium scale software project development using SOA principles. (Apply)
CO3	Develop SOA messages from business use cases. (Apply)
CO4	Design and implementation of modern SOA and SOA-specific methodologies, technologies and standards. (Apply)
CO5	Create composite services by applying composition style. (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	M	L	S	M	L							S	S	
CO2	L	M	L	S	M	L							S	S	
CO3	L	M	L	M	M	L							S	S	
CO4	L	M	L	L	M	L							S	S	
CO5	L	M	L	S	M	L							S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Introducing SOA: Fundamental SOA, Common characteristics of contemporary SOA, common misperceptions about SOA, common tangible benefits of SOA, common pitfalls of adopting SOA

The evolution of SOA: An SOA timeline (from XML to Web services to SOA), The continuing evolution of SOA (Standards organizations and Contributing vendors), The roots of SOA (comparing SOA to Past architectures).

Web Services and Primitive SOA: The Web services framework, Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP).

Web Services and Contemporary SOA (Part-I: Activity Management and Composition) – Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. Web Services and Contemporary SOA

(Part-II: advanced messaging, Metadata and Security): Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing.

Principles of Service - Orientation: Services orientation and the enterprise; Anatomy of a service-oriented architecture; Common Principles of Service orientation; how service orientation principles interrelate; Service orientation and object orientation; Native Web service support for service orientation principles.

Service Layers: Service orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios.

An overview of JEE Web Services: The JEE platform, The Technologies of Web Services, The JEE Web Service APIs.

XML Basics Primer Namespaces, Wrapping Up, The W3C XML Schema Language:XML Schema Basics, Advanced XML Schema.

XML based RPC, JSON, UDDI

SOAP and WSDL: The Basic Structure of SOAP, SOAP Namespaces, SOAP Headers, SOAP Body, SOAP Message Models, SOAP Faults, SOAP over HTTP.

Reading List:

1. Thomas Erl, “Service-Oriented Architecture: Concepts, Technology and Design”, Prentice Hall Publication, 2005.
2. Richard Monson, Haefel, “J2EE Web Services”, Pearson Education,2004
3. The Java EE 6 Tutorial: https://docs.oracle.com/javaee/6/tutorial/doc/gijti.html

CS433	Cloud Computing	DEC	3-0-0	3 Credits

Pre-requisites:

i. Computer Networks (CS352)
ii. Operating Systems (CS202)

Course Outcomes: At the end of this course, students will be able to:

CO1	Illustrate the basic principles of Cloud Computing, technologies, architectures and implementation. (Apply)
CO2	Construct solutions for complex engineering problems using Cloud with a comprehension of the principles of cloud virtualization, cloud storage, data management and data visualization. (Apply)
CO3	Apply different cloud programming platforms and tools to develop and deploy applications on cloud (Apply)
CO4	Construct suitable service and resource management and provisioning schemes for cloud with a comprehension of the techniques for task scheduling, load balancing and service tolerance. (Apply)
CO5	Construct cloud-based applications with an objective to minimize and control authentication, confidentiality and privacy issues. (Apply)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	M	S								S	S	M
CO2	S	M	L	M	S								S	S	M
CO3	S	M	M	S	S								S	S	M
CO4	S	M	L	L	S								S	S	M
CO5	S	M	M	L	S								S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction: Overview of Computing paradigm: Grid computing, Cluster computing, Distributed computing, Utility computing, P2P computing and so forth, Cloud Fundamentals: Cloud definition, Evolution, System models for cloud computing: Benefits, challenges and Risks in cloud computing, Introduction to Infrastructure as a Service (IaaS), Platform as a service (PaaS), Software as a service (SaaS): From IAAS to PaaS.

Cloud Infrastructure/Architecture:

Study of Cloud computing Systems like Amazon EC2 and S3, Google App Engine, and Microsoft Azure, Build Private/Hybrid Cloud using open-source tools, Deployment of Web Services from Inside and Outside a Cloud Architecture. MapReduce and its extensions to Cloud Computing, HDFS, and GFS

Cloud Resource Virtualization:

Introduction to virtualization: Different approaches to virtualization Hypervisors, Machine Image Virtual Machine (VM), Process VM vs System VM, Resource virtualization: Server, Storage, Network Full Virtualization Vs Para Virtualization Operating System, Operating System support for Virtualization, Virtual Machine (Resource), Provisioning and Management, VM Placement, VM migration.

Cloud Resource Management and Scheduling:

Policies and mechanism for resource management, Application of control theory to cloud resource allocation, Stability of a two-level resource allocation architecture, Proportional thresholding, A utility-based model for cloud-based web-based services, Scheduling algorithms for computing clouds: Fair queuing, Start Time fair queuing, borrowed virtual time, cloud scheduling subject to deadlines scheduling, Map Reduce Application: Subject to deadlines

Storage System:

Storage models, File Systems and databases, Distributed file systems, General Parallel file system, Google file system, Apache Hadoop, Transaction Processing and NOSQL Databases, Bigtable, Megastore

Cloud Security:

Vulnerability Issues and Security Threats, Application-level Security, Data level Security, and Virtual Machine level Security, Infrastructure Security, and Multi-tenancy Issues. IDS: host-based and network-based, Security-as-a-Service. Trust Management, Identity Management, and Access Controls Techniques

Reading List:

1. Marinescu, Dan C. “Cloud computing: theory and practice”. Morgan Kaufmann Elsevier, 2022.
2. Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. “Mastering cloud computing: foundations and applications programming”, McGraw Hill, 2013.
3. Sosinsky Barrie. “Cloud computing bible”, John Wiley & Sons, 2010 Dec 10.
4. Miller, Michael. “Cloud computing: Web-based applications that change the way you work and collaborate online”, Que publishing, Pearson Education, 2008.
5. Mather, T., Kumaraswamy, S., & Latif, S. (2009). “Cloud security and privacy: an enterprise perspective on risks and compliance”, O'Reilly Media, Inc., 2009.

CS434	Blockchains	DEC	2 – 0 – 2	3 Credits

Pre-requisites:

None

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

CO1	Familiarize the functional/operational aspects of cryptocurrency eco-system. (Apply)
CO2	Design Smart Contracts for different application scenarios (Apply)
CO3	Analyze different consensus algorithms used in the design of blockchains (Apply)
CO4	Design Blockchain solutions to address Privacy and Anonymity in transaction networks (Apply)
CO5	Implement smart-contract based solutions for different application scenarios using Ethereum (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Cryptographic basics for cryptocurrency - a short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography

Bitcoin - Wallet - Blocks - Merkley Tree - hardness of mining - transaction verifiability - anonymity - forks - double spending - mathematical analysis of properties of Bitcoin.

Ethereum - Ethereum Virtual Machine (EVM) - Wallets for Ethereum - Solidity - Smart Contracts - some attacks on smart contracts, Attacks on Ethereum smart contracts, Merkle Trie data structure for Ethereum, Mining in ethereum

Nakamoto Consensus on permission-less, nameless, peer-to-peer network, Selfish Mining, Known attacks on Bitcoin

Consensus in Distributed Systems: The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis, PBFT, paxos, 2PC, 3PC protocols, RAFT, Honeybadger

Abstract Models for BLOCKCHAIN - GARAY model - RLA Model - Proof of Work (PoW) as random oracle - formal treatment of consistency, liveness and fairness - Proof of Stake (PoS) based Chains - Hybrid models (PoW + PoS) .

Anonimity and Privacy in Block Chains - Zero Knowledge proofs and protocols in Blockchain - Succinct non interactive argument for Knowledge (SNARK) - pairing on Elliptic curves - Zcash.

List of Experiments:

1. Establishing bitcoin network and demonstration of smart contracts
2. Implementation of attacks on bitcoin network
3. Establishing Ethereum network
4. Implementing escrow application in Ethereum
5. Attacks on smart-contracts in ethereum
6. Project on real-world application of smart-contracts using Ethereum
7. Implementing anonymity in blockchain applications

Reading List:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016. (Free download available)
2. Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015 (article available for free download)
3. Roger Wattenhofer: The Science of the Blockchain Cengage publishers, 2016
4. J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOl 9057, (VOLII), pp 281-310. (Also available at eprint.iacr.org/2016/1048)
5. R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks , EUROCRYPT 2017, (eprint.iacr.org/2016/454)
6. R Pass et al, Fruitchain, a fair blockchain, PODC 2017 (eprint.iacr.org/2016/916)

CS435	Network Security	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Computer Networks (CS352)

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

CO1	Assess the suitable cryptographic techniques and number theory concepts to design network security protocols. (Analyze)
CO2	Constrcut secure protocols using cryptographic algorithms. (Apply)
CO3	Assess system vulnerabilities of communication protocols. (Analyze)
CO4	Analyze various types of security systems and applications. (Analyze)
CO5	Develope robust applications and analyze its security. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	S
CO2	S	M	L					L					S	S	M
CO3	S	M	L										S	S	M
CO4	S	S	S	M				M					S	S	M
CO5	S	M	L	M				M					S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Security mechanisms: Key management and distribution – Public key infrastructure; Authentication protocols.

Introduction to network security: Denial-of- service/Distributed denial-of-service attacks; Spoofing, Man-in-the-middle, Replay, TCP/Hijacking, Fragmentation attacks, Weak keys, social engineering, Port scanning, Birthday attacks, Password guessing, Software exploitation, Inappropriate system use, TCP sequence number attacks, War dialing/demon dialing attacks.

Network Defence Tools: Firewalls; Design of firewalls; VPN, Filtering, Intrusion detection

Security protocols – Network and transport layer security- SSL/TLS, IPsec IKE; Kerberos; S/MIME; PGP.

Reading List:

1. William Stallings, Cryptography and Network Security, 6th edition Pearson Education, 2014
2. A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 2004.
3. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security: Private Communication in a Public World, Prentice Hall, 2002.

CS436	Secure Software Engineering	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Software Engineering (CS351)
ii. Cryptography and Engineering Secure Systems (CS401)

Course Outcomes: At the end of the course the student will be able to:

CO1	Assess the secure software engineering problems, including the specification, design, implementation, and testing of software systems. (Analyze)
CO2	Analyze and specify security requirements through SRS. (Analyze)
CO3	Construct software solutions to security problems using various paradigms. (Apply)
CO4	Implement the secure software systems using Unified Modeling Language Sec (UML Sec). (Apply)
CO5	Implement testing strategies for Secure software applications. (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	S
CO2	S	S	S	M									S	S	S
CO3	S	M	L										S	S	M
CO4	S	M	L										S	S	M
CO5	S	M	L										S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Software assurance and software security, threats to software security, sources of software insecurity, benefits of detecting software security, managing secure software development

Defining properties of secure software, how to influence the security properties of software, how to assert and specify desired security properties

Secure software Architecture and Design: Software security practices for architecture and design: Architectural risk analysis, software security knowledge for Architecture and

Design: security principles, security guidelines, and attack patterns, secure design through threat modeling

Writing secure software code: Secure coding techniques, Secure Programming: Data validation, Secure Programming: Using Cryptography Securely, Creating a Software Security Programs.

Secure Coding and Testing: code analysis- source code review, coding practices, static analysis, software security testing, security testing consideration through SDLC, vulnerability assessment and penetration testing (VAPT) tools - metasploit and nmap.

Reading List:

1. Julia H Allen, Sean J Barnum, Robert J Ellison, Gary McGraw, Nancy R Mead, Software Security Engineering: A Guide for Project Managers, Addison Wesley, 2008.
2. Ross J Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 2nd Edition, Wiley, 2008.
3. Howard, M. and LeBlanc, D., Writing Secure Code, 2nd Edition, Microsoft Press, 2003.
4. Robert C. Seacord, Secure Coding in C and C++, Addison-Wesley, Second Edition, 2005.

CS461	Deep Learning for NLP	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Natural Language Processing (CS374)

Course Outcomes: At the end of the course, the students will be able to:

CO1	Apply suitable word embeddings (word and subword models) for the NLP task (Apply)
CO2	Build a language modeling system and evaluate its performance (Apply)
CO3	Construct a model and evaluate for Machine Translation task (Apply)
CO4	Build and evaluate a model for Reading Comprehension (Apply)
CO5	Build and evaluate a model for Natural Language Generation (Apply)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	L		L					S	M	M
CO2	S	S	S	M	M	L		L					S	M	M
CO3	S	S	S	M	M	L		L					S	M	M
CO4	S	S	S	M	M	L		L					S	M	M
CO5	S	S	S	M	M	L		L					S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Unsupervised word embeddings: word2vec – CBOW, Skip-gram, Negative sampling, Hierarchical Softmax, Glove, Comparison of word embeddings to SVD, Evaluation methods for word embeddings.

Neural Networks Review: Backpropagation, activation functions, regularization, and optimization.

Sequence models and Language Modelling: Recurrent and Recursive Neural Nets, n-gram language models, RNNs and LSTMs for language modeling, handling long-term dependencies, Evaluation methods for language models.

Machine Translation: Statistical and Neural, Attention, Beam search, Teacher forcing, Exposure bias, Performance metrics.

Sub word models and contextual embeddings: Byte pair embeddings, FastText, and ELMO.

Transformers and Transfer Learning: Self Attention, Multi-head Attention, Encoder-Decoder of Transformer, BERT, GPT, and T5, pre-train fine-tune paradigm for downstream tasks.

Reading Comprehension: Attention models, Transformer models, Performance metrics-beyond Accuracy, Open Domain Question Answering.

Natural Language Generation: Summarization and Dialogue systems, Training and Decoding approaches, Weaknesses of existing evaluation methods.

Introduction to ethics and bias.

Reading List:

1. Jacob Eisenstein, “Natural Language Processing”, MIT Press, 2018, https://raw.githubusercontent.com/jacobeisenstein/gt-nlp-class/master/notes/eisenstein-nlp-notes.pdf
2. Yoav Goldberg, “A Primer on Neural Network Models for Natural Language Processing”, Arxiv, 2015, https://u.cs.biu.ac.il/~yogo/nnlp.pdf
3. Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research, 2011, https://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, “Deep Learning”, MIT Press, 2016, https://www.deeplearningbook.org/

CS462	Social network Analytics	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Data Structures and algorithms (CS201)

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

CO1	Compute network measures for a social media dataset. (Apply)
CO2	Analyze different varieties of networks from a social media dataset. (Analyze)
CO3	Evaluate various diffusion models and influence techniques. (Evaluate)
CO4	Construct techniques for community detection. (Create)
CO5	Construct network measures for recommender systems. (Create)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	L	L	L								S	M	L
CO2	S	S	M	M	L								S	M	L
CO3	S	S	M	M	M								S	S	L
CO4	S	S	S	M	M								S	S	L
CO5	S	S	S	M	M								S	S	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to Social Network Mining, Graph Models:

Graph Essentials, Degree and Degree Distribution, Graph Representations, Types of Graphs, Connectivity in Graphs, Graph/Tree Traversal, Shortest Path Algorithms, Minimum Spanning Trees, Network Flow Algorithms

Network Measures:

Centrality, Degree Centrality, Eigenvector Centrality, Katz Centrality, PageRank, Topic-Specific PageRank, Betweenness Centrality, Closeness Centrality, Group Centrality, Transitivity and Reciprocity, Balance and Status, Similarity, Structural Equivalence, Regular Equivalence

Network Models:

Properties of Real-World Networks, Degree Distribution, Clustering Coefficient, Average Path Length, Random Graphs, Modeling Real-World Networks with Random Graphs, Small-World model, Modeling Real-World Networks with the Small-World Model, Preferential Attachment Model, Modeling Real-World Networks with the Preferential Attachment Model

Communities and Interactions:

Community Detection, Community Detection Algorithms, Member-Based Community Detection, Group-Based Community Detection, Community Evolution, Community Detection in Evolving Networks.

Information Diffusion in social media:

Information Cascades, Independent Cascade Model (ICM), Maximizing the Spread of Cascades, Diffusion of Innovations, Modeling Diffusion of Innovations, Epidemics, SI Model, SIR Model, SIS Model, SIRS Model

Influence and Homophily:

Measuring Assortativity, Measuring Assortativity for Nominal Attributes, Measuring Assortativity for Ordinal Attributes, Influence, Homophily, Distinguishing Influence and Homophily

Recommendation in social media:

Classical Recommendation Algorithms, Content-Based Methods, Collaborative Filtering (CF), Extending Individual Recommendation to Groups of Individuals, Recommendation Using Social Context, Extending Classical Methods with Social Context, Evaluating Recommendations, Evaluating Relevancy of Recommendations, Evaluating Ranking of Recommendations

Reading List:

1. Reza Zefarani, Mohammad Ali Abbasi, Huan Liu, “Social Media Mining: An Introduction.” Cambridge University Press, 2014. ISBN: 978-1107018853.
2. Jure Leskovec, Anand Rajaraman and Jeffrey David Ullman, Mining of massive datasets, Cambridge University Press, 2014

CS463	Information Retrieval	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)
ii. Applied Machine Learning (CS373)

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

CO1	Identify the efficient IR model for an application or its dataset. (Apply)
CO2	Apply text processing, indexing, clustering and classification techniques to a corpus of texts for a specific information need (Apply)
CO3	Apply the techniques for web searching. (Apply)
CO4	Evaluate how techniques of IR integrate with recommender system. (Evaluate)
CO5	Evaluate how techniques of IR integrate with Question Answering. (Evaluate)

Course Articulation Matrix:

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	L									S	M	L
CO2	S	M	M	L									S	M	L
CO3	S	M	M	L									S	M	L
CO4	S	S	M	M									S	S	M
CO5	S	S	M	M									S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to IR and Text pre-processing:

Introduction, Data vs Information Retrieval, Logical view of the documents, Architecture of IR System, Web search system, Tokenization, Text Normalization, Stop-word removal, Morphological Analysis, Word Stemming, Lemmatization, Index term selection, Inverted indices, Positional Inverted index, Basic NLP tasks – POS tagging; shallow parsing

IR Models:

Classes of Retrieval Model, Boolean model, Term weighting mechanism – TF, IDF, TF-IDF weighting, Cosine Similarity, Vector space model, Probabilistic models, Non-Overlapping Lists, Proximal Nodes Model

Evaluation of IR:

Precision, Recall, F-Measure, MAP (Mean Average Precision), Discounted Cumulative Gain, Known-item Search Evaluation

Query Operations and Languages

Relevance feedback and pseudo relevance feedback, Query expansion (with a thesaurus or WordNet and correlation matrix), Spelling correction, Query languages

Web Search:

Search engines, Spidering (Structure of a spider, Simple spidering algorithm, multithreaded spidering, Bot), Directed spidering, Crawlers (Basic crawler architecture), Link analysis (HITS, Page ranking), Query log analysis, Handling “invisible” Web – Snippet generation, Cross Language Information Retrieval

Text Categorization and Clustering:

Categorization, Learning for Categorization, General learning issues, Learning algorithms: Bayesian (naïve), Decision tree, KNN, Rocchio), Clustering algorithms (Hierarchical clustering, k-means, k-medoid, Expectation maximization (EM), Text shingling)

Recommender System:

Personalization, Collaborative filtering recommendation, Content-based recommendation

Question Answering

Information bottleneck, Information Extraction, Ambiguities in IE, Architecture of QA system, Question processing, Paragraph retrieval, Answer processing

Reading List:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. http://nlp.stanford.edu/IR-book/information-retrieval-book.html
2. Modern Information Retrieval. Baeza-Yates Ricardo and Berthier Ribeiro-Neto. 2nd edition, Addison-Wesley, 2011.

CS471	Security and Privacy	DEC	3 – 0 – 0	3 Credits

Pre-requisites:

i. Computer Networks (CS352)
ii. Cryptography and Engineering Secure Systems (CS401)

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

CO1	Assess the risks and vulnerabilities in the data communications. (Analyze)
CO2	Analyze privacy related aspects of data uses. (Analyze)
CO3	Assess proposed technical mechanisms for privacy protection. (Analyze)
CO4	Construct an efficient technique for privacy preserving data analysis. (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	S
CO2	S	M	L					L					S	S	M
CO3	S	M	L					L					S	S	M
CO4	S	M	L					L					S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Introduction to Security – Vulnerabilities and Attacks; Introduction to Privacy - Policies and Practices; Data Privacy Taxonomy and Frameworks; Security Design Principles; Threat Modeling; Access Control; Anonymity Models; Authentication Beyond Passwords; Enterprise Roles – Technology and Law; Security Policies; Standards and Best Practices; Privacy in Cloud Infrastructure and Big Data.

Reading List:

1. J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC press, 2008.
2. C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy, now Publishers, 2014.
3. April Falcon Doss, Cyber Privacy: Who Has Your Data and Why You Should Care, BenBella Books, 2020.

CS472	Cyber Laws and Intellectual Property Rights	DEC	3 - 0 - 0	3 Credits

Pre-requisites:

i. Network Security (CS435)
ii. Security and Privacy (CS471)

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

CO1	Assess the cyber laws in general and Indian IT act in particular. (Analyze)
CO2	cybercrimes and their culpability under various sections of the act. (Analyze)
CO3	Examine cyber case laws and recall various cases for developing solutions. (Analyze)
CO4	Assess the intellectual property rights in Indian context. (Analyze)
CO5	Extract illegal knowledge related to computer-based activities and diffuse such knowledge. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M				L					S	S	S
CO2	S	S	S	M				L					S	S	S
CO3	S	S	S	M				L					S	S	S
CO4	S	S	S	M				L					S	S	S
CO5	S	M	L					L					S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Cyber Space- Fundamental definitions -Interface of Technology and Law

Jurisprudence and-Jurisdiction in Cyber Space - Indian Context of Jurisdiction - Enforcement agencies – Need for IT act - UNCITRAL – E-Commerce basics

Information Technology Act, 2000 - Aims and Objects — Overview of the Act – Jurisdiction - Electronic Governance – Legal Recognition of Electronic Records and Electronic Evidence - Digital Signature Certificates - Securing Electronic records and secure digital signatures - Duties of Subscribers - Role of Certifying Authorities -

Regulators under the Act -The Cyber Regulations Appellate Tribunal - Internet Service Providers and their Liability – Powers of Police under the Act – Impact of the Act on other Laws.

Cyber Crimes -Meaning of Cyber Crimes –Different Kinds of Cybercrimes – Cybercrimes under IPC, CRPC and Indian Evidence Law - Cybercrimes under the Information Technology Act, 2000 - Cybercrimes under International Law - Cyber Stalking, Virus Dissemination,

Software Piracy, Internet Relay Chat (IRC) Crime, Credit Card Fraud, Net Extortion - Cyber Terrorism - Violation of Privacy on Internet - Data Protection and Privacy – Indian Court Case Studies

Intellectual Property Rights

Copyrights- Software – Copyrights vs Patents debate - Authorship and Assignment Issues - Copyright in Internet - Multimedia and Copyright issues - Software Piracy –

Trademarks - Trademarks in Internet – Copyright and Trademark cases

Patents - Understanding Patents - European Position on Computer related Patents - Legal position on Computer related Patents - Indian Position on Patents – Case Law

Domain names - Registration - Domain Name Disputes-Cyber Squatting-IPR cases

Reading List:

1. Justice Yatindra Singh: Cyber Laws, Universal Law Publishing Co., New Delhi
2. Farouq Ahmed, Cyber Law in India, New Era publications, New Delhi
3. S.R.Myneni: Information Technolgy Law (Cyber Laws), Asia Law House, Hyderabad
4. Chris Reed, Internet Law-Text and Materials, Cambride University Press
5. Pawan Duggal: Cyber Law- the Indian perspective Universal Law Publishing Co., New Delhi

Course offered by ECE department to CSE students

EC237	DIGITAL LOGIC DESIGN	ESC	2 – 0 – 2	4 Credits

Pre-requisites: EC101-Basic Electronic Engineering

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand number representation and Boolean algebra. Design digital components.
CO2	Compile and Simulate Verilog models of digital circuits using CAD tool.
CO3	Analyze digital systems and improve the performance by reducing complexities.
CO4	Design of combinational and sequential logic circuits and develop Verilog models.

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	S	S	S	L	L	L	-	-	-	-	-	-	S	S
CO2	S	S	M	L	-	-	-	-	-	-	-	-	S	S
CO3	S	S	S	M	L	M	-	-	-	-	-	-	S	S
CO4	S	S	S	S	M	L	-	-	-	-	-	L	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed syllabus:

Digital Hardware, Design Process, Structure of a computer, Digital representation of Information.

Variables and functions, Logic Gates and networks, Boolean algebra, Synthesis using AND, OR, NOT gates, Design examples, Introduction to CAD tools and Verilog HDL. Implementation of Logic functions, Minimization and k-maps, Product-of-sums Form, incompletely specified functions. Multiple output circuits. Positional Number representation, Additions of unsigned and signed number, FastAdders, Design of Arithmetic Circuits using CAD tools. Combinational Circuit Building blocks – Multiplexers, Decoders, encoders, Code converters, Arithmetic Comparison Circuits, Verilog for Combinational circuits.

Flip-Flops, Registers and Counters – Basic Latch, SR and D latches, Edge triggered D Flip-flop, T and JK Flip Flops. Registers, Synchronous and Asynchronous Counters. Reset Synchronization, using storage elements with CAD tools, Using Verilog constructs for Registers and Counters.

Synchronous Sequential circuits – Basic design steps,State Assignment problem. Moore and Mealy State models, Design of Finite State Machines using CAD tools.

Reading List:

1. S. Brown, Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*,McGrawHill, third edition, 2014.
2. W. I Fletcher, *An Engineering approach to Digital Design*, Eastern Economy edition, PHI Limited, 2000.
3. J. Bhasker, Verilog Primer, 3rd edition, Prentice-Hall India, 1998.

4. S. Palnitkar, Verilog HDL: A guide to digital Design and Synthesis, 2nd edition, Pearson, 2003.

EC337	Microprocessor	PCC	3-0-2	4 Credits

Pre-requisites:

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

CO1	Understand the evolution of microprocessors
CO2	Understand assembly language programming basics for 8085
CO3	Understand the basic serial I/O and interrupt mechanism used in 8085
CO4	Understand the data transfer techniques using microprocessor
CO5	Understand the typical 16-bit microprocessor

Course Articulation Matrix:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**M**	**S**	**-**	**S**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**M**	**-**
CO2	**-**	**S**	**-**	**S**	**-**	**-**	**-**	**-**	**L**	**-**	**-**	**-**	**M**	**-**
CO3	**-**	**S**	**-**	**-**	**-**	**-**	**-**	**-**	**L**	**-**	**-**	**-**	**M**	**-**
CO4	**-**	**M**	**-**	**S**	**-**	**-**	**-**	**-**	**L**	**-**	**-**	**-**	**M**	**-**
CO5	**-**	**M**	**-**	**S**	**-**	**-**	**-**	**-**		**-**	**-**	**-**	**M**	**-**

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to Microprocessors: History and Evolution, types of microprocessors, Microcomputer Programming Languages, Microcomputer Architecture, Intel 8085 Microprocessor, Register Architecture, Bus Organization, Registers, ALU, Control section, Instruction set of 8085, Instruction format, Addressing modes, Types of Instructions**.**

Assembly Language Programming and Timing Diagram: Assembly language programming in 8085, Macros, Labels and Directives, Microprocessor timings, Micro instructions, Instruction cycle, chine cycles, T-states, State transition diagrams, Timing diagram for different machine cycles.

Serial I/O and Interrupts: Serial I/O using SID, SOD. Interrupt in 8085, RST instruction, Issues in implementing interrupts, Multiple interrupts and priorities, Daisy chaining, interrupt handling in 8085, Enabling, Disabling & masking of interrupts.

Data Transfer techniques: Data transfer techniques, Parallel & Programmed data transfer using 8155. Programmable parallel ports & handshake input/output, Asynchronous and Synchronous data transfer using 8251. PIC (8259), PPI (8255), DMA controller (8257).

Architecture of Typical 16-Bit Microprocessors: Introduction to a 16-bit microprocessor, Memory address space and data organization, Segment registers and Memory segmentation, generating a memory address, I/O address space, addressing modes, Comparison of 8086 & 8088, Basic configurations of 8086/8088, Min. Mode, Max. Mode & System timing, Introduction to Instruction Set of 8086.

Practical:

1. Write a program using 8085 Microprocessor for Decimal, Hexadecimal addition and subtraction of two Numbers.
2. Write a program using 8085 Microprocessor for addition and subtraction of two BCD numbers.
3. To perform multiplication and division of two 8-bit numbers using 8085.
4. To find the largest and smallest number in an array of data using 8085 instructions set. Interfacing
5. To write a program to arrange an array of data in ascending and descending order.
6. To convert given Hexadecimal number into its equivalent ASCII number and vice versa using 8085 instructions set.
7. To write a program to initiate 8251 and to check the transmission and reception of character.
8. To interface 8253 programmable interval timers to 8085 and verify the operation of 8253 in six different modes.
9. To interface a 7 segment LED with 8085.
10. To interface a stepper motor with 8085.
11. To interface a DAC with 8085.
12. To interface a ADC with 8085.

Text books:

1. R.S. Gaonkar, Microprocessor Architecture, Programming & Applications with the 8085/8080A, 5th Ed., Prentice Hall, 2002.
2. A.H. Mukhopadhyay, Microprocessor, Microcomputer and Their Applications, 3rd Edition Alpha Science International, Ltd.
3. M. Rafiquzzman: Microprocessors: Theory & Applications (Intel & Motorola), PHI.
4. Berry .B. Bray INTEL 8086/88, 80186, 286, 386, 486, Pentium Pro & Pentium IV.
5. D. V. Hall , Microprocessor and Interfacing', 3rd Edition, TMH.

Service Courses offered by CSED to Other Departments

CS285	Data Structures and Algorithms	ESC	2 – 1 – 2	4 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct solutions for problems using linear data structures such as Linked List, Stacks and Queues. (Apply)
CO2	Construct solutions for problems using non-linear Data Structures such as Trees and Graphs. (Apply)
CO3	Implement solutions for problems that requires sorting and searching as a sub-routine. (Apply)
CO4	Analyze, evaluate and choose appropriate data structures and algorithms for a specific application. (Analyze)
CO5	Analyze algorithms with respect to their time and space complexities. (Analyze)

Course Articulation Matrix:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L		S			S	S	M		
CO2	S	M	L		S			S	S	M		
CO3	S	M	L		S			S	S	M		
CO4	S	M	M	L	S			S	S	M		
CO5	S	M	M	L	S			S	S	M		

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction to Data Structures, Algorithm Analysis and Examples based on Asymptotic Notations, Abstract Data Types (ADTs), Stacks, Queues, Circular Queues and Linked List (Singly Linked, Doubly Linked and Circular).

Trees: Representation of Trees, Binary Trees, Binary Search Trees.

Priority Queues, Binary Heap and applications, Hash Tables and Operations, Collision Resolution: Open Addressing and Chaining.

Graphs: Representation of Graphs, Graph Traversal Techniques, Minimum Cost Spanning Trees: Prim's and Kruskal's Algorithms, Shortest Path Algorithms: Dijkstra's Algorithm and Floyd-Warshall Algorithm.

Sorting Algorithms: Merge Sort, Heap Sort, Quick Sort and Counting Sort.

List of Experiments:

1. Implementation of Stacks and Queues using arrays.
2. Implementation of Stack and Queue based applications.
3. Implementation of Single Linked List, Double Linked List and Circular Linked List.
4. Implementation of Stacks and Queues using Linked List.
5. Implementation of Circular Queues.
6. Implementation of Binary Search Trees with its operations.
7. Implementation of Priority Queues.
8. Implementation of Hashing with open addressing and separate chaining methods.
9. Implementation of Graph Traversal techniques: BFS and DFS.
10. Implementation of Minimum cost spanning tree algorithms.
11. Implementation of Dijkstra and Floyd-Warshall Algorithms.
12. Implement the following sorting algorithms: Merge sort, Heap sort, Quick sort, Counting sort.

Reading List:

1. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Fourth Edition, 2014.
2. Data structures and algorithms in C++, 4th Edition, Adam Drozdek, Thomson, Cengage, 2012.
3. Data structures and Algorithms in C++, Michael T. Goodrich, R. Tamassia, and Mount, Second Edition, Wiley, 2011.
4. Data Structures: A Pseudocode Approach with C++, Richard F. Gilberg, Behrouz A. Forouzan, Pacific Grove, CA: Brooks/Cole, 2001.

Open Elective Courses offered by CSED*

1. **CS340** Object Oriented Programming
2. **CS390** Database Management Systems
3. **CS440** Web Programming

*Students who are pursuing Minor in Software Engineering are not allowed to opt for the Open Elective Courses offered by CSED. The syllabus for these courses is available at the Minor in Software Engineering courses.

Courses for Minor in Software Engineering

CSM251	Algorithmics And Programming	PCC	2 – 0 – 2	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)
ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct and implement suitable data structures to meet the given time and space constraints. (Apply)
CO2	Construct and implement algorithms using techniques namely divide and conquer, greedy and dynamic programming. (Apply)
CO3	Infer the time and space complexities of the given algorithm. (Apply)
CO4	Translate algorithms into efficient programs using the Application Programming Interfaces of the runtime environment, with a comprehension of the underlying language features and trade-offs involved. (Apply)
CO5	Demonstrate problem solving and programming skills while solving unknown problems. (Apply)

Course Articulation Matrix:

PO / CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	S	M	L		M				S			
CO2	S	M	L		M				S			
CO3	S	M	L						S			
CO4	S	M	L		S				S			S
CO5	S	M	L		S				S	S		S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Data Structures: Abstract Data Types (ADT), Arrays and Records, Memory Layout and implementation of Stack, Queue, Linked List (Single, Double and Circular), Applications of Stack, Queue and Linked Lists, Trees: Binary Search Trees, Traversals and

Applications, Hash Tables: Open and Closed Addressing, Priority Queues: Heaps, Graphs: Memory Representation, Traversals and illustrative applications.

Algorithmics: Asymptotic notions, Step-counting, Computing the time and space complexities of simple iterative and recursive algorithms.

Divide and Conquer: Introduction, Merge Sort and Quick Sort.

Greedy: Greedy Choice and Optimal Substructure, Making Change and Activity Selection Problems, Prim's and Kruskal's algorithm, Dijkstra's Algorithm.

Dynamic Programming: Design template, overlapping sub-problems, Making Change problem and Floyd-Warshall's Algorithm.

List of Experiments: (Language Used: C++)

1. Implementation of Stack using Array.
2. Implementation of Queue using Array.
3. Implementation of Singly Linked List.
4. Implementation of Doubly Linked List.
5. Implementation of Stack and Queue using Linked Lists.
6. Implementation of Circular Linked List.
7. Implementation of a Divide and Conquer based solution for a given problem.
8. Implementation of a Greedy based solution for a given problem.
9. Implementation of a Dynamic Programming based solution for a given problem.
10. Implementation of a Dynamic Dictionary.

Reading List:

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia, and Mount, Wiley student edition, John Wiley and Sons, Second Edition, Wiley, 2011.
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Fourth Edition, 2014.
3. Data structures and algorithms in C++, Fourth Edition, Adam Drozdek, Thomson, Cengage.
4. Data Structures: A Pseudocode Approach with C++, Richard F. Gilberg, Behrouz A. Forouzan, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

CSM301/ CS340	Object Oriented Programming	PCC/ OPC	3– 0 – 0	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)

ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Construct programs using Object Oriented Design principles like abstraction, polymorphism and inheritance and typing. (Apply)
CO2	Develop applications with handlers for user-defined exceptions, according to the given requirements. (Apply)
CO3	Develop efficient multi-threaded applications with synchronization constructs. (Apply)
CO4	Develop interactive GUI applications with event handling to provide rich user experience. (Apply)
CO5	Develop applications that use file input and output. (Apply)

Course Articulation Matrix:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	M		L				L			L
CO2	S	M	M	L	L				L			L
CO3	S	M	M	L	L				L			L
CO4	S	M	M	L	M				L			L
CO5	S	M	M	L	M				L			L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Overview of Object-Oriented Programming and its need, Java Programming Elements: Classes and Objects, Data types, Constructors, Input-Output Handling, Control structures, Method overloading and overriding, Abstraction and Inheritance, Interfaces, final and static: classes, blocks and methods, Packages.

Exception Handling: Types of Exceptions, Exception classes, try, catch, throw, throws and finally, Exception Handling with Method Overriding, Custom Exceptions.

Multithreaded Programming: Introduction to multitasking through processes and threads, The Java Thread Model, creating threads, thread life cycle, thread scheduling, thread priorities, daemon thread, synchronization.

Garbage Collection, Runtime class and Memory management in Java.

String handling: String, StringBuffer, StringBuilder and tokenizer.

Generics: The Collections framework: List, Set and Map interfaces, Enumerator.

Event handling: Event, Listeners and adapter classes, anonymous inner classes.

Abstract Windowing Toolkit (AWT): Button, Label, Checkbox, CheckboxGroup, TextField, TextArea, Choice, List, Menu, Panel, Scrollbar and Layout managers.

File I/O: Character based Streams, Readers and Writers, RandomAccess, Scanner.

Reading List:

1. Java: The Complete Reference, Herbert Schildt, 11th edition, Mc Graw Hill, 2019.
2. Head First Java, Kathy Sierra & Bert Bates, 3rd edition, O'Reilly, 2005.
3. Clean Code, Robert C Martin, Pearson, 2012.
4. Object Oriented Programming with Java, Timothy Budd, Updated Edition, Pearson Education, 2020.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: https://cse.iitkgp.ac.in/~dsamanta/java/index.htm, Accessed on: August 2021.

CSM302	Object Oriented Programming Lab	PCC	0– 1 – 2	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)

ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Develop programs using object-oriented design principles. (Apply)
CO2	Develop and test programs that can handle exceptions. (Apply)
CO3	Develop and test efficient multi-threaded programs with thread synchronization. (Analyze)
CO4	Develop GUI applications with event handling that provide rich user experience. (Apply)
CO5	Construct programs using the suitable data structures and interfaces among List, Set and Map for efficient modeling of the objects and entities of the program. (Apply)
CO6	Develop and test programs using data available in files and write data to files with sequential and random access. (Apply)

Course Articulation Matrix:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	M		L			S	S	M		L
CO2	S	M	M	L	L			S	S	M		L
CO3	S	M	M	L	L			S	S	M		L
CO4	S	M	M	L	M			S	S	M		L
CO5	S	M	M	L	M			S	S	M		L
CO6	S	M	M	L	M			S	S	M		L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

List of Experiments:

1. Develop programs to familiarize with object-oriented design concepts.
2. Implement programs to illustrate overloading and overriding.
3. Implement program to study inheritance, polymorphism and data abstraction.
4. Implement abstract classes and Interfaces.
5. Implement programs using Static classes, blocks and methods.
6. Implement programs to compare shallow and deep copy.
7. Develop programs using Exception Handling.
8. Develop programs using multi-threading and synchronization.
9. Implement programs to familiarize with the Testing and Debugging facilities.
10. Implement programs using Generics.
11. Implement programs using Linked data structures, Heaps, priority queues, and binary search trees.
12. Implement programs using String processing using String, StringBuffer and StringBuilder.
13. Develop Event-driven programs for GUI using common interactive elements for rich user experience.
14. Develop Event-driven programs for GUI to develop gaming application.
15. Implement programs using Streams and File I/O system.
16. Implement a program to figure out if someone has won in a game of tic-tac-toe.
17. Implement a program to find all pairs of integers within an array which sum to a specified value.
18. Implement a program to find the frequency of occurrences of any given word in a book.

Reading List:

1. Java: The Complete Reference, Herbert Schildt, 11th edition, Mc Graw Hill, 2019.
2. Head First Java, Kathy Sierra & Bert Bates, 3rd edition, O'Reilly, 2005.
3. Clean Code, Robert C Martin, Pearson, 2012.
4. Object Oriented Programming with Java, Timothy Budd, Updated Edition, Pearson Education, 2020.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: https://cse.iitkgp.ac.in/~dsamanta/java/index.htm, Accessed on: August 2021.

CSM351/ CS390	Database Management Systems	PCC/ OPC	2 – 0 – 2	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)

ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

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

CO1	Construct database schema using Database models at conceptual level identifying entities and relationships among entities using E-R and relational models. (Apply)
CO2	Construct database and implement queries using SQL constructs for a given requirement specification. (Apply)
CO3	Construct database design using Normalization and Functional Dependencies to store information without redundancy. (Analyze)
CO4	Design and develop an application using stored procedures for a given requirement specification. (Apply)
CO5	Design and develop an application with database connectivity for a given requirement specification. (Apply)
CO6	Implement database maintenance and control using authorization access control, transaction and concurrency management and recover constructs. (Apply)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	L	S				M	L			S	S	L
CO2	S	M	L	L	S				M	L			S	S	L
CO3	S	M	S	L	S				M	L			S	S	M
CO4	S	M	M	L	S			S	S	M			S	S	L
CO5	S	M	M	L	S	S	S	S	S	M	L	L	S	S	S
CO6	S	M	M	L	S			S	S	M			S	S	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Introduction: Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Database Design, Database Engine, Database and Application Architecture

Database Users and Administrators, History of Database Systems

Introduction to the Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, The Relational Algebra

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database

Intermediate SQL: Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Index Definition in SQL, Authorization

Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, Advanced Aggregation Features

Database Design Using the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Complex Attributes, Mapping Cardinalities, Primary Key, Removing Redundant Attributes in Entity Sets, Reducing E-R Diagrams to Relational Schemas, Extended E-R Features, Entity-Relationship Design Issues, Alternative Notations for Modeling Data

Relational Database Design: Features of Good Relational Designs, Decomposition Using Functional Dependencies, Normal Forms, Functional-Dependency Theory, Algorithms for Decomposition Using Functional Dependencies, Decomposition Using Multivalued

Dependencies, More Normal Forms, Atomic Domains and First Normal Form, Database-Design Process, Modeling Temporal Data

Introduction to Indexing, Query processing and optimization, Transactions, concurrency control and recovery

List of Experiments:

1. E-R diagrams
2. SQL Query Language, Data Definition Language
3. SQL Queries, Operations, Set Operations, Null Values
4. Aggregate Functions
5. Nested Subqueries
6. Join Expressions
7. Views
8. Transactions
9. Integrity Constraints

10. SQL Data Types and Schemas
11. Index Definition in SQL
12. Authorization
13. Accessing SQL from a Programming Language
14. Functions and Procedures
15. Triggers
16. Recursive Queries

Reading List:

1. Abraham Silberschatz; Henry F Korth; S Sudarshan, Database System Concepts, 7th Edition, New York, NY: McGraw-Hill Education 2020
2. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 2nd Edition, McGraw-Hill, Inc., 2000

CSM401/ CS440	Web Programming	PCC/ OPC	2 – 0 – 2	3 Credits

Pre-requisites:

i. Introduction to Algorithmic Thinking and Programming (CS101)

ii. Introduction to Algorithmic Thinking and Programming Lab (CS102)

Course Outcomes: At the end of this course, students will be able to:

CO1	Discover the basic web components and development environment. (Apply)
CO2	Design and develop client-side scripting techniques (Apply)
CO3	Design and develop server-side scripting techniques (Apply)
CO4	Build real world applications using client side and server-side scripting languages (Apply)
CO5	Analyze SOAP and RESTful web Services. (Analyze)

Course Articulation Matrix

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L	M								M	L	L
CO2	S	M	S	L	S				M		L		S	M	L
CO3	S	M	S	L	S				M		L		S	M	L
CO4	S	M	S	L	S				S		M		S	M	L
CO5	S	M	S	M	S				M				S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

HTML5 - List - Tables - Images - Forms - Frames - Cascading Style sheets, XML Document type definition - XML Schemas, Document Object model.

Java Script - Control statements - Functions - Arrays - Objects - Events –Random number Generation - Dynamic HTML with Java Script – Ajax, JSON – Introduction, Responsive Web Design, Front end framework- Angular JS.

Different kinds of servers - web servers – Apache & nginx, File servers – time servers- DB servers. Server-side scripting languages Introduction: PHP basics - PHP server applications (No DB). Introduction to NodeJS – asynchronous nature of nodejs – simple apps Integrating application with DB – DB drivers- Integrate NodeJS with NOSQL, Integrate NodeJS with SQL.

Web Architecture & Web services: MVC introduction- thin clients Vs Thick clients. Web services – Introduction- SOAP, REST – writing a RESTful service (nodejs + expres). SOAP Vs REST.

List of Experiments:

1. Construct static web pages by make use of basic HTML5.
2. Construct fascinating web pages by make use of CSS3.
3. Construct XML documents – formatting, styling, and schema.
4. Construct an interactive web page by make use of client-side scripting.
5. Create a required environment to develop a complete web application.
6. Develop a backend application for any specific real-world application.
7. Construct a static web application by make use of framework.
8. Develop a complete dynamic web application for any real-world application using framework.

Reading List:

1. Paul Deitel, Harvey Deitel, Abbey Deitel, “Internet and World Wide Web How to Program”, Pearson, Fifth Edition, 2011.
2. Jeffrey C. Jackson, “Web Technologies – A Computer Science Perspective”, Pearson Education, Fourth Edition, 2012.
3. Anthony, Accomazzo, Murray Nathaniel, Lerner Ari, “Fullstack React: The Complete Guide to React JS and Friends”, Fullstack.io, 2017.
4. Brown, Ethan, “Web Development with Node and Express: Leveraging the JavaScript Stack”, O'Reilly Media, 2019.
5. Dayley B., “Node.js, MongoDB, and AngularJS Web Development”, Addison-Wesley Professional, 2014.

CSM402	Software Engineering	PCC	1-0-2	3 Credits

Pre-Requisites:

None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply the methods of software project management in order to manage qualy product. (Apply)
CO2	Develop design document and compute effort estimates for a software project. (Create)
CO3	Design UML Diagrams in order to implement forward and reverse engineering. (Create)
CO4	Develop quality test cases to improve the quality assurance. (Create)
CO5	Apply test driven development approach to execute efficient test cases. (Apply)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	M	L	L	M	L		M	M	M		M		
CO2	M	L	M			M	M		M	M			M		
CO3	M	M	M	L		L	L			M	L		M		
CO4	M	M	M		M	M	M		M	M	L		M		
CO5	M	M	M		M	M	M		M	M	L		L		

M: Medium correlation, L: Low correlation

Detailed Syllabus:

Software Life Cycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Spiral Model, Agile Development Models. Software Project Management: Responsibilities of a Software Project Manager- Job Responsibilities for Managing Software Projects, Skills Necessary for Managing Software Projects, Project Planning- Sliding Window Planning, The SPMP Document of Project Planning.

Requirements gathering and analysis, software requirement specification- users of SRS document- characteristics of a good SRS document, Attributes of Bad SRS documents, important categories of customer Requirements, Functional requirements, Traceability, organization of the SRS document.

Software Design and Modelling: Software Design: Approaches to software design- function oriented design- object oriented design. Object Modelling Using UML: Basic object orientation Concepts, Use case Model, Class diagram, Interaction diagrams, Activity Diagram, state chart Diagram, Component and Deployment diagrams.

Testing, Black-Box Testing- Equivalence class partitioning- Boundary value analysis, White-Box Testing- Basic concepts- statement coverage- branch coverage- multiple condition coverage- path coverage- McCabe's cyclomatic complexity metric, Integration testing, System Testing.

List of Experiments:

1. Develop a Software Project Plan for Indian railway online unreserved ticket issuing system using Microsoft Project.
2. Develop a SRS Document for Online certification portal using Rational Requisite Pro Tool.
3. Design UML Diagrams for Online internet banking using Umbrello.
4. Develop a java code with minimum of 6 functions and design JUnit test cases using eclipse.
5. Construct a java code with minimum of 8 functions and apply Test Driven Development approach on each and every function to execute test cases. Develop test cases on chosen web application using test case execution sheet.
6. Automate chosen web application and develop both negative and positive test case using selenium web driver.

Reading Lists:

1. Ian Sommerville, "Software Engineering", 7th edition, Pearson education.7/e, 2005.
2. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Publication, 2009.
3. Timothy C Lethbridge, Object-Oriented Software Engineering, Practical software development using UML and Java, McGraw-Hill Education, 2nd edition, 2004.
4. Roger S. Pressman, Software Engineering, A Practitioner's approach, McGraw Hill, 8th edition, 2014.
5. Pankaj Jalote, "Software Engineering, A Precise Approach", Wiley India, 2010.

Courses for Honors in Data Science

CSH301	Foundations of Data Science	PCC	4 – 0 – 0	4 Credits

Students of this course upon completion will learn how to solve standard distributed computing problems. Students will also learn consensus algorithms. Further students shall also learn designing widely used algorithms for map-reduce/spark paradigms of distributed computing.

Pre-requisites:

i. Artificial Intelligence (CS303)

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

CO1	Construction, analysis and evaluation of learning objectives for Regression (Analyze)
CO2	Analyze efficiency large scale matrix factorizations for different data science scenarios (Analyze)
CO3	Apply basic machine learning algorithms (Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes) for predictive modeling (Apply)
CO4	Designing Recommender System for different applications. (Analyze)
CO5	Designing feature reduction objectives using PCA (Apply)
CO6	Analyze various clustering objectives K-Means-EM-SpectralClustering (Analyze)

Course Articulation Matrix:

PO/PSO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Basics of Optimization in Data Science: Linear Programming Objectives, Primal-Dual Methods, Quadratic Programming, Convex Optimization, Gradient Descent, Adaptive

Learning Rate, Quasi Newton's method, Constrained Optimization, KKT Conditions for Optimization Problems

Probability theory including random variables, conditional probability, Bayes law, concentration of measure, linear algebra including eigenvalues, norms, elementary spectral graph theory.

Regression: Least Squares Objective for Multiple Linear Regression, Gauss-Markov Theorem, Statistical Tests, Weighted Least Squares, Box-Cox Transformation, Polynomial & Spline Regression, Ridge Regression, Bias-Variance Tradeoff, Subset Selection, LASSO, Adaptive LASSO, Elastic Net, Dantzig Selector, SLOPE and sorted penalties.

Matrix Factorizations used in Data Science:

Cholesky Decomposition and QR Factorization: Concepts, Applications

Eigen Vector Decomposition, Solving Large Scale Value Problems using Lanczos Method, Arnoldi's Iteration

SVD: Geometric Interpretation, Best Rank-K Approximation using SVD, Power Method for SVD, Efficient methods for SVD, Applications of SVD

PCA: PCA learning objective: Application of PCA for dimensionality reduction

Recommendation Systems: Collaborative Filtering using gradient Descent and Alternating Least Squares for recommender systems.

Classification: GLM methods for classification, SVM, Naïve Bayes, Evaluation of classification methods – Confusion matrix, Students T-tests and ROC curves, Feature Selection for Classification

Clustering: Choosing distance metrics – Different clustering approaches – hierarchical agglomerative clustering, k-means (Lloyd's algorithm), EM Algorithm for clustering, Spectral Clustering: Graph Laplacian, Properties of Graph Laplacian, Application of Spectral Clustering for Transfer-learning.

Link Analysis: Random Walks, Markov Chains, Stationary Distribution, Metro Polis Hastings Algorithm, Gibbs Sampling, Convergence of Random Walks on Undirected Graphs, Page Rank, HITS, combating spam, personalized page rank

High Dimensional Space High dimensional sphere, volumes of high dimensional solids, gaussians in high dimension, high dimensional point sets, Johnson-Lindenstrauss theorem.

Sampling and VC-dimension Random walks and graph sampling, MCMC algorithms, learning, linear and non-linear separators, VC-dimension and connection to sampling, PAC learning.

Sketching and Sparsification Locality sensitive hashing, min-wise hashing, Bloom filters, count-min sketches, compressed sensing, graph sparsification techniques, pseudo-random generators with application to approximating norms.

Reading List:

1. Avrim Blum, John Hopcroft, and Ravindran Kannan: Foundations of Data Science Cambridge University Press, 2020
2. Jianqing Fan, Runze Li, Cun-Hui Zhang, Hui Zou Statistical Foundations of Data Science, CRC Press 2020
3. Jure Leskovec, Anand Rajaraman, Jeff Ullman: Mining of Massive Datasets, Cambridge University Press, 2016
4. G. Strang. Introduction to Linear Algebra, Wellesley-Cambridge Press, Fifth edition, USA, 2016
5. Relevant Research articles shared by the instructor

CSH302	Advanced Computational Statistics	PCC	4 – 0 – 0	4 Credits

Students of this course upon completion will gain a broad comprehension of the importance of computation in statistics and machine learning. Focus of this course will be on the mathematical and statistical underpinnings of why and how seminal modern statistical methods and inference works.

Pre-requisites:

i. Probability, Statistics and Stochastic Processes (MA239)
ii. Differential and Integral Calculus (MA101)
iii. Matrices and Differential Equations (MA151)

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

CO1	Apply theorems and probability axioms in constructing statistical models and inferring their parameters. (Apply)
CO2	Apply CDF estimation techniques, hypothesis testing, Bayesian inference, variational inference and statistical decision theory in performing inference with respect to parametric and non-parametric models. (Apply)
CO3	Construct statistical models using stochastic processes with a comprehension of their underlying representational ability in modeling the uncertainty in the problem domain. (Apply)
CO4	Apply stochastic optimization techniques and Bayesian modeling / inference techniques with a comprehension of how randomness and uncertainty in the domain is modeled. (Apply)
CO5	Apply Monte Carlo methods with an understanding of the role of various random sampling strategies in solving a given problem and with a comprehension of the important implementation issues in MC methods. (Apply)

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	S
CO2	S	M	L										S	S	S
CO3	S	M	L										S	S	S
CO4	S	M	L										S	S	S

CO5	S	M	L										S	S	S
CO6	S	M	L										S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Probability:

Review: Probability (Sample spaces, Conditional probability, independent events, Bayes' theorem).

Review: Random variables (Distribution and probability functions, Discrete and continuous random variables, Bivariate, Marginal and conditional distributions, Multivariate distributions and IID samples, Expectation, Variance, Covariance, Conditional expectation, Moment generating functions).

Probability Inequalities, Convergence types of Random variables, Law of large numbers, Central limit theorem, The Delta method, L1 Convergence.

Statistical Inference:

Parametric and non-parametric models, Point estimation, Confidence sets, Hypothesis testing – a review.

CDF estimation, Statistical functionals, Bootstrap: Simulation, Variance estimation, Confidence intervals, Percentile intervals.

Parametric inference: Method of moments, Maximum likelihood estimators: Properties, Consistency and Equivariance, Asymptotic normality, Optimality, Multiparameter models, Parametric bootstrap, Sufficiency, Exponential families and Conditional maximum likelihood estimators. Case study on Noise Contrastive Estimation.

Hypothesis testing and p-values: Wald test, Chi-square distribution, Pearson's Chi-square test for multinomial data, Permutation test, Likelihood ratio test, Multiple testing, Goodness-of-fit tests.

Bayesian Inference: Functions of parameters, Simulation, Large sample properties of Bayes' procedures, Flat priors, improper priors and non-informative priors, Multiparameter problems, Bayesian testing, Strengths and weaknesses of Bayesian inference.

Statistical Decision theory: comparing risk functions, Bayes estimators, Minmax rules, Admissibility, Stein's paradox.

Introduction to Variational Inference.

Statistical Processes and Methods:

Stochastic Processes: Markov processes, Poisson processes, Birth-death processes.

Stochastic optimization: Robbins-Monro and Kiefer-Wolfowitz algorithms, simulated annealing, stochastic gradient methods.

Simulation methods:

Monte Carlo methods: Rejection sampling, importance sampling, variance reduction methods (Rao-Blackwellization, stratified sampling).

MCMC methods: Gibbs sampling, Metropolis-Hastings, Langevin methods, Hamiltonian Monte Carlo, slice sampling. Implementation issues: burnin, monitoring convergence.

Sequential Monte Carlo (particle filtering)

Reading List:

1. Larry Wasserman, All of Statistics, First Edition, Springer-Verlag, 2004.
2. Geof H. Givens and Jennifer A. Hoeting, Computational Statistics, Second Edition, Wiley, 2005.
3. Christian P. Robert, George Casella, Monte Carlo Statistical Methods, Springer, First Edition, 2004

CSH351	Applied Machine Learning	PCC	4 – 0 – 0	4 Credits

Pre-requisites:

i. Artificial Intelligence (CS303)

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

CO1	Constructing learning objectives and associated algorithms for Regression and Classification using Probabilistic Approaches (Apply)
CO2	Constructing learning objectives and algorithms for regression and classification using the maximum margin principle (Apply)
CO3	Constructing algorithms for mixture models using Expectation Maximization (Apply)
CO4	Designing models for Sequence Labelling problems (Apply)
CO5	Applying feature reduction and clustering strategies for different learning objectives (Apply)
CO6	Design of efficient numerical algorithms for solving maximum margin objectives

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Basics of Linear Algebra, Probability Theory and Optimization: Vectors, Inner product, Outer product, Inverse of a matrix, Eigenanalysis, Singular value decomposition, Probability distributions – Discrete distributions and Continuous distributions; Independence of events, Conditional probability distribution and Joint probability distribution, Bayes theorem, Unconstrained optimization, Constrained optimization – Lagrangian multiplier method.

Methods for Function Approximation: Linear models for regression, Parameter estimation methods - Maximum likelihood method and Maximum a posteriori method; Regularization, Ridge regression, Lasso, Bias-Variance decomposition, Bayesian linear regression.

Probabilistic Models for Classification (Generative): Bayesian decision theory, Bayes classifier, Minimum error-rate classification, Normal (Gaussian) density – Discriminant functions, Decision surfaces, Maximum-Likelihood estimation, Maximum a posteriori estimation; Naive Bayes classifier, non-parametric techniques for density estimation -- Parzen-window method, K-nearest neighbors method,

Probabilistic Models for Classification (Discriminative): Softmax(Maximum Entropy) Classification, efficient parameter estimation for large-scale softmax based classifier, Multi-Layer Perceptron for classification, backpropagation, optimizing neural network par

Maximum Margin Approaches: Support Vector Regression: Primal and Dual forms, Support Vector Classification: Primal and Dual forms, Linear vs Non-Linear Kernels, SVM for Non-Linear Decision boundary, Multi-Class SVM, Parameter estimation for SVM using SMO, cutting-plane method for linear svm, sequential dual method for struct-svm.

Mixture Models: Gaussian mixture models -- Expectation-Maximization method for parameter estimation; Mixture of Multinomials, EM algorithm for semi-supervised learning, derivation of EM steps for probabilistic latent semantic indexing

Sequence Labelling Models: Hidden Markov models (HMMs) for sequential pattern classification: forward backward algorithm, Viterbi algorithm, derivation of HMM parameters using EM, Conditional Random Fields (CRF), Parameter estimation for CRF, POS Tagging/Named Entity Recognition using HMM/CRF

Dimensionality Reduction Techniques: Principal component analysis, Fisher discriminant analysis, Multiple discriminant analysis, Probabilistic PCA

Pattern Clustering: Criterion functions for clustering, Techniques for clustering -- K-means clustering, Hierarchical clustering, Density based clustering and Spectral clustering; Cluster validation, Spectral Clustering for transfer learning

Reading List:

1. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006
2. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001
3. S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, 2009
4. E. Alpaydin, Introduction to Machine Learning, Prentice-Hall of India, 2010
5. G. James, D. Witten, T. Hastie and R. Tibshirani, Introduction to Statistical Learning, Springer, 2013.
6. Charles Sutton and Andrew McCallum. 2012. An Introduction to Conditional Random Fields. Found. Trends Mach. Learn. 4, 4 (April 2012), 267–373.

7. L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989, doi: 10.1109/5.18626.
8. Joachims, T., Finley, T. & Yu, CN.J. Cutting-plane training of structural SVMs. Mach Learn 77, 27–59 (2009). https://doi.org/10.1007/s10994-009-5108-8
9. Thomas Hofmann, Learning the Similarity of Documents: an information-geometric approach to document retrieval and categorization, Advances in Neural Information Processing Systems 12, pp-914-920, MIT Press, 2000
10. S. Sathiya Keerthi, S. Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. 2008. A sequential dual method for large scale multi-class linear svms. KDD-08 408–416

CSH352	Advanced Data Mining	PCC	4 – 0 – 0	4 Credits

Pre-requisites:

i. Design and Analysis of Algorithms (CS252)
ii. Database Management Systems (CS255)
iii. Database Management Systems Lab (CS257)

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

CO1	Identify and characterize of sequence families
CO2	Design models for analyzing stream data
CO3	Summarize effectively which arts of the graph stand out.
CO4	Identify interest patterns in Web logs, spatial databases and temporal data
CO5	Analyze the large-scale data that is derived from social networks

Course Articulation Matrix:

CO \ PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	L							S	S	S
CO2	S	S	S	M	M	L							S	S	S
CO3	S	S	S	M	M	L							S	S	S
CO4	S	S	S	M	M	L							S	S	S
CO5	S	S	S	M	M	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Sequential Pattern Mining concepts: Frequent and Closed Sequence Patterns: Sequential Patterns, GSP: An Apriori-like Method, PrefixSpan: A Pattern-growth, Depth-first Search Method, Mining Sequential Patterns with Constraints, Mining Closed Sequential Patterns;

Classification, Clustering, Features and Distances of Sequence Data: Three Tasks on Sequence Classification/Clustering, Sequence Features, Distance Functions over Sequences, Classification of Sequence Data, Clustering Sequence Data.

Mining Data Streams: The Stream Data Model, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Counting Ones in a Window, Decaying Windows

Graph Mining: Patterns in Static Graphs: Heavy-tailed Degree Distribution, Eigenvalue Power Law (EPL), Small Diameter, Triangle Power Laws (TPL, DTPL); Patterns **in Evolving Graphs:** Shrinking Diameters, Densification Power Law (DPL), Diameter-plot and Gelling Point, Oscillating NLCCs Sizes, Principal Eigenvalue Ova-Time; **Patterns in Weighted Graphs:** Snapshot Power Laws (SPL) – Fortification, Weight Power Law (WPL), Weighted Principal Eigenvalue Over Time.

Web Mining: Introduction, Web Content Mining, Web Structure Mining, Web Usage Mining

Spatial Mining: Introduction, Spatial Data Overview, Spatial Data Mining Primitives, Generalization and Specialization, Spatial Rules, Spatial Classification Algorithm, Spatial Clustering Algorithms

Temporal Mining: Introduction, Modeling Temporal Events, Time Series, Prediction,

Pattern Detection, Sequences, Temporal Association Rules

Mining Social-Network Graphs: Social Networks as Graphs, Clustering of Social-Network Graphs, Direct Discovery of Communities, Partitioning of Graphs, Finding Overlapping Communities, Simrank, Counting Triangles, Neighborhood Properties of Graphs

Reading List:

1. G Dong and J Pei, Sequence Data Mining, Springer, 2007
2. Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman "Mining of Massive Datasets", Cambridge University Press, 2014
3. Dunham, Margaret H., Data Mining: Introductory and Advanced Topics, Prentice Hall PTR, USA, 2002.
4. Deepayan Chakrabarti, Christos Faloutsos, Graph mining laws tools and case studies, Morgan and Claypool, 2012.

CSH401	Deep Learning	PCC	4 – 0 – 0	4 Credits

Pre-requisites:

i. Artificial Intelligence (CS303)
ii. Applied Machine learning (CS373)

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

CO1	Designing Strategies for Optimizing Neural Networks-based Learning (Apply)
CO2	Constructing Sequence Labelling approaches for Language and Vision applications (Apply)
CO3	Constructing CNN architectures suitable for diverse vision applications such as classification, image denoising, image enhancement, Semantic Segmentation, Object Detection (Apply)
CO4	Construct and Analyze different Deep Generative Models applicable for application scenarios such as Image Generation, Music Generation, Natural Language Generation (Analyze)
CO5	Analyze existing attention models for diverse application scenarios such as captioning, sequence labeling, and mage representation (Analyze)

Course Articulation Matrix:

PO/PSO \ CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	L	L
CO2	S	M	M	L									S	M	L
CO3	S	S	M	L									S	M	L
CO4	S	S	M	L									S	M	L
CO5	S	S	S	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

Detailed Syllabus:

Deep Learning:

Introduction of Deep Learning, Multi-layer Perceptron, Backpropagation, Optimizing Gradient Descent for Neural Network Learning using momentum, RMSPROP, AdamOpt, Choice of activation functions, Local Response Normalization, Batch Normalization, Dropout Regularization

Convolutional Neural Networks (CNNs):

Introduction to CNNs; Evolution of CNN Architectures: Alex Net, ZF Net, VGG, Inception Net, Res Net, Dense Net.

Visualization and Understanding CNNs:

Visualization of Kernels; Backprop-to-image/Deconvolution Methods; CNNs for Recognition and Verification: Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss; CNNs for Detection: Background of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, Retina Net; CNNs for Segmentation: FCN, Seg Net, U-Net, Mask-RCNN

Recurrent Neural Networks (RNNs):

Introduction to RNNs;

Sequence models and Language Modelling: Recurrent and Recursive Neural Nets, n-gram language models, RNNs and LSTMs for language modeling, handling long-term dependencies,

CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition,

Attention Models:

Introduction to Attention Models in Vision; Vision and Language: Image Captioning, Visual QA, Visual Dialog;

Transformers and Transfer Learning: Self Attention, Multi-head Attention, Encoder-Decoder of Transformer, BERT, GPT, and T5, pre-train fine-tune paradigm for downstream tasks. Spatial Transformers; Transformer Networks

Representation Learning and Deep Generative Models:

Unsupervised word embeddings: word2vec – CBOW, Skip-gram, Negative sampling, Hierarchical Softmax, Glove, Comparison of word embeddings to SVD, Evaluation methods for word embeddings

Popular Deep Generative Models: GANs, VAEs; Other Generative Models: Pixel RNNs, NADE, Normalizing Flows.

Reading List:

1. Goodfellow I, Bengio Y, and Courville A, Deep Learning, MIT Press, 2016
2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.
3. Michael Nielsen, Neural Networks and Deep Learning, 2016
4. Simon Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012.
5. Jacob Eisenstein, "Natural Language Processing", MIT Press, 2018, https://raw.githubusercontent.com/jacobeisenstein/gt-nlp-class/master/notes/eisenstein-nlp-notes.pdf
6. Yoav Goldberg, "A Primer on Neural Network Models for Natural Language Processing", Arxiv, 2015, https://u.cs.biu.ac.il/~yogo/nnlp.pdf

7. Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research, 2011, https://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf