NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

B.Tech. in

COMPUTER SCIENCE AND ENGINEERING

SCHEME OF INSTRUCTION AND SYLLABI

(Effective from 2021-22)

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

VISION

Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society

MISSION

- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitive environment.
- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.
- Fostering product-oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Attaining global recognition in Computer Science & Engineering education, research and training to meet the growing needs of the industry and society.

MISSION

- Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.
- Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of Computer Science and Engineering.
- Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit.

Department of Computer Science and Engineering:

Brief about the Department:

The Department of Computer Science & Engineering was established in the year 1991. The department offers high quality undergraduate, postgraduate and doctoral programs. The B. Tech (Computer Science and Engineering) program was started in the year 1983 with an intake of 20 students. The intake was subsequently increased to 120 in 2008. M. Tech (Computer Science and Engineering) program was started in 1987 with an intake of 18 and subsequently increased to 20 in 2008. M. Tech (Information Security) was introduced in the year 2008 Under ISEAP sanctioned by Ministry of Communication and Information Technology (MCIT), DOE, GOI, New Delhi with intake of 20. Later, it was renamed as Computer Science and Information Security. The Master of Computer Applications (MCA) program was started in 1986 with an intake of 30 and increased to 46 from 2008. B.Tech, M.Tech (CSE) and M.Tech (CSIS) programs were accredited in 2014 by NBA as per Washington Accord.

The department has distinguished and committed faculty members with PhD from reputed institutes. It has close rapport with MICROSOFT, TCS, INFOSYS, ORACLE, TRDDC-Pune, SUN Microsystems, EMC^2, ACCENTURE, C-DAC, MOTOROLA, HONEYWELL, NOVELL, PHILIPS, SAMSUNG, and IBM-Bangalore. The department has MOUs with TCS, IBM, C-DAC and INFOSYS, for training students and faculty on latest cutting edge technologies and also to pursue Research and Development activities. The Department has been selected as a Remote Center under Information Security Awareness Project (ISEAP) Phase II, MCIT Dept of IT. Along with department of ECE, it is associated in E&ICT Academy project sanctioned by Department of Electronics and Information Technology, Govt of India for Rs.25 crores to train faculty from Engineering and Polytechnic Streams across Telangana, Andhra Pradesh, Karnataka States and Pudicherry, Andaman & Nicobar islands and Lakshadweep UTs.

List of Programs offered by the Department:

Program	**Title of the Program**
B.Tech.	Computer Science and Engineering
M.Tech.	Computer Science and Engineering
	Computer Science and Information Security
M.C.A.	Master of Computer Applications
Ph.D.	Computer Science and Engineering

Note: Refer to the following weblink for Rules and Regulations of B.Tech. program:
https://www.nitw.ac.in/media/uploads/2021/08/27/btech_rules-and-regulations-2021-22.pdf

B.Tech. – Computer Science and Engineering

Program Educational Objectives

PEO-1	Apply computer science theory blended with mathematics and engineering to model computing systems.
PEO-2	Design, implement, test and maintain software systems based on requirement specifications
PEO-3	Communicate effectively with team members, engage in applying technologies and lead teams in industry.
PEO-4	Assess the computing systems from the view point of quality, security, privacy, cost, utility, etiquette and ethics.
PEO-5	Engage in lifelong learning, career enhancement and adapt to changing professional and societal needs

Program Articulation Matrix

Mission Statements / **PEO**	**PEO1**	**PEO2**	**PEO3**	**PEO4**	**PEO5**
Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.	3	3	2	3	2
Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of Computer Science and Engineering.	2	3	3	3	2
Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit.	2	3	2	3	2

1-Slightly; 2-Moderately; 3-Substantially

B.Tech. – COPUTER SCIENCE AND ENGINEERING

Program Outcomes (POs)

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

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and computer science and engineering to the solution of complex engineering problems
PO2	**Problem analysis:** Identify, formulate, research literature, and analyse 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 computer science and 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 computer science and 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.

Program Specific Outcomes (PSOs)

PSO1	Design algorithms for real world computational problems and analyze their complexities.
PSO2	Design and develop interfaces among subsystems of computing.
PSO3	Analyze large data samples and discover knowledge to provide solutions to engineering problems.
PSO4	Assess security, privacy, quality and cost parameters in developing software systems.

SCHEME OF INSTRUCTION

B.Tech. (Computer Science and Engineering) Course Structure

Induction Program (Two Weeks) - Mandatory

I – Year: I – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	IC001	Induction Program *				0	MNC
2	MA137	Linear Algebra, Calculus and Differential Equations	3	0	0	3	BSC
3	HS132	English for Technical Communication	2	0	2	3	HSC
4	PH131	Applied Physics	3	0	0	3	BSC
5	SM131	Economics and Financial Analysis	3	0	0	3	HSC
6	CS101	Problem Solving and Computer Programming	3	0	0	3	ESC
7	PH132	Applied Physics Laboratory	0	0	4	2	BSC
8	CS102	Problem Solving and Computer Programming Laboratory	0	1	4	3	ESC
9	IC101	Extra Academic Activity - I *	0	0	2	0	MNC
		Total	**14**	**1**	**12**	**20**	

* MNC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/mnc_1st-year.pdf

I – Year: II – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	MA187	Integral Calculus and Transforms	3	0	0	3	BSC
2	EC181	Analog and Digital Electronics	3	0	0	3	ESC
3	CS151	Discrete Mathematics	3	0	2	4	BSC
4	CS152	Data Structures	3	0	0	3	ESC
5	EC182	Analog and Digital Electronics Laboratory	0	1	2	2	ESC
6	CS153	Data Structures Laboratory	0	2	2	3	ESC
7	IC151	Extra Academic Activity - II *	0	0	2	0	MNC
		Total	**12**	**3**	**8**	**18**	

* MNC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/mnc_1st-year.pdf

SCHEME OF INSTRUCTION

B.Tech. (Computer Science and Engineering) Course Structure

II – Year: I – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	MA237	Probability, Statistics and Queuing Theory	3	0	0	3	BSC
2	EE231	Network Analysis	3	0	0	3	ESC
3	CS201	Microprocessors	3	0	2	4	ESC
4	CS202	Object Oriented Programming	3	0	0	3	PCC
5	CS203	Theory of Computation	3	0	2	4	PCC
6	CS204	Design and Analysis of Algorithms	3	0	0	3	PCC
7	CS205	Object Oriented Programming Laboratory	0	1	2	2	PCC
8	BT231	Bioengineering	2	0	0	2	BSC
9		Mandatory Non-Credit Course *	1	0	0	0	MNC
		Total	**20**	**1**	**8**	**24**	

* MNC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/mnc_2nd-year.pdf

II – Year: II – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS251	Statistical Foundations of Computer Science (R)	2	0	2	3	PCC
2	CS252	Modelling and Optimization Techniques	3	0	0	3	PCC
3	CS253	Computer Architecture	3	0	0	3	PCC
4	CS254	Database Management Systems	3	0	0	3	PCC
5	CS255	Language Processors	3	0	0	3	PCC
6	CS256	Modelling and Optimization Techniques Laboratory	0	1	2	2	PCC
7	CS257	Database Management Systems Laboratory	0	1	3	2.5	PCC
8	CS258	Language Processors Laboratory	0	1	3	2.5	PCC
		Total	**14**	**3**	**12**	**22**	

SCHEME OF INSTRUCTION

B.Tech. (Computer Science and Engineering) Course Structure

III – Year: I – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS301	Operating Systems	3	0	0	3	PCC
2	CS302	Data Warehousing and Data Mining	3	0	0	3	PCC
3	CS303	Software Engineering	3	0	0	3	PCC
4	CS304	Computer Networks	3	0	0	3	PCC
5		Department Elective - 1	3	0	0	3	PEC
6	CS305	Operating Systems Laboratory	0	1	2	2	PCC
7	CS306	Knowledge Engineering Laboratory	0	0	2	1	PCC
8	CS307	CASE Tools Laboratory	0	1	2	2	PCC
9	CS308	Computer Networks Laboratory	0	1	2	2	PCC
10		Mandatory Non-Credit Course *	1	0	0	0	MNC
		Total	**15**	**3**	**8**	**22**	

* MNC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/mnc_3rd-year.pdf

III – Year: II – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS351	Mobile Computing	3	0	0	3	PCC
2	CS352	Cryptography	3	0	0	3	PCC
3	CS353	Machine Learning	3	0	0	3	PCC
4	CS354	Advanced Algorithms	3	0	0	3	PCC
5		Department Elective - 2	3	0	0	3	PEC
6		Department Elective - 3	3	0	0	3	PEC
7		Open Elective - 1 #	3	0	0	3	OEC
8	CS355	Cryptography Laboratory	0	0	3	1.5	PCC
9	CS356	Machine Learning Laboratory	0	0	3	1.5	PCC
		Total	**21**	**0**	**6**	**24**	

OEC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/open-elective-1_vi-sem.pdf

SCHEME OF INSTRUCTION

B.Tech. (Computer Science and Engineering) Course Structure

IV – Year: I – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS401	Deep Learning	3	0	0	3	PCC
2	CS402	Network Security	2	0	2	3	PCC
3		Department Elective - 4	3	0	0	3	PEC
4		Department Elective - 5	3	0	0	3	PEC
5		Department Elective - 6	3	0	0	3	PEC
6		Open Elective – 2 #	3	0	0	3	OEC
7	CS449	Summer Internship/EPICS				2	PCC
			17	**0**	**2**	**20**	

OEC weblink: https://www.nitw.ac.in/media/uploads/2021/10/22/open-elective-2_vii-sem.pdf

IV – Year: II – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS451	Internet of Things (Online/Conventional)	2	0	0	2	PCC
2		Dept. Elective – 7 (Online/Conventional)	3	0	0	3	PEC
3	CS498	Seminar	0	0	2	1	SEM
4	CS499	Project Work @	0	0	8	4	PW
		Total	**5**	**0**	**10**	**10**	

@ **NOTE: Refer to the following link for the guidelines to prepare dissertation report:**
https://www.nitw.ac.in/media/uploads/2021/08/27/ug_project-report-format_55vW5pL.pdf

Note:

BSC – Basic Science Courses
ESC – Engineering Science Courses
PCC – Professional Core Courses
PEC – Professional Elective Courses
OEC – Open Elective Courses
HSC – Humanities and Social Science Courses
MNC – Mandatory Non-credit Courses
SEM – Seminar
PW – Project Work

Credits in Each Semester									
Cat. Code	**Sem-I**	**Sem-II**	**Sem-III**	**Sem-IV**	**Sem-V**	**Sem-VI**	**Sem-VII**	**Sem-VIII**	**Total**
BSC	8	7	5	0	0	0	0	0	20
ESC	6	11	7	0	0	0	0	0	24
PCC	0	0	12	22	19	15	8	2	78
PEC	0	0	0	0	3	6	9	3	21
OEC	0	0	0	0	0	3	3	0	6
HSC	6	0	0	0	0	0	0	0	6
MNC	0	0	0	0	0	0	0	0	0
PW	0	0	0	0	0	0	0	4	4
SEM	0	0	0	0	0	0	0	1	1
Total	**20**	**18**	**24**	**22**	**22**	**24**	**20**	**10**	**160**

Professional Elective Courses:

Semester	Elective Number	Course Code	Course Title
III/I	1	CS311	Web Technologies
		CS312	Graph Algorithms
		CS313	Advanced Data Structures
		CS314	Programming Language Concepts
		CS315	Artificial Intelligence
		CS316	Advanced Computer Architecture
		CS317	Computational Number Theory
		CS318	Game Theory
III/II	2,3	CS361	Advanced Theoretical Computer Science
		CS362	Distributed Computing
		CS363	Cyber Laws and IPR
		CS364	Software Metrics and Software Project Management
		CS365	Software Testing
		CS366	Quantum Computing
		CS367	Advanced Data Mining
		CS368	Bio-Informatics
		CS369	Human Computer Interaction
		CS370	Intrusion Detection Systems
		CS371	Data Networks
		CS372	Semantic Web
		CS373	Heterogeneous Computing
		CS374	Cloud Computing
		CS375	Computational Geometry
		CS376	Model Driven Frameworks
		CS377	Software Reliability Techniques
		CS378	High Performance Computing
		CS379	Advanced Operating Systems
		CS380	Privacy Preserving Data Publishing
IV/I	4, 5, 6	CS411	Algorithmic Game Theory
		CS412	Security and Privacy
		CS413	Information Retrieval
		CS414	Biometrics
		CS415	Secure Multi-party Computation
		CS416	Natural Language Processing
		CS417	Algorithmic Coding Theory
		CS418	Malware Detection and Mitigation
		CS419	Formal Methods in Software Engineering
		CS420	Advanced Databases
		CS421	Computational Learning Theory
		CS422	Advanced Software Testing
		CS423	Service Oriented Architecture
		CS424	Secure Software Engineering
		CS425	Design Patterns
		CS426	Program Analysis and Verification
		CS427	Algorithmic Techniques for Big Data
		CS428	Design of Secure Protocols

		CS5113	Computer Vision and Image Processing
		CS5151	Advanced Computer Networks
IV/II	7	CS461	Cyber Security
		CS462	Mining of Massive Datasets
		CS463	Real Time Systems
		CS464	IoT Security
		CS465	Cluster Computing
		CS466	Intelligent Agents
		CS467	Virtual Reality and Augmented Reality
		CS468	Social Networks
		CS469	Fog and Edge Computing

DETAILED SYLLABUS

MA137	LINEAR ALGEBRA, CALCULUS AND DIFFERENTIAL EQUATIONS	Credits 3-0-0: 3

Pre-Requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand the concepts of vector space and linear transformations
CO2	Apply orthogonal and congruent transformations to a quadratic form
CO3	Find the maxima and minima of multivariable functions
CO4	Solve arbitrary order linear differential equations with constant coefficients
CO5	Apply the concepts in solving physical problems arising in engineering

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	1	2	1	-	-	-	-	-	-	-	1	1	1	1
CO2	3	3	1	2	1	-	-	-	-	-	-	-	-	-	1	-
CO3	3	3	1	2	1	-	-	-	-	-	-	-	1	-	1	1
CO4	3	3	1	2	1	-	-	-	-	-	-	-	-	-	-	1
CO5	3	3	1	2	1	-	-	-	-	-	-	-	2	1	1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Linear Algebra: Vector Space; Linear dependence and independence of vectors; Linear Transformations; Inner-products and norms; 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 and congruent transformation; Properties of complex matrices - Hermitian, skew-Hermitian and Unitary matrices

Differential Calculus: Taylor's theorem with remainders; Taylor's and Maclaurin's expansions; Functions of several variables - 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

Ordinary Differential Equations: Geometric interpretation of solutions of first order ODE $y' = f(x,y)$; Exact differential equations; integrating factors; orthogonal trajectories; 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

Text Books / Reference Books / Online Resources:

1. R. K. Jain and S. R. K. Iyengar, *"Advanced Engineering Mathematics"*, Fifth Edition, Narosa Publishing House, 2016
2. Erwin Kreyszig, *"Advanced Engineering Mathematics"*, Eighth Edition, John Wiley and Sons, 2015
3. George B. Thomas and Ross L. Finney, "*Calculus and Analytic Geometry*", Pearson, Ninth Edition, 2020
4. Dennis G. Zill, "*Advanced Engineering Mathematics*", Jones & Bartlett Learning, Sixth Edition, 2018
5. B. S. Grewal, *"Higher Engineering Mathematics"*, 42^{nd} Edition, Khanna Publications, 2012

HS132	ENGLISH FOR TECHNICAL COMMUNICATION	Credits 2-0-2: 3

Pre-Requisites:

English proficiency above B1 level as per the CEFR (Common European Framework of Reference) for languages.

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

CO1	Write an effective paragraph using devices of coherence & cohesion, idioms, and phrasal verbs in context.
CO2	Construct an effective résumé and cover letter
CO3	Demonstrate the ability to employ a range of critical reading skills.
CO4	Employ reported speech, active and passive voice in engineering and scientific contexts to compile technical reports
CO5	Distinguish technical reports from other types of reports such as business reports, analytical reports, and progress reports.
CO6	Interpret technical data presented in the form of graphs, pie charts, and diagrams.
CO7	Demonstrate use of English speech sounds, stress and intonation in day-to-day situations/conversations/interactions

Syllabus:

Grammar Principles & Effective Sentence Construction (Correction of sentences, Concord) and Vocabulary Building (synonyms and antonyms): Idioms and Phrasal verbs--patterns of use and suggestions for effective employment in varied contexts.-Strategies for bringing variety and clarity in sentences- removing ambiguity - editing long sentences for brevity and clarity- **Reported speech**-contexts for use of reported speech - its impact on audiences and readers- active and passive voice-reasons for preference for passive voice in scientific English.
Paragraph-writing: Definition of paragraph and types- features of a good paragraph - unity of theme-coherence- linking devices- direction- patterns of development. **Note-making** - definition- the need for note-making - its benefits - various note formats- like tree diagram, block or list notes, tables, etc.
Letter-Writing: Its importance in the context of other channels of communication- qualities of effective letters-types -personal, official, letters for various purposes- emphasis on letter of application for jobs - cover letter and resume types -examples and exercises.
Reading techniques: Definition- Skills and sub-skills of reading- Skimming and Scanning - their uses and purposes- examples and exercises. **Reading Comprehension** - reading silently and with understanding- process of comprehension- types of comprehension questions.
Features of Technical English - description of technical objects and process- Report-Writing-definition- purpose -types- structure- formal and informal reports- stages in developing report- proposal, progress and final reports-examples and exercises.
Book Reviews- Oral and written review of a chosen novel/play/movie- focus on appropriate vocabulary and structure - language items like special vocabulary and idioms used

Language laboratory:

1**. English Sound System** -vowels, consonants, Diphthongs, phonetic symbols- using dictionary to decode phonetic transcription-- Received Pronunciation, its value and relevance- transcription of exercises-

2. Stress and Intonation –word and sentence stress - their role and importance in spoken English- Intonation in spoken English -definition, patterns of intonation- –falling, rising, etc.-use of intonation in daily life-exercises

3.Introducing oneself in formal and social contexts- Role plays. - their uses in developing fluency and communication in general.

4. Oral presentation - definition- occasions- structure- qualities of a good presentation with emphasis on body language and use of visual aids.

5. Listening Comprehension -Challenges in listening, good listening traits, some standard listening tests- practice and exercises.

6. Debate/ Group Discussions-concepts, types , Do's and don'ts- intensive practice.

Readings:

1. English for Engineers and Technologists (Combined edition , Vol. 1 and 2) Orient Blackswan 2010.
2. Ashraf, M Rizvi. Effective Technical Communication. Tata McGraw-Hill, 2006
3. Meenakshi Raman and Sangeetha Sharma. Technical Communication: Principles and Practice 2nd Edition, Oxford University Press, 2011.

Software:

1.Clear Pronunciation – Part-1 *Learn to Speak English.*
2. Clear Pronunciation – Part-2 *Speak Clearly with Confidence*
3. Study Skills
4.English Pronunciation

PH131	APPLIED PHYSICS	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Apply the concepts of wave and particle nature of matter and energy for solving problems
CO2	Understand the applications of Interference, diffraction, optical fibers, holography and lasers in engineering
CO3	Understand the basics of semiconductors, magnetism, super conductivity, nano materials and their applications in engineering.
CO4	Comprehend sensing technologies and their applications in computer science and engineering

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	1	1	-	-	-	-	-	-	-	-	-	1	1	-
CO2	3	3	1	1	-	-	-	-	-	-	-	-	-	1	1	-
CO3	3	3	1	1	-	-	-	-	-	-	-	-	-	1	1	-
CO4	3	3	1	1	-	-	-	-	-	-	-	-	-	1	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Wave Optics and Modern Physics:
Interference: Concept of Interference of Light-Division of Amplitude and Wave front with examples-Michelson and Fabry perot Interferometers- Applications
Diffraction: Fraunhofer’s Class of Diffraction at Single, Double and Multiple Slits-Gratings and Applications
Polarization: Production and Detection of Polarised Light—Wave Plates- Optical Activity-Laurent’s. half shade polarimeter
Lasers: Interaction of Radiation with Matter-Spontaneous and Stimulated Emissions-Basic requirements for the construction of Lasers-Construction and working of He-Ne, CO2, Nd-YAG and Semiconductor Lasers, Holography and HNDT
Optical Fibers: Principle and working of optical Fiber, structure, Classification and advantages of optical fiber, Light guiding mechanism in Optical Fibers -Numerical Aperture, Signal Degradation, Attenuation, Absorption, Inter and intra modal Dispersions. Fiber optics sensors and optical fiber communications
Quantum Physics (8 Classes)
Quantum Mechanics - Introduction to quantum theory, concepts and experiments led to the discovery, wave particle duality-Davisson-Germer experiment, Heisenberg uncertainty principle, Schrodinger time independent wave equation, the free particle problem - particle in an infinite and finite potential well, quantum mechanical tunnelling – applications; Hydrogen Atom Wave Functions, Angular Momentum Operators, Identical Particles, Quantum Optics - Introduction to quantum optics and Quantum Imaging.
Engineering Materials (14 Classes)
Magnetic Materials: Weiss Theory of Ferromagnetism –Properties – Domains – Curie Transition - Hard and soft magnetic materials – Ferrites – Structure, Classification, Applications in Computers.
Superconductors: Introduction to superconductivity, Meissner effect - Type-I and Type-II Superconductors – Applications in Computers.
Semiconductor Materials and Devices: Types of semiconductor materials, temperature and concentration effects on band gap, Hall effect, PN junction diode, photodiode, LED, junction transistor, phototransistor.
Nanomaterials – Introduction to Nanomaterials and Nano technology, Nano computers
Sensors and Sensing Technologies (5 Classes)

Introduction, The Human Body as a Sensor System, Passive and Active sensors, the sensor as part of a measurement system, sensor properties, Classification of Sensors – Infrared Sensor, Bio Sensors, Piezoelectric Sensors, Thermal Sensors, Quantum Sensors and Applications in Computer Science and Engineering.

Text Books / Reference Books / Online Resources:

1. Ajoy K.Ghatak, "*Optics*", Tata McGraw Hill, Sixth Edition, 2017.
2. Gerd Keiser, "*Optical Fibre communications*", McGraw Hill, 4th Edition
3. Arthur Beiser, Shobhit Mahajan, S. Rai Choudhury, "*Concepts of Modern Physics*", McGraw Hill Publications, Sixth edition,2009
4. M.N. Avadhanulu, P.G. Khirsagar, "*A Text Book of Engineering Physics*", 9th edition,2011
5. John Vetelino and Aravind Reghu, "*Introduction to Sensors*", CRC Press, 1st Edition, 2010.
6. Narciso Garcia, Arthur Damask and Steven Schwarz, "*Physics for Computer Science Students*", Springer, 2012, 2nd Edition.
7. Jeff Hecht, "*Understanding Lasers An Entry-Level Guide*", Wiley Publications, Fourth edition, 2018.
8. Hugh D. Young, Roger A. Freedman, "*University Physics with Modern Physics*", Pearson Education, 2014.
9. https://nptel.ac.in/courses/122/107/122107035

SM131	ECONOMICS AND FINANCIAL ANALYSIS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand various methods of Economic Analysis and apply
CO2	Calculate Depreciation using various methods
CO3	Sensitize to Macro Economic Environment and understand the growth of IT and ITES
CO4	Analyze the financial statements with ratio's for investment decisions
CO5	Analyze costs and their role in pricing
CO6	To develop effective presentation skills

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	-	2	-	-	-	-	-	-	-	-	-	-	2	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	-	2	2	3	-
CO3	-	2	-	-	-	-	-	-	-	-	-	-	2	-	-	1
CO4	-	3	-	-	-	-	-	-	-	-	-	-	2	2	-	-
CO5	-	3	-	-	-	-	-	-	-	-	-	-	2	2	-	-
CO6	1	-	-	-	-	-	-	-	-	-	-	-	2	-	3	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

ECONOMICS

1. Introduction to Engineering Economics, Fundamental concepts, Time value of money, Cash flow and Time Diagrams, Choosing between alternative investment proposals, Methods of Economic analysis (Pay back, ARR, NPV, IRR and B/C ratio),
2. The Effect of borrowing on investment, Equity Vs Debt Financing, Concept of leverage, Income tax and leverage
3. Depreciation and methods of calculating depreciation (Straight line, Sum of the years digit method, declining Balance Method, Annuity Method, Sinking Fund method).
4. National Income Accounting, Methods of Estimation, Various Concepts of National Income, Significance of National Income Estimation and its limitations.
5. Inflation, Definition, Process and Theories of Inflation and Measures to Control,
6. Balance of payments and its impact on exchange rate.
7. New Economic Policy 1991, Growth of IT and ITES in Indian economy, Start-up culture and initiatives by Government

FINANCIAL ANALYSIS

8. Analysis of financial statements, income statements and balance sheet (simple ratios).

9. Cost Accounting, Introduction, Classification of costs, Methods of Costing, Techniques of Costing, Cost sheet and preparation cost sheet, Breakeven Analysis, Meaning and its application, Limitation.

Presentations/ Group Discussions on current topics.

Text Books / Reference Books / Online Resources:

1. D N Dwivedi, "*Managerial Economics*", Vikas Publishing House Private Limited, 2018
2. Agrawal AN, "*Indian Economy*" Wiley Eastern Ltd, New Delhi, 2014
3. R.K Sharma and Sashi K Gupta,"*Financial Management*", Kalyani Publications
4. Arora, M.N., "*Cost Accounting*", Vikas Publication. , 2012
5. Latest trends in Indian Economy.
6. Capitaline Plus Database – http://www.capitaline.com/
7. Ministry of Finance – http:/finmin.nic.in/
8. Database of Indian Economy - http://dbie.rbi.org.in
9. Statistics of India – www.indiastat.com/ or http://mospi.nic.in/Website reference links

CS101	PROBLEM SOLVING AND COMPUTER PROGRAMMING	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Design algorithms for solving simple mathematical problems including computing, searching and sorting
CO2	Compare and contrast algorithms in terms of space and time complexity to solve simple mathematical problems
CO3	Explore the internals of computing systems to suitably develop efficient algorithms
CO4	Examine the suitability of data types and structures to solve specific problems
CO5	Apply control structures to develop modular programs to solve mathematical problems
CO6	Apply object oriented features in developing programs to solve real world problems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	1	1	1	2	1	-	-	-	-	-	3	3	3	3	3
CO2	2	1	2	1	2	3	-	-	-	-	-	3	3	2	2	2
CO3	1	2	2	2	2	1	-	-	-	-	-	3	3	2	2	1
CO4	2	2	2	2	2	2	-	-	-	-	-	2	3	2	1	2
CO5	2	2	3	1	2	2	-	-	-	-	-	2	3	2	2	2
CO6	2	2	3	2	2	2	-	-	-	-	-	2	3	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Fundamentals of Computers, Historical perspective, Early computers, Components of a computers, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

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

Number systems and data representation, Basics of C++, Basic data types.

Numbers, Digit separation, Reverse order, Writing in words, Development of Elementary School Arithmetic Testing System, Problems on Date and factorials, Solutions using flow of control constructs, Conditional statements - If-else, Switch-case constructs, Loops - while, do-while, for.

Functions – Modular approach for solving real time problems, user defined functions, library functions, parameter passing - call by value, call by reference, return values, Recursion, Introduction to pointers.

Sorting and searching algorithms, Large integer arithmetic, Single and Multi-Dimensional Arrays, passing arrays as parameters to functions

Magic square and matrix operations using Pointers and Dynamic Arrays, Multidimensional Dynamic Arrays

String processing, File operations.

Structures and Classes - Declaration, member variables, member functions, access modifiers, function overloading, Problems on Complex numbers, Date, Time, Large Numbers.

Text Books / Reference Books / Online Resources:

1. Walter Savitch, "*Problem Solving with C++*", Ninth Edition, Pearson, 2014.
2. Cay Horstmann, Timothy Budd, "*Big C++*", Wiley, 2nd Edition, 2009.
3. R.G. Dromey, "*How to solve it by Computer*", Pearson, 2008.

PH132	APPLIED PHYSICS LABORATORY	Credits 0-0-4: 2

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Use CRO, signal generator, spectrometer, polarimeter and GM counter for making measurements
CO2	Test optical components using principles of interference and diffraction of light
CO3	Determine the selectivity parameters in electrical circuits
CO4	Determine the width of narrow slits, spacing between close rulings using lasers and appreciate the accuracy in measurements

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	3	-	-	-	-	3	2	-	-	-	1	-	-
CO2	3	2	-	3	-	-	-	-	3	2	-	-	-	1	-	-
CO3	3	2	-	3	-	-	-	-	3	2	-	-	-	1	-	-
CO4	3	2	-	3	-	-	-	-	3	2	-	-	-	1	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
1. Determination of Wavelength of Sodium light using Newton’s Rings.
2. Determination of Wavelength of He-Ne laser – Metal Scale.
3. Measurement of Width of a narrow slit using He- Ne Laser.
4. Determination of Specific rotation of Cane sugar by Laurent Half-shade Polarimeter.
5. Determination of capacitance by using R-C circuit.
6. Determination of resonating frequency and bandwidth by LCR circuit.
7. Measurement of half-life of radioactive source using GM Counter.
8. Diffraction grating by normal incidence method.
9. Measurement of numerical aperture of optical fiber.

Text Books / Reference Books / Online Resources:

1. "*Physics Laboratory Manual*" by Physics Department, NIT Warangal,2021.
2. P.R. Sasi Kumar, "*Practical Physics*", PHI publications, first edition,2011
3. G.L.Squire, "*Practical Physics*", Cambridge University press, fourth edition,2001.
4. Dr.S.K.Gupta Krishna,"*Engineering Physics Practical*", Prakashan Publications, ninth edition,2010.
5. https://nptel.ac.in/courses/115/105/115105110/

CS102	PROBLEM SOLVING AND COMPUTER PROGRAMMING LABORATORY	Credits 0-0-4: 2

Pre-requisites: None

Course Outcomes:

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

CO1	Design and test programs to solve mathematical and scientific problems
CO2	Develop and test programs using control structures
CO3	Implement modular programs using functions
CO4	Develop programs using classes

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	1	2	1	-	-	-	-	-	-	3	3	3	3
CO2	1	1	2	1	2	2	-	-	-	-	-	-	3	3	2	2
CO3	1	2	3	2	2	1	-	-	-	-	-	-	3	3	2	2
CO4	2	2	2	2	2	3	-	-	-	-	-	-	2	3	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Laboratory:

1. Programs on conditional control constructs.
2. Programs on loops (while, do-while, for).
3. Programs using user defined functions and library functions.
4. Programs on arrays, matrices (single and multi-dimensional arrays).
5. Programs using pointers (int pointers, char pointers).
6. Programs on structures.
7. Programs on classes and objects.

Text Books / Reference Books / Online Resources:

1. Walter Savitch, "*Problem Solving with C++*", Ninth Edition, Pearson, 2014.
2. Cay Horstmann, Timothy Budd, "*Big C++*", Wiley, 2nd Edition, 2009.
3. R.G. Dromey, "*How to solve it by Computer*", Pearson, 2008.

MA187	INTEGRAL CALCULUS AND TRANSFORMS	Credits 3-0-0: 3

Pre-requisites: MA137: Linear Algebra, Calculus and Differential Equations

Course Outcomes:

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

CO 1	Analyze improper integrals
CO 2	Evaluate multiple integrals in various coordinate systems
CO 3	Apply the concepts of gradient, divergence and curl to formulate engineering problems
CO 4	Convert line integrals into surface integrals and surface integrals into volume integrals
CO 5	Apply Laplace transforms and Fourier transforms to solve physical problems arising in engineering

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO2	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO3	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO4	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO5	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

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.

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; Stoke's theorem; Gauss Divergence theorem.

Laplace Transforms: Laplace transforms; inverse Laplace transforms; Properties of Laplace transforms; Laplace transforms of unit step function, impulse function, periodic function; Convolution theorem;

Fourier Transforms: Fourier transformation and inverse transforms - sine, cosine transformations and inverse transforms - simple illustrations.

Text Books / Reference Books / Online Resources:

1. R. K. Jain and S. R. K. Iyengar, "*Advanced Engineering Mathematics*", Narosa Publishing House, 5th Edition, 2016.
2. Erwin Kreyszig, "*Advanced Engineering Mathematics*", John Wiley and Sons, 8th Edition, 2015.
3. George B. Thomas and Ross L. Finney, "*Calculus and Analytic Geometry*, Pearson, Ninth Edition, 2020
4. Dennis G. Zill, "*Advanced Engineering Mathematics*", Jones & Bartlett Learning, Sixth Edition, 2018
5. B. S. Grewal, "*Higher Engineering Mathematics*", 42^{nd} Edition, Khanna Publications, 2012

EC181	ANALOG AND DIGITAL ELECTRONICS	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Design digital components including - decoders, multiplexers, arithmetic circuits
CO2	Design of synchronous sequential circuits
CO3	Analyze digital systems and improve the performance by reducing complexities.
CO4	Test digital systems and analyze faults.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	1	1	1	-	-	-	-	-	-	3	3	-	-
CO2	3	3	3	-	2	1	-	-	-	-	-	-	3	3	-	-
CO3	3	3	3	2	1	2	-	-	-	-	-	-	3	3	-	-
CO4	3	3	3	3	2	1	-	-	-	-	-	1	3	3	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Part-A: Analog Electronics (28hrs)
1.Introduction to electronics: Overview of semiconductor physics, review of P-N junction operation, Characteristics of PN-Junction diode, Zener diode, Schottky diode, LED, LCD. Applications od diodes: Rectifiers with filters (C, L, LC and π), Zener diode as a voltage regulator, Problem solving.(4)
2.BJT- CE, CB and CC configurations, CE Transistor Characteristics, Low frequency h-parameter model, Transistor Biasing, Stability and Thermal Runaway; JFET Characteristics, MOSFET: Enhancement and depletion mode characteristics. Applications: Single stage RC coupled amplifier and it's frequency response, gain-band width product, differential amplifier, CMRR, Transistor as a switch. Problem Solving. (6)
3.Feedback in Amplifiers: feedback topology-negative feedback, voltage series and shunt feedbacks, positive feedback, Bark-Hausen criterion, Oscillators- RC and LC Oscillators(4)
4. Integrated circuits: Introduction, Operational amplifiers(μA 741)- Ideal and practical characteristics and linear applications, V-I Converter and I-V Converters, Precision Rectifiers, Log and Antilog Amplifiers, Comparator Principle, Astable and monostable multivibrators, Sine wave and Triangular wave generators 555 Timer: Functional block diagram, Astable and monostable multivibrators Problem Solving. (10)
5.Regulated power supply: Shunt and series voltage regulators, UPS, Linear voltage regulator IC723, high and low voltage design , current boosting and current folding operation ,3-terminal regulator ICs(4)
Part- B : Digital Electronics(28hrs)
1.Number system and codes:
Analog versus digital system, number systems, base conversions, complements of numbers weighted and unweighted codes and error detecting and correcting codes, Alpha numeric code (ASCII), Error detecting and correcting codes.
2.Switching algebra and switching functions: Boolean algebra, postulates, theorems and switching algebra, completely and incompletely specified switching functions, Representation of Boolean functions in sum of products form and product of sums form, minimization of Boolean functions using Karnaugh map and Quine McCluskey methods. Problem solving. (6)
3.Combinational logic circuits and logic families: Logic gates, Logic gates operation using discrete components, Universal Logic gates, Logic design of combinational circuits: adders, Code converters, Comparators, multiplexers, de-multiplexers, encoders, decoders, buffers, tri-state buffers. Transistor as an inverter, Classification of logic families and their developments. TTL NAND gate analysis, ECL and CMOS logic family. Comparison TTL CMOS and ECL logic families.(8)
4.Sequential Logic circuits: Flip-Flops- RS Flip flop, Clocked RS flip-flop, JK flip-flop, T-flip-flop, JK flip-flops and M/S JK flip flop, Conversion of flip-flops. Registers: Buffer Register, Controlled buffer

register, Shift Registers,Universal shift register: SISO, SIPO, PISO, PIPO, Ring counter and twisted ring counter, Counters: Design of Asynchronous and Synchronous counters.(8)
Semiconductor Memories: RAM, ROM (Cell Structures and Organization on Chip) and their ICs(3)
Data Conversion Circuits: D/A converters- specifications, A/D converters- specifications, D/A converters such as DAC 0808, DAC 1408/1508, Integrated circuit A/D Converters ADC 0808, ICL 7106/7107.(3)

Text Books / Reference Books / Online Resources:

1. Jacob Millman, Christos Halkias, Chetan Parikh, "*Millman's Integrated Electronics*", 2nd Edition, McGraw Hill Education, 2017
2. Bhargava N. N., D C Kulshreshtha and S C Gupta, "*Basic Electronics & Linear Circuits*", Tata McGraw Hill, 2nd Edition, 2013
3. D.Roy Choudhury, Shail B. Jain, "*Linear Integrated Circuits*", 4th Edition, New Age International Publishers, 2018
4. Ramakant A. Gayakwad, "*Operational Amplifiers and Linear IC Technology*", PHI,1987
5. Donald P. Leach, Malvino and Saha, "*Digital Principles and Applications (SIE)*", 8th Edition Paperback, McGraw Hill Education, 2014
6. Zvi Kohavi, Niraj K Jha, "*Switching and Finite Automata Theory*", 8th Edition, McGraw-Hill Eighth edition
7. Robert L.Morris John R.Miller, "*Designing with TTL Integrated Circuits*", McGraw-Hill
8. Herbert Taub, Donald Schilling, "*Digital Integrated Electronics*", McGraw-Hill

CS151	DISCRETE MATHEMATICS	Credits 3-0-2: 4

Pre-requisites: MA181-Mathematics II

Course Outcomes:

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

CO1	Apply formal methods of proof to solve discrete problems
CO2	Apply Propositional logic and First order logic to solve problems
CO3	Apply techniques for counting the occurrences of discrete events including permutations, combinations with or without repetitions
CO4	Formulate and solve graph problems including searching and spanning
CO5	Formulate and solve recurrence relations

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	2	1	-	-	-	-	1	-	1	3	-	-	-
CO2	3	2	1	1	1	-	-	-	-	1	-	-	3	-	-	-
CO3	3	3		1	1	-	-	-	-	-	-	-	3	-	-	2
CO4	3	3	2	2	1	-	-	-	-	1	-	-	3	-	2	-
CO5	3	2	1	1	1	-	-	-	-	1	-	-	3	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Review of Sets, Relations and Functions.

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, Sub lattices, 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

Trees: Definition, Binary Tree Traversals, Spanning Trees, Minimum Spanning Tree Algorithms

Text Books / Reference Books / Online Resources:

1. Joe L. Mott, Abraham Kandel, Theodore P. Baker, "*Discrete Mathematics for Computer Scientists and Mathematicians*", Second Edition, PHI, 2001.
2. Tremblay J. P. and Manohar R., "*Discrete Mathematical Structures*", MGH, 1997.
3. Kenneth H. Rosen, "*Discrete Mathematics and Its Applications with Combinatorics and Graph Theory*", Seventh Edition, MGH, 2011.

CS152	**DATA STRUCTURES**	**Credits 3-0-0: 3**

Pre-requisites: CS101 – Problem Solving and Computer Programming

Course Outcomes:

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

CO1	Understand the concept of ADT, identify data structures suitable to solve problems
CO2	Develop and analyze algorithms for stacks, queues
CO3	Develop algorithms for binary trees and graphs
CO4	Implement sorting and searching algorithms
CO5	Implement symbol table using hashing techniques and multi-way search trees

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	1	1	1	-	-	-	-	-	-	3	3	2	3
CO2	3	3	3	2	1	-	-	-	-	-	-	-	3	3	2	3
CO3	3	3	3	2	1	2	-	-	-	-	-	-	3	3	2	3
CO4	3	3	3	3	2	1	-	-	-	-	-	-	3	3	2	3
CO5	3	3	3	3	1	2	-	-	-	-	-	-	3	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Iterative and Recursive Algorithms

Abstract Data Types (ADTs), Implementation and Applications of Stacks, Operations and Applications of Queues, Array Implementation of Circular Queues, Implementation of Stacks using Queues, Implementation Queues using Stacks, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues

Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs, Splay Trees, B-trees, Trie, C-Trie

Hashing: Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations

Priority Queues: Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

Sorting Algorithms: Stability and In Place Properties, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Lower Bound for Comparison Based Sorting Algorithms, Linear Sorting Algorithms: Counting Sort, Radix Sort, Bucket Sort

Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths, Biconnected Components.

Text Books / Reference Books / Online Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", Second Edition, PHI, 2009.
2. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C++*", Third Edition, Pearson Education, 2006
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "*Fundamentals of Computer Algorithms*", Second Edition, Universities Press, 2011.
4. Michael T.Goodrich and Roberto Tamassia, "*Algorithm Design: Foundations, Analysis and Internet Examples*", Second Edition, Wiley-India, 2006.

EC182	ANALOG AND DIGITAL ELECTRONICS LABORATORY	Credits 0-1-2: 2

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Carryout the experiments and understand characteristics of Si, Ge, diodes & Common Emitter Transistor characteristics
CO2	Execute rectifiers & filters & study ripple factor & regulation performance.
CO3	Implement BJT amplifier (CE), RC phase shift oscillator experiment.
CO4	Implement Op-amp applications –inverse amplifier
CO5	Implement logic gates and test their functionality

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
1. Study and Operation of Instruments, IC Trainer and IC testers.
2. BJT Characteristics in CE configuration mode.
3. Single stage BJT amplifier and it’s frequency response.
4. RC Phase Shift Oscillator
5. Measurement of Op Amp parameters
6. Op Amplifier applications as adder , subtractor and multivibrators
7. 555 Timer as Multivibrators
8. Combinational Logic Circuits : adders, subtractors & code converters
9. Adders, Multiplexers and Decoders : i) 4-bit Adder/Subtractor using 7483 and 7486 ii) Full adder and Full subtractor using Mux 74153 and 7404 iii) Full adder and Full subtractor using decoder 74138
10. Study of Flip-flops 7474 ,7476; 4-bit shift register using 7474 ICs; and implement synchronous counter using 7476ICs
11. Implementation of various modes of counters using 7490,7492 and 7493 ICs.
12. Implementation of a 4-bit shift register, self-generating ring counter and Johnson counter using 74195/7495/7498/7419

The following ICs are used in Digital Experiments:
ICs: 7400, 7402, 7404, 7408, 7432. 7486; Multiplexer ICs:74151, 74153 ; Decoder IC 74138; BCD to 7 segment decoder IC:7447; Flip-flop ICs:7474, 7476; Asynchronous Counter ICs: 7490,7492,7493; Synchronous counter ICs: 74190,74192,74193, Register ICs:7491,7495,74195.

Text Books / Reference Books / Online Resources:
1. IC Manuals

CS153	DATA STRUCTURES LABORATORY	Credits 0-2-2: 3

Pre-requisites: CS101- Problem Solving and Computer Programming

Course Outcomes:

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

CO1	Develop ADT for stack and queue applications
CO2	Implement tree and graph algorithms
CO3	Implement and analyze internal and external sorting algorithms
CO4	Design and implement symbol table using hashing technique

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	1	1	1	-	-	-	-	-	-	3	3	-	3
CO2	3	3	3	2	2	2	-	-	-	-	-	-	3	3	2	3
CO3	3	3	3	3	1	1	-	-	-	-	-	1	3	3	2	3
CO4	3	3	3	2	1	1	-	-	-	-	-	1	3	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Write a program to implement stack using arrays and evaluate a given postfix expression
2. Write a program to implement circular queue using arrays
3. Write a program to implement double ended queue (de queue) using arrays
4. Write programs for applications based on stacks and queues.
5. Write programs to implement the following data structures and their applications
(a) Single linked list (b) Double linked list
6. Write programs to implement a stack and a queue using linked lists
7. Write a program to create a binary search tree (BST) by considering the keys in given order and perform the following operations on it.
(a) Minimum key (b) Maximum key (c) Search for a given key
(d) Find predecessor of a node (e) delete a node with given key (f) applications of BST
8. Write a program to construct an AVL tree for the given set of keys. Also write function for deleting a key from the given AVL tree.
9. Write a program to implement hashing with (a) Separate Chaining and (b) Open addressing methods.
10. Implement the following sorting algorithms:
(a) Insertion sort (b) Merge sort (c) Quick sort (d) Heap sort
(e) Radix sort (f) Shell sort
11. Write programs for implementation of graph traversals by applying: (a) BFS (b) DFS
12. Write programs to find out a minimum spanning tree of graph by applying:
(a) Prim's algorithm (b) Kruskal's algorithm c) any other algorithms
13. Write a program to implement Dijkstra's algorithm using priority queue.
14. Write a program to find Euler's path.
15. Write a programs to find Biconnected components and strongly Connected components.
16. Write program for creation, insertion, and printing functions of a Treap
17. Write program for creation, insertion, deletion and printing functions of a B_d-Tree.
18. Write program for creation, insertion, deletion and printing functions of B_d+-Tree, B*-Tree.
19. Write program for creation, insertion, deletion and printing functions of a Trie.
20. Write program for creation, insertion, deletion and printing functions of a C-Trie.

Text Books / Reference Books / Online Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", Second Edition, PHI, 2009.
2. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C++*", Third Edition, Pearson Education, 2006

3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "*Fundamentals of Computer Algorithms"*, Second Edition, Universities Press, 2011.
4. Michael T.Goodrich and Roberto Tamassia, "*Algorithm Design: Foundations, Analysis and Internet Examples"*, Second Edition, Wiley-India, 2006.

MA237	PROBABILITY, STATISTICS AND QUEUING THEORY	Credits 3-0-0: 3

Pre-requisites: MA187: Integral Calculus and Transforms

Course Outcomes:

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

CO1	Find mean and variance of a given probability distribution
CO2	Test the hypothesis for small and large samples
CO3	Find the coefficient of correlation and lines of regression
CO4	Understand the characteristics of a queuing model

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO2	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO3	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-
CO4	3	3	1	2	1	-	-	-	-	-	-	-	-	1	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Random variables and their distributions:

Introduction to Probability, random variables (discrete and continuous), probability functions, density and distribution functions, mean and variance, special distributions (Binomial, Hyper geometric, Poisson, Uniform, exponential and normal), Chebyshev's inequality, parameter and statistic, estimation of parameters by maximum Likelihood Estimation method

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), Karl Pearson coefficient of correlation, lines of regression.

Queuing theory:

Concepts, applicability, classification, birth and death process, Poisson queues, Characteristics of queuing models - single server (with finite and infinite capacities) model, multiple server (with infinite capacity only) model.

Text Books / Reference Books / Online Resources:

1. R. A. Johnson, Miller and Freund's "*Probability and Statistics for Engineers*", Pearson Publishers, 9th Edition, 2017
2. John E. Freund, Benjamin M. Perles, "*Modern Elementary Statistics*", 12th Edition, Pearson, 2013
3. Hamdy A. Taha, "*Operations Research: An Introduction*", Pearson, 2017, Tenth Edition
4. S.C.Gupta and V.K.Kapoor, "*Fundamentals of Mathematical Statistics*", 12th Edition, S.Chand & Co, 2020
5. Kantiswarup, P.K.Gupta and Manmohan Singh, "*Operations Research*", Sultan Chand & Sons, 2014

EE231	NETWORK ANALYSIS	Credits 3-0-0: 3

Pre-requisites: Differential Equations, Laplace Transforms

Course Outcomes:

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

CO1	Simplify DC networks and analyze them using loop and node equations and determine the dual of a given network
CO2	Analyze magnetic circuits, and electric circuits with sinusoidal excitation along with the phenomenon of resonance
CO3	Formulate the dynamic equations of electric circuits using differential equations and simplify their solutions using Laplace.
CO4	Simplify the analysis of electric circuits using network theorems
CO5	Analyze and simplify two-port networks using their properties and interrelationships

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	3	3	2	1	1	1	1	-	2	_	_	-	3
CO2	3	3	3	3	3	2	1	1	1	1	-	2	_	_	-	-
CO3	3	3	3	3	3	2	1	1	1	1	-	2	_	2	-	-
CO4	3	3	3	3	3	2	1	1	1	1	-	2	_	_	-	-
CO5	3	3	3	3	3	2	1	1	1	1	-	2	_	_	1	-
CO6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Circuit Elements and Relations:

Types of circuit components, Types of Sources and Source Transformations, Star-delta transformation, KVL and KCL with dependent and independent Sources, DC circuit analysis, Formation of loop and node equations. Graph of a network- incidence matrix, Dual networks. [6]

Magnetic Circuits: Concept of MMF, flux and magnetic reluctance, self and mutual inductances, Dot convention, coefficient of coupling and coupled circuits. [4]

Steady State Analysis of Circuits for Sinusoidal Excitations: Concept of phasor, Single phase Series, Parallel, Series Parallel circuits, Concept of power factor, Solution of AC networks using mesh and nodal analysis, Phasor diagrams. Resonance, Series and Parallel resonance, Bandwidth, Q-factor and selectivity. [10]

Time Domain Analysis: Solution of network equations in time domain, Classical differential- equations approach, Initial conditions & evaluation, applications to simple RLC circuits only. [6]

Applications of Laplace Transforms in Circuit Theory: Laplace transforms of various signals of excitation, Laplace transformed networks. [3]

Network Theorems: Superposition theorem, Thevenin's theorem, Norton's theorem, Maximum power transfer theorem, Millman's theorem, Tellegen's theorem [5]

Two port parameters : Relationship of two port variables, Short circuit admittance parameters, open circuit impedance parameters, Transmission parameters, Hybrid parameters, Interrelationships, parallel connections of two port networks.[6]

Text Books / Reference Books / Online Resources:

1. M.E.Van Valken Burg, "*Network Analysis*", 3/e, PHI, 2015
2. William H. Hayt, Jack Kemmerly, Steven M. Durbin, "*Engineering Circuit Analysis*", 8th Edition, Tata McGraw-Hill, 2013
3. M.L.Soni and J.C. Gupta, "*A Course in Electrical Circuit Analysis*", Dhanpat Rai & Co. (P), 2001.
4. G.K.Mithal and Ravi Mittal, "*Network Analysis*", Khanna Pub., 2003.
5. Gopal G Bhise, Prem R Chadha &Durgesh C. Kulshreshtha Gopal, "*Engineering Network Analysis and Filter Design*", Umesh Publications, 2012
6. S.R. Paranjothi, "*Electric Circuit Analysis*", New Age International Pub.,2002.
7. De Carlo & Lin, "*Linear circuit Analysis*", Oxford University Press, 2nd Edition,2010.

CS201	MICROPROCESSORS	Credits 3-0-2: 4

Pre-requisites: EC181-Analog and Digital Electronics

Course Outcomes:

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

CO1	Understand the interfacing of the different components of personal computer system
CO2	Identify the suitable instructions of x86 microprocessor and develop assembly language programs for performing basic arithmetic, logic, I/O and control operations
CO3	Develop driver programs for the operation of keyboard microcontroller
CO4	Implement programs to operate timer and interrupt microcontroller chips with required configurations

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	1	2	2	2	-	2	-	2	2	2	-	2	1	2	-
CO2	2	2	1	2	2	-	-	-	-	2	2	-	2	1	2	-
CO3	2	2	2	2	2	1	2	-	-	2	2	2	2	1	3	-
CO4	2	2	2	2	2	1	2	-	-	1	1	2	2	1	3	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction of Microcomputer System: CPU, I/O devices, clock, memory, bussed architecture, tristate logic, address bus, data bus and control bus.

Architecture of 16-bit Microprocessor: Intel 8086 microprocessor, Pin description and internal architecture.

Operation and Control of Microprocessor: Timing and control unit, op-code fetch machine cycle, memory read/write machine cycles, I/O read/write machine cycles, interrupt acknowledge machine cycle, state transition diagram.

Instruction Set: Addressing modes; Data transfer, arithmetic, logical, branch, stack and machine control groups of instruction set, macro RTL and micro RTL flow chart of few typical instructions; Unspecified flags and instructions.

Assembly Language Programming: Assembler directives, simple examples; Subroutines, parameters

Interfacing: Interfacing of memory chips, address allocation technique and decoding; Interfacing of I/O devices, LEDs and toggle-switches as examples, memory mapped and isolated I/O structure; Input/Output techniques: CPU initiated unconditional and conditional I/O transfer, device initiated interrupt I/O transfer.

Interrupts: Interrupt structure of 8086 microprocessor, processing of vectored and nonvectored interrupts, latency time and response time; Handling multiple interrupts, Programmable Interrupt Controller 8259A, Pin configuration, configuring 8259A, cascading multiple 8259A chips.

Programmable Peripheral Interface: Intel 8255, pin configuration, internal structure of a port bits, modes of operation, bit SET/RESET feature, programming; ADC and DAC chips and their interfacing.

Programmable Interval Timer: Intel 8253, pin configuration, internal block diagram of counter and modes of operation, counter read methods, programming, READ-BACK command of Intel 8254.

Text Books / Reference Books / Online Resources:

1. Barry B. Brey, "*The Intel Microprocessors-Architecture, Programming and Interfacing*", 8th Edition, Pearson Prentice Hall publishers, 2009
2. Hall D. V., “*Microprocessor and Interfacing-Programming and Hardware*”, 2nd Ed., Tata McGraw-Hill Publishing Company Limited, 2008.

CS202	OBJECT ORIENTED PROGRAMMING	Credits 3-0-0: 3

Pre-requisites: CS101-Problem Solving and Computer Programming
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand object oriented paradigms: abstraction, encapsulation, inheritance, polymorphism
CO2	Learn java concepts like exception handling, interfaces, object classes and various libraries.
CO3	Design object oriented solutions for real world problems.
CO4	Implement the applications using the learnt concepts

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	2	-	-	1	-	3	-	-	-	2	1	-	-	3	-
CO2	1	1	2	3	3	3	-	-	-	3	-	-	-	-	-	-
CO3	2	1	1	1	1	1	-	-	3	-	3	3	1	3	2	2
CO4	1	-	1	1	1	2	2	-	3	3	2	2	1	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Object Oriented Thinking – A way of Viewing the World, Computation as Simulation, Messages and Methods; - A Brief History Of Object - Oriented Programming - The History of Java, The White Paper Description; – Object - Oriented Design - Responsibility Implies Noninterference, Programming in the Small and in the Large, Why Begin with Behavior? A Case Study in RDD, CRC Cards – Recording Responsibility, Components and Behavior, Software Components, Formalizing the Interface; - A Paradigm - Program Structure, The Connection to the Java World, Types, Access Modifiers, Lifetime Modifiers; - Ball Worlds – Data Fields, Constructors, Inheritance, The Java Graphics Model, The Class Ball, Multiple Objects of the Same Class; - A Cannon Game – The Simple Cannon Game, Adding User Interaction; Pinball Game Construction Kit – First Version of Game, Adding Targets : Inheritance and Interfaces, Pinball Game Construction Kit : Mouse Events Reconsidered; - Understanding Inheritance – An Intuitive Description of Inheritance, The Base Class Object, Subclass, Subtype, and Substitutability – Forms of Inheritance, Modifiers and Inheritance, The Benefits of Inheritance, The Costs of Inheritance; - A Case Study : Solitaire – The Class Card, The Game – Card Piles- Inheritance in Action, The Application Class, Playing the Polymorphic Game, Building a More Complete Game; - Polymorphism - Varieties of Polymorphism, Polymorphic Variables, Overloading, Overriding, Abstract Methods, Pure Polymorphism; - The AWT – The AWT Class Hierarchy, The Layout Manager, User Interface Components, Panels, Dialogs, The Menu Bar; - Input And Output Streams - Streams versus Readers and Writers, Input Streams, Stream Tokenizer, Output Streams, Object Serialization, Piped Input and Output; -
Understanding Graphics - Colour, Rectangles, Fonts, Images, Graphic Contexts, A Simple Painting Program; - Applets And Web Programming – Applets and HTML, Security Issues, Applets and Applications, Obtaining Resources Using an Applet, Combining Applications and Applets.

Text Books / Reference Books / Online Resources:

1. Timothy Budd, "*Object Oriented Programming with JAVA*", Updated Edition, Pearson Education, 2009.
2. Herbert Schildt, "*Java 2 Complete Reference*", TMH, 2010.

CS203	THEORY OF COMPUTATION	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand formal machines, languages and computations
CO2	Design finite state machines for acceptance of strings
CO3	Design context free grammars for formal languages
CO4	Develop pushdown automata accepting strings
CO5	Design Turing machine
CO6	Distinguish between decidability and undecidability

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO2	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO3	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO4	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO5	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO6	2	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Finite Automata - Structural Representations. Automata and Complexity, The Central Concepts of Automata Theory, Alphabets, Strings, Languages, Enabling the Automata to Ignore Actions, Deterministic, non-deterministic, Finite Automata with Epsilon-Transitions, Uses of e-Transitions.

Regular expressions - The Operators of Regular Expressions, Building Regular Expressions, Precedence of Regular-Expression Operators, Finite Automata and Regular Expressions, From DFA's to Regular Expressions, Converting DFA's to Regular Expressions by Eliminating States, Converting Regular Expressions to Automata, Applications of Regular Expressions, Regular Expressions in UNIX, Lexical Analysis, Finding Patterns in Text, Algebraic Laws for Regular Expressions, Associativity and Commutativity, Identities and Annihilators, Distributive Laws, The Idempotent Law, Laws Involving Closures, Pumping Lemma

Context Free Grammars - Derivations Using a Grammar, Leftmost and Rightmost Derivations, The Language of a Grammar, Sentential Forms, Parse Tress, Constructing Parse Trees, The Yield of a Parse Tree, Applications of Context-Free Grammars, Parsers, The YACC Parser-Generator, Ambiguity in Grammars and Languages, Ambiguous Grammars, Removing Ambiguity From Grammars

Push Down Automata - Definition of the Pushdown Automaton, A Graphical Notation for PDA's, Instantaneous Descriptions of a PDA, The Languages of a PDA, Acceptance by Final State, Acceptance by Empty Stack, Equivalence of PDA's and CFG's, Context Free Languages - Properties, Normal Forms for Context-Free Grammars, Eliminating Useless Symbols.

Turing Machines - Introduction to Turing Machines, Problems That Computers Cannot Solve, Notation for the Turing Machine, Instantaneous Descriptions for the Turing Machines, Transition Diagrams for Turing Machines, The Language of a Turing Machine, Turing Machines and Halting, 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.

Undecidability - A Language that is Not Recursively Enumerable, Enumerating the Binary Strings, Codes for Turing Machines, The Diagonalization Language, and An Undecidable Problem That is RE, Complements of Recursive and RE Languages, The Universal Language, and Undecidability of the Universal Language

Text Books / Reference Books / Online Resources:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D Ullman, "*Introduction to Automata Theory, Languages and Computation*", 2nd Edition, Pearson, 2001
2. Michael Sipser, "*Introduction to Theory of Computation*", 3rd Edition, Course Technology, 2012.

CS204	DESIGN AND ANALYSIS OF ALGORITHMS	Credits 3-0-0: 3

Pre-requisites: CS101: Problem Solving and Computer Programming, CS152: Data Structures and Algorithms

Course Outcomes:

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

CO1	Analyze time and space complexities of algorithms.
CO2	Identify algorithm design methodology to solve problems.
CO3	Distinguish between P and NP classes of problems.
CO4	Design and analyze approximation algorithms for NP-hard problems.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	1	-	-	-	1	-	-	-	-	1	3	1	1	2
CO2	-	-	-	-	-	-	-	-	-	-	1	1	3	1	2	2
CO3	-	-	-	-	-	-	-	-	-	-	3	-	3	1	1	2
CO4	3	3	1	-	-	-	-	-	-	-	-	-	3	-	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Algorithm Analysis, Asymptotic Notations, Divide and Conquer – Master Theorem, Maximum Element in an Unimodal Array, Maximum Subarray Sum Problem, Expected Running Time of Randomized Quick Sort, Strassen's Matrix Multiplication Algorithm, Karatsuba's Large Integer Multiplication, Selection in Worst Case Linear Time, Dynamic Programming - Matrix Chain Multiplication Problem, Optimal Binary Search Tree, Rod-Cutting Problem, 0-1 Knapsack Problem, Travelling Salesman Problem, All-Pairs Shortest Paths Problem, Optimal Vertex Cover of a Tree, Greedy Method - Activity Selection Problem, Fractional Knapsack Problem, Correctness and Running Time Analysis of Prim's and Kruskal's Algorithms for Finding Minimum Spanning Tree, Complexity Classes - P, NP, NP-hard, NP-complete, Example NP-complete Problems – Clique, Independent Set, Vertex Cover, Approximation Algorithms - Vertex Cover Problem

Text Books / Reference Books / Online Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", Third Edition, PHI, 2009.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "*Fundamentals of Computer Algorithms*", Second Edition, Universities Press, 2011.
3. Michael R. Garey and David S. Johnson, "*Computers and Intractability: A Guide the theory of NP-Incompleteness*", W.H. Freeman & Co., 1979.
4. Herbert S. Wilf, "*Algorithms and Complexity*", AK Peters Ltd., 2003.
5. https://www.algorist.com/

CS205	OBJECT ORIENTED PROGRAMMING LABORATORY	Credits 0-1-2: 2

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Develop programs using objects and inheritance in Java Language.
CO2	Design and implement GUI programs using components in Java Language

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3	3	1	2	-	2	2	1	-	-	1	-	-	1
CO2	1	3	3	3	1	2	-	2	2	-	-	-	1	-	-	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Java Programming:
1. Ball games
2. Cannon game
3. Pinball game
4. Cards game
5. User interface dialogs related programs
6. I/O processing programs.

Text Books / Reference Books / Online Resources:

1. Timothy Budd, "*Object Oriented Programming with JAVA*", Updated Edition, Pearson Education, 2009.
2. Herbert Schildt, "*Java 2 Complete Reference*", TMH, 2010.

BT231	BIOENGINEERING	Credits 2-0-0: 2

Pre-Requisites: None

Course Outcomes:

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

CO1	Realize the significance of biomolecules for sustaining life
CO2	Identify the difference between unicellular to multi-cellular organisms
CO3	Understand heredity, variation and central dogma of life
CO4	Analyze and understand the concepts of biology for engineering the cell

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	1	1	-	2	2	-	-	1	-	1	1	1	-
CO2	2	2	1	1	-	2	2	-	-	1	-	1	1	1	-
CO3	2	2	1	1	-	2	2	-	-	1	-	1	1	1	-
CO4	2	2	1	1	-	2	2	-	-	1	-	1	1	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Molecules of Life, Water and Carbon: Chemical basis of life, protein structure and function, nucleic acids and the RNA world, carbohydrates, lipids, membranes and first cells.

Cell Structure and Function: Inside the cell, cell–cell interactions, cellular respiration and fermentation, photosynthesis, cell cycle, biological signal transduction.

Gene Structure and Expression: Mitosis, Meiosis, Mendel and the gene, DNA and the gene: synthesis and repair, how genes work, transcription, RNA processing, and translation, control of gene expression, analysing and engineering genes, genomics.

Engineering Concepts in Biology: Genetic engineering, disease biology, stem cell engineering, metabolic engineering, systems biology, biological data types; biocomputing; DNA based data storage, synthetic biology: synthetic gene regulatory circuit.

Text Books / Reference Books / Online Resources:

1. Biological Science, Quillin, Allison Scott Freeman, Kim Quillin and Lizabeth Allison, Pearson Education, 2016, 1st Edition.
2. Biotechnology for Beginners, Reinhard Renneberg, Viola Berkling and Vanya Loroch, Academic Press, 2017, 1st Edition.
3. https://nptel.ac.in/courses/121/106/121106008/

CS251	STATISTICAL FOUNDATIONS OF COMPUTER SCIENCE	Credits 2-0-2: 3

Pre-requisites: MA131-Mathematics I, MA181- Mathematics II

Course Outcomes:

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

CO1	Apply statistical methods to data for inferences.
CO2	Access online resources for R and import new function packages into the R workspace
CO3	Import, review, manipulate and summarize data-sets in R
CO4	Perform descriptive analytics over large scale data and apply appropriate statistical tests using R
CO5	Explore data-sets to create testable hypotheses and identify appropriate statistical tests

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	-	1	1	-	-	-	1	1	1	-	1	1	-	1
CO2	2	-	-	-	-	1	-	-	-	-	2	-	-	-	-	-
CO3	-	-	-	2	2	-	1	-	-	-	2	1	-	-	-	-
CO4	2	-	1	1	1	1	-	-	2	1	-	1	1	2	-	1
CO5	-	1	2	1	2	2	2	-	1	2	1	2	1	-	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Data Science and data visualization; Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes**.**

R Programming Structures, Control Statements, Loops, - Looping Over Nonvector Sets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion, A Quicksort Implementation-Extended Extended Example: A Binary Search Tree.

Doing Math and Simulation in R, Math Function, Extended Example Calculating Probability- Cumulative Sums and Products-Minima and Maxima- Calculus, Functions Fir Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product- Extended Example: Finding Stationary Distribution of Markov Chains, Set Operation, Input /output, Accessing the Keyboard and Monitor, Reading and writer Files,

Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot () Function –Customizing Graphs, Saving Graphs to Files.

Probability Distributions, Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,-ANOVA.

Linear Models, Simple Linear Regression, -Multiple Regression Generalized Linear Models, Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis, Nonlinear Models, Splines- Decision- Random Forests,

Text Books / Reference Books / Online Resources:

1. Norman Matloff, "*The Art of R Programming*", Cengage Learning
2. Lander, "*R for Everyone*", Pearson
3. Paul Teetor, "*R Cookbook*", Oreilly
4. Rob Kabacoff, "*R in Action*", Manning

CS252	MODELLING AND OPTIMIZATION TECHNIQUES	Credits 3-0-0: 3

Pre-requisites: CS101 – Problem Solving and Computer Programming, CS152: Data Structures and Algorithms

Course Outcomes:

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

CO1	Prepare and solve linear programming model.
CO2	Model transportation and flow through networks and compute optimal parameters
CO3	Optimize inventory levels
CO4	Solve real life problems using Meta-heuristic techniques
CO5	Generate random numbers and random variates
CO6	Verify and validate simulation models

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	-	-	-	1	-	-	-	-	1	1	1	1	2
CO2	3	3	3	2	2	-	1	-	-	-	1	1	1	2	2	2
CO3	2	2	2	1	1	-	1	-	-	-	3	-	2	1	1	2
CO4	2	2	2	1	2	-	1	-	-	-	2	1	2	1	1	2
CO5	2	2	2	1	1	-	-	-	-	-	-	1	1	1	2	-
CO6	2	2	1	-	-	-	-	-	-	-	-	-	2	2	1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Modelling with linear programming – The Simplex method, Sensitivity Analysis, Integer linear programming: Branch and Bound technique – Transportation Model and its variants, Network Model: CPM and PERT - Deterministic and non-deterministic inventory models.

Heuristic and Meta Heuristic Programming : Simulated Annealing, Genetic Algorithm, Particle warm Optimization algorithm and Teaching learning based optimization algorithm - Non Linear Programming algorithms.

Introduction to Quadratic Programming, Constrained Optimization Problem Solving, Convex Optimization Methods.

Simulation Modelling – Random number generation, Random variate generation – Verification and Validation of simulation models, Simulation of Computer Systems and Computer Networks.

Text Books / Reference Books / Online Resources:

1. Hamdy A Taha – "*Operations Research-An Introduction*", 9th Ed, Pearson, 2017 (Chs 1-8, 12, 14, 17)
2. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, "*Discrete Event Simulation*", 5th Ed, Pearson, 2010 (Chs 8 – 12, 14, 15)
3. NPTEL Video Lecturers for Meta Heuristic Techniques by Prof. Prakash Kotecha (IIT Guwahati)

CS253	COMPUTER ARCHITECTURE	Credits 3-0-0: 3

Pre-requisites: EC231-Analog and Digital Electronics

Course Outcomes:

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

CO1	Identify functional units, bus structure and addressing modes
CO2	Design the hardwired and micro-programmed control units.
CO3	Identify memory hierarchy and performance.
CO4	Design Arithmetic Logic Unit.
CO5	Interface I/O devices

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	1	2	1	-	-	1	-	-	-	1	-	2	1	1	2
CO2	2	1	2	-	-	-	1	-	-	-	-	-	2	1	1	1
CO3	2	1		-	1	1	-	-	-	-	-	-	2	2	1	1
CO4	2	1	2	-	-	1	1	-	-	-	-	-	2	1	1	1
CO5	1	1	2	1	-	-	1	-	-	-	1	-	1	1	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Basic Structures of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multicomputers, Historical Perspective

Machine instructions and Programs: Numbers, Arithmetic Operations and Characters, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes,

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, 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

Large Computer Systems: Forms of Parallel Processing, Array Processors, the Structure of General-Purpose Multiprocessors, Interconnection Networks

Text Books / Reference Books / Online Resources:

1. Carl Hamacher, "*Computer Organization*", 5th Edition, McGraw Hill Publishers, 2002.
2. Wiiliam Stallings, "*Computer Organization and Architecture Designing for Performance*", 8th Edition, Pearson Education, 2010
3. John P Hayes, "Computer Architecture and Organization", 3rd revised Ed., McGraw-Hill, 1998

CS254	DATABASE MANAGEMENT SYSTEMS	Credits 3-0-0: 3

Pre-requisites: CS152 – Data Structures and Algorithms

Course Outcomes:

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

CO1	Understand functional components of the DBMS.
CO2	Devise queries using Relational Algebra, Relational Calculus and SQL.
CO3	Design database schema.
CO4	Develop E-R model.
CO5	Evaluate and optimize queries.
CO6	Analyze transaction processing, concurrency control and recovery techniques.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1	1
CO2	2	1	-	-	2	1	-	1	-	2	2	1	-	1	-	-
CO3	1	2	3	2	2	1	2	1	-	2	2	-	1	2	2	2
CO4	1	2	3	2	2	1	2	1	-	2	2	-	1	2	2	2
CO5	2	-	-	1	-	-	1	-	-	1	-	-	1	2	-	-
CO6	-	1	2	1	1	-	1	1	-	1	2	-	-	2	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to DBMS: Historical perspective, File Versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS, Different Data Models;

Entity Relationship(ER) model: Features of ER model, conceptual design using ER model, design for large enterprises; Relational model–structure and operations, Integrity constraints over relations;

Query languages: Relational Algebra, Relational Calculus and SQL– Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, triggers and Embedded SQL;

Database Design: Mapping ER model to Relational form; Functional Dependency–Closer of functional dependencies, closer of attributes, canonical cover and Properties of Decompositions; Normalization process – 1NF, 2NF, 3NF and BCNF; Multivalued dependency– Closer properties of Multivalued dependency and 4NF; Join dependency– PJNF, Decomposition Algorithms;

Transaction Management: ACID properties, transactions, schedules and concurrent execution of transactions; Concurrency control – lock based protocol, Serializability, recoverability, dealing with deadlocks and Concurrency control without locking;

Query Processing: Overview of Query Evaluation, operator evaluation; Algorithms for relational operations– Selection operation, General selection condition, Projection operation, Join operation, set operation and aggregate operation, Evaluation of relational operations; Query optimization: Alternative plans, functions of query optimizer, translating SQL queries into relational algebra, estimating the cost of a plan, relational algebra equivalences, and other approaches to query optimization;

Database Recovery: Failure classification, Recovery and atomicity, Log-based recovery shadow paging and Advanced Recovery Techniques:

Security and Authorization: Access control, direct access control and Mandatory access control, Role of DBA, Application development.

Text Books / Reference Books / Online Resources:

1. Elamsri, Navathe, Somayajulu and Gupta, "*Fundamentals of Database Systems*", 6th Edition, Pearson Education, 2011.
2. Raghu Ramakrishnan, Johannes Gehrke, "*Database Management Systems*", 3nd Edition, McGraw Hill, 2003.
3. Silberschatz, Korth and Sudharshan, "*Database System Concepts*", 6th Edition, McGraw Hill, 2010.

CS255	LANGUAGE PROCESSORS	Credits 3-0-0: 3

Pre-requisites: CS203: Theory of Computation

Course Outcomes:

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

CO1	Understand phases in the design of compiler
CO2	Design top-down and bottom-up parsers
CO3	Identify synthesized and inherited attributes
CO4	Develop syntax directed translation schemes
CO5	Develop algorithms to generate code for a target machine

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	-	-	-	-	-	-	-	-	-	-	1	2	-	-
CO2	1	-	-	3	-	-	-	-	-	-	-	-	1	-	-	-
CO3	2	2	2	-	-	2	-	-	-	-	-	-	1	-	-	-
CO4	2	1	-	-	2	-	-	-	-	-	-	-	2	-	-	-
CO5	1	-	-	1	-	-	-	-	-	-	-	-	3	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

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.

Text Books / Reference Books / Online Resources:

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.

CS256	**MODELLING AND OPTIMIZATION TECHNIQUES LABORATORY**	**Credits 0-1-2: 2**

Pre-requisites: CS101 – Problem Solving and Computer Programming, CS152: Data Structures and Algorithms

Course Outcomes:

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

CO1	Implement linear programming model.
CO2	Implement transportation and network model
CO3	Design and implement inventory model
CO4	Design random number generator and random variate generator
CO5	Develop simulation models to solve real world problems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	1	-	-	-	1	-	-	-	-	1	1	1	1	2
CO2	2	3	2	2	2	-	1	-	-	-	1	1	1	2	2	2
CO3	2	2	2	1	1	-	1	-	-	-	2	-	2	1	1	2
CO4	2	2	2	1	1	-	-	-	-	-	-	1	1	1	2	-
CO5	1	2	1	-	-	-	-	-	-	-	-	-	2	1	1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Write a program to solve a problem using Simplex method
2. Write a program to solve a transportation problem.
3. Write a program to solve a network problem
4. Implement Inventory models
5. Write heuristic programming to solve real world problems
6. Write a program to solve travelling sales person problem using simulated annealing algorithm
7. Write a program to solve constraint optimization problem
8. Write a program to solve convex optimization problems
9. Write a program to design a pseudo random number generator using Linear congruential and combined congruential method
10. Write a program to design a random variate generator which generates random numbers following different distributions

Text Books / Reference Books / Online Resources:

3. Hamdy A Taha – “Operations Research-An Introduction”, 9th Ed, Pearson, 2017 (Chs 1-8, 12, 14, 17)
4. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, “Discrete Event Simulation 5th Ed, Pearson, 2010 (Chs 8 – 12, 14, 15)

CS257	DATABASE MANGEMENT SYSTEMS LABORATORY	Credits 0-1-3: 2.5

Pre-requisites: CS152-Data Structures and Algorithms

Course Outcomes:

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

CO1	Design and Implement a database schema
CO2	Devise queries using DDL, DML, DCL and TCL commands.
CO3	Develop application programs using PL/SQL
CO4	Design and implement a project using embedded SQL and GUI.
CO5	Apply modified components for performance tuning in open source software.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	3	2	1	-	1	-	-	1	2	2	2	3	2	2
CO2	2	-	2	1	2	-	2	1	1	1	1	-	-	1	1	-
CO3	2	-	2	1	2	-	2	1	1	1	1	3	3	2	-	3
CO4	-	1	2	1	2	-	1	-	-	2	2	2	2	2	1	2
CO5	1	2	1	1		-	-	1	-	1	1	2	2	1	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Familiarization of Oracle RDBMS, SQL*Plus and Oracle developer,

SQL: query-structure; DDL-create, alter, drop, rename and Truncate; DML-select, insert, update, delete and lock; Set operations- union, intersection and except; join; Aggregate Operations- group-by and having; nested sub-queries and views; DCL-grant and revoke, TCL-Commit, save point, rollback and set transaction.

PL/SQL: Environment, block structure, variables, operators, data types, control structures; Cursors structures- Implicit and Explicit; Bulk statements- Bulk collect into and forall; Exception handling- Compilation and Run-time, user-defined; Stored procedures- creation options, pass-by-value and functions-pass-by-value; Packages-package specification, body, package creation and usage; Triggers- Data definition language triggers, Data manipulation triggers, Compound

triggers and trigger restrictions; Large objects-CLOB, NCLOB, BLOB and BFILE; Implementation of applications using GUI; group project;

Text Books / Reference Books / Online Resources:

1. James, Paul and Weinberg, Andy Oppel, "*SQL: The Complete Reference*", 3rd Edition, McGraw Hill, 2011.
2. Michael McLaughlin, "*Oracle Database 11g PL/SQL Programming*", Oracle press.

CS258	LANGUAGE PROCESSORS LABORATORY	Credits 0-1-3: 2.5

Pre-requisites: CS203: Theory of Computation

Course Outcomes:

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

CO1	Implement simple lexical analyzers
CO2	Generate predictive parsing table for a CFG
CO3	Apply Lex and Yacc tools
CO4	Design and Implement LR parser
CO5	Implement Intermediate code generation for subset C language

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	-	-	-	-	-	-	-	-	-	-	-	2	2	-	1
CO2	1	1	-	2	-	-	-	-	-	-	-	-	2	2	-	1
CO3	3	-	-	2	2	-	-	-	-	-	-	-	2	2	-	1
CO4	2	-	3	2	-	-	-	-	-	-	-	-	2	2	-	1
CO5	1	-	-	-	-	-	-	-	-	-	-	-	2	2	-	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Lex and Yacc - Generation of Intermediate Code for Expression Grammar - Construction of Predictive Parsing Table - LR Parsing Tables - Parsing Actions.
2. Implement CYK algorithm (from Motwani's book)
3. Using lex/yacc tools generate assembly language code for a block of assignment and arithmetic statements.
4. Implement elimination of left recursion and left factoring algorithms for any given grammar and generate predictive parsing table.
5. Write a program for generating a parser program using lex and yacc for a language with integer identifiers, binary arithmetic expressions and assignments. (Input is grammar and output is parser in C language)
6. Write a program for generating SLR Parsing table and also write a parser.
7. Write a program for generating derivation sequence for a given terminal string using parsing table.
8. Using back-patching method generate three address code for while, if and Boolean expressions.
9. Major assignment: Intermediate code generation for subset C language.

Text Books / Reference Books / Online Resources:

1. Alfred V. Aho, Monical S. Lam, Ravi Sethi, and Jeffrey D. Ullman, "*Compilers - Principles, Techniques and Tools*", 2nd Edition, Pearson, 2007.
2. John R Levine, Tony Mason, Doug Brown, "*Lex and Yacc*", Orielly, 2nd Edition, 2009.

IC251	ENVORONMENTAL SCIENCE AND ENGINEERING	Credits 2-0-0: 2

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Identify environmental problems arising due to engineering and technological activities and the science behind those problems.
CO2	Estimate the population - economic growth, energy requirement and demand.
CO3	Analyse material balance for different environmental systems.
CO4	Realize the importance of ecosystem and biodiversity for maintaining ecological balance.
CO5	Identify the major pollutants and abatement devices for environmental management and sustainable development

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	-	-	2	3	1	-	-	-	-	-	1	2	-
CO2	3	3	3	-	-	2	3	1	-	-	-	-	-	1	2	-
CO3	3	3	3	-	-	2	3	1	-	-	-	-	-	1	2	-
CO4	3	3	3	-	-	2	3	1	-	-	-	-	-	1	2	-
CO5	3	3	3	-	-	2	3	1	-	-	-	-	-	1	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction to Environmental Science: Environment and society, major environmental issues: Ozone layer depletion, Acid rains, global climate change etc., sustainable development, Environmental impact assessment, environmental management
Natural Resources Utilization and its Impacts: Energy, minerals, water and land resources, Resource consumption, population dynamics, urbanization.
Ecology and Biodiversity: Energy flow in ecosystem, food chain, nutrient cycles, eutrofication, value of biodiversity, biodiversity at global, national and local levels, threats for biodiversity, conservation of biodiversity
Water Pollution: Sources, types of pollutants and their effects, water quality issues, contaminant transport, self-purification capacity of streams and water bodies, water quality standards, principles of water and wastewater treatment.
Air Pollution: Sources, classification and their effects, Air quality standards, dispersion of pollutants, control of air pollution, automobile pollution and its control.
Solid Waste Management: Sources and characteristics of solid waste, effects, Collection and transfer system, disposal methods

Text Books / Reference Books / Online Resources:

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. W P Cunningham, M A Cunningham, "*Principles of Environmental Science, Inquiry and Applications*", Tata McGraw Hill, 8th Edition, 2016.
4. M. Chandrasekhar, "*Environmental Science*", Hi Tech Publishers, 2009

CS301	OPERATING SYSTEMS	Credits 3-0-0: 3

Pre-requisites: CS253 - Computer Architecture, CS152- Data Structures and Algorithms

Course Outcomes:

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

CO1	Distinguish functional architectures of operating systems and file systems
CO2	Develop algorithms for subsystem components
CO3	Design device drivers and multi-threading libraries for an OS
CO4	Develop application programs using UNIX system calls
CO5	Design and solve synchronization problems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	3	2	-	2	-	2	2	2	2	2	3	-	2
CO2	2	2	3	3	2	-	2	-	2	2	2	2	2	3	-	2
CO3	2	2	2	3	2	-	2	-	2	2	2	2	2	3	-	2
CO4	3	2	2	3	2	-	2	-	2	2	2	2	1	3	-	-
CO5	2	2	2	3	1	-	2	-	2	2	2	2	2	3	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, UNIX system introduction and commands

Operating system structures: Computer system structure, Network structure, I/O Structure, Storage Structure, Dual mode operation, System components, Operating-System Services, System Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generation

Processes and Threads : Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Communication in Client – Server Systems, Multithreading Models, Threading Issues, Pthreads Basic Concepts,

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation, Process Scheduling Models

Process Synchronization: Synchronization Background, the Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Critical Regions, Monitors, OS Synchronization

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

Memory Management : Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations

File System: File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File System, NFS

I/O Systems : Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure , Disk Scheduling , Disk Management, Swap-Space Management, RAID Structure , Disk Attachment, Stable-Storage Implementation, Tertiary-Storage Structure

Text Books / Reference Books / Online Resources:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "*Operating System Principles*", Wiley, 8/e
2. Richard Stevens, Stephen Rago, "*Advanced Programming in the UNIX Environment*", Pearson Education, 2/e

CS302	**DATA WAREHOUSING AND DATA MINING**	**Credits 3-0-0: 3**

Pre-requisites: CS254 – Database Management Systems

Course Outcomes:

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

CO1	Understand stages in building a Data Warehouse
CO2	Apply preprocessing techniques for data cleansing
CO3	Analyze multi-dimensional modeling techniques
CO4	Analyze and evaluate performance of algorithms for Association Rules.
CO5	Analyze Classification and Clustering algorithms

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	3	2	1	-	1	-	-	1	2	2	1	2	3	2
CO2	2	-	2	1	2	-	2	1	1	1	1	-	2	1	3	1
CO3	2	-	2	1	2	-	2	1	1	1	1	3	2	1	3	1
CO4	-	1	2	1	2	-	1	-	-	2	2	2	3	1	3	1
CO5	1	2	1	1	-	-	-	1	-	1	1	2	2	1	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

KDD Process, Introduction to Data Warehouse, Data Preprocessing- Data Cleaning methods, Descriptive Data Summarization, Data Reduction, Data Discretization and Concept hierarchy generation, Overview of ETL and OLAP OLTP integration – comparison of OLAP with OLTP systems, ROLAP, MOLAP and DOLAP, Data Cube Computation methods, Advanced SQL support for OLAP, multi-dimensional modeling, Attribute-oriented Induction, Data Warehouse architecture and implementation - Parallel execution, Materialized views.

Data Mining Techniques: Basic concepts of Association Rule Mining, Frequent Item set mining, Mining various kinds of association rules, Classification by decision tree induction, Bayesian Classification, Rule-based Classification, Classification Back-propagation, Associative Classification, Lazy Learners, Rough set approach, Clustering methods, Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Measuring Data Similarity and Dissimilarity Partition based Clustering, Hierarchical based clustering, Density based clustering.

Text Books / Reference Books / Online Resources:

1. Jiawei Han and M Kamber, "*Data Mining Concepts and techniques*", Third Edition, Elsevier Publications, 2011; chapters 1-8
2. Data Warehouse Video Content
3. NMIECT – Data Warehousing Video Content

CS303	SOFTWARE ENGINEERING	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Comprehend software development life cycle
CO2	Prepare SRS document for a project
CO3	Apply software design and development techniques
CO4	Identify verification and validation methods in a software engineering project
CO5	Implement testing methods for software
CO6	Analyze and Apply project management techniques for a case study

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	1	1	2	1	-	2	2	2	-	1	2	1	2
CO2	2	1	2	-	-	2	2	-	2	2	-	-	-	1	-	2
CO3	2	2	2	1	-	1	1	-	-	2	1	-	2	2	1	2
CO4	2	2	2	-	2	2	2	-	2	2	1	-	-	1	-	3
CO5	2	2	2	-	2	2	2	-	2	2	1	-	1	2	1	2
CO6	2	2	2	1	1	1	1	-	1	2	2	1	1	2	-	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The Software Problem - Cost, Schedule, and Quality, Scale and Change ; Software Processes- Process and Project , Component Software Processes; Software Development Process Models - Waterfall Model, Prototyping , Iterative Development , Rational Unified Process , Time boxing Model , Extreme Programming and Agile Processes , Using Process Models in a Project. Software Requirements Analysis and Specification - Value of a Good SRS , Requirement Process , Requirements Specification; Formal Specification- Formal Specification in the Software process, Sub-system interface specification, Behavioural Specification; Desirable Characteristics of an SRS - Components of an SRS, Structure of a Requirements Document; Functional Specification with Use Cases - Basics , Examples , Extensions, Developing Use Cases; Other Approaches for Analysis - Data Flow Diagrams , ER Diagrams , Validation; Software Architecture - Role of Software Architecture, Architecture Views - Component and Connector View - Components, Connectors, An Example. Architecture Styles for C&C View - Pipe and Filter, Shared-Data Style , Client-Server Style, Some Other Styles, Documenting Architecture Design - Evaluating Architectures; Design - Design Concepts - Coupling , Cohesion , The Open-Closed Principle . Function-Oriented Design (from Pressman) - Structure Charts, Structured Design Methodology, An Example. Object-Oriented Design (from Jalote)- OO Concepts, Unified Modeling Language (UML) , A Design Methodology , Examples; Detailed Design - Logic/Algorithm Design, State Modeling of Classes; Verification - Metrics - Complexity Metrics for Function-Oriented Design, Complexity Metrics for OO Design; Coding and Unit Testing -Programming Principles and Guidelines - Structured Programming , Information Hiding, Some Programming Practices, Coding Standards; Incrementally Developing Code - An Incremental Coding Process ,Test-Driven Development, Pair Programming; Managing Evolving Code - Source Code Control and Build, Refactoring; Unit Testing - Testing Procedural Units, Unit Testing of Classes; Code Inspection - Planning, Self-Review, Group Review Meeting; Metrics - Size Measures, Complexity Metrics; Testing - Testing Concepts - Error, Fault, and Failure, Test Case, Test Suite, and Test Harness , Psychology of Testing , Levels of Testing; Testing Process - Test Plan, Test Case Design, Test Case Execution; Black-Box Testing - Equivalence Class Partitioning, Boundary Value Analysis , Pairwise Testing, Special Cases, State-Based Testing; White-Box Testing - Control Flow-Based Criteria, Test Case Generation and Tool Support; Metrics - Coverage Analysis, Reliability, Defect Removal Efficiency.

Text Books / Reference Books / Online Resources:

1. Pankaj Jalote, "*Software Engineering Precise Approach*" , Wiley Publishers, 2012
2. Ian Sommerville, "*Software Engineering*", *8/e* Pearson Publishers, 2012.
3. Roger Pressman, "*Software Engineering*", 5th edition, MCgrawHill, 2002

CS304	**COMPUTER NETWORKS**	**Credits 3-0-0: 3**

Pre-requisites: CS301-Operating Systems
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand OSI and TCP/IP models
CO2	Analyze MAC layer protocols and LAN technologies
CO3	Design applications using internet protocols
CO4	Implement routing and congestion control algorithms
CO5	Develop application layer protocols

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	-	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-
CO3	3	2	3	-	3	-	-	-	-	-	-	2	2	2	-	2
CO4	1	2	2	1	-	-	-	-	-	-	-	-	2	2	-	-
CO5	3	3	3	-	3	-	1	1	-	-	-	2	3	3	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

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-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control
Network layer – network layer design issues - Routing algorithms - Congestion control algorithms – Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing
Transport layer - Elements of transport protocol - Congestion control – The Internet’s Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD sockets - client-server applications
Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol - World wide web - Hypertext transfer protocol -Presentation formatting and data compression-
Introduction to Network security - Web Services architectures for developing new application protocols.

Text Books / Reference Books / Online Resources:

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, 2010.

CS305	OPEARING SYSTEMS LABORATORY	Credits 0-1-2: 2

Pre-requisites: CS152- Data Structures and Algorithms

Course Outcomes:

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

CO1	Implement elementary UNIX system commands
CO2	Develop programs to test synchronization problems
CO3	Design and develop user level thread library
CO4	Design and implement file system.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2	2	-	2	-	2	2	2	2	-	2	-	-
CO2	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2
CO3	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2
CO4	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

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. 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 Sales Person problem)
4. 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.
5. Implement all CPU Scheduling Algorithms using your thread library
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk image file.

Text Books / Reference Books / Online Resources:

1. Richard Stevens, Stephen Rago, "*Advanced Programming in the UNIX Environment*", Pearson Education, 2/e

CS306	KNOWLEDGE ENGINEERING LABORATORY	Credits 0-0-2: 1

Pre-requisites: None

Course Outcomes:

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

CO1	Build data cubes with SQL
CO2	Implement data preprocessing techniques on data.
CO3	Implement OLAP operations and multi-dimensional modeling
CO4	Implement data mining algorithms

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	1	1	2	1	-	2	2	2	-	3	2	3	2
CO2	2	1	2	-	-	2	2	-	2	2	-	-	2	-	3	2
CO3	2	2	2	1	-	1	1	-	-	2	1	-	2	-	3	2
CO4	2	2	2	-	2	2	2	-	2	2	1	-	3	2	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Advanced SQL Analytic functions
2. Implementation of OLAP operations
3. Data preprocessing techniques
4. Cube computation methods
5. Concept hierarchy method
6. Write a program in any programming language to generate at least 10,000 transactions in a text file with at least three items.
7. Write a program to implement the *APRIORI* algorithm and also test it thoroughly.
8. Write a program for each of the following to improve *APRIORI*
   A. Hash based Technique. B. Dynamic Item set Counting Algorithm. C. Partition Based Approach.
9. Write a program for *FPGROWTH* algorithm and also test it.
10. Write a program to construct an optimized DECISION TREE for a given training data and by using any attribute selection measure.
11. Write a program for NAÏVE BAYESIAN algorithm for classifying the data.
12. Implement the K-Means Clustering algorithm for clustering the given data.

Text Books / Reference Books / Online Resources:

1. Jiawei Han and M Kamber, "*Data Mining Concepts and techniques*", Third Edition, Elsevier Publications, 2011

CS307	CASE TOOLS LABORATORY	Credits 0-1-2: 2

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Prepare Software Requirement Specification document
CO2	Prepare design document and compute effort estimates for a software project
CO3	Design UML diagram for a case study
CO4	Design and Develop test cases for a software

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	1	2	-	-	2	2	-	2	2	-	-	2	1	-	2
CO2	2	2	2	1	-	1	1	-	-	2	1	-	2	2	-	2
CO3	2	2	2	-	2	1	1	-	1	2	1	-	2	2	-	2
CO4	2	2	2	-	2	2	2	-	2	2	1	-	1	1	1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
1. Develop a Software Project Plan Using Microsoft Project (Planning involves estimation – work break down structure – how much money- how much effort – how many resources – how much time it will take to a specific software –based system or product)
2. Develop a SRS Document using Rational Requisite Pro Tool. (This Lab is for mastering the software requirements in this regard the documents like Vision Document- Use Case Document – SRS Documents must be submitted for the Problem given to you)
3. Designing a Software using UML – Tool used is Rational Rose- In this lab, the student is supposed to solve the problem using OO analysis and design Methodology
4. Testing using tools Rational Robot, Rational Purify, Rational Quantify, Rational Pure Coverage etc., (Functional and Structural Testing techniques were discussed and necessary programs will be tested)
5. Writing a programs for the following : Quality Metrics and OO Metrics, Finding the coupling and cohesion intensity in java code, Reverse Engineering Problems
6. Web site Testing , Security Testing , System Testing

Text Books / Reference Books / Online Resources:

1. Rational Online Documentation
2. Booch, Jackobson and Rambaugh, "*UML Guide*" , Pearson Edu, 1999
3. IEEE Standards for SRS Documents, IEEE Std. 830.
4. Fenton NE, "*Software Metrics: A Rigorous Approach*", Chapman and Hall, 1991

CS308	COMPUTER NETWORKS LABORATORY	Credits 0-1-2: 2

Pre-requisites: CS305: Operating Systems Laboratory

Course Outcomes:

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

CO1	Develop programs for client-server applications
CO2	Perform packet sniffing and analyze packets in network traffic.
CO3	Implement error detecting and correcting codes

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	3	-	2	-	2	-	-	-	-	2	2	2	-	-
CO2	2	3	2	2	1	1	-	2	-	-	-	1	1	2	-	2
CO3	1	1	1	2	-	-	-	-	-	-	-	-	2	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
1. Programs to implement error detection and correction
2. Client-Server applications using inter process communication mechanisms a) FIFO b) Message queues c) Shared memory
3. Connection-oriented Client-Server applications based on BSD sockets
4. Connectionless Client-Server applications
5. Implementation of Chat servers and mail Servers
6. Implementation of routing algorithms
7. Programs using Remote Procedure Call (RPC)
8. Client-Server applications based on Raw Sockets, IP Spoofing
9. Implementation of application layer protocols
10. Datalink layer Access, Packet Sniffing

Text Books / Reference Books / Online Resources:

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.*

CS311	**WEB TECHNOLOGIES**	**Credits 2-0-2: 3**

Pre-requisites: None

Course Outcomes:

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

CO1	Understand, analyze and build dynamic and interactive web sites
CO2	Understand current and evolving Web languages for integrating media and user interaction in both front end and back end elements of a Web site
CO3	Analysis and reporting of web data using web analytics
CO4	Applying different testing and debugging techniques and analyzing the web site effectiveness.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	2	1	2	1	-	-	-	2	2	1	2	3	1	1
CO2	1	1	-	-	-	-	2	1	1	-	1	1	-	2	1	-
CO3	1	1	-	2	3	-	-	1	-	1	1	-	2	1	1	-
CO4	1	-	-	1	-	-	-	1	-	1	-	-	-	2	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to XHTML: Editing XHTML, First XHTML Example, W3C XHTML Validation Service, Headers, Linking, Images, Special Characters and More Line Breaks, Unordered Lists, Nested and Ordered Lists, Internet and World Wide Web Resources;

Dynamic HTML: Object Model and Collections- Introduction, Object Referencing, Collections all and children, Dynamic Styles, Dynamic Positioning, Using the frames Collection, navigator Object, Summary of the DHTML Object Model; Event Model- vent onclick, Event onload, Error Handling with onerror, Tracking the Mouse with Event onmousemove, Rollovers with onmouseover and onmouseout; Form Processing- Form Processing with onfocus and onblur, More Form Processing with onsubmit and onreset, Event Bubbling, More DHTML Events; Filters and transitions; Data binding with tabular data control, Structured graphics and active X control;

JavaScript: Functions; Program Modules in JavaScript, Programmer Defined Functions, Function Definitions, Random-Number Generation, Duration of Identifiers, Scope Rules, JavaScript Global Functions, Recursion, JavaScript arrays, JavaScript objects;

XML: Structuring Data, XML Namespaces, Document Type Definitions (DTDs) and Schemas, Document Type Definitions, W3C XML Schema Documents, XML Vocabularies, Chemical Markup Language (CML), Other Markup Languages, Document Object Model (DOM), DOM Methods, Simple API for XML (SAX), Extensible Style sheet Language (XSL), Simple Object Access Protocol (SOAP);

Web Servers: HTTP Request Types, System Architecture, Client-Side Scripting versus Server-Side Scripting, Accessing Web Servers, Microsoft Internet Information Services (IIS), Microsoft Personal Web.

Server-side Scripting: Introduction to PHP, String Processing and Regular Expressions, Form processing and Business logic, Dynamic content, Database connectivity, Applets and Servlets, JDBC connectivity, JSP and Web development Frameworks.

Text Books / Reference Books / Online Resources:

1. Deitel, Deitel and Nieto, "*Internet and Worldwide Web - How to Program*", 5th Edition, PHI, 2011.
2. Bai and Ekedhi, "*The Web Warrior Guide to Web Programming*", 3rd Edition, Thomson, 2008.

CS312	GRAPH ALGORITHMS	Credits 3-0-0: 3

Pre-requisites: Discrete Mathematics, Data Structures and Algorithms, Design and Analysis of Algorithms

Course Outcomes:

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

CO1	Analyze time and space complexities of algorithms using asymptotic analysis.
CO2	Formulate and solve graph problems.
CO3	Identify algorithm design methodology to solve problems.
CO4	Design efficient polynomial time algorithms for restricted classes of intractable problems.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	1	-	-	-	1	-	-	-	-	1	3	1	1	2
CO2	3	2	2	2	-	-	-	-	-	-	-	-	2	-	1	-
CO3	-	-	-	-	-	-	-	-	-	-	1	1	3	1	2	2
CO4	3	2	2	2	-	-	-	-	-	-	-	-	2	-	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Analysis of algorithms, Degree sequences, Havel–Hakimi algorithm, Erdös-Gallai theorem, Cheriton-Tarjan algorithm for minimum spanning tree, Connectivity, Blocks, Algorithms for finding the blocks of a graph, Matching, Hungarian algorithm for maximum matching, Perfect matchings and 1-factorizations, Network flows, Ford-Fulkerson algorithm, Matchings and flows, Hamiltonian cycle, Euler cycle, Chinese postman problem, Strong components, Tournaments, 2-Satisfiability, Perfect graphs, Greedy graph coloring algorithm, Some special classes of graphs, Algorithms for recognition of chordal graphs, Unit disk graphs, Dominating sets, Complexity of dominating and connected dominating sets, Algorithms for dominating sets, Hardness of some graph problems, Algorithms for independent sets and cliques, Random graphs, Social network models.

Text Books / Reference Books / Online Resources:

1. Martin Charles Golumbic, "*Algorithmic Graph Theory and Perfect Graphs*", Academic Press, 1980.
2. M.E.J. Newman, *"Networks: An Introduction"*, Oxford University Press, 2010.
3. Teresa W. Haynes, Stephen T. Hedetniemi and Peter J. Slater, "*Fundamentals of Domination in Graphs*", Marcel Dekker, Inc., 1998.
4. William Kocay and Donald L. Kreher, "*Graphs, Algorithms, and Optimization*", CRC Press, 2005.

CS313	ADVANCED DATA STRUCTURES	Credits 3-0-0: 3

Pre-requisites: CS152 – Data Structures and Algorithms

Course Outcomes:

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

CO1	Understand implementation of symbol table using hashing techniques
CO2	Develop and analyze algorithms for red-black trees, B-trees and Splay trees
CO3	Develop algorithms for text processing applications
CO4	Identity suitable data structures and develop algorithms for computational geometry problems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	1	1	1	-	-	-	-	-	-	-	2	1	-	-
CO2	2	1	1	1	-	-	-	-	-	-	-	-	2	1	-	-
CO3	3	2	2	1	2	-	-	-	-	-	-	-	3	1	2	1
CO4	3	3	2	1	1	-	-	-	-	-	-	-	3	1	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries.

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.

Skip Lists: Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists

Trees: Binary Search Trees (BST), AVL Trees

Red Black Trees: Height of a Red Black Tree, Red Black Trees Bottom-Up Insertion, Top-Down Red Black Trees, Top-Down Deletion in Red Black Trees, Analysis of Operations.

2-3 Trees: Advantage of 2-3 trees over Binary Search Trees, Search and Update Operations on 2-3 Trees, Analysis of Operations.

B-Trees: Advantage of B- trees over BSTs, Height of B-Tree, Search and Update Operations on 2-3 Trees, Analysis of Operations.

Splay Trees: Splaying, Search and Update Operations on Splay Trees, Amortized Analysis of Splaying.

Text Processing: String Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.

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

Text Books / Reference Books / Online Resources:

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.

CS314	PROGRAMMING LANGUAGE CONCEPTS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understanding the concepts of evolution of programming languages.
CO2	Understanding the concepts of object oriented languages, functional and logical programming languages
CO3	Analyzing the methods and tools to define syntax and semantics of a languages
CO4	Analyzing the design issues involved in various constructs of programming languages
CO5	Apply the concepts and identify the issues involved in other advanced features of programming languages

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	1	2	1	1	1	1	1	1	1	1	2	1	3	3
CO2	1	2	2	3	2	1	1	1	1	1	1	1	2	1	3	3
CO3	3	3	2	3	1	1	1	1	2	1	1	1	2	1	3	3
CO4	2	3	1	3	3	2	1	1	1	1	1	1	2	2	2	3
CO5	2	2	1	2	2	2	1	1	1	1	1	1	2	1	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction- The Origins of Programming Languages- Abstractions in Programming Languages - Computational Paradigms -Language Definition - Language Translation

Language Design Criteria – Efficiency, regularity, security and extensibility - C++: An Object-Oriented Extension of C-Python: A General-Purpose Scripting Language –

Functional Programming - Programs as Functions - Scheme: A Dialect of Lisp - ML: Functional Programming with static typing -Delayed Evaluation- Haskell- Overloading.

Logic Programming-Logic and Logic Programs - Horn Clauses -Resolution and Unification- The Language Prolog - Problems with Logic Programming

Object-Oriented Programming- Software Reuse and Independence Smalltalk Java C++ - Design Issues in Object-Oriented Languages - Implementation Issues in Object-Oriented

Languages

Syntax-Lexical Structure of Programming Languages -Context-Free Grammars and BNFs -Parse Trees and Abstract Syntax Trees - EBNFs and Syntax Diagrams - Parsing Techniques and Tools-Lexics vs. Syntax vs. Semantics

Basic Semantics -Attributes, Binding, and Semantic Functions - Declarations, Blocks, and Scope - The Symbol Table - Name Resolution and Overloading - Allocation, Lifetimes, and the Environment Variables and Constants Aliases, Dangling References, and Garbage

Data Types-Data Types and Type Information - Simple Types - Type Constructors - Type Nomenclature in Sample Languages -Type Equivalence- Type Checking -Type Conversion-Polymorphic Type Checking- Explicit Polymorphism

Control Expressions and Statements –Expressions - Conditional Statements and Guards- Exception Handling- Procedure Definition and Activation-Procedure Semantics- Parameter-Passing Mechanisms-Procedure Environments, Activations, and Allocation-Dynamic Memory Management- Exception Handling and Environments

Abstract Data Types and Modules - The Algebraic Specification of Abstract Data Types- Abstract Data Type Mechanisms and Modules -Separate Compilation in C, C++ Namespaces, and Java Packages-Ada Packages -Modules in ML- Problems with Abstract Data Type Mechanisms

Formal Semantics- A Sample Small Language- Operational Semantics -Denotational Semantics-Axiomatic Semantics- Proofs of Program Correctness-

Parallel Programming- Introduction to Parallel Processing- Parallel Processing and Programming Languages- Threads – Semaphores- Monitors –Message Passing.

Text Books / Reference Books / Online Resources:

1. Kenneth C. Louden, "*Programming Language Principles and Practices*", 2nd Edition, Thomson 2003.
2. Carlo Ghezzi, Mehdi Jazayeri, "*Programming Language Concepts*", 3rd Edition, John Wiley & Sons, 1997.

CS315	ARTIFICIAL INTELLIGENCE	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Solve searching problems using A*, Mini-Max algorithms.
CO2	Create logical agents to do inference using first order logic.
CO3	Understand Bayesian Networks to do probabilistic reasoning.
CO4	Perform Statistical learning using EM algorithm.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	2	1	1	-	1	-	1	1	1	-	2	1	2	-
CO2	2	2	1	1	1	-	-	-	-	1	1	-	2	1	2	-
CO3	2	2	2	2	1	1	1	-	-	1	1	1	2	1	3	-
CO4	2	2	2	2	1	1	1	-	-	1	1	1	2	1	3	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

INTRODUCTION – Agents and Objects – Evaluation of Agents – Agent Design Philosophies - Multi-agent System – Mobile Agents – Agent Communication – Knowledge query and Manipulation Language – Case Study. What is AI? , The Foundations of Artificial Intelligence; - INTELLIGENT AGENTS – Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents; - SOLVING PROBLEMS BY SEARCH – Problem-Solving Agents, Formulating problems, Searching for Solutions, 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; - LOGICAL AGENTS – Knowledge-Based agents, The Wumpus World, Logic, Propositional Logic: A Very Simple Logic, Reasoning Patterns in Propositional Logic, Resolution, Forward and Backward chaining; - FIRST ORDER LOGIC – Syntax and Semantics of First-Order Logic, Using First-Order Logic , Knowledge Engineering in First-Order Logic; - INFERENCE IN FIRST ORDER LOGIC – Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution ; - UNCERTAINTY – Acting under Uncertainty, Basic Probability Notation, The Axioms of Probability, 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; STATISTICAL LEARNING METHODS – Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: EM Algorithm.

Text Books / Reference Books / Online Resources:

1. Stuart Russell, Peter Norvig, "*Artificial Intelligence -A Modern Approach"*, 3/e, Pearson, 2003.
2. Nils J Nilsson, "*Artificial Intelligence: A New Synthesis"*, Morgan Kaufmann Publications, 2000.

CS316	ADVANCED COMPUTER ARCHITECTURE	Credits 3-0-0: 3

Pre-requisites: Computer Architecture, Microprocessors

Course Outcomes:

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

CO1	Measure the performance of a computer system
CO2	Apply instruction scheduling techniques to improve the performance of a code segment
CO3	Understand the working of shared and distributed memory systems and cache coherence protocols
CO4	Evaluate the I/O system performance

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	2	1	1	-	1	-	1	1	1	-	2	1	2	-
CO2	2	2	1	1	1	-	-	-	-	1	1	-	2	1	2	-
CO3	2	2	2	2	1	1	1	-	-	1	1	1	2	1	3	-
CO4	2	2	2	2	1	1	1	-	-	1	1	1	2	1	3	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Fundamentals of Computer Design: Trends in technology, power, integrated circuits, cost. Dependability, measuring the performance. Pipelining basic concepts, pipeline hazards and other issues, Memory hierarchy, Cache performance, virtual memory, memory protection

Instruction level parallelism: Basic compiler techniques, reducing branch costs, overcoming data hazards, multiple issue and static scheduling, dynamic scheduling, speculation, Limitations of instruction level parallelism, Multithreading and thread level parallelism, VLIW and Vector processors.

Multiprocessors and Thread-level parallelism: Symmetric shared memory architectures, performance of Symmetric shared memory multiprocessor, Distributed shared memory and directory based coherence, Synchronization, Memory consistency

Advanced memory hierarchy design: optimizations of cache performance, memory technology optimizations, protection, virtual memory and virtual machines, case studies

Storage Systems: Advanced topics in disk storage, real faults and failures, I/O performance, Reliability measures, design and evaluation of I/O system

Text Books / Reference Books / Online Resources:

1. John L. Hennessy and David A. Patterson, "*Computer Architecture: A Quantitative Approach*" 4th Edition, Elsevier, 2007
2. Kai Hwang, "*Advanced Computer Architecture*", Second Edition, Tata McGraw Hill, 2008

CS317	COMPUTATIONAL NUMBER THEORY	Credits 3-0-0: 3

Pre-requisites: CS204-Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Analyse large integer computations in Zn
CO2	Analyse primality testing and integer factorization algorithms
CO3	Develop algorithms for computations in groups, rings and fields
CO4	Develop algorithms for computations in polynomial rings
CO5	Develop algorithms for computations in finite fields

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	2	1	1	-	-	-	-	1	-	-	-	1	-	-
CO2	2	3	2	2	3	-	-	1	-	1	-	1	-	-	2	-
CO3	2	1	3	2	2	1	1	-	2	-	1	1	-	1	-	1
CO4	2	2	2	1	2	1	1	-	1	1	-	-	2	-	-	1
CO5	2	3	3	3	3	1	-	1	1	-	1	1	1	-	-	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Basic properties of the integers - Divisibility and primality, Ideals and greatest common divisors, unique factorization, Congruences - Basic properties, Solving linear congruences, Residue classes, Euler’s phi function, Fermat’s little theorem, Arithmetic functions and Mobius inversion

Computing with large integers - Asymptotic notation, Machine models and complexity theory, Basic integer arithmetic, Computing in Zn, Faster integer arithmetic; Euclid’s algorithm - basic Euclidean algorithm, extended Euclidean algorithm, Computing modular inverses and Chinese remaindering, Speeding up algorithms via modular computation, Rational reconstruction and applications

The distribution of primes - Chebyshev’s theorem on the density of primes, Bertrand’s postulate, Mertens’ theorem, sieve of Eratosthenes, prime number theorem and beyond

Probabilistic algorithms – definitions, Approximation of functions, generating a random number from a given interval, generating a random prime, generating a random non-increasing sequence, generating a random factored number, RSA cryptosystem

Algebraic Structures - Subgroups, Cosets and quotient groups, Group homomorphisms and isomorphisms, Cyclic groups, structure of finite abelian groups, Rings - Definitions, basic properties, and examples, Polynomial rings, Ideals and quotient rings, Ring homomorphisms
and isomorphisms; Modules and vector spaces - Submodules and quotient modules, Module homomorphisms and isomorphisms, Linear independence and bases, Vector spaces and dimension; Matrices - linear maps, inverse of a matrix, Gaussian elimination, Applications of Gaussian elimination; Algebras - The field of fractions of an integral domain, Unique factorization of polynomials, Polynomial congruences, Polynomial quotient algebras, General properties of extension fields, Formal power series and Laurent series, Unique factorization domains.

Primality testing - Trial division, structure of Z_n, The Miller–Rabin test, Generating random primes using the Miller–Rabin test, Perfect power testing and prime power factoring, Factoring and computing Euler’s phi function; Deterministic primality testing - The algorithm and its analysis

Finding generators and discrete logarithms - Finding a generator for Z_p, Computing discrete logarithms Z_p, The Diffie–Hellman key establishment protocol

Quadratic residues and quadratic reciprocity - Quadratic residues, Legendre symbol, Jacobi symbol; Computational problems related to quadratic residues - Computing the Jacobi symbol, Testing quadratic residuosity, Computing modular square roots, The quadratic residuosity assumption.

Subexponential-time discrete logarithms and factoring - Smooth numbers, algorithm for discrete logarithms, algorithm for factoring integers, Practical improvements,

Polynomial arithmetic and applications - Basic arithmetic, Computing minimal polynomials, Euclid’s algorithm, Computing modular inverses and Chinese remaindering, Rational function

reconstruction and applications, Faster polynomial arithmetic; Linearly generated sequences and applications - Basic definitions and properties, Computing minimal polynomials, Solving sparse linear systems, The algebra of linear transformations,
Finite fields - The existence of finite fields, The subfield structure and uniqueness of finite fields, Conjugates, norms and traces; Algorithms for finite fields - Testing and constructing irreducible polynomials, Computing minimal polynomials in F[X]/(f), Factoring polynomials: the Cantor–Zassenhaus algorithm, Factoring polynomials: Berlekamp's algorithm, Deterministic factorization algorithms, Faster square-free decomposition.

Text Books / Reference Books / Online Resources:

1. Victor Shoup, "*A Computational Introduction to Number Theory and Algebra"*, Cambridge University Press, 2008
2. Henri Cohen, "*A Course in Computational Algebraic Number Theory"*, Springer-Verlag, 2000
3. Abhijit Das, "*Computational Number Theory*", Cambridge University Press, 2013
4. Eric Bach and Jeffrey Shallit, "*Algorithmic Number Theory, Volume* 1: *Efficient Algorithms"*, MIT Press, 1996
5. J. P. Buhler, P. Stevenhagen, "*Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography"*, Cambridge University Press, 2008

CS318	GAME THEORY	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Analyze games based on complete and incomplete information about the players
CO2	Analyze games where players cooperate
CO3	Compute Nash equilibrium
CO4	Apply game theory to model network traffic
CO5	Analyze auctions using game theory

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	1	1	1	-	1	-	-	1	-	-	3	1	1	-
CO2	2	3	2	2	3	-	-	1	-	1	-	1	2	2	1	-
CO3	2	1	3	2	2	1	1	-	2	-	1	1	2	2	2	1
CO4	2	2	2	1	2	1	1	-	1	1	-	-	2	1	2	1
CO5	2	3	2	3	3	1	-	1	1	-	-	1	2	2	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Noncooperative Game Theory: Games in Normal Form - Preferences and utility, examples of normal-form, Analyzing games: Pareto optimality, Nash equilibrium, Maxmin and minmax strategies, dominated strategies, Rationalizability, Correlated equilibrium

Computing Solution Concepts of Normal-Form Games: Computing Nash equilibria of two-player, zero-sum games, Computing Nash equilibria of two-player, general-sum games, Complexity of computing Nash equilibrium, Lemke–Howson algorithm, Searching the space of supports, Computing Nash equilibria of n-player, general-sum games, Computing maxmin and minmax strategies for two-player, general-sum games, Computing correlated equilibria

Games with the Extensive Form: Perfect-information extensive-form games, Subgame-perfect equilibrium, Computing equilibria, Imperfect-information extensive-form games, Sequential equilibrium

Other Representations: Repeated games: Finitely repeated games, Infinitely repeated games, automata, Stochastic games Bayesian games: Computing equilibria

CoalitionalGameTheory: Transferable Utility, Analyzing Coalitional Games, Shapley Value, the Core

Mechanism Design: Strategic voting, unrestricted preferences, Implementation, quasilinear setting, efficient mechanisms, and Computational applications of mechanism design, Task scheduling, Bandwidth allocation in computer networks

Auctions: Single-good auctions, Canonical auction families, Bayesian mechanisms, Multiunit auctions, combinatorial auctions,

Text Books / Reference Books / Online Resources:

1. Shoham, Y. and Leyton–Brown, K., "*Multiagent Systems: Algorithmic, Game Theoretic, and Logical Foundations*". Cambridge University Press, 2008.
2. Osborne, M. J., and Rubinstein, A., "*A Course in Game Theory", Cambridge*, MA: MIT Press, 1994.
3. D. Fudenberg and J. Tirole, "*Game Theory*", The MIT Press, 2005

CS351	MOBILE COMPUTING	Credits 3-0-0: 3

Pre-requisites: CS304-Computer Networks

Course Outcomes:

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

CO1	Identify mobile computing societal applications and communication constraints in wireless environment
CO2	Analyze mobile IPv4 and IPv6 architectures with agents and proxies.
CO3	Design MAC protocols for wireless networks.
CO4	Evaluate the performance of TCP protocols in Wireless Networks with mobile nodes.
CO5	Design and analyze the existing routing protocols for multi-hop wireless networks.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	2	2	2	-	-	-	1	1	2	2	1	1	-	2
CO2	1	2	2	2	2	-	-	-	1	1	2	2	1	2	-	2
CO3	2	2	3	2	3	-	2	-	1	1	2	2	2	3	-	3
CO4	2	2	3	2	3	-	2	-	1	1	2	2	1	3	-	3
CO5	2	2	3	2	3	-	2	-	1	1	2	2	2	3	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Basic communication Technologies, Introduction to Mobile Networks, Introduction to different categories of Wireless networks (MANET: Mobile ad-hoc networks- Communication Architectures of a typical MANET, Applications of MANET, WSN: Wireless Sensor Networks- topologies in WSN- Linear, Grid and Cluster based topologies, communication architectures in a WSN, applications of WSNs, VANET: Vehicular Ad-hoc Networks- communication architectures in VANET, Applications of VANET, PAN: Personal Area Networks- the Bluetooth technology, the blue tooth specifications, DTN: Delay Tolerant Network-delay tolerant network architecture, applications of DTN), Wireless Communication Fundamentals, Cellular Wireless Networks.

Medium Access Control Layer- Hidden terminal problem, Exposed terminal problem, Collision avoidance, Congestion Avoidance, Congestion control, Energy Efficiency, MACA and MACAW protocols, Wireless LAN and IEEE 802.11- Network architecture, the physical layer, the MAC layer, security.

Detailed network layer functionalities in multi-hop wireless networks- Mobile Ad-hoc Networks- broadcasting in a MANET, flooding generated broadcast storm problem, rebroadcasting schemes, Issues in providing multicasting in MANET, Multicast routing protocols, Geocasting- Geocast routing protocols. Mobile Network Layer (Mobile IP), DHCP (Dynamic host configuration protocol), Routing in Mobile Ad hoc Networks (MANET)- Topology-based versus position based approaches, Proactive routing protocols, Reactive routing protocols, Hybrid routing protocols, position based routing issues and forwarding strategies, AODV (Ad-hoc On-Demand Distance Vector Routing Protocol)- Analysis of AODV under mobility and Faults in a network, DSR (Dynamic Source Routing)-Analysis of DSR under mobility and Faults in a network, Secure routing protocols in MANET, Wireless Sensor Networks: (Routing protocols, Localization methods, Sensor Deployment Strategies), traffic flow pattern in WSN- one to many, many to one and many to many, Routing protocols for Delay Tolerant Networks, Routing protocols for Vehicular Ad-hoc Networks, Wireless Access Protocol, GPS (Global positioning system) and applications, RFID and its applications.

Text Books / Reference Books / Online Resources:

1. Jochen Schiller, "*Mobile Communications*", Second Edition, Pearson Education, 2003.
2. C D M Cordeiro, D. P. Agarwal, "*Adhoc and Sensor Networks: Theory and applications*", World Scientific, 2006.

Asoke K Talukder and Roopa R. Yavagal, "*Mobile Computing – Technology, Applications and Service Creation*"; TMH Pub., New Delhi, 2006

CS352	CRYPTOGRAPHY	Credits 3-0-0: 3

Pre-requisites: none
Course Outcomes:
At the end of the course, the student will be able to

CO1	Analyze number theoretic concepts
CO2	Analyze encryption algorithms
CO3	Explore key management issues and solutions
CO4	Apply cryptographic algorithms to build secure protocols

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3	3	-	2	-	-	-	1	-	1	2	1	2	-	2
CO2	-	3	3	2	3	1	-	-	1	-	1	2	-	-	3	3
CO3	3	2	3	3	3	1	1	1	3	2	3	3	3	2	3	3
CO4	-	3	2	1	3	2	1	1	2	2	3	3	-	-	1	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Number theory: Introduction to number theory – Modular Arithmetic; Finite fields; Number theory properties – Primality testing; Fetmat's and Euler's theorem; Chinese remainder theorem; Integer factorization; discrete logarithm; Introduction to security - security architecture; security attacks; security services; security mechanisms; Classical and modern ciphers; pseudorandomness; statistical properties of random sequences; discrete probability; Symmetric key and public key cryptosystems; General design principles of block ciphers; substitution-permutation networks; General design principles of stream ciphers; linear feedback shift-register sequences; boolean functions; canonical examples - DES, 3DES, AES, RC4, RC5, RC6, A5/1,2; Analysis methodologies - differential, linear, square, algebraic techniques. Public key cryptosystems Diffie Hellman key exchange, public key encryption, digital signatures, Knapsack, RSA, ElGamal, Rabin schemes; Functionalities of entity, content authentication; message digests and hashing schemes; Key management and Distribution-Certificate authorities; PKI; MAC; Hashing; Digital Signatures-Authentication protocols; Digital Signature Standard; Cryptographic embedding in different layers of network stack; applications, protocols and standards; social, economic and geo-political

Text Books / Reference Books / Online Resources:

1. William Stallings, "*Cryptography and Network Security*", 6th edition Pearson Education, 2014
2. Cryptography: Theory and Practice, Third Edition (Discrete Mathematics and Its Applications) by Douglas R. Stinson, CRC Press
3. Applied Cryptography - Protocols, Algorithms, and Source Code in C, by Bruce Schneier
4. A. Menezes, P. Van Oorschot, S. Vanstone, "*Handbook of Applied Cryptography*", CRCPress, 2004

CS353	MACHINE LEARNING	Credits 3-0-0: 3

Pre-requisites: CS152- Data Structures and Algorithms

Course Outcomes:

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

CO1	Understand instance based learning algorithms
CO2	Design neural network to solve classification and function approximation problems
CO3	Build optimal classifiers using genetic algorithms
CO4	Design convolutional networks to solve classification problems

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	1	2	2	-	1	-	-	1	-	-	2	1	2	2
CO2	2	1	2	2	2	-	1	-	-	1	-	-	2	2	2	2
CO3	2	1	1	2	2	-	1	-	-	1	-	-	2	2	2	2
CO4	2	2	2	2	2	1	2	-	-	1	1	-	2	2	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; - The Concept Learning Task - General-to-specific ordering of hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias - Decision Tree Learning - Decision tree learning algorithm-

Inductive bias- Issues in Decision tree learning; - Artificial Neural Networks – Perceptrons, Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of backpropagation rule-Backpropagation Algorithm- Convergence, Generalization; – Evaluating Hypotheses – Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms; - Bayesian Learning – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm; - Computational Learning Theory – Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning; - Instance-Based Learning – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning - Genetic Algorithms – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Reinforcement Learning - The Learning Task, Q Learning, Support vector Machines, Deep learning networks – Deep Feedforward Networks – Regularization for Deep Learning – Optimization for Training Deep Models – Convolutional Network

Text Books / Reference Books / Online Resources:

Reading:

1. Tom.M.Mitchell, "*Machine Learning*", McGraw Hill International Edition, 1997
2. C Bishop, "*Pattern Recognition and Machine Learning*", Springer, 2006.
3. Ian Goodfellow, Yoshua Bengio, AaronCourville, "*Deep Learning*", The MIT Press Cambridge, Massachusetts, London, England, 2016

CS354	ADVANCED ALGORITHMS	Credits 3-0-0: 3

Pre-requisites: CS152-Data structures and Algorithms, CS204-Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Analyze time and space complexities of algorithms using asymptotic analysis.
CO2	Analyze amortized time complexity.
CO3	Classify problems into different complexity classes.
CO4	Analyze approximation algorithms and determine approximation factor.
CO5	Design FPT algorithms for some intractable problems.
CO6	Design and analyze efficient randomized algorithms.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	1	-	-	-	1	-	-	-	-	1	3	1	1	2
CO2	2	2	1	-	-	-	-	-	-	-	-	-	3	1	1	2
CO3	2	2	2	1	1	-	1	-	-	-	-	-	3	1	2	2
CO4	3	2	2	2	-	-	1	-	-	-	-	-	2	-	1	-
CO5	3	2	2	2	-	-	-	-	-	-	-	-	2	-	1	-
CO6	3	2	2	2	-	-	1	-	-	-	-	-	2	-	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Algorithm design techniques: Overview, Tree related dynamic programming, Backtracking algorithm for finding a maximum independent set, Enumerating independent Sets,
Amortized Analysis: Three approaches to amortized analysis, Disjoint-sets, Fibonacci heap
NP-completeness: Polynomial-time solvability, Polynomial-time verification, NP-completeness and reducibility, Example NP-complete Problems – Set cover, Graph Coloring, Domination.
Approximation Algorithms: One way of coping with NP-hardness, The set cover problem, The traveling-sales person problem, The bin packing problem.
Fixed-Parameter Algorithms: Another way of coping with NP-hardness, Bounded search tree, Kernelization, The vertex cover problem.
Randomized Algorithms: Las Vegas and Monte Carlo algorithms, Karger's min cut algorithm and improvements to it by Karger and Stein, Randomized selection.

Text Books / Reference Books / Online Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", Third Edition, PHI, 2009.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "*Fundamentals of Computer Algorithms*", Second Edition, Universities Press, 2011.
3. Herbert S. Wilf, "*Algorithms and Complexity*", AK Peters Ltd., 2003.
4. Vijay V. Vajirani, "*Approximation Algorithms*", Springer, 2001.
5. Rodney G. Downey and M. R. Fellows, "*Parameterized Complexity*", Springer, 2012.
6. Rajeev Motwani and Prabhakar Raghavan, "*Randomized Algorithms*", Cambridge University Press, 1995.
7. Michael R. Garey and David S. Johnson, "*Computers and Intractability: A Guide the theory of NP-Incompleteness*", First edition, W.H. Freeman & Co., 1979.
8. Michael T. Goodrich and Roberto Tamassia, "*Algorithm Design: Foundations, Analysis and Internet Examples*", Second Edition, Wiley-India, 2006.

CS355	CRYPTOGRAPHY LABORATORY	Credits 0-0-3: 1.5

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Implement number theoretic algorithms
CO2	Apply number theoretic concepts in cryptographic algorithms
CO3	Implement cryptographic algorithms
CO4	Design and Implement cryptographic tools

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3	3	1	-	-	-	-	1	-	1	2	2	-	-	2
CO2	2	3	3	3	3	1	-	-	2	-	2	3	2	1	-	3
CO3	-	2	-	2	3	1	1	-	3	2	3	3	-	-	1	2
CO4	2	1	3	3	3	2	1	2	3	2	3	3	2	2	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Cycle 1:
Number theory and Cryptographic experiments - 1) Euclidean and extended Euclidean algorithms 2) Modular Arithmetic over Zn 3) Polynomial Arithmetic over GF(2^n) 4) Substitution Techniques 5) DES 6) AES 7) Chinese Remainder Theorem 8) RSA 9) Diffie-Hellman Key Exchange 10) Elgamal Cryptographic System 11) Elliptic curve cryptography 12) Elgamal and DSS Digital signature scheme 13) MAC 14) Cryptographic hashing schemes 15)Authentication Protocols
Cycle 2:
Explore cryptographic tools (ex openssl)-Design and Implement crypto tool to offer security services

Text Books / Reference Books / Online Resources:

1. GNU-MP Manual
2. A. Menezes, P. Van Oorschot, S. Vanstone, "*Handbook of Applied Cryptography*", CRC Press, 2004
3. William Stallings, "*Cryptography and Network Security*", 6th edition Pearson Education, 2014

CS356	MACHINE LEARNING LABORATORY	Credits 0-0-3: 1.5

Pre-requisites: CS152- Data Structures and Algorithms

Course Outcomes:

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

CO1	Design and implement algorithm using least means square learning rule to play checkers game
CO2	Design a classifier using Genetic Algorithm..
CO3	Apply supervised learning using perceptron
CO4	Design and implement Convolutional Networks to solve classification problem.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	1	2	2	-	1	-	-	1	-	-	2	1	2	1
CO2	2	1	2	2	2	-	1	-	-	1	-	-	2	1	2	2
CO3	2	1	1	2	2	-	1	-	-	1	-	-	1	1	1	1
CO4	2	2	2	2	2	1	2	-	-	1	1	-	2	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Design and implement machine learning algorithm using least means square learning rule to play checkers game. The training experience should be generated by the system playing game with itself.
2. Implement a genetic algorithm program to successfully classify examples in the restaurant domain problem. Data for RESTAURANT domain given in this table.

Example	Friday	Patrons	Price	Rain	Res	Est	Type	Will Wait
X1	No	Some	High	No	Yes	0-10	French	Yes
X2	No	Full	Low	No	No	30-60	Thai	No
X3	No	Some	Low	No	No	0-10	Burger	Yes
X4	Yes	Full	Low	Yes	No	10-30	Thai	Yes
X5	Yes	Full	High	No	Yes	>60	French	No
X6	No	Some	Med	Yes	Yes	0-10	Italian	Yes
X7	No	None	Low	Yes	No	0-10	Burger	No
X8	No	Some	Med	Yes	Yes	0-10	Thai	Yes
X9	Yes	Full	Low	Yes	No	>60	Burger	No
X10	Yes	Full	High	No	Yes	10-30	Italian	No
X11	No	None	Low	No	No	0-10	Thai	No
X12	Yes	Full	Low	No	No	30-60	Burger	Yes

Res indicates reservation made and Est means estimated waiting time to get table, WillWait target attribute indicates whether to wait for table in that restaurant.

3. Design a feed forward neural network using back propagation algorithm to to solve and written character recognition problem for A to Z and 0 to 9 letters.
4. Implement perceptron learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gate problems.
5. Implement Adaline learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gate problems.
6. Implement a Genetic algorithm to generate solutions for 8-Queens problem.
7. Implement a machine learning program to play 5× 5 Tic tac toe game. The program should use least means square learning rule.
8. Design a Convolutional network for handwritten character recognition problem for A to Z and 0 to 9 digits.

Text Books / Reference Books / Online Resources:

1. Tom.M.Mitchell, "*Machine Learning*", McGraw Hill International Edition, 2017
2. C Bishop, "*Pattern Recognition and Machine Learning* ", Springer, 2006.
3. Ian Goodfellow, Yoshua Bengio, AaronCourville, "*Deep Learning*", The MIT Press Cambridge, Massachusetts, London, England., 2016

CS361	ADVANCED THEORETICAL COMPUTER SCIENCE	Credits 3-0-0: 3

Pre-requisites: CS203 – Theory of Computation

Course Outcomes:

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

CO1	Identify and explain fundamental mathematical constraints for developing algorithms to solve problems in high dimensional space
CO2	Develop methods to study how to draw good samples efficiently and how to estimate statistical and linear algebra quantities, with such Samples.
CO3	Apply learning models and algorithms with provable guarantees on learning error and time.
CO4	Build models to understand and to capture essential properties of large structures, like the web and social networks
CO5	Build applications for ranking and social choice as well as problems of sparse representations such as compressed sensing

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2	2	1	-	1	-	-	1	-	-	3	1	2	-
CO2	2	3	2	1	3	-	-	1	-	1	-	1	3	1	3	-
CO3	2	2	3	2	2	1	1	-	2	-	1	1	3	2	2	1
CO4	2	2	2	2	3	1	1	-	1	1	-	-	3	3	2	1
CO5	2	3	3	3	3	1	-	1	1	-	-	1	3	3	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

High-Dimensional Space - The Law of Large Numbers, The Geometry of High Dimensions, Properties of the Unit Ball, Volume of the Unit Ball, Volume Near the Equator, Generating Points Uniformly at Random from a Ball, Gaussians in High Dimension, Random Projection and Johnson-Lindenstrauss Lemma, Separating Gaussians, Fitting a Spherical Gaussian to Data

Best-Fit Subspaces and Singular Value Decomposition (SVD) - Singular Vectors, Best Rank-k Approximations, Left Singular Vectors, Power Method for Singular Value Decomposition, A Faster Method, Singular Vectors and Eigenvectors, Applications of Singular Value Decomposition, Centering Data, Principal Component Analysis, Clustering a Mixture of Spherical Gaussians, Ranking Documents and Web Pages, Application of SVD to a Discrete Optimization Problem

Random Walks and Markov Chains - Stationary Distribution, Markov Chain Monte, Metropolis-Hasting Algorithm, Gibbs Sampling, Areas and Volumes, Convergence of Random

Walks on Undirected Graphs, Using Normalized Conductance to Prove Convergence, Electrical Networks and Random Walks, Random Walks on Undirected Graphs with Unit Edge Weights, Random Walks in Euclidean Space, The Web as a Markov Chain,

Machine Learning - The Perceptron algorithm, Kernel Functions, Generalizing to New Data,

Overfitting and Uniform, illustrative Examples and Occam's Razor, Learning Disjunctions, Occam's, Application: Learning Decision Trees, Regularization: Penalizing Complexity, Online, Online to Batch Conversion, Support-Vector Machines, VC-Dimension, Strong and Weak Learning – Boosting, Stochastic Gradient Descent, Combining (Sleeping) Expert Advice, Deep Learning, Further Current Directions.

Algorithms for Massive Data Problems: Streaming, Sketching, and Sampling - Frequency Moments of Data Streams, Matrix Algorithms using, Sketches of Documents

Clustering - Two General Assumptions on the Form of Clusters, k-Center, Spectral Clustering, Approximation Stability, High-Density Clusters, Kernel Methods, Recursive Clustering based on Sparse Cuts, Dense Submatrices and Communities, Community Finding and Graph Partitioning, Spectral clustering applied to social

Random Graphs - The G (n; p) Model, Phase Transitions, Giant, Cycles, Phase Transitions for Increasing, Branching Processes, CNF-SAT, Nonuniform Models of Random Growth Models, Small World Graphs

Topic Models, Nonnegative Matrix Factorization, Hidden Markov Models,
and Graphical Models - Topic Models, An Idealized Model, Nonnegative Matrix Factorization – NMF, NMF with Anchor Terms, Hard and Soft Clustering, The Latent Dirichlet Allocation Model for Topic, The Dominant Admixture, Finding the Term-Topic Matrix, Hidden Markov, Graphical Models and Belief, Bayesian or Belief, Markov Random, Factor, Tree Algorithms, Message Passing in General, Graphs with a Single Cycle, Belief Update in Networks with a Single, Maximum Weight Matching, Warning

Other Topics - Ranking and Social Choice, Compressed Sensing and Sparse, Applications, Uncertainty Principle

Text Books / Reference Books / Online Resources:

1. Brian Steele, John Chandler, Swarna Reddy, "*Algorithms for Data Science"*, Springer, 2016
2. Noga Alon and Joel H Spenser, "*Probabilistic Method"*, Third Edition,. John Wiley & Sons, 2008.
3. Rajeev Motwani and Prabhakar Raghavan, "*Randomized Algorithms*", Cambridge University Press, 1995

CS362	DISTRIBUTED COMPUTING	Credits 3-0-0: 3

Pre-requisites: CS301: Operating Systems, CS304: Computer Networks

Course Outcomes:

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

CO1	Understand models of distributed computing
CO2	Analyze algorithms for coordination, communication and synchronization in distributed systems
CO3	Analyze distributed shared memory models
CO4	Design and Implement distributed file systems
CO5	Design distributed algorithms for handling deadlocks

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	3	2	2	-	-	-	-	-	-	3	2	-	-
CO2	2	3	2	1	2	3	-	-	-	-	-	-	3	3	-	3
CO3	3	3	3	2	2	2	-	-	-	-	-	-	2	2	-	3
CO4	1	2	3	3	3	2	-	-	-	-	-	-	3	3	-	3
CO5	3	3	2	2	2	3	-	-	-	-	-	-	3	3	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Distributed Computing Introduction: Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges

A Model of Distributed Computations: A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication.

Logical Time: Logical clocks, scalar time, vector time, Matrix time, virtual time, Physical clock synchronization - NTP

Global state and snapshot recording algorithms: System model, Snapshot algorithms for FIFO channels, Variations of Chandy-Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal delivery system, Monitoring global state, Necessary and sufficient conditions for consistent global snapshots, Finding consistent global snapshots in a distributed computation.

Message ordering and group communication: Message ordering paradigms, Group communication, Causal order (CO), Total order, Multicast, Propagation trees for multicast,

Application-level multicast algorithms, Fault-tolerant group communication, Multicast algorithms at the network layer.

Termination detection: System model of a distributed computation, Termination detection using distributed snapshots, weight throwing and spanning-tree-based algorithms, Message-optimal termination detection, and Termination detection in a general distributed computing model, Termination detection in the atomic computation model, Termination detection in a faulty distributed system

Distributed mutual exclusion algorithms: Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Lodha and Kshemkalyani's fair mutual exclusion algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Token-based algorithms, Suzuki–Kasami's broadcast algorithm, Raymond's tree-based algorithm,.

Deadlock detection in distributed systems: System model, Models of deadlocks, Knapp's classification of distributed deadlock detection algorithms, Mitchell and Merritt's algorithm for the single resource model, Chandy–Misra–Haas algorithm for the AND model, Chandy–Misra–Haas algorithm for the OR model, Kshemkalyani–Singhal algorithm for the P-out-of-Q model

Distributed shared memory: Abstraction and advantages, Memory consistency models, Shared memory mutual exclusion, Wait-freedom, Register hierarchy and wait-free simulations, Wait-free atomic snapshots of shared objects

Check pointing and rollback recovery : Introduction, Background and definitions, Issues in failure recovery, Checkpoint-based recovery, Log-based rollback recovery, Koo–Toueg coordinated check pointing algorithm, Juang–Venkatesan algorithm for asynchronous check pointing and recovery, Manivannan–Singhal quasi-synchronous check pointing algorithm, Peterson–Kearns algorithm based on vector time, Helary–Mostefaoui–Netzer–Raynal communication-induced protocol
Consensus and agreement algorithms: Problem definition, Overview of results, Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures, Wait-free shared memory consensus in asynchronous systems
Failure detectors: Unreliable failure detectors, The consensus problem, Atomic broadcast, A solution to atomic broadcast, The weakest failure detectors to solve fundamental agreement problems, An implementation of a failure detector, An adaptive failure detection protocol

Text Books / Reference Books / Online Resources:

1. Ajay D. Kshemakalyani, Mukesh Singhal, "*Distributed Computing*", Cambridge University Press, 2008
2. Andrew S. Tanenbaum, Maarten Van Steen, "*Distributed Systems - Principles and Paradigms*", PHI, 2004

CS363	CYBER LAWS AND INTELLECTUAL PROPERTY RIGHTS	Credits 3-0-0: 3

Pre-requisites: Discrete Mathematics, Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand Cyber laws in general and Indian IT Act in particular.
CO2	Analyze cybercrimes and their culpability under various sections of the act.
CO3	Understand Cyber case law and an ability to recall various cases for developing solutions
CO4	Understand Intellectual property rights in Indian context
CO5	Extract illegal knowledge related to computer based activities and diffuse such knowledge.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	1	-	-	-	-	1	-	-	-	1	-	-	1	1	-
CO2	-	1	-	-	2	-	2	-	-	-	1	-	-	-	2	-
CO3	-	-	2	-	1	-	-	-	1	1	1	1	1	-	2	1
CO4	-	-	-	-	-	-	-	2	-	-	1	-	-	-	-	1
CO5	1	-	-	1	1	1	-	-	-	1	-	2	-	-	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

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, Cr.P.C and Indian Evidence Law - Cybercrimes under the Information Technology Act, 2000 - Cybercrimes under

International Law - Hacking Child Pornography, Cyber Stalking, and Denial of service Attack, Virus Dissemination,

Software Piracy, Internet Relay Chat (IRC) Crime, Credit Card Fraud, Net Extortion, Phishing etc - Cyber Terrorism - Violation of Privacy on Internet - Data Protection and Privacy – Indian Court cases

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

Text Books / Reference Books / Online Resources:

1. Justice Yatindra Singh, "Cyber Laws", Universal Law Publishing Co., New Delhi, 2016
2. Farouq Ahmed, "*Cyber Law in India"*, New Era publications, New Delhi, 2017
3. S.R.Myneni, "*Information Technology Law (Cyber Laws)"*, Asia Law House, Hyderabad, 2018
4. Chris Reed, "*Internet Law-Text and Materials"*, Cambridge University Press, 2004
5. Pawan Duggal, "*Cyber Law- the Indian Perspective"*, Universal Law Publishing Co., New Delhi, 2009

CS364	SOFTWARE METRCIS AND SOFTWARE PROJECT MANAGMENET	Credits 3-0-0: 3

Pre-requisites: CS303- Software Engineering

Course Outcomes:

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

CO1	Determine the software measurement attributes and metrics
CO2	Plan and evaluate software projects
CO3	Analyze factors involved in implementation of software projects.
CO4	Understand project monitoring and control techniques

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	2	1	2	1	1	1	1	1	1	1	1	2	1	3	3
CO2	1	2	2	3	2	1	1	1	1	1	1	1	2	1	3	3
CO3	1	1	2	3	1	1	1	1	2	1	1	1	2	1	3	3
CO4	1	1	1	3	3	1	1	1	1	1	1	1	2	2	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The Basics of Measurement: Measurement in software engineering, The scope of software metrics, the representational theory of measurement, Measurement and models, Measurement scales and scale types, Meaningfulness' in measurement

Goal-based framework for software measurement: Classifying software measures, determining what to measure, applying the framework, Software measurement validation

Empirical investigation: Principles of investigation, planning formal experiments, planning case studies

Measuring internal product attributes: Aspects of software size, Length, reuse

Measuring internal product attributes; Types of structure measures, Control-flow structure, and Modularity and information flow attributes, Object-oriented metrics

Measuring external product attributes: Modeling software quality, measuring aspects of quality

Making process predictions: Good estimates, Cost estimation - problems and approaches, Models of effort and cost, Problems with existing modeling methods, Dealing with problems of current estimation methods, Implications for process prediction

Software Project Management: General management, introduction to software project management, Conventional software management, project initiation, feasibility study, project planning, project evaluation, resource allocation, project monitoring, project control, case studies

Text Books / Reference Books / Online Resources:

1. Norman E. Fenton, Shari Lawrence Pfleeger, "*Software Metrics - A Rigorous and Practical Approach*", 2nd Edition, PWS Pub, 1996.
2. Walker Royce, "*Software Project Management*", Addison Wesley, 1998.
3. Pankaj Jalote, "*Software Project Management in Practice*", Pearson Education Inc. Delhi, 2002.

CS365	**SOFTWARE TESTING**	**Credits 3-0-0: 3**

Pre-requisites: CS303 – Software Engineering

Course Outcomes:

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

CO1	Analyze Various test processes and continuous quality improvement
CO2	Analyze Types of errors and fault models
CO3	Modeling the behaviour using FSM
CO4	Application of software testing techniques in commercial environments
CO5	Analyze various test tools

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	1	2	1	1	1	2	2	2	2	1	2	1	3	3
CO2	2	2	2	3	2	1	1	1	1	1	2	1	2	1	3	3
CO3	3	1	2	3	2	1	1	1	2	1	2	1	3	3	3	3
CO4	2	2	1	3	3	2	2	2	2	1	2	1	3	2	2	3
CO5	2	2	1	2	1	1	1	1	1	2	1	1	2	1	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction, The Taxonomy Of Bugs, Flowgraphs And Path Testing, Transaction-Flow Testing, Data-Flow Testing, Graph Coverage, Logic Coverage, Input Space Partitioning, Domain Testing, Metrics And Complexity, Paths, Path Products, And Regular Expressions, Syntax Testing, Logic-Based Testing, States, State Graphs, And Transition Testing, Graph Matrices And Applications Applying Criteria In Practice: Practical Considerations, Engineering Criteria For Technologies, Building Testing Tools, Challenges In Testing Software

Text Books / Reference Books / Online Resources:

1. Baris Beizer, "*Software Testing techniques*", Dreamtech, second edition, 1990
2. Paul Ammann George and Jeff Offutt , "*Introduction to Software Testing*", Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Second Edition, 2008
3. Naresh Chauhan, "*Software Testing- Principles and practices* ", Oxford University Press, Second Edition, 2016

CS366	QUANTUM COMPUTING	Credits 3-0-0: 3

Pre-requisites: Discrete Mathematics, Applied Physics, Computational Number Tjeory, Mathematics-I, Design and Analysis of Algorithms

Course Outcomes:

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

CO1	Demonstrate fundamental concepts of quantum computing
CO2	Explain the basic architecture of Quantum computing using qubit
CO3	Analyze the applications of Quantum Computing Algorithms
CO4	Develop applications using Quantum Programming

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	1	-	-	-	-	-	-	-	-	-	-		-	-	-
CO2	2	1		-	-	-	-	-	-	-	-	-	-	-	-	-
CO3	2	3	2	2	3	-	-	-	-	-	-	-	3	3	3	1
CO4	-	1	3	3	3	-	-	-	-	-	-	-	3	3	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Mathematics: Complex Numbers; Complex Vector Spaces

Leap from Classical to Quantum: Classical Deterministic Systems; Probabilistic Systems, Quantum Systems, Assembling Systems

Basic Quantum Theory: Quantum States, Observables; Measuring; Dynamics; Assembling Quantum Systems

Architecture: Bits and Qubits; Classical Gates; Reversible Gates; Quantum Gates

Algorithms: Deutsch's Algorithm; Deutsch–Jozsa Algorithm; Simon's Periodicity Algorithm; Grover's Search Algorithm; Shor's Factoring Algorithm

Programming Languages: Programming in a Quantum World; Quantum Assembly Programming; Toward Higher-Level Quantum Programming; Quantum Computation Before Quantum Computers

Text Books / Reference Books / Online Resources:

1. Noson S. Yanofsky, Mirco A. Mannucci, "*Quantum Computing for Computer Scientists*", Cambridge University Press, 1st Edition, 2008
2. Eleanor G. Rieffel, Wolfgang H. Polak, "*Quantum Computing: A Gentle Introduction*", MIT Press, 2011

CS367	ADVANCED DATA MINING	Credits 3-0-0: 3

Pre-requisites: Data Warehousing and Data Mining
Course Outcomes:
At the end of the course, the student will be able to

CO1	Analyze Algorithms for sequential patterns
CO2	Determine patterns from time series data
CO3	Develop algorithms for Temporal Patterns
CO4	Distinguish computing frameworks for Big Data analytics.
CO5	Apply Graph mining algorithms to Web Mining.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	1	1	1	-	1	-	-	1	-	-	3	-	3	2
CO2	2	3	2	2	3	-	-	1	-	1	-	1	3	-	3	2
CO3	2	1	3	2	2	1	1	-	2	-	1	1	3	-	3	1
CO4	2	2	2	1	2	1	1	-	1	1	-	-	2	2	3	1
CO5	2	3	2	3	3	1	-	1	1	-	-	1	3	2	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Sequential Pattern Mining concepts, primitives, scalable methods; Transactional Patterns and other temporal based frequent patterns, Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis; Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph Mining, Mining frequent subgraphs, finding clusters, hub and outliers in large graphs, Graph Partitioning; Web Mining, Mining the web page layout structure, mining web link structure, mining multimedia data on the web, Automatic classification of web documents and web usage mining; Distributed Data Mining, Distribute data mining framework, Distributed data source, Distributed data mining techniques, Distributed classifier learning, distributed clustering, distributed association rule mining and Challenges of distributed data mining; Social Network Analysis, characteristics of social Networks.

Text Books / Reference Books / Online Resources:

1. J Han and M Kamber, "*Data Mining Concepts and Techniques*", 2nd Edition, Elsevier, 2011
2. Pang Ning Tan, M Steinbach, Vipin Kumar, "*Introduction to Data Mining*", Addision Wesley, 2006
3. G Dong and J Pei, "*Sequence Data Mining*", Springer, 2007.

CS368	BIO-INFORMATICS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand the theoretical basis behind bioinformatics.
CO2	Compute homologues, analyse sequences, construct and interpret evolutionary trees.
CO3	Analyse protein sequences, identify proteins, and retrieve protein structures from databases.
CO4	Understand homology modelling and computational drug design.
CO5	Determine and model biological information and apply this to the solution of biological problems in any arena involving molecular data.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	-	2	-	3	-	-	1	-	-	-	-	1	1	2	-
CO2	-	2	3	-	-	1	2	3	-	2	-	-	1	1	3	1
CO3	3	-	2	-	3	-	-	-	3	-	-	3	-	2	3	1
CO4	2	3	2	3	2	-	-	3	-	1	2	-	1	1	3	1
CO5	-	1	3	-	1	-	3	-	2	2	3	3	1	2	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Bioinformatics: What is a Data Base, Types of Databases, Biological Databases, Pitfalls of Biological Databases, Information Retrieval from Biological Databases, Pair wise Sequence Alignment: Evolutionary Basics, Sequence homology versus similarity, Sequence similarity versus Identity, Scoring Matrices, Statistical Significance of Sequence alignment, Database similarity searching: Unique requirement of Database searching, Heuristic Database searching, Basic alignment search tool: Comparison of FASTA and BLAST, Multiple Sequence Alignment, Scoring Function, Exhaustive Algorithms, Heuristic Algorithms, Gene Prediction, Categories of gene prediction programs, Gene prediction in prokaryotes and Eukaryotes, Phylogenetics Basics Molecular phylogenetics and molecular basics Gene phylogeny versus species phylogeny, Forms of tree representation, Why finding a true tree is difficult, Phylogenetic tree construction methods and programs Protein structure basics: Amino acid, peptide formation, Dihedral Angles, Hierarchy, Secondary structures, Tertiary structure, Determination of protein 3-D structure, Protein structure data base, Genome mapping, assembly and comparison, Genome mapping, Genome sequencing, Genome sequence assembly, Genome Annotation, Comparative genomics, Functional Genomics, Sequence based approaches, Microarray based approaches, Comparisons of SAGE and DNA microarray.

Text Books / Reference Books / Online Resources:

1. Jin Xiong, "*Essential Bioinformatics*", 1st Edition, Cambridge University Press, 2011.
2. Arthur M Lesk, "*Introduction to Bioinformatics*", 2nd Edition, Oxford University Press, 2007.

CS369	HUMAN COMPUTER INTERACTION	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand usability and the factors affecting universal usability.
CO2	Apply the principles of design process in human computer interaction.
CO3	Analyze and evaluate the interaction styles in human computer interaction.
CO4	Analyze the design issues in human computer interaction.
CO5	Create an interface based on the concepts of human computer interaction.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	-	2	-	-	-	-	-	-	-	-	-	-	1	1
CO2	2	2	2	2	2	1	-	-	-	-	-	-	1	2	1	1
CO3	2	2	2	2	2	-	-	-	-	-	-	-	2	-	1	1
CO4	2	3	2	2	2	-	-	-	-	-	-	-	1	-	2	2
CO5	2	1	3	1	2	-	-	-	-	-	-	-	2	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

INTRODUCTION:

Usability of Interactive Systems: Usability goals and measures, usability motivations, goals for Professional.

Universal Usability: Variations in physical abilities and physical workplaces, diverse cognitive and perceptual abilities, cultural and international diversity, users with disabilities, older adult users, children, accommodating hardware and software diversity.

DESIGN PROCESS:

Managing design processes: Introduction, organizational support for design, design process, design frameworks, design methods, social impact analysis, legal issues.

Evaluation of user experience: Expert reviews and heuristics, usability testing and laboratories, survey instruments.

INTERACTION STYLES:

Direct manipulation and immersive environments: Direct manipulation, examples of direct manipulation, 2D and 3D interfaces, teleoperation and presence, augmented and virtual reality.

Fluid navigation: Introduction, navigation by selection, small displays, content organization, audio menus, form fill-in and dialog boxes.

Expressive human and command languages: Introduction, speech recognition, speech production, human language technology, traditional command languages.

Devices: Introduction, Keyboards and Keypads, Pointing Devices, Displays.

DESIGN ISSUES:

Timely user experience: Models of system response-time impacts, expectations and attitudes, user productivity and variability in system response time, frustrating experiences.

Advancing the user experience: Introduction, display design, view management, animation, awebpage design, nonanthropomorphic design, error messages.

Documentation and user support: Shaping the content of the documentation, accessing the documentation, reading from displays versus reading from paper, online tutorials and animated documentations, online communities and other avenues for user support, the development process.

Information Search: Five stage search framework, dynamic queries and faceted search, command languages and natural language queries, multimedia document search and other specialized search.

Text Books / Reference Books / Online Resources:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos, "*Designing the User Interface: Strategies for Effective Human Computer Interaction"*, 6 Ed, Pearson, 2017.
2. Wilbert O Galitz, "*The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*", 3 Ed, Wiley, 2007.

CS370	INTRUSION DETECTION SYSTEMS	Credits 3-0-0: 3

Pre-requisites: CS301-Operating systems, CS304- Computer Networks
Course Outcomes:
At the end of the course, the student will be able to

CO1	Explore the concepts of Network Protocol Analysis, and analyze information systems and networked systems
CO2	Identify system vulnerabilities and attacks, and troubleshoot system problems
CO3	Design and Develop intrusion detection systems& intrusion prevention systems and identify their signatures
CO4	Select technologies and tools for intrusion detection and intrusion prevention
CO5	Exercises and use cases for testing and evaluating various IDS techniques.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	1	-	-	-	1	-	2	-	-	-	2	-	2	1	-
CO2	2	1	-	1	-	1	-	1	-	-	-	2	-	1	1	2
CO3	1	1	1	1	2	-	1	-	1	2	-	-	1	1	-	2
CO4	-	2	-	-	1	-	-	-	1	2	-	1	-	1	-	1
CO5	2	-	1	2	2	-	1	-	-	-	1	-	2	-	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Firewall Planning and Design, Developing a Security Policy, System Configuration Strategies, Working with Proxy Servers and Application-Level Firewalls, Authenticating Users, Encryption and Firewalls.
Intrusion detection, Audit, Internal and external threats to data, attacks, Information sources - Host based information sources, and Network based information sources; Types and classification of IDS.
Intrusion Prevention Systems, Network Systems, Network IDs protocol based IDs, Hybrid IDs, Analysis schemes, models for intrusion analysis, techniques, mapping responses to policy vulnerability analysis, credential analysis non-credential analysis
IDS using SNORT, NIDS, NNID and HIDS;
Discovery and Detection: Identify IDS signatures such as anomaly detection, pattern matching and statistical analysis; Machine Learning models for IDS, Distributed IDS models; Architecture models of Intrusion Detection and intrusion prevention.

Text Books / Reference Books / Online Resources:

1. RafeeqRehman: "*Intrusion Detection with SNORT, Apache, MySQL, PHP and ACID*", Prentice Hall, 2003.
2. Christopher Kruegel, FredrikValeur, Giovanni Vigna, "*Intrusion Detection and Correlation Challenges and Solutions*", Springer, 2005.
3. Carl Endorf, Eugene Schultz and Jim Mellander, "*Intrusion Detection & Prevention*", Tata McGraw-Hill, 2004.
4. Stephen Northcutt, Judy Novak, "*Network Intrusion Detection*", New Riders Publishing, 2002.

CS371	DATA NETWORKS	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms, Discrete mathematics

Course Outcomes:

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

CO1	Analyze layered network architecture and passage of data over communication links
CO2	Analyze delay models in Data Networks using Queueing Systems for messaging and delay sensitive applications
CO3	Design and analyze routing algorithms for Internet and multi-hop autonomous networks
CO4	Analyze flow and rate control algorithms between a sender and receiver in wide area networks
CO5	Design and analyze software defined networking algorithms for data forwarding through Internet

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	2	-	-	-	2	2	2	3	2	2	-	2
CO2	3	3	3	3	2	-	-	-	2	2	2	3	2	2	-	3
CO3	3	3	3	3	2	-	-	-	2	2	2	3	3	3	-	3
CO4	3	3	3	3`	2	-	-	-	2	2	2	3	3	3	-	3
CO5	3	3	3	3	3	-	-	-	2	2	2	3	3	3	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Physical Layer Channels: Frequency- and Time- Division Multiplexing; Error Detection; ARQ; Framing, *Point-to-Point Protocols at Network Layer:* Error Recovery, The X.25 Network Layer Standard, The Internet Protocol, *Transport Layer*: Transport Layer Standards, Addressing and Multiplexing TCP, Error Recovery in TCP; Flow Control in TCP/IP, *Asynchronous Transfer Mode (ATM),* Delay Models in Data Networks: The *M/M/1* Queueing System, *M/M/m, M/M/∞, M/M/m/m* and other Markov Systems, Networks of Transmission Lines, Networks of Queues – Jackson's Theorem, Multi-Access Communication: Packet Radio Networks, Splitting Algorithms, Carrier Sensing, Multi-access Reservations, Routing in Data Networks: Wide-Area Network Routing, Interconnected Network Routing, Network Algorithms and Shortest Path Routing, Broadcasting Routing Information, Flow Models, Optimal Routing and Topological Design; Characterization of Optimal Routing, Flow Control: Main Objectives of Flow Control, Window

Flow Control, Rate Control Schemes, Rate Adjustment Algorithms, Classification of TCPs, Software Defined Networks: Fundamental Characteristics of SDN, the *OpenFlow* Specification, SDN via Hypervisor-Based Overlays, SDN in the Data Center, SDN in Wide Area Networks, SDN in Mobile Networks, SDN Ecosystem and Network Virtualization.

Text Books / Reference Books / Online Resources:

1. Bertsekas, Dimitri, and Robert Gallager, "*Data Networks*" (2nd Edition). Upper Saddle River, NJ, USA, Second Edition, Pearson Education/Prentice Hall, 1992
2. Walrand and Varaiya, "*High Performance Communication Networks*", San Francisco, CA: Morgan Kaufmann Publishers, 1996
3. Stevens, "*TCP/IP Illustrated. Reading*", MA: Addison-Wesley Pub. Co., c1994-c1996.
4. Paul Goransson, Chuck Black, "*Software Defined Networking: a comprehensive approach*", Morgan Kaufmann (Elsevier), 2014

CS372	SEMANTIC WEB	Credits 3-0-0: 3

Pre-requisites: CS311-Web Technologies, CS152-Data Structures and Algorithms

Course Outcomes:

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

CO1	Understand the standards and data formats used in the Semantic Web
CO2	Comprehend technologies including XML and XSLT
CO3	Design semantic web meta data and RDF schema
CO4	Develop ontology programming with Jena API

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	-	-	-	1	-	-	2	-	2	-	-	-	-	1	1
CO2	-	1	2	-	2	1	2	-	-	-	-	2	-	1	2	1
CO3	2	1	1	1	2	1	-	1	2	1	2	1	1	1	2	1
CO4	-	-	1	1	1	2	1	-	1	-	-	1	2	1	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The Semantic Web Vision, overview of techniques and standards, Semantic Web Architecture, XML with Document Type Definitions and Schemas, Transformation/Inference rules in XSLT, RuleML and RIF, metadata with RDF (Resource Description Framework); metadata taxonomies with RDF Schema; Ontology languages, Ontology Development using Protege editor, Ontology Querying, Ontology Reasoning and Description Logic (DL), Semantic Web Application Areas, Ontology programming with Jena API, Ontology Engineering.

Text Books / Reference Books / Online Resources:

1. Grigoris Antoniou and Frank van Harmelen, "*A Semantic Web Primer*", 1st Edition, MIT Press, 2004.
2. John Hebeler, Matthew Fisher, Ryan Blace and Andrew Perez-Lopez, "*Semantic Web Programming*", 1st Edition, Wiley, 2009.

CS373	HETEROGENEOUS COMPUTING	Credits 3-0-0: 3

Pre-requisites: CS301- Operating Systems, CS304- Computer Networks
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand different models of parallel programming and the usage of MPI and OpenMP libraries
CO2	Analyze for the performance of GPU memory hierarchy
CO3	Develop parallel programs using OpenCL library
CO4	Generate parallel programs for matrix, graph and sorting problems using Cuda library
CO5	Develop mixed mode programs for Multicore and GPGPU systems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3	2	2	2	2	-	-	-	-	-	-	2	2	-	-
CO2	3	2	2	2	2	2	-	-	-	-	-	-	2	2	-	3
CO3	3	3	3	3	2	2	-	-	-	-	-	-	3	3	2	2
CO4	3	3	3	3	3	3	-	-	-	-	-	-	3	3	-	2
CO5	2	3	3	3	3	2	-	-	-	-	-	-	3	3	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks.

GPU Computing - Introduction : Introduction to General Purpose Computing on Graphic Processing Units (GPGPU); GPU as parallel computers – CUDA enabled NVIDIA GPUs; AMD-ATI-OpenCL, GPGPU Architecture of a Modern GPU – Threaded Streaming Multi-processors; communication bandwidth; Unified Graphics and Computing Processors; GPGPU- GPU computing – Scalable GPUs; Speed-up & Parallelism; CPU/GPU programing; SPMD programming model

CUDA APIs & CUDA Threads - GPUs-Data Parallelism; GPU-CUDA Program Structure; GPU device memories & Data transfer; Kernel functions and threading; CUDA Runtime API; CUDA Thread Execution; CUDA Thread organization; Synchronization; Thread Scheduling;

OpenCL (Open Computing Language) : Heterogeneous Computing – Programming; Data Parallelism Model – OpenCL; OpenCL, Device Architecture; OpenCL Kernel Functions; OpenCL APIs – Matrix-Matrix, Computations using different partitioning techniques– OpenCL; OpenCL – Device Management and Kernel launch; Compilation Model and programming features of OpenCL – Device query; Object Query, and task parallelism model

Mixed Programming - Multi-Core Processors & GPUs : Heterogeneous computing – mixed programming (Message Passing-MPI and Shared Memory Programming (Pthreads, OpenMP); Heterogeneous computing - mixed programming – CPU (Pthreads, OpenMP) & GPU (CUDA, OpenCL); MPI-OpenCL & MPI-CUDA ; Programming for Dense Matrix Computations

Text Books / Reference Books / Online Resources:

1. Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, "*Heterogeneous Computing with OpenCL*", MGH, 2011

2. Jason Sanders, Edward Kandrot, "*CUDA By Example – An Introduction to General-Purpose GPU Programming*", Addison Wesley, 2011
3. Michael J Quinn, "*Parallel Programming in C with MPI and OpenMP*", TMH, 2004

CS374	CLOUD COMPUTING	Credits 3-0-0: 3

Pre-requisites: CS304-Computer Networks, CS301-Operating Systems

Course Outcomes:

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

CO1	Determine Cloud Computing Architectures and Services for various societal applications
CO2	Analyze Cloud infrastructure including Google Cloud and Amazon Cloud.
CO3	Develop private and hybrid cloud for organizations to execute customized applications
CO4	Analyze authentication, confidentiality and privacy issues in Cloud computing environment.
CO5	Determine financial and technological implications for selecting cloud computing platforms

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	2	-	2	-	2	-	-	-	-	2	2	-	-	2
CO2	3	3	2	2	2	1	2	-	2	2	-	2	3	3	2	2
CO3	3	3	2	3	2	2	2	-	2	2	2	2	3	3	2	2
CO4	3	3	2	2	2	1	2	-	2	2	-	2	3	2	-	3
CO5	-	-	3	-	3	2	2	-	-	2	2	2	2	-	1	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction - Cloud Computing Architecture, Cloud Delivery Models, The SPI Framework, SPI Evolution, The SPI Framework vs. the Traditional IT Model, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS) Google Cloud Infrastructure - Google File System – Search engine – MapReduce - Amazon Web Services - REST APIs - SOAP API - Defining Service Oriented Architecture, Combining the cloud and SOA, Characterizing SOA, Loosening Up on Coupling, Making SOA Happen, Catching the Enterprise Service Bus, Telling your registry from your repository, Cataloguing services, Understanding Services in the Cloud. Serving the Business with SOA and Cloud Computing, Query API - User Authentication- Connecting to the Cloud - OpenSSH Keys - Tunneling / Port Forwarding - Simple Storage Service - S3, EC2 - EC2 Compute Units, Platforms and storage, EC2 pricing, EC2 customers Amazon Elastic Block Storage - EBS - Ubuntu in the Cloud - Apache Instances in EC2 – Amazon Cloud Services- Amazon Elastic Compute Cloud (Amazon EC2), Amazon SimpleDB, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon Simple Queue Service (Amazon SQS), Amazon Elastic MapReduce, Amazon Relational Database Service (Amazon RDS) , EC2 Applications - Web application design - AWS EC2 Capacity Planning – Apache Servers - Mysql Servers - Amazon Cloud Watch - Monitoring Tools.

Text Books / Reference Books / Online Resources:

1. Anothony T Velte, Toby J Velte, Robert Elsenpeter, "*Cloud Computing: A Practical Approach*", MGH, 2010.
2. Arshdeep Bahga, Vijay Madisetti, "*Cloud Computing: A Hands-on Approach*", Universities Press (India) Private Limited, 2014
3. Gautam Shroff, "*Enterprise Cloud Computing*", Cambridge, 2010.
4. Rajkumar Buyya, James Broberg, Andrzej Goscinski, "*Cloud Computing Principles and Paradigms*", Wiley, 2011
5. Ronald Krutz and Russell Dean Vines, "*Cloud Security*", 1st Edition, Wiley, 2010.

CS375	COMPUTATIONAL GEOMETRY	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Summarize combinatorial geometry.
CO2	Analyse algorithms for efficiently solving geometric problems involving 2D polygons and 3D polyhedrons
CO3	Choose and apply data structures for geometric combinatorial problems.
CO4	Build algorithms for applications to such areas as computer graphics, big data analytics and pattern recognition, geometric databases, numerical taxonomy, and robotics

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	1	1	-	1	-	-	1	-	-	2	1	-	1
CO2	2	3	2	2	3	-	-	1	-	1	1	1	3	2	3	1
CO3	2	1	3	2	2	1	1	-	1	-	1	1	2	2	2	1
CO4	3	3	2	3	3	1	1	-	1	1	1	-	3	3	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Combinatorial geometry: Polygons, polytopes, triangulations and simplicial complexes, planar and spatial subdivisions. Constructions: triangulations of polygons and point sets, convex hulls, intersections of halfspaces, Voronoi diagrams, Delaunay triangulations, restricted Delaunay triangulations, arrangements of lines and hyperplanes, Minkowski sums, Reeb graphs and contour trees; relationships among them. Geometric duality and polarity. Upper Bound Theorem, Zone Theorem.
Algorithms and analyses: Sweep algorithms, incremental construction, divide-and-conquer algorithms, randomized algorithms, backward analysis. Numerical predicates and constructors, geometric robustness. Construction of triangulations, convex hulls, halfspace intersections, Voronoi diagrams, Delaunay triangulations, arrangements, Minkowski sums, and Reeb graphs.
Geometric data structures: Doubly-connected edge lists, quad-edges, face lattices, trapezoidal maps, conflict graphs, history DAGs, spatial search trees (a.k.a. range search), segment trees, binary space partitions, quadtrees and octrees, visibility graphs.
Applications: Line segment intersection and overlay of subdivisions for cartography and solid modeling. Triangulation for graphics, interpolation, and terrain modeling. Nearest neighbor search, small-dimensional linear programming, database queries, point location queries, windowing queries, discrepancy and sampling in ray tracing, curve reconstruction and surface reconstruction, robot motion planning.

Text Books / Reference Books / Online Resources:

1. Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, "*Computational Geometry: Algorithms and Applications*", Third edition, Springer-Verlag, 2008
2. F. P. Preparata and M. I. Shamos, "*Computational Geometry: An Introduction*", Springer-Verlag, 1985
3. Ketan Mulmuley, "*Computational Geometry: An Introduction through Randomized Algorithms*", Prentice Hall, 1994.
4. H. Edelsbrunner, "*Algorithms in Combinatorial Geometry*", Springer-Verlag, 1987.

CS376	MODEL DRIVEN FRAMEWORKS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Construct domain specific languages
CO2	Construct model transformations
CO3	Synthesize model metrics
CO4	Understand contemporary approaches to model driven engineering
CO5	Apply domain specific modeling approach to authentic cases

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	-	3	1	2	-	2	-	-	3	1	3	-	2
CO2	3	3	3	-	3	-	1	-	2	-	-	1	1	3	-	2
CO3	3	3	3	2	3	1	1	-	1	-	-	1	1	3	-	2
CO4	3	3	3	-	3	-	1	-	2	-	-	2	1	3	-	2
CO5	3	3	3	-	3	-	1	-	2	-	-	2	1	3	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Traditional software engineering approach: Drawbacks, Software processes, modular-based software design

Model Driven Software Engineering (MDSE) Principles: MDSE basis, Overview of MDSE Technology, Criticism of MDSE, MDSE use cases

Model driven Architecture (MDA): MDA Definitions and Assumptions, the modeling levels-CIM, PIM, PSM, mapping, general purpose and domain specific language in MDA, architecture Driven modernization

Integration of MDSE in a development process: introducing MDSE in a software development process, traditional development process and MDSE, Domain driven design and MDSE, Test driven Development and MDSE

Modeling Language at a glance: Anatomy of modeling language, general purpose vs. domain specific modeling language, General purpose modeling- the case of UML, UML profile

platforms, software artefacts using UML standard modeling language, defining modeling constraints, automated GUI generation

Transformations: Model to model transformations, model to text transformations

MDA Practice, Usage of QVT, Kermeta, etc., MDA Transformation Languages, model editors, model valuators, model metrics, modeling framework, middleware to support transformations, MDA applications

Text Books / Reference Books / Online Resources:

1. Thomas Stahl, Markus Voelter, "*Model-Driven Software Development: Technology, Engineering, Management*", Wiley, 2006.
2. Anne Kleppe, Jos Warmer, and Wim Bast, "*MDA Explained - The Model Driven Architecture: Practice and Promise*", Pearson Education, Boston, USA, 2003.

CS377	SOFTWARE RELIABILITY TECHNIQUES	Credits 3-0-0: 3

Pre-requisites: Software Engineering, Probability and Statistics

Course Outcomes:

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

CO1	Understand Software Reliability during different phases of Software Development Life Cycle
CO2	Analyze Software Reliability parameters using Markovian Modelling
CO3	Estimate Software Reliability parameters using Maximum Likelihood and Least Square Method
CO4	Evaluate performance of Binomial-Type, Poison-Type and Markovian Models
CO5	Predict Software Reliability using Intelligent Techniques

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2
CO2	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	3
CO3	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	3
CO4	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2
CO5	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Software Reliability: The need for Software Reliability, Some Basic Concepts, Software Reliability and Hardware Reliability, Availability, Modelling and General Model Characteristics.

Software Reliability Modeling: Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Markov Structure Models, Time Series Models, Non-homogeneous Poison Process Models.

Markovian Models: General Concepts, General Poison-Type Models, Binomial -Type Models, Poison-Type Models, Comparison of Binomial-Type and Poison-Type Models, Fault Reduction Factor for Poison-Type Models.

Descriptions of Specific Models: Finite Failure Category Models, Infinite Failure Category Models.

Parameter Estimation: Maximum Likelihood Estimation, Least Squares Estimation, Bayesian Inference.

Comparison of Software Reliability Models: Comparison Criteria, Comparison of Predictive Validity of Model Groups, Evaluation of other Criteria.

Software Reliability Prediction: Problems associated with different Software Reliability Models, Software Reliability prediction parameters, Intelligent Techniques for Software Reliability Prediction.

Text Books / Reference Books / Online Resources:

1. M. Xie, "*Software Reliability Modelling*", World Scientific; 1991.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, "*Software Reliability Measurement, Prediction, Application*", McGraw-Hill Book Company; 1987.
3. Hoang Pham, "*System Software Reliability*", Springer; 2005

CS378	HIGH PERFORMANCE COMPUTING	Credits 3-0-0: 3

Pre-requisites: CS301 – Operating Systems, CS304 – Computer Networks
Course Outcomes:
At the end of the course, the student will be able to

CO1	Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
CO2	Optimize the performance of a parallel program to suit a particular platform.
CO3	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.
CO4	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	3	2	2	-	-	-	-	-	-	3	2	2	2
CO2	3	2	3	3	2	2	-	-	-	-	-	-	2	2	-	3
CO3	3	3	3	2	2	2	-	-	-	-	-	-	3	3	-	2
CO4	2	3	3	2	2	2	-	-	-	-	-	-	2	3	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction: Implicit parallelism, Limitations of memory system performance, control structure, communication model, physical organization, and communication costs of parallel platforms, Routing mechanisms for interconnection networks, mapping techniques.
Parallel algorithm design: Preliminaries, decomposition techniques, tasks and interactions, mapping techniques for load balancing, methods for reducing interaction overheads, parallel algorithm models.
Basic communication operations: Meaning of all-to-all, all-reduce, scatter, and gather, circular shift and splitting routing messages in parts. Analytical modeling of parallel programs: sources of overhead, performance metrics, the effect of granularity on performance, scalability of parallel systems, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs.
Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators
Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization
constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks.
Basic parallel algorithms: prefix sums, Tree traversal algorithms, basic operations (insertion deletion and search) on trees, merging, maximum, graph colouring list ranking, Planar geometry and String algorithms
Dense Matrix Algorithms: matrix vector multiplication, matrix-matrix multiplication, solving system of linear equations, Sorting: Sorting networks, Bubble sort, Quick sort, Bucket sort and other sorting algorithms Graph algorithms: Minimum spanning tree, single source shortest paths, all-pairs shortest paths, Transitive closure, connected components, algorithms for sparse graphs.

Text Books / Reference Books / Online Resources:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "*Introduction to Parallel Computing*", Second Edition Pearson Education – 2007
2. Michael J. Quinn (2004), "*Parallel Programming in C with MPI and OpenMP*", McGraw-Hill International Editions, Computer Science Series,
3. Joseph Jaja, "*An Introduction to Parallel Algorithms*", Addison-wesley, 1992

CS379	ADVANCED OPERATING SYSTEMS	Credits 3-0-0: 3

Pre-requisites: CS301-Operating Systems

Course Outcomes:

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

CO1	Design and implement Unix kernel data structures and algorithms
CO2	Analyze synchronization problems in uniprocessor and multiprocessor systems
CO3	Evaluate the scheduling requirements of different types of processes and find their solutions
CO4	Implement user level thread library and mimic the behaviour of Unix kernel for scheduling, synchronization and signals

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	2	1	2	2	-	-	-	-	-	1	-	3	3	-	2
CO2	1	2	3	3	3	2	-	-	-	-	1	-	3	3	-	2
CO3	2	2	3	3	1	1	-	-	-	-	1	-	3	2	-	3
CO4	1	2	3	3	2	2	-	-	-	-	1	-	3	3	-	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to UNIX: The process and the kernel, Mode, space and context, Process abstraction, kernel mode, synchronization by blocking interrupts, process scheduling.

Introduction to Threads: Fundamental abstractions, Lightweight process design, Issues to consider, User level thread libraries, scheduler activations

Signals: Signal generation and handling, Unreliable signals, Reliable signals, Signals in SVR4, Signals implementation, Exceptions, Process Groups

Process Scheduling: Clock interrupt handling, Scheduler Goals, Traditional UNIX scheduling, scheduling case studies

Synchronization and Multiprocessing: Introduction, Synchronization in Traditional UNIX Kernels, Multiprocessor Systems, Multiprocessor synchronization issues, Semaphores, spin locks, condition variables, Read-write locks, Reference counts

Introduction to Intel X86 Protected Mode: Privilege Levels, Flat memory model, Descriptors - Segment, Task, Interrupt; GDT, LDT and IDT, Initializing to switch to protected mode operation, Processor Exceptions.

Kernel Memory Allocators: Resource map allocator, Simple power-of-two allocator, McKusick-Karels Allocator, Buddy system, SVR4 Lazy Buddy allocator, OSF/1 Zone Allocator, Hierarchical Allocator, Solaris Slab Allocator

File system interface and framework : The user interface to files, File systems, Special files, File system framework, The Vnode/Vfs architecture, Implementation Overview, File System dependent objects, Mounting a file system, Operations on files

File System Implementations : System V file system (s5fs) implementation, Berkeley FFS, FFS functionality enhancements and analysis, Temporary file systems, Buffer cache and other special-purpose file systems

Distributed File Systems: Network File System (NFS), Remote File Sharing (RFS)

Advanced File Systems : Limitations of traditional file systems, Sun-FFS, Journaling approach 4.4 BSD, Log-Structured file system, Meta logging Episode FS, Watchdogs, 4.4 BSD portal FS, Stackable FS layers, 4.4 BSD FS interface.

Text Books / Reference Books / Online Resources:

1. Uresh Vahalia, "*UNIX Internals*", Pearson Education, 2005.
2. Richard Stevens, Stephen A. Rago, "*Advanced Programming in the UNIX Environment*", Pearson Education, 2/e, 2005.

CS380	PRIVACY PRESERVING DATA PUBLISHING	Credits 3-0-0: 3

Pre-requisites: CS301-Operating systems, CS304-Computer Networks

Course Outcomes:

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

CO1	Apply anonymization methods for sensitive data protection.
CO2	Apply state-of art techniques for data privacy protection
CO3	Design privacy preserving algorithms for real-world applications.
CO4	Identify security and privacy issues in OLAP systems
CO5	Apply information metrics for Maximizing the preservation of information in the anonymization process.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	-	-	2	-	1	2	-	-	-	-	1	2	-	2	1
CO2	-	1	-	-	2	-	-	-	-	1	-	1	1	-	2	-
CO3	1	-	1	1	1	-	-	-	2	1	1	-	1	1	1	-
CO4	-	2	-	2	-	1	1	2	2	-	2	-	-	1	-	2
CO5	1	-	2	1	-	2	-	-	-	-	1	-	-	2	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

1. Privacy issues and privacy models
2. Anonymization: Operations on Anonymization, Information metrics, Anonymization methods for the transaction data, trajectory data, social networks data, and textual data. Collaborative Anonymization
3. Access control methods for outsourced data
4. Use of Fragmentation and Encryption to Protect Data Privacy;
5. Security and Privacy in OLAP systems.
6. Extended Data publishing Scenarios
7. Anonymization for Data Mining and social media data

Text Books / Reference Books / Online Resources:

1. Benjamin C.M. Fung, Ke Wang, Ada Wai-Chee Fu and Philip S. Yu, "*Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*", 1st Edition, Chapman & Hall/CRC, 2010.
2. Charu C. Aggarwal, "*Privacy-Preserving Data Mining: Models and Algorithms*", 1st Edition, Springer, 2008.

CS401	INTERNET OF THINGS	Credits 2-0-2: 3

Pre-requisites: Design and Analysis of Algorithms, Data Structures and Algorithms, Computer Networks

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
CO2	Develop smart IoT Applications using smart sensor devices and cloud systems
CO3	Development of smart mobile apps for societal applications
CO4	Design secure protocols for IoT systems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	3	-	-	-	-	-	-	2	2	2	2	2	2
CO2	3	3	3	3	3	2	2	-	2	2	2	2	2	2	3	3
CO3	3	3	3	3	3	2	2	-	2	2	2	2	2	3	2	3
CO4	2	3	2	2	2	-	-	-	-	-	2	2	2	3	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Internet of Things-Concepts, Characteristics, Challenges, M2M and Smart IoT Technology, Applications of IoT, IoT Architectures- Reference Model, Protocol Architecture, Overview of WSN-Sensor Technology, RFID, Sensor-based Solutions and RFIDs in IoT. Network Technology- Data Link Layer Protocols- 802.11, 802.15.1, 802.15.4, ZigBee, BLE and 6LoWPAN and other Communication Standards, Internet Layer Protocols- IPv6 Protocol, Addressing, BLE Over IPv6, ZigBee Over IPv6 and 802.11 Over IPv6, Internet Layer-Routing Protocols- RPL, AODV, Transport Protocols- UDP in IoT, Application Layer Communication Protocols- MQTT, CoAP, AMQT and other Protocols, Review of Edge Computing and Fog Computing, Cloud Computing in IoT, Smart Applications-Smart City, Health-Care, Vulnerabilities, Security Requirements, and Threat Analysis, IoT Security Tomography and Layered Attacker model, , Access control methods, Privacy Issues in IoT Architectures, Hardware Platforms, Operating System of IoT, Web APIs, Mobile app developments for IoT, Programming with Raspberry Pi, Implementing smart IoT applications through web APIs

Lab Assignments:

1. Design a Wireless Sensor network to perform end-to-end data transfer by creating a network topology of 10 Nodes and establish the connection between them and test to check the connection by sending ICMPv6 echo request flows from node-1 through the network to node-10 back with ICMPv6 echo reply flows from node-10 through the network to node-1. Trace the packet queues and analyse the packet flow by creating a file "wsn-ping6.tr". (Hint: Use 6LowPAN Module in NS3 and Validation using Wireshark, Use NetAnim for Visualization).
2. Design a network to perform end-to-end data transfer of IPv6 Packets over IEEE 802.15.4 Networks? Create a network topology of 10 Nodes and test to check the connection between different two UDP clients and the verify the correctness of the received packets. (Hint: Use 6LowPAN Module in NS3 and Validation using Wireshark, Use NetAnim for Visualization).
3. Design a network to perform end-to-end data transfer of IPv6 Packets over Bluetooth Low Energy Networks? Create a network topology of 10 Nodes and test to check the connection between different two UDP clients and the verify the correctness of the received packets. (Hint: Use 6LowPAN Module in NS3 and Validation using Wireshark, Use NetAnim for Visualization).
4. Design an IoT Network that consists of three RPL Border Router. Each RPL Border Router has two different edge network and each RPL Border Router is used to connect a regular IP network with a RPL 6LoWPAN edge network. To perform this activity create a network topology of 50 Nodes and establish the connection between them. Display the data and visualise the data (Hint: Contiki OS, Cooja Emulator, and Wireshark and Use Web-Sense Motes and Sky Motes).

5. Design an MQTT-SN that exchanges the messages between the node and the broker of the network. Create a simulation with a RPL Border Router device to connect a regular IP network with a MQTT-SN network. To perform this activity create a network topology of 50 Nodes and establish the connection between them. Display the data and visualise the data (Hint: Contiki OS, Cooja Emulator, and Wireshark).
6. Design an IoT Network that consists of one RPL Border Router and two different edge network and RPL Border Router is used is used to connect a regular IP network to two RPL 6LoWPAN edge network. Create the first RPL 6LoWPAN edge network with groups of clients UDP or the "sensors" that will send data to the gateway and another RPL 6LoWPAN edge network. Next, create the UDP Server in node.js which is outside of the 6LoWPAN network. Establish UDP communication between network (Groups of Clients) and server that reports every client information to the UDP Server.
7. Design a Wireless Sensor network that generates the data, transmit the data through internet and stores into IoT Cloud. To perform this activity create a network topology of 10 Nodes and establish the connection between them, Display the data and visualise the data (Hint: Contiki OS, Cooja Emulator, and Ubidots).
8. Design an IoT Network that consists of three RPL Border Router. Each RPL Border Router has two different edge network and each RPL Border Router is used to connect a regular IP network with a RPL 6LoWPAN edge network. Store the generated data from the edges devices into IoT Cloud. To perform this activity create a network topology of 10 Nodes and establish the connection between them, Display the data and visualise the data You can pause the simulation and examine the packets and console output at your own pace, simply click the Pause button at the Simulation Control panel. When you are done click Restart. (Hint: Contiki OS, Cooja Emulator, and Ubidots; Use Web-Sense Motes and Sky Motes).
9. Implement the Energy Aware Routing Protocol that use the RPL that reduces energy consumption in a Wireless Sensor Network. Test the IoT model that uses UDP/IP/RPL/6LoWPAN/802.15.4 in a simulated environment. Use ns3 and Cooja simulators/Whitefield Framework. Analyse the Network formation and Dynamics of parent switching behaviour of RPL network.
10. Design an LORAWAN wireless sensor network. Assume the LoRaWAN network contains the sensor on the left, which is connected to the network server on the right. It sends the sensing data to the Gateway using the LoRa technique over long distance. At the Gateway, a higher-throughput IP network takes over and forwards all the data to the network server. Analyse the performance the LORAWAN networks end-to-end reliability packet delivery ratio. Hint Use NS3

Text Books / Reference Books / Online Resources:

1. Olivier Hersent, "*The Internet of Things Key Applications and Protocols*", Wiley, 2012
2. Sudip Misra, "*Introduction to IoT*", Cambridge University Press; First edition , 2021
3. David Hanes, "*IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*", Cisco Press; 1st edition 2017
4. Arshdeeep Bahga, Vijay Madisetti, "*Internet of Things: A Hands-on Approach*", Universities Press, 2015
5. Raj Kamal, "*Internet of Things: Architecture and Design Principles*", McGraw Hill Education private limited, 2017
6. Kai Hwang, Min Chen, "*Big Data Analytics for Cloud, IoT and Cognitive Computing*", Wiley, 2018

CS401	DEEP LEARNING	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Design Multi-Layer neural network to solve Supervised Learning problems
CO2	Design Autoencoders to solve Unsupervised Learning problems
CO3	Apply Regularization methods Early stopping, data augmentation, dropout etc. for optimization results
CO4	Apply Classical Supervised methods CNN'S, FCN, RCNN etc. for Image Denoising, Segmentation and Object detection problems.
CO5	Use Long Shot Term Memory (LSTM) Networks, GRU for time series analysis classification problems.
CO6	Apply Generative Adversarial Networks, GAN, VAE to solve Supervised and Unsupervised Learning Problems

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3	3	2	3	1	-	-	1	1	1	-	3	2	2	1
CO2	2	3	3	2	3	1	-	-	1	1	1	-	3	2	3	2
CO3	3	2	2	3	3	2	-	-	1	1	1	-	2	2	2	1
CO4	3	2	2	3	3	2	-	-	1	1	1	-	3	2	3	2
CO5	2	3	3	3	3	2	-	-	1	1	1	-	3	2	3	2
CO6	3	2	2	3	3	2	-	-	1	1	1	-	3	2	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
History of Deep Learning, Deep Learning Success Stories, McCulloch Pitts Neuron, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Gradient Descent, Feed Forward Neural Networks, Back propagation, Optimizers: Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD; Principal Component Analysis and its interpretations, Singular Value Decomposition; Auto encoders and relation to PCA, Regularization in auto encoders, Denoising auto encoders, Sparse auto encoders; Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Dropout, Drop connect; Greedy Layer wise Pre-training, Better activation functions, Better weight initialization methods, Batch Normalization; Learning Vectorial Representations of Words; Convolutional Neural Networks, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet; Recurrent Neural Networks, Back propagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs; Semantic Segmentation, Instance Segmentation, FCN, Unet ; Object Localization, Region Proposal Networks, RCNN, RFCN, DeYolo; Encoder Decoder Models, Generative Adversarial Networks, GAN, VAE, One Shot Learning, Deep Reinforcement Learning, Attention Mechanism, Attention over images.

Text Books / Reference Books / Online Resources:

1. Ian Goodfelllow, Yoshua Benjio, Aaron Courville, "*Deep Learning*", The MIT Press.
2. Christopher Bishop, "*Pattern Recognition and Machine Learning*", Springer, 2006.
3. Charu C. Aggarwal, "*Neural Networks and Deep Learning*", Springer, 2018.
4. https://nptel.ac.in/courses/106/106/106106184/

CS402	NETWORK SECURITY	Credits 2-0-2: 3

Pre-requisites: Computer Networks, Cryptography
Course Outcomes:
At the end of the course, the student will be able to

CO1	Design and analyse authentication protocols
CO2	Understand different types of network attacks and propose counter measures
CO3	Analyse vulnerabilities in network layer and apply IP Security mechanisms
CO4	Analyse application level threats and design security mechanisms
CO5	Analyse network traffic and detect intrusions
CO6	Implement security algorithms to handle attacks at different layers

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	-	-	-	-	-	-	-	-	-	2	-	-	3
CO2	3	3	3	-	-	2	-	-	-	-	-	-	2	2	-	3
CO3	3	3	3	-	-	2	-	-	-	-	-	-	2	2	-	3
CO4	3	3	3	-	-	2	-	-	-	-	-	-	2	2	-	3
CO5	3	3	3	-	-	2	-	-	-	-	-	-	2	2	2	3
CO6	3	3	3	3	3	2	-	-	-	-	-	-	2	2	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
User Authentication: Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos Systems, Remote User Authentication Using Asymmetric Encryption. Malicious Software: Viruses, Worms, System Corruption, Attack Agents, Information Theft, Phishing, Spyware Payload Stealthing, Backdoors, Rootkits, Distributed Denial of Service Attacks.
DoS Attacks and Network Defenses, Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange (IKE).
DNS, Web Security Considerations, Secure Sockets Layer, Transport Layer Security, HTTPS standard, Secure Shell (SSH) application. Electronic Mail Security- Pretty Good Privacy, Analyzing and defending against web-based malware.
Firewalls and Intrusion Detection Systems: Intrusion Detection Password Management, Firewall Characteristics Types of Firewalls, Firewall Basing, Firewall Location and Configurations. Overview of Blockchain,
Programming assignments on analysing network vulnerabilities and attacks, Programming assignments involves the use of tools, libraries and packages such as Wireshark, OpenSSL and Snort.

Text Books / Reference Books / Online Resources:

1. M. Speciner, R. Perlman, C. Kaufman, "*Network Security: Private Communications in a Public World*", Prentice Hall, 2002.
2. William Stallings, "*Cryptography and Network Security: Principles and Practice*", 6th Edition, 2014, Pearson, ISBN13:9780133354690.
3. J. Michael Stewart, "*Network Security, Firewalls And VPNs*", Jones & Bartlett Learning, 2013
4. Michael Gregg, "*The Network Security Test Lab: A Step-By-Step Guide*", Dreamtech Press, 2015

CS411	ALGORITHMIC GAME THEORY	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms, Game Theory

Course Outcomes:

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

CO1	Computation of equilibrium based on complete and incomplete information about the players
CO2	Develop Combinatorial Algorithms for Market Equilibria
CO3	Analyse mechanism design
CO4	Design and analyse combinatorial auctions
CO5	Design of Scalable Resource Allocation Mechanisms

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	2	3	1	3	1	3	3	3	1	3	3	2	3
CO2	3	2	3	2	3	1	3	1	3	3	3	1	3	3	2	3
CO3	2	3	2	3	2	1	3	1	2	2	2	1	3	2	3	2
CO4	3	2	3	2	2	1	3	1	3	3	3	1	3	3	2	3
CO5	3	2	3	2	2	1	3	1	3	3	3	1	3	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Basic Solution Concepts and Computational Issues - Games, Strategies, Costs, and Payoffs, Basic Solution Concepts, Finding Equilibria and Learning in Games, Refinement of Nash: Games with Turns and Subgame Perfect Equilibrium, Nash Equilibrium without Full Information: Bayesian Games, Cooperative Games, Markets and Their Algorithmic Issues

The Complexity of Finding Nash Equilibria - NP-Completeness, the Lemke–Howson Algorithm, the Class PPAD, Succinct Representations of Games, Reduction, Correlated Equilibria

Equilibrium Computation for Two-Player Games in Strategic and Extensive Form - Bimatrix Games and the Best Response Condition, Equilibria via Labeled Polytope, Lemke–Howson Algorithm, Integer Pivoting, Degenerate Games, Extensive Games and Their Strategic Form, Subgame Perfect Equilibria, Reduced Strategic Form, Sequence Form, Computing Equilibria with the Sequence Form

Learning, Regret Minimization, and Equilibria - External Regret Minimization, Regret Minimization and Game Theory, Generic Reduction from External to Swap Regret, The Partial Information Model, On Convergence of Regret-Minimizing Strategies to Nash Equilibrium in Routing Games

Combinatorial Algorithms for Market Equilibria - Fisher’s Linear Case and the Eisenberg–Gale Convex Program, Checking If Given Prices Are Equilibrium Prices, Primal-Dual Schema in the Enhanced Setting, Tight Sets and the Invariant, Balanced Flows, Linear Case of the Arrow–Debreu Model, Auction-Based Algorithm, Resource Allocation Markets, Algorithm for Single-Source Multiple-Sink Markets

Computation of Market Equilibria by Convex Programming - Fisher Model with Homogeneous Consumers, Exchange Economies Satisfying WGS, Specific Utility Functions, Models with Production

Graphical Games - Computing Nash Equilibria in Tree Graphical Games, Graphical Games and Correlated Equilibria, Graphical Exchange Economies.

Cryptography and Game Theory - Cryptographic Notions and Settings, Game Theory Notions and Settings, Contrasting MPC and Games, Cryptographic Influences on Game Theory, Game Theoretic Influences on Cryptography

Mechanism Design - Social Choice, Mechanisms with Money, Implementation in Dominant Strategies, Characterizations of Incentive Compatible Mechanisms, Bayesian–Nash Implementation

Combinatorial Auctions - The Single-Minded Case, Walrasian Equilibrium and the LP Relaxation, Bidding Languages, Iterative Auctions: The Query Model, Communication Complexity, Ascending Auctions

Inefficiency of Equilibria - Fundamental Network Examples, Inefficiency of Equilibria as a Design Metric
Routing Games - Models and Examples, Existence, Uniqueness, and Potential Functions, Price of Anarchy of Selfish Routing, Reducing the Price of
Network Formation Games and the Potential Function Method - Local Connection Game, Potential Games and a Global Connection Game, Facility Location
Selfish Load Balancing - Pure Equilibria for Identical Machines, Pure Equilibria for Uniformly Related Machines, Mixed Equilibria on Identical Machines, Mixed Equilibria on Uniformly Related Machines

Text Books / Reference Books / Online Resources:

1. N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, "*Algorithmic Game Theory*", Cambridge University Press, 2007..
2. Y. Shoham and K. Leyton-Brown, "*Multi-agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*", Cambridge University Press, 2009.
3. T. Roughgarden, "*Twenty Lectures on Algorithmic Game Theory*", Cambridge University. Press, 2016.

CS412	SECURITY AND PRIVACY	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Evaluate the risks and vulnerabilities in protocols/Standards.
CO2	Design and security analysis of cryptographic algorithms.
CO3	Design and security analysis of authentication, message integrity and authenticated encryption protocols
CO4	Develop techniques for Privacy preserving Data analysis
CO5	Compute lower bounds for differential privacy

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	2	2	3	2	2	1	2	2	2	1	2	2	2	2
CO2	3	2	3	2	2	2	2	1	3	2	2	1	3	3	2	2
CO3	3	2	3	2	2	2	2	1	3	2	2	1	3	3	2	2
CO4	3	3	3	3	3	2	2	1	3	2	3	1	2	2	2	3
CO5	3	3	3	3	3	2	2	1	3	2	3	1	2	2	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction to Security – risks, threats and vulnerabilities, Cryptography.
Symmetric key Cryptography – Encryption, Block ciphers, Chosen plaintext attacks, Stream Ciphers – One-time Pad (OTP), Perfect secrecy, Pseudo-random generators (PRG), Attacks on stream ciphers and OTP, Real world stream ciphers, Semantic security, Case Study- RC4, Salsa 20, CSS in DVD encryption, A5 in GSM, Block ciphers- DES, attacks, AES, Block ciphers from PRG, Modes of operation – one-time key and many-time keys, CBC, CTR modes,
Message Integrity – MAC, MAC based on PRF, NMAC, PMAC, Collision resistance – Birthday attack, Merkle-Damgard construction, HMC, Case study:SHA-256, Authenticated encryption, Key exchange algorithms,
Public key cryptosystems – Public key tools, Public key encryption, Chosen ciphertext secure public-key encryption, Digital signature, Fast hash based signatures, RSA, ElGamal, Elliptic curve cryptosystems – PKC, key exchange, IBE, Analysis of number theoretic assumptions, Case studies – HTTPS – SSL/TLS, SSH, IPSec, 802.11i WPA.
Protocols - Protocols for identification and login, Identification and signatures from sigma protocols, Proving properties in zero-knowledge, Authenticated key exchange, Key establishment with online trusted third parties, Two-party and multi-party secure computation
Privacy preserving Data analysis - Basis Techniques - Randomized response, the Laplace mechanism, the exponential mechanism, Composition theorems, and sparse vector technique.
Releasing Linear Queries with Correlated Error, Mechanisms via α-nets, Iterative construction mechanism, Boosting for queries algorithm, Stability and privacy, Lower bounds for differential privacy.
System design and analysis – Survivable distributed storage system, Electronic voting system, Digital Cash, Bit coin.

Text Books / Reference Books / Online Resources:

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. Van Oorschot, Paul Scott, A Vanstone, A J Menezes, "*Handbook of Applied Cryptography*", CRC Press, 2004.

CS413	INFORMATION RETRIEVAL	Credits 3-0-0: 3

Pre-requisites: Data Warehousing and Data Mining, Data Structures and Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand the concepts of information retrieval and their application to locate relevant information in large corpus of documents
CO2	Design and develop information retrieval systems for retrieval from web and other resources
CO3	Develop skills to analyze the performance of retrieval systems
CO4	Explore real-life case studies in different domains

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2	2	2	1	1	2	2	2	1	2	2	2	2
CO2	3	2	3	2	3	2	1	1	3	2	3	1	3	3	2	2
CO3	3	3	3	3	2	2	1	1	2	2	2	1	2	2	3	2
CO4	3	3	2	3	2	2	1	1	2	2	2	1	2	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Information Retrieval (IR) problem, Search and Browse, Efficient text indexing, inverted index, Metrics – relevance, effectiveness, precision, recall; Term vocabulary and postings lists, Dictionaries and tolerant retrieval, Index construction, Storage of indices, Storage of documents, Inverted file creation, Dictionary compression techniques, Inverted file compression techniques, Document compression techniques, Scoring, term weighting and the vector space model, Retrieval models: Probabilistic IR - the binary independence model; Boolean and vector-space retrieval models; Evaluation and interface issues, IR techniques for the web, including crawling, link-based algorithms, and metadata usage
Document clustering and Document classification -, Text classification (Naive Bayes, kNN, decision boundaries, Support Vector Machine), Clustering in information retrieval, Matrix decompositions and latent semantic indexing, Traditional and machine learning-based ranking approaches; personalization, link analysis, information visualization, Parallel and distributed IR, Multimedia IR – Search and Indexing.

Text Books / Reference Books / Online Resources:

1. C. Manning, P. Raghavan, and H. Schütze, "*Introduction to Information Retrieval*", Cambridge University Press, 2008.
2. D. Grossman and O. Frieder, "*Information Retrieval: Algorithms and Heuristics*", Springer, 2004.
3. R. Baeza-Yates and B. Ribeiro-Neto, "*Modern Information Retrieval*", Addison-Wesley, ACM Press, 1999.

CS414	BIOMETRICS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand various biometrics
CO2	Derive features from different biometric traits
CO3	Identify the acceptance issues associated with the design and implementation of biometric systems.
CO4	Identify various Biometric security issues.
CO5	Design biometric systems in various applications

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	2	1	-	-	-	-	-	-	-	1	1	1	1
CO2	3	3	1	2	1	-	-	-	-	-	-	-	-	-	1	-
CO3	3	3	1	2	1	-	-	-	-	-	-	-	1	1	-	-
CO4	3	2	1	2	1	-	-	-	-	-	-	-	1	-	1	1
CO5	2	3	1	3	1	-	-	-	-	-	-	-	-	-	-	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Person Recognition, Biometric Systems, Biometric Functionalities, Biometric System Errors, Design cycle of Biometric Systems, Applications of Biometric Systems, Security and Privacy issues.

Fingerprint recognition, friction and ridge pattern, fingerprint acquisition, sensing techniques, feature extraction, ridge orientation and frequency estimation, singularity extraction, minutiae extraction, matching, fingerprint indexing, synthesis, palmprint recognition

Face recognition sensors, cameras, Video sequences, Viola Jones face detection, appearance based , model based, texture based face recognition, Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Heterogeneous face recognition

Iris Recognition, Design, segmentation methods, normalization methods, encoding and matching

Additional biometric traits, Ear, Gait, Hand geometry, Ocular biometrics, Online and offline Signature authentication, soft biometrics.

Multibiometrics, multi-sensor, multi-algorithm, multi-instance, multi-sample, multi-modal systems, processing architectures, processing sequence, sensor level, feature level, score level, rank level, and decision level fusion, Attacks on biometric systems, adversary attacks, user interface, processing, template databases.

Text Books / Reference Books / Online Resources:

1. Anil K. Jain, Arun A. Ross, Karthik Nandakumar, "*Introduction to Biometrics*", Springer, 2011.
2. Arun A.Ross, Karthik Nandakumar, Anil K.Jain, "*Handbook of Multibiometrics*", Springer, 2006.
3. John Chirillo, Scott Blaul, "*Implementing Biometric Security"*, 1st Edition, Wiley Eastern Publication, 2005.
4. Anil K Jain, Patrick Flynn and Arun A Ross, "*Handbook of Biometrics"*, Springer,USA,2010.
5. Samir Nanavati, Michael Thieme and Raj Nanavati, "*Biometrics – Identity Verification in a Networked World"*, John wiley & Sons, New Delhi, 2003.
6. Paul Reid, "*Biometrics for Network Security"*, Pearson Education, New Delhi, 2004.
7. David D Zhang, "*Automated Biometrics: Technologies and Systems"*, Kluwer Academic Publishers, New Delhi, 2000.

CS415	SECURE MULTI-PARTY COMPUTATION	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understanding real world notions related to secure multi party computation
CO2	Distinguishing various models with respect to secure computation
CO3	Applying semi honest and active security secure computation protocols on solving real world problems
CO4	Apply zero knowledge proofs

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2	2	2	2	1	2	2	2	1	3	2	3	2
CO2	2	2	2	2	3	1	2	1	2	2	2	1	3	2	3	3
CO3	3	3	3	3	3	1	2	1	3	3	3	1	3	3	2	3
CO4	3	3	3	3	3	1	2	1	3	3	3	1	3	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction to secure computing: Definition-Secure Computation-Computational/statistical Indistinguishability; Real-Ideal World or Simulation based Security notions, secret sharing;
Models of Secure Computation: Honest vs. Dishonest majority settings; semi-honest vs active (malicious) adversary; static vs. adaptive computation; computational vs. information theoretic security; synchronous vs. asynchronous network
Oblivious Transfer and its extension: Oblivious transfer protocols - Definitions, constructions, and applications
Secure computation with semi-honest security: Honest-majority Setting- Secret Sharing, BenOr-Goldwasser-Wigderson (BGW) Construction, Optimizations (MPC in preprocessing mode and circuit randomization), Cramer-Damgaard-Neilsen (CDN) Construction; Dishonest majority Setting-Oblivious Transfers (OT), two-party Goldreich-Micali-Wigderson (GMW) construction, Optimizations of GMW (Random input OT and OT extension), Yao construction, BMR construction and multi-party GMW construction.
Secure computation with Active security: Honest Majority Setting.-Verifiable Secret Sharing, BGW Construction with active security, Hyper-invertible Matrices and Beerliova-Hirt (BH) Construction, Information Checking Protocol; Dishonest majority Setting-Commitment Schemes, Zero-knowledge, GMW Compiler for active corruption, Cut-and-Choose OT and Lindell-Pinkas Construction.
Zero-knowledge proof systems: zero-knowledge proofs of knowledge; non-interactive zero-knowledge
Broadcast & Byzantine Agreement : Dolev-Strong Broadcast; Exponential Information Gathering (EIG) construction for BA; Berman-Garay-Perry (BGP) construction for BA; Multi-valued Broadcast and BA.
Practical Secure Computation: Secure Set Intersection; Privacy Preserving Biometrics & Genomics; Secure Cloud Computing

Text Books / Reference Books / Online Resources:

1. Manoj M. Prabhakaran, Amit Sahai, "*Secure Multi-Party Computation, Cryptography and Information security series Vol 10*", IOS Press, 2013
2. Ronald Cramer, Ivan Damgaard and JesperBuus Nielsen, "*Secure Multiparty Computation and Secret Sharing - An Information Theoretic Approach*", Cambridge Press. (Book Draft), 2015

CS416	NATURAL LANGUAGE PROCESSING	Credits 3-0-0: 3

Pre-requisites: Mathematics-I, Probability, Statistics and Queuing Theory, Machine Learning
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand language modeling with N-Grams.
CO2	Apply syntactic parsing to produce parse trees.
CO3	Analyze semantics with dense vectors.
CO4	Apply lexical semantics with word senses.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	-	-	3	-	-	-	-	-	-	-	3	2	2	1
CO2	3	2	1	3	3	-	-	-	-	-	-	-	3	3	2	1
CO3	3	2	-	-	3	-	-	-	-	-	-	-	3	3	3	1
CO4	3	2	1	3	3	-	-	-	-	-	-	-	3	3	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction, Regular Expressions, Text Normalization and Edit Distance. Finite State Transducers, Language Modeling with N-Grams, Spelling Correction and the Noisy Channel, Naive Bayes Classification and Sentiment, Part-of-Speech Tagging, Syntactic Parsing, Statistical Parsing, Dependency Parsing, Vector Semantics, Hidden Markov Models, Syntactic parsing, Semantic Analysis, LSTM Recurrent Neural Networks, Lexicons for Sentiment and Affect Extraction, Information Extraction, Semantic Role Labeling and Argument Structure, Seq2seq Models and Machine Translation, Dialog Systems and Chatbots, Speech Recognition and Synthesis

Text Books / Reference Books / Online Resources:

1. Daniel Jurafsky and James H. Martin, "*Speech and Language Processing*", (3rd ed.)
2. Allen, James, "*Natural Language Understanding*", Second Edition, Benjamin/ Cumming, 1995.
3. Charniack, Eugene, "*Statistical Language Learning*", MIT Press, 1993.
4. Uday Kamath, John Liu, James Whitaker, "*Deep Learning for NLP and Speech Recognition*", Springer, 2020
5. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, "*Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*", O'REILY, 2020

CS417	ALGORITHMIC CODING THEORY	Credits 3-0-0: 3

Pre-requisites: Theory of Computation, Design and Analysis of Algorithms, Cryptography
Course Outcomes:
At the end of the course, the student will be able to

CO1	Apply Shannon's noisy coding theorem, Shannon capacity and entropy
CO2	Design of error correcting codes and decoding algorithms
CO3	Design and Analyze of light weight and code based cryptosystems
CO4	Design of network coding algorithms for communication networks

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO2	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO3	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO4	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Shannon Theorem, Shannon capacity, Hamming's Theory, Error correcting codes, Linear codes, Impossibility results for codes, Mac Williams Identities, Linear programming bound, The asymptotic perspective, Encoding, Decoding from erasures, Decoding RS codes, List decoding, linear time decoding, LDPC codes, Sipser-Spielman codes, Linear time encoding and decoding, Linear time and near optimal error decoding, Expander based constructions of efficiently, decodable codes, Some NP hard coding theoretic problems, Applications in complexity theory, Cryptography with error correcting codes, Lossless Multicast Network Coding, Network coding in Lossy Networks, Security against adversarial errors, Error correction bounds for centralized network coding.

Text Books / Reference Books / Online Resources:

1. Tom Richardson, Rudiger Urbanke, "*Modern Coding Theory*", Cambridge University Press, 2008
2. John b. Anderson and Seshadri Mohan, "*Source and Channel Coding: An Algorithm Approach"*, Springer, 1991.
3. G. Kabatiansky, E. Krouk and S. Semenov, "*Error Correcting Coding and Security for Data Networks"*, John Wiley & Sons Ltd., 2005.
4. Jiri Adamek, "*Foundations of Coding"*, Wiley Interscience Publication, John Wiley & Sons, *1991*

CS418	MALWARE DETECTION AND MITIGATION	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Possess the skills necessary to carry out independent analysis of modern malware samples using both static and dynamic analysis techniques.
CO2	Have an intimate understanding of executable formats, Windows internals and API, and analysis techniques.
CO3	Extract investigative leads from host and network based indicators associated with a malicious program
CO4	Apply techniques and concepts to unpack, extract, decrypt, or bypass new anti-analysis techniques in future malware samples.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	1	1	2	-	-	-	-	-	-	-		-	1	-
CO2	3	2	2	2	2	-	-	-	-	-	-	-	-	-	2	-
CO3	3	3	2	3	3	-	-	-	-	-	-	-	-	-	3	-
CO4	3	3	3	3	3	-	-	-	-	-	-		3	3	3	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction

Introduction to malware, OS security concepts, malware threats, evolution of malware, malware types- viruses, worms, rootkits, Trojans, bots, spyware, adware, logic bombs, malware analysis, static malware analysis, dynamic malware analysis.

Advanced Static Analysis

X86 Architecture- Main Memory, Instructions, Opcodes and Endianness, Operands, Registers, Simple Instructions, The Stack, Conditionals, Branching, Rep Instructions, C Main Method and Offsets. Analyzing Windows programs, Portable executable file format, disassembling malicious executable programs. Anti-static analysis techniques- obfuscation, packing, metamorphism, polymorphism.

Advanced Dynamic Analysis

Debugging malware- ollydbg, windbg, setting virtual environments- sandboxes, emulators, Hypervisors, virtual machines, live malware analysis, dead malware analysis, analyzing traces of malware- system-calls, api-calls, registries, network activities. Anti-dynamic analysis techniques-anti-vm, runtime-evasion techniques.

Malware Functionality

Downloaders, Backdoors, Credential Stealers, Persistence Mechanisms, Privilege Escalation, Covert malware launching- Launchers, Process Injection, Process Replacement, Hook Injection, Detours, APC injection.

Malware Detection Techniques

Signature-based techniques: malware signatures, packed malware signature, metamorphic and polymorphic malware signature. Non-signature based techniques: similarity-based techniques, machine-learning methods, invariant-inferences.

Text Books / Reference Books / Online Resources:

1. Sikorski, Michael, and Andrew Honig, "*Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software*", no starch press, 2012.
2. Filiol, Eric, "*Computer Viruses: From Theory to Applications*", Springer Science & Business Media, 2006.
3. Ligh, Michael, Steven Adair, Blake Hartstein, and Matthew Richard, "*Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code*", Wiley Publishing, 2010.

CS419	FORMAL METHODS IN SOFTWARE ENGINEERING	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Model the state of a software component using the unifying concept of mathematical relation
CO2	Design of automatic verification tools to establish the validity of a given software property
CO3	Apply automatic software verification tools based on model checking
CO4	Design tools for the deductive verification of programs annotated with contracts

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	3	2	2	2	1	3	2	3	1	3	3	2	3
CO2	3	2	3	2	2	2	2	1	3	3	3	1	3	3	2	3
CO3	3	2	3	2	3	2	2	1	3	2	3	1	3	3	2	3
CO4	3	2	3	2	3	2	2	1	3	3	3	1	3	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Specification and Modeling:
Introduction: the role of formal methods in software engineering; the role of abstraction in formal modeling; propositional and first-order logic.
Requirement-Specification-Analysis, Formal Specification,Petri Net.
Relational logic: syntax and semantics; modeling using relations; introduction to the relational calculus; taxonomy and relational algebra.
Alloy: specification of invariants and operations using pre- and post-conditions using relational logic; idioms for modeling dynamic behaviour; semantics and type system; automatic verification techniques; comparison with other modeling languages.
Specification of reactive systems: temporal logic (LTL and CTL); SPIN; explicit state model checking; symbolic model checking; tools for model checking.
Theorem proving: Introduction to the interactive construction of proofs.
First order theories: employing SMT solvers.
Deductive verification: program logics; verification condition generation; behavioral interface specification languages and design by contract. Tools covered: Dafny; Frama-C; SPARK.
Model Checking: symbolic model checking; partial order reduction; bounded model checking. Tool covered: SMV. Software Model Checking: bounded model checking of software; existential abstraction mechanisms; predicate abstraction; abstraction refinement. Tools covered: CBMC; BLAST
Fuzzing: Black-box, Grey-box, and White-box fuzzings. Smart Fuzzers. Tool: AFL
Symbolic Execution: Dynamic Symbolic Execution, Dynamic Symbolic Execution with pruning. Tools: KLEE, TracerX.
Text Books / Reference Books / Online Resources:

1. Daniel Jackson, "*Software Abstractions: Logic, Language, and Analysis*", Revised edition, MIT Press, 2012.
2. Christel Baier and Joost-Pieter Katoen, "*Principles of Model Checking*", MIT Press, 2008.
3. Michael Huth and Mark Ryan, "*Logic in Computer Science: Modelling and Reasoning about Systems*", Cambridge University Press, New York, NY, USA. 2004.
4. Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled, "*Model Checking*", MIT Press, Cambridge, MA, USA. 2000.
5. José Bacelar Almeida, Maria João Frade, Jorge Sousa Pinto, and Simão Melo de Sousa, "*Rigorous Software Development: An Introduction to Program Verification*", (1st ed.). Springer Publishing Company, Incorporated. 2011.
6. Aaron R. Bradley and Zohar Manna, "*The Calculus of Computation: Decision Procedures with Applications to Verification*" (1st ed.). Springer Publishing Company, Incorporated 2010.

CS420	ADVANCED DATABASES	Credits 3-0-0: 3

Pre-requisites: Database Management systems
Course Outcomes:
At the end of the course, the student will be able to

CO1	Design distributed database for application development.
CO2	Apply query optimization principles for optimizing query performance in centralized and distributed database systems
CO3	Design distributed database schema using principles of fragmentation and allocation.
CO4	Apply distributed transaction principles for handling transactions in distributed database applications.
CO5	Apply distributed database administration principles for managing distributed database.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	3	1	1	-	-	1	1	1	1	1	3	2	-	1
CO2	2	2	2	1	1	-	-	1	-	-	1	1	1	1	-	2
CO3	1	2	3	1	1	-	-	1	1	1	1	1	3	2	-	1
CO4	1	1	1	1	2	-	-	1	1	1	1	1	1	2	1	1
CO5	-	1	1	1	2	-	-	1	1	1	1	1	1	2	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Distributed Databases: Introduction to Distributed Database Systems, Distributed Database System Architecture; Top-Down Approach, Distributed Database Design Issues, Fragmentation, Allocation, Database Integration, Bottom-up approach, Schema Matching, Schema Integration, Schema Mapping; Data and Access Control, View Management, Data Security; Query processing problem, Objectives of Query processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing; Query Decomposition, Normalization, Analysis, Elimination of Redundancy and Rewriting; Localization of Distributed Data, Reduction for primary Horizontal, Vertical, derived Fragmentation; Distributed Query Execution, Query Optimization, Join Ordering, Static & Dynamic Approach, Semi-joins, Hybrid Approach;
Distributed Concurrency Control: Taxonomy of Concurrency control Mechanisms, Lock-Based Concurrency Control, Timestamp-Based Concurrency Control, Optimistic Concurrency Control, Deadlock Management; Heterogeneity issues Advanced Transaction Models, Distributed systems 2PC & 3PC protocols, Replication protocols, Replication and Failures, HotSpares; Parallel Databases: Introduction to Parallel Databases, Parallel Database System Architectures, Parallel Data Placement, Full Partitioning; Parallel Query Processing, Query Parallelism; Parallel Query Optimization, Search Space, Cost Model, Search Strategy; Load Balancing.

Text Books / Reference Books / Online Resources:

1. M T Ozsu, Patrick Valduriez, "*Principles of Distributed Database Systems*", Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, "*Distributed Database System Principles and Systems*", MGH, 1985.

CS421	**COMPUTATIONAL LEARNING THEORY**	**Credits 3-0-0: 3**

Pre-requisites: Design and Analysis of Algorithms, Machine Learning

Course Outcomes:

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

CO1	Develop algorithms and methods to analyse learning algorithms
CO2	Design algorithms that are computationally efficient with limited amount of data to understand statistical and computational trade-offs in learning algorithms
CO3	Design models to address relevant practical questions of the day, such as learning with limited memory, communication, and labelled and unlabelled data
CO4	Develop algorithms to address privacy in learning algorithms.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	3	2	2	1	3	3	3	1	3	2	3	2
CO2	3	3	3	3	2	2	2	1	2	2	2	1	3	3	2	3
CO3	3	3	3	3	2	2	2	1	2	2	2	1	3	3	2	3
CO4	3	3	3	3	3	2	2	1	3	3	3	1	3	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The Probably Approximately Correct Learning Model - General Mode, Learning Boolean Conjunctions Intractability of Learning 3-Term DNF Formulae, Using 3-CNF, Formulae to Avoid Intractability

Occam's razor - Occam Learning and Succinctness, Improving the Sample Size for Learning Conjunctions, Learning Conjunctions with Few Relevant Variables, Learning Decision Lists

Vapnik-Chervonenkis Dimension - Learning with a Finite Sample, Examples of the VC Dimension, Polynomial Bound - A Polynomial Bound on the Sample Size for PAC Learning - Importance of f-Nets, A Small f-Net from Random Sampling, Sample Size Lower Bounds, Application to Neural Networks

Weak and Strong Learning - Relaxed Definition of Learning, Boosting the Confidence, Boosting the Accuracy - A Modest Accuracy Boosting Procedure, Error Analysis for the Modest Procedure, A Recursive Accuracy Boosting Algorithm, Bounding the Depth of the Recursion, Analysis of Filtering Efficiency

Learning in the Presence of Noise - Classification Noise Model, Algorithm for Learning Conjunctions from Statistics, Statistical Query Learning Model, Simulating Statistical Queries in the Presence of Noise - Nice Decomposition of Px, Solving for an Estimate of Px. Guessing and Verifying the Noise Rate, Description of the Simulation Algorithm

Inherent Unpredictability - Representation Dependent and Independent Hardness, Discrete Cube Root Problem, Difficulty of Discrete Cube Roots, Discrete Cube Roots as a Learning Problem, Small Boolean Circuits Are Inherently Unpredictable, Reducing the Depth of Inherently Unpredictable Circuits- Expanding the Input, General Method and Its Application to Neural Networks

Reducibility in PAC Learning - Reducing DNF to Monotone DNF, General Method for Reducibility, Reducing

Boolean Formulae to Finite Automata - Learning Finite Automata by Experimentation, Active and Passive Learning, Exact Learning Using Queries, Exact Learning of Finite Automata - Access Strings and Distinguishing Strings, Efficiently Computable State Partition, Tentative Hypothesis, Using a Counterexample, Algorithm for Learning Finite Automata, Running Time Analysis; Learning without a Reset - Using a Homing Sequence to Learn, Building a Homing Sequence Using Oversized Generalized, Classification Trees, No-Reset Algorithm, Making Sure L_sigma Builds Generalized Classification Trees.

Text Books / Reference Books / Online Resources:

1. Michael Kearns and Umesh Vazirani, "*An Introduction to Computational Learning Theory*", MIT Press, 1994.

2. David Mackay, "*Information Theory, Inference and Learning Algorithms*", Cambridge University Press, 2003.
3. Shai Shalev-Shwartz and Shai Ben-David, "*Understanding Machine Learning: From Theory to Algorithms"*, Cambridge University Press, 2014.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, "*The Elements of Statistical Learning*", Second edition, Springer, 2008

CS422	ADVANCED SOFTWARE TESTING	Credits 3-0-0: 3

Pre-requisites: Software Engineering

Course Outcomes:

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

CO1	Analyze Various test processes and continuous quality improvement
CO2	Analyze Types of errors and fault models
CO3	Modeling the behavior using FSM
CO4	Application of software testing techniques in commercial environments
CO5	Analyze various test tools

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	1	2	1	1	1	2	2	2	2	1	2	1	3	3
CO2	2	2	2	3	2	1	1	1	1	1	2	1	2	1	3	3
CO3	3	1	2	3	2	1	1	1	2	1	2	1	3	3	3	3
CO4	2	2	1	3	3	2	2	2	2	1	2	1	3	2	2	3
CO5	2	2	1	2	1	1	1	1	1	2	1	1	2	1	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Software Testing: Introduction, Evolution of Software Testing, Software Testing—Myths and Facts, Goals of Software Testing, Psychology for Software Testing, Software Testing Definitions, Model for Software Testing, Effective Software Testing vs. Exhaustive Software Testing, Effective Testing is Hard, Software Testing as a Process, Schools of Software Testing, Software Failure Case Studies. Software Testing Terminology and Methodology: Software Testing Terminology, Software Testing Life Cycle (STLC), Software Testing Methodology. Verification and Validation: Verification and Validation (V&V) Activities, Verification, Verification of Requirements, Verification of High-level Design, Verification of Low-level Design, How to Verify Code?, Validation. Dynamic Testing: Black-Box Testing Techniques: Boundary Value Analysis (BVA), Equivalence Class Testing, State Table-Based Testing, Decision Table-Based Testing, Cause-Effect Graphing Based Testing, Error Guessing. Dynamic Testing: White-Box Testing Techniques: Need of White-Box Testing, Logic Coverage Criteria, Basis Path Testing, Graph Matrices, Loop Testing, Data Flow Testing, Mutation Testing. Static Testing: Inspections, Structured Walkthroughs, Technical Reviews. Validation Activities: Unit Validation Testing, Integration Testing, Function Testing, System Testing, Acceptance Testing. Regression Testing: Progressive vs. Regressive Testing, Regression Testing Produces Quality Software, Regression Testability, Objectives of Regression Testing, When is Regression Testing Done?, Regression Testing Types, Defining Regression Test Problem, Regression Testing Techniques. Test Management: Test Organization, Structure of Testing Group, Test Planning, Detailed Test Design and Test Specifications. Software Metrics: Need of Software Measurement, Definition of Software Metrics, Classification of Software Metrics, Entities to be Measured, Size Metrics. Testing Metrics for Monitoring and Controlling the Testing Process: Measurement Objectives for Testing, Attributes and Corresponding Metrics in Software Testing, Attributes, Estimation Models for Estimating Testing Efforts, Architectural Design Metric Used for Testing, Information Flow Metrics Used for Testing, Cyclomatic Complexity Measures for Testing, Function Point Metrics for Testing, Test Point Analysis (TPA), Some Testing Metrics. Efficient Test Suite Management: Why Does a Test Suite Grow?, Minimizing the Test Suite and its Benefits, Defining Test Suite Minimization Problem, Test Suite Prioritization, Types of Test Case Prioritization, Prioritization Techniques, Measuring the Effectiveness of a Prioritized Test Suite

Text Books / Reference Books / Online Resources:

1. Naresh Chauhan, "*Software Testing- Principles and practices* ", Oxford University Press, Second Edition, 2016

Jamie L. Mitchell and Rex Black, "*Advanced Software Testing*" Vol. 3, Second Edition, 2015

CS423	SERVICE-ORIENTED ARCHITECTURE	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand software oriented architectures
CO2	Design medium scale software project development using SOA principles
CO3	Develop SOA messages from business use cases
CO4	Design and implementation of modern SOA and SOA-specific methodologies, technologies and standards
CO5	Create composite services by applying composition style

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3	2	3	1	2	2	1	2	2	2	1	3	2	3	2
CO2	2	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3
CO3	3	2	3	2	2	2	3	1	3	3	3	1	2	3	2	3
CO4	3	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3
CO5	3	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction to SOA, Evolution of SOA: Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits 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 – I Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. Web Services and Contemporary SOA-2: 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.
Business Process Design: WS-BPEL language basics; WS Coordination overview; Service oriented business process design; WS addressing language basics; WS Reliable Messaging language basics.
SOA Platforms: SOA platform basics; SOA support in J2EE; SOA support in. ET; Integration considerations

Text Books / Reference Books / Online Resources:

1. Thomas Erl, "*Service-Oriented Architecture: Concepts, Technology and Design*", Prentice Hall Publication, 2005.
2. Michael Rosen, Boris Lublinsky, "*Applied SOA Service Oriented Architecture and Design Strategies*", Wiely India Edition, 2008.

CS424	SECURE SOFTWARE ENGINEERING	Credits 3-0-0: 3

Pre-requisites: None
Course Outcomes:
At the end of the course, the student will be able to

CO1	Evaluate secure software engineering problems, including the specification, design, implementation, and testing of software systems
CO2	Elicit, analyze and specify security requirements through SRS
CO3	Design and Plan software solutions to security problems using various paradigms
CO4	Model the secure software systems using Unified Modeling Language Sec(UMLSec)
CO5	Develop and apply testing strategies for Secure software applications

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	2	2	1	1	1	2	1	1	3	3	2	-
CO2	2	2	1	2	2	2	1	1	1	2	1	1	2	2	3	-
CO3	2	3	3	3	2	2	2	1	2	3	2	1	3	3	2	-
CO4	2	2	3	2	3	2	2	1	2	2	2	1	3	2	2	-
CO5	2	2	3	2	3	2	2	1	2	2	2	1	3	3	2	-

1 - Slightly; 2 - Moderately; 3 – Substantially

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

Text Books / Reference Books / Online Resources:

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.

CS425	DESIGN PATTERNS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand common design patterns in the context of incremental/iterative development
CO2	Evaluate and retractor software source code using patterns
CO3	Analyze and combine design patterns to work together in software design
CO4	Implement the design patterns in an object oriented language.
CO5	Understand the benefits of a pattern approach over program in a software application.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3	2	3	2	2	2	1	2	3	2	1	2	2	2	1
CO2	2	2	3	2	2	2	2	1	3	3	3	1	2	2	2	2
CO3	2	3	3	3	2	2	3	1	2	3	2	1	1	2	3	2
CO4	2	2	3	2	3	3	2	1	3	3	3	1	3	3	2	1
CO5	2	2	2	2	1	2	2	1	2	2	2	1	2	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction: What is a Design Pattern, Design Patterns in Smalltalk MVC, Describing Design Patterns, the Catalogue of Design Patterns, Organizing the catalogue, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

A Case Study: Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, and Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation.

Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton.

Structural Pattern: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.

Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, a Brief History, and the Pattern Community

Text Books / Reference Books / Online Resources:

1. Erich Gamma, "*Design Patterns*", Addison-Wesley, 1994.
2. Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "*Pattern-Oriented Software Architecture: A System of Pattern*", John Wiley & Sons; 1996.

CS426	PROGRAM ANALYSIS AND VERIFICATION	Credits 3-0-0: 3

Pre-requisites: Discrete Mathematics, Programming Language Concepts, Language Processors
Course Outcomes:
At the end of the course, the student will be able to

CO1	Apply the theory of abstract interpretation.
CO2	Examine existing techniques
CO3	Combine algorithms for program analysis
CO4	Experiment with Soot and Java software packages

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2	2	2	1	1	3	2	3	1	3	3	2	2
CO2	2	2	2	2	3	2	1	1	3	2	3	1	2	2	3	1
CO3	3	3	3	3	3	2	1	1	3	2	3	1	3	2	2	2
CO4	2	2	2	2	3	2	1	1	2	2	2	1	2	2	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:
Introduction - Nature of Program Analysis, Data Flow Analysis, Equational Approach, and Constraint Based, Type and Effect Systems, Effect Systems, Algorithms.
Data Flow Analysis – Intraprocedural Analysis, Available Expressions Analysis, Reaching Definitions Analysis, Very Busy Expressions Analysis, Live Variables Analysis, Structural Operational Semantics, Correctness of Live Variables Analysis, Monotone Frameworks, Equation Solving, Interprocedural Analysis, Shape Analysis.
Constraint Based Analysis - Abstract 0-CFA Analysis, Theoretical Properties, Constraint Based 0-CFA Analysis, Adding Context Information.
Abstract Interpretation – Correctness, Approximation of Fixed Points, Galois Connections, Induced Operations.
Type and Effect Systems - Control Flow Analysis, Theoretical Properties, Inference Algorithms, Effects, Behaviours.
Algorithms - Worklist Algorithms, Iterating in Reverse Postorder.

Text Books / Reference Books / Online Resources:

1. Flemming Nielson, Hanne R. Nielson and Chris Hankin, "*Principles of Program Analysis*", Springer, 2005.
2. Edmund M. Clarke, Jr., Orna Grumberg, Daniel Kroening, Doron Peled and Helmut Veith, "*Model Checking*", MIT Press, Second Edition, 2018
3. Aaron R. Bradley and Zohar Manna, "*The Calculus of Computation*", Springer, 2007
4. Daniel Kroening and Ofer Strichman, "*Decision Procedures: An Algorithmic Point of View*", Springer, 2008

CS427	ALGORITHMIC TECHNIQUES FOR BIG DATA	Credits 3-0-0: 3

Pre-requisites: CS301-Operating systems, CS304- Computer Networks
Course Outcomes:
At the end of the course, the student will be able to

CO1	Identify various statistical information required to be drawn from Big Data
CO2	Develop mathematical models for Big Data
CO3	Design sub-linear randomized algorithms and analyze their quality of answer.
CO4	Derive lower bounds for randomized algorithms

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	3	2	2	1	2	1	1	-	2	3	3	2
CO2	3	2	2	3	3	3	3	1	1	1	1	-	2	3	3	1
CO3	3	3	3	3	3	3	3	2	1	1	1	-	3	2	3	3
CO4	2	2	2	2	2	2	1	2	1	1	1	-	2	1	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Preliminaries: The Data Model, Basic Setup, Quality of an Algorithm's Answer, Variations of the Basic Setup

Finding Frequent Items Deterministically: Problem, Frequency Estimation, The Misra–Gries Algorithm, Analysis of the Algorithm,

Estimating the Number of Distinct Elements: Problem, Tidemark Algorithm, Quality of the Algorithm's Estimate, Median Trick, A Better Estimate for Distinct Elements - BJKST Algorithm, Analysis: Space Complexity, Analysis: The Quality of the Estimate, Optimality

Approximate Counting: Problem, Algorithm, Quality of the Estimate, Median-of-Means Improvement

Finding Frequent Items via (Linear) Sketching: Problem, Sketches and Linear Sketches, Count Sketch, Quality of the Basic Sketch's Estimate, Count-Min Sketch, Quality of the Algorithm's Estimate, Comparison of Frequency Estimation Methods

Estimating Frequency Moments: Background and Motivation, AMS Estimator for Fk, Analysis of the Basic Estimator, Estimator and Space Bound, Soft-O Notation, Tug-of-War Sketch, Sketch, Quality of the Estimate Geometric Interpretation

Estimating Norms Using Stable Distributions: Median of a Distribution and its Estimation Accuracy of the Estimate

Weight-Based Sampling: Problem, Lo-Sampling, Idealized Algorithm, Quality of the Output. L2 Sampling - L2-sampling Algorithm, Analysis

Finding the Median: Problem, Munro–Paterson Algorithm, Computing a Core, Utilizing a Core, Analysis: Pass/Space Tradeoff

Geometric Streams and Coresets: Extent Measures and Minimum Enclosing Ball, Coresets and Their Properties, Coreset for MEB , Data Stream Algorithm for Coreset, Construction

Metric Streams and Clustering: Metric Spaces, Cost of a Clustering: Summarization Costs, Doubling Algorithm, Metric Costs and Threshold Algorithms, Guha's Cascading Algorithm, Space Bounds, Quality of the Summary.

Graph Streams: Basic Algorithms, Streams that Describe Graphs, Semi-Streaming Space Bounds , Connectedness Problem, Bipartiteness Problem, Shortest Paths and Distance Estimation via Spanners, Quality of the Estimate, Space Complexity: High-Girth Graphs and the Size of a Spanner .

Finding Maximum Matchings: Maximum Cardinality Matching, Maximum Weight Matching

Graph Sketching: Value of Boundary Edges, Testing Connectivity Using Boundary Edges , Testing Bipartiteness, The AGM Sketch: Producing a Boundary Edge, Counting Triangles, Sampling-Based Algorithm, Sketch-Based Algorithm

Communication Complexity and Lower Bounds: Communication Games, Protocols, Complexity, Specific Two-Player Communication Games, Definitions, Results and Some Proofs: Deterministic Case, Proofs: Randomized Case, Data Streaming Lower Bounds, Lower Bound for Majority, Lower

Bound for Frequency Estimation, Further Reductions and Lower Bounds - Importance of Randomization, Multi-Pass Lower Bounds for Randomized Algorithms, Graph Problems, importance of approximation, Space verses approximation quality.

Text Books / Reference Books / Online Resources:

1. Oded Goldreich, "*Introduction to Property Testing"*, Cambridge University Press, 2017.
2. Jure Leskovec, Anand Rajaraman, Jeff Ullman, "*Mining of Massive Datasets*", Dreamtech Press, 2016
3. Amit Chakrabarthy, "*Data Stream Algorithms"*, Lecture Notes, Dartmouth College, 2020
4. Relevant research papers

CS428	DESIGN OF SECURE PROTOCOLS	Credits 3-0-0: 3

Pre-requisites: Design and Analysis of Algorithms, Cryptography, Network Security, Computer Networks, Operating System

Course Outcomes:

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

CO1	Identify security goals and risks,
CO2	Analysis of risks and threat modeling
CO3	Integrate different technologies to achieve security goals
CO4	Develop security protocols and policies
CO5	Implement security protocols and secure coding

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO2	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO3	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO4	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO5	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Needham-Schroeder public-key protocol. Introduction to finite-state checking, SSL/TLS case study, IP security. Internet Key Exchange (IKE) protocol, Introduction to process algebra, Just Fast Keying (JFK) protocol, Security as observational equivalence. JFK protocol in applied pi calculus, Protocols for anonymity, Probabilistic model checking, Probabilistic contract signing protocols, Floyd-Hoare logic. Compositional protocol logic, Paulson's inductive method, Analyzing SET with the inductive method, Symbolic constraint solving, Formal definitions of security for symmetric ciphers, Formal model for secure key exchange, Simulatability-based proofs of protocol security, Probabilistic polynomial-time process calculus, Formal analysis of denial of service, Formal verification of routing protocols, Computational soundness of formal models, Multicast security, Spoofing and identity theft, Fair exchange and contract signing protocols, Trusted computing, Privacy preserving data mining, Automatic proofs of strong secrecy, Game-based verification of contract signing protocols, Wireless security, Game-based analysis of denial-of-service protection, Analysis of Internet voting protocols, Privacy-preserving graph algorithms, Universal composability framework, Analysis of Group Diffie-Hellman protocols

Text Books / Reference Books / Online Resources:

1. Oded Goldreich, "*Foundations of Cryptography, Vol. I and II*", Cambridge University Press, 2007.
2. Jonathan Katz and Yahuda Lindell, "*Introduction to Modern Cryptography*", CRC press, 2008.
3. Van Oorschot, Paul Scott, A Vanstone, A J Menezes, "*Handbook of Applied Cryptography*", CRC Press, 2004.

CS5113	COMPUTER VISION AND IMAGE PROCESSING	Credits 3-0-0: 3

Pre-requisites: Mathematics-I, Mathematics-II

Course Outcomes:

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

CO1	Understand Image representation and modeling
CO2	Apply Image transformation methods
CO3	Implement image processing algorithms
CO4	Design of face detection and recognition algorithms
CO5	Analyze the features and propose new features of images.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2	2	2	2	1	2	2	2	1	3	2	3	2
CO2	3	2	3	2	2	2	3	1	3	3	3	1	3	3	2	2
CO3	3	2	3	2	3	2	3	1	3	3	3	1	2	3	2	2
CO4	2	2	3	2	2	2	3	1	3	3	3	1	3	3	2	2
CO5	2	3	2	3	1	2	2	1	2	2	2	1	3	2	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The image model and acquisition, image shape, sampling, intensity images, color images, range images, image capture, scanners. Statistical and spatial operations, Gray level transformations, histogram equalization, multi image operations. Spatially dependent transformations, templates and convolution, window operations, directional smoothing, other smoothing techniques. Segmentation and Edge detection, region operations, Basic edge detection, second order detection, crack edge detection, edge following, gradient operators, compass & Laplace operators. Morphological and other area operations, basic morphological operations, opening and closing operations, area operations, morphological transformations. Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression. Representation and Description, Object Recognition, 3-D vision and Geometry, Digital Watermarking. Texture Analysis.

Text Books/Reference Books/Online Resources:

1. D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, PHI Learning, 2009.
2. Milan Soanka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision, Cengage Learning, 2014
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Pearson Education, 2007.

CS5151	ADVANCED COMPUTER NETWORKS	Credits 3-0-0: 3

Pre-requisites: CS304 – Computer Networks

Course Outcomes:

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

CO1	Analyze computer network architectures and estimate quality of service
CO2	Design application-level protocols for emerging networks
CO3	Analyze TCP and UDP traffic in data networks
CO4	Design and Analyze medium access methods, routing algorithms and IPv6 protocol for data networks
CO5	Analyze Data Center Networks and Optical Networks

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	2	3	2	-	-	-	-	-	-	-	-	-	-	-
CO2	2	1	2	2	-	-	-	-	-	-	-	-	-	-	-	-
CO3	1	-	2	2	-	-	-	-	-	-	-	-	-	-	-	-
CO4	1	-	2	2	1	-	-	-	-	-	-	-	-	-	-	-
CO5	1	-	2	2	1	-	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Network Architecture, Performance: Bandwidth and Latency, High Speed Networks, Network-Centric View, Error Detection, Reliable Transmission, Ethernet and Multiple Access Networks, Overlay Networks: Routing Overlays, Peer-to-Peer Networks and Content Distribution Networks, Client-Server Networks, Delay-Tolerant Networks, Switching: Circuit-Switched Networks, Datagram Networks, Virtual-Circuit Networks, Message-Switched Networks, Asynchronous Transfer Mode: Evolution, Benefits, Concepts, Exploring Broadband Integrated Services Digital Network, Layer and Adaptation Layer, IPv4: Address Space, Notations, Classful, Classless, Network Address Translation, Datagram, Fragmentation and Checksum IPv6 Addresses: Structure, Address Space, Packet Format and Extension Headers, ICMP, IGMP, ARP, RARP, Congestion Control and Resource Allocation: Problem, Issues, Queuing, TCP Congestion Control, Congestion-Avoidance Mechanisms and Quality of Service, Internetworking: Intra-Domain and Inter-Domain Routings, Unicast Routing Protocols: RIP, OSPF and BGP, Multicast Routing Protocols: DVMRP, PIM-DM, PIM-SM, CBT, MSDP and MOSPF, Spanning Tree Algorithm, Optical Networking: SONET/SDH Standards, Traffic Engineering: Requirement, Traffic Sizing, Characteristics, Protocols, Time and Delay Considerations, Connectivity, Availability, Reliability and Maintainability and Throughput, Multimedia Over Internet: Transmission, IP Multicasting and VoIP, Domain Name System: Name Space, Domain Name Space, Distribution, Domains, Resolutions and Dynamic Domain Name System, SNMP, Security: IPSec, SSL/TLS, PGP and Firewalls, Datacenter Design and Interconnection Networks.

Text Books / Reference Books / Online Resources:

1. Larry L. Peterson and Bruce S. Davie, "*Computer Networks: A System Approach*", Fifth Edition, Morgan Kaufmann, Elsevier, 2012.
2. Behrouz A. Forouzan, "*Data Communications and Networking*", McGraw Hill, Fifth Edition, 2017.
3. Chwan-Hwa (John) Wu, J. David Irwin, "*Introduction to Computer Networks and Cyber Security*", CRC press, Taylor & Francis Group,2014
4. Andrew S. Tanenbaum, David J. Wetherall, "*Computer Networks*", Pearson, 5th Edition, 2014.
5. G. Wright and W. Stevens, "*TCP/IP Illustrated, Volume 1 and Volume 2*", Addison-Wesley, 1996.
6. Dayanand Ambawade, Deven Shah, Mahendra Mehra and Mayank Agarwal, "*Advanced Computer Networks*", Dreamtech Press, 2016.
7. R. Srikant, The Mathematics of Internet Congestion Control, Springer, 2004.
8. J. L. Boudec and P. Thiran, Network Calculus, Springer, 2011.

CS461	CYBER SECURITY	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand cyber security fundamentals
CO2	Analyze cyber security threats and vulnerabilities in Information Systems and apply security measures to real time scenarios
CO3	Design and implement appropriate security techniques and cyber policies to protect computers and digital information.
CO4	Demonstrate the use of standards and cyber laws to enhance information security in the development process and infrastructure protection

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	-	1	-	-	2	-	2	-	-	-	1	-	-	-	2	-
CO2	-	-	2	-	1	-	-	-	1	1	1	1	1	-	2	1
CO3	-	-	-	-	-	-	-	2	-	-	1	-	-	-	-	1
CO4	1	-	-	1	1	1	-	-	-	1	-	2	-	-	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Cyber Security: Overview of Cyber Security, Internet Governance – Challenges and Constraints, Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber terrorism-Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Need for an International convention on Cyberspace.

Cyber Security Vulnerabilities and Safeguards: Cyber Security Vulnerabilities-Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Unprotected Broadband communications, Poor Cyber Security Awareness. Cyber Security Safeguards- Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security policy, Threat Management.

Securing Web Application, Services and Servers: Introduction, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges.

Intrusion Detection and Prevention: Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation.

Cyber Forensics and Laws: Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, Conducting disk-based analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating E-mail header information, Tracing Internet access, Tracing memory in real-time. Cyberspace and the Law: Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, National Cyber Security Policies.

Text Books / Reference Books / Online Resources:

1. James Graham, Richard Howard, Ryan Olson, "*Cyber Security Essentials*", CRC Press, 2016.
2. Nina Godbole and Sunit Belapure, "*Cyber Security*", Wiley India, 2012.

CS462	MINING OF MASSIVE DATASETS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand and Design Similar items
CO2	Design Regression model to solve the classification and prediction problems
CO3	Design models for Recommendation Systems
CO4	Build Large-scale Machine Learning algorithms for classification and clustering of the data

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	2	2	3	1	1	1	1	-	3	2	2	1
CO2	3	3	3	3	2	3	2	1	1	1	1	-	3	2	3	2
CO3	3	3	2	3	2	2	2	1	1	1	1	-	2	2	2	1
CO4	3	3	3	3	2	3	2	1	1	1	1	-	3	2	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Data Mining: Data Mining Overview, **MapReduce Overview, Finding Similar Items:** Applications of Near-Neighbour Search, Shingling of Documents, Similarity-Preserving Summaries of sets, Locality-Sensitive Hashing for Documents, Distance Measures, The Theory of Locality-Sensitive Functions, Applications of Locality-Sensitive Hashing, Methods for High Degrees of Similarity; **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 Window; **Link Analysis:** PageRank, Efficient Computation of PageRank, Topic-Sensitive PageRank, Link Spam, Hubs and Authorities; **Frequent Itemsets:** Handling Larger Datasets in Main Memory, Limited-Pass Algorithms, Counting Frequent Items in a Stream; Clustering, **Recommendation Systems:** A Model for Recommendation Systems, Content-Based Recommendations, Collaborative Filtering, Dimensionality Reduction; Large-scale Machine Learning, Variety, Volume, Velocity, Scalable Parallel Artificial Neural Networks, Adaptive computation graph, ; **Mining Social-Network Graphs**

Text Books / Reference Books / Online Resources:

1. Jure Leskovec, Anand Rajaraman, Jeff Ullman, "*Mining of Massive Datasets*", Dreamtech Press, 2016

CS463	**REAL TIME SYSTEMS**	**Credits 3-0-0: 3**

Pre-requisites: None

Course Outcomes:

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

CO1	Understand the use of multi tasking techniques in real time systems.
CO2	Evaluate the performance of soft and hard real time systems.
CO3	Analyze multi task scheduling algorithms for periodic, aperiodic and sporadic tasks.
CO4	Design real time operating systems.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	-	2	-	-	-	-	-	-	-	-	2	-	-	-
CO2	3	3	3	-	-	3	-	2	-	-	-	-	2	-	-	2
CO3	2	2	3	2	-	3	-	-	-	-	-	-	2	-	-	2
CO4	3	3	3	3	-	3	-	2	2	-	-	-	2	2	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Real-Time Systems, Typical Real-Time Applications, Hard Versus Soft Real-Time Systems, A Reference Model of Real-Time Systems.

Commonly Used Approaches to Hard Real-Time Scheduling, Clock-Driven Scheduling, Priority-Driven Scheduling of Periodic Tasks, Scheduling Aperiodic and Sporadic Jobs in Priority- Driven Systems.

Resources and Resource Access Control, Multiprocessor Scheduling and Resource Access Control.

Scheduling Flexible Computations and Tasks with Temporal Distance Constraints.

Real-Time Communications, Operating Systems.

Text Books / Reference Books / Online Resources:

1. Jane Liu, “*Real-Time Systems*”, Prentice Hall, 2000.
2. Philip.A.Laplante, “*Real Time System Design and Analysis*”, 3rd Edition, PHI, 2004.

CS464	IOT SECURITY	Credits 3-0-0: 3

Pre-requisites: Internet of Things, Mobile Computing, Cryptography, Network Security

Course Outcomes:

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

CO1	Describe IoT general models and security challenges
CO2	Recognize IoT security and vulnerability threats.
CO3	Understand different IoT protocols and their security measures.
CO4	Apply techniques to secure an IoT environment
CO5	Interpret different IoT types of attacks.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	3	-	-	-	-	-	-	2	2	2	2	2	2
CO2	3	3	3	3	3	2	2	-	2	2	2	2	2	2	3	3
CO3	3	3	3	3	2	2	-	2	2	2	2	2	2	3	2	3
CO4	2	2	2	2	2	-	-	-	-	-	2	2	2	3	-	3
CO5	2	2	2	2	2	-	-	-	-	-	2	2	2	3	-	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

IoT Reference Model- Introduction -Functional View, IoT Security Challenges-Hardware Security Risks - Hardcoded/Default Passwords -Resource Constrained Computations -Legacy Assets Connections - Devices Physical Security, Software Security Risks -Software Vulnerabilities -Data Interception - Identification of Endpoints -Tamper Detection, Lack of Industrial Standards.

IoT- Security &Vulnerability Issues: IoT Security Requirements -Data Confidentiality -Data Encryption -Data Authentication -Secured Access Control –IoT-Vulnerabilities – Secret-Key, Authentication/Authorization for Smart Devices - Constrained System Resources -Device Heterogeneity -Fixed Firmware.

Secured Protocols For IoT Infrastructure-IPv6 -LowPAN , Identification-Electronic Product Code -uCode, , Cryptographic controls for IoT communication protocols- ZigBee, Bluetooth-LE, Near field communication (NFC), Cryptographic controls for IoT messaging protocols- MQTT, CoAP, DDS, REST, Multi-layer Frameworks-Alljoyn,-IoTivity

Securing Internet of Things Environment IoT Hardware -Test Device Range-Latency and Capacity -Manufacturability Test -Secure from Physical Attacks, IoT Software -Trusted IoT Application Platforms, -Secure Firmware Updating -Network Enforced Policy -Secure Analytics, Visibility and Control.

IoT Attacks - Side-channel Attacks -Reconnaissance -Spoofing -Sniffing -Neighbour -Discovery -Rogue Devices-Man-in-Middle, CASE STUDY MIRAI Botnet Attack -Iran's Nuclear FacilityStuxnet Attack -TeslaCryptojacking Attack -The TRENDnet WebcamAttack -The JeepSUV Attack -The Owlet Wi-Fi Baby Heart Monitor Vulnerabilities -St.Jude_Hackable Cardiac Devices.

Text Books / Reference Books / Online Resources:

1. B. Rusell and D. Van Duren, “*Practical Internet of Things Security*”, Packt Publishing, 2016
2. T. Alpcan and T. Basar, “*Network Security: A Decision and Game-theoretic Approach*”, Cambridge University Press, 2011.
3. Olivier Hersent, “*The Internet of Things Key Applications and Protocols*”, Wiley, 2012.
4. Research Papers

CS465	CLUSTER COMPUTING	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand the architecture of cluster computer and its components
CO2	Know the setup of cluster and its administration
CO3	Analyze dependability trade-offs and the limits of computer system dependability
CO4	Gain knowledge in sources of faults and their prevention and impacts
CO5	Analyze load sharing and balancing algorithms and their performance.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	-	3	3	-	-	-	-	-	-	-	-	-	-	-	-
CO2	3	-	3	3	-	-	-	-	-	-	-	-	-	-	-	-
CO3	3	-	3	3	-	-	-	-	-	-	-	-	-	-	-	-
CO4	3	-	3	3	-	-	-	-	-	-	-	-	-	-	-	-
CO5	3	-	3	3	-	-	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Cluster Computing, Scalable Parallel Computer Architectures, Low Cost Parallel Computing, Cluster Computer and its Architecture, Classifications, Components for Clusters, Network Services, Cluster Middleware and Single System Image, Resource Management and Scheduling, Programming Environments and Tools, Applications, Representative Cluster Systems, Cluster of SMPs, Cluster Setup and its Administration: Setting up the Cluster, Security, System Monitoring and System Tuning, Constructing Scalable Services: Environment, Resource Sharing, Locality, Prototype Implementation and Extension, Dependable Clustered Computing: Dependable Parallel Computing, Critical Computing, Dependability, Cluster Architectures, Detecting and Masking Faults and Recovering from Faults, High Throughput Computing Cluster: Condor, Software Development and System Administration, Performance Models and Simulation: Performance Issues and Cost Model for Effective Parallel Computing, Metacomputing: Evolution, Design Objectives and Issues, Projects and Environments. Process Scheduling, Load Sharing and Balancing: Evolution, Job and Resource Management Systems, State-of-the-Art in RMS and Challenges, Scheduling Parallel Jobs on Clusters: Job Types, Requirements, Rigid Jobs with Process Migration, Malleable Jobs with Dynamic Parallelism, Communication-Based Coscheduling and Batch Scheduling, Load Sharing and Fault Tolerance Manager: Fault Tolerance by Means Checkpointing and Integration, Parallel Program Scheduling Techniques: Scheduling Problem for Network Computing, Scheduling Tasks to Machines Connected via Fast Networks: ISH, MCP and ETF, Scheduling Tasks to Arbitrary Processors Networks: MH, DLS and BSA, CASCH, Dynamic Load Balancing, Mapping and Scheduling, Task Granularity and Partitioning, Static and Dynamic Scheduling.

Text Books / Reference Books / Online Resources:

1. R. Buyya, "*High Performance Cluster Computing: Architectures and Systems*", Vol. 1, 2007.
2. I. Koren and C. M. Krishna, "*Fault Tolerant Systems*", Morgan Kauffman, 2007.

CS466	INTELLIGENT AGENTS	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Demonstrate the basic concepts, representations, and algorithms for Planning.
CO2	Develop reinforcement learning model for real world problems.
CO3	Make use of language models for various NLP tasks.
CO4	Model Agent's inputs perception and preprocessing techniques.
CO5	Design and implement robot hardware and software.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	-	-	-	-	-	-	-	1	2	-	-	2
CO2	2	2	2	2	1	1	-	-	-	-	-	1	2	2	-	1
CO3	2	1	1	2	1	1	-	-	-	-	-	1	1	1	-	1
CO4	2	1	2	2	1	-	-	-	1	-	-	1	1	1	-	1
CO5	2	2	2	1	1	1	1	1	1			1	2	2		1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

INTELLIGENT AGENTS – Agents and environments, Good behavior: The concept of rationality, The nature of environments, The structure of agents. CLASSICAL PLANNING - Definitions, Algorithms, Planning graphs, Classical planning approaches, Analysis. PLANNING AND ACTING IN THE REAL WORLD – Time, schedule and resources, Hierarchical planning, Planning and acting in nondeterministic domains, Multiagent planning. KNOWLEDGE REPRESENTATION – Ontological engineering, Categories and objects, Events, Mental events and mental objects, Reasoning systems for categories, Reasoning with default information, The Internet shopping world. REINFORCEMENT LEARNING – Introduction, Passive and active reinforcement learning, Generalization in reinforcement learning, Policy search, Applications of reinforcement learning. NATURAL LANGUAGE PROCESSING – Language models, Text classification, Information retrieval, Information extraction. NATURAL LANGUAGE FOR COMMUNICATION – phrase structure grammars, syntactic analysis, Augmented grammars and semantic interpretation, Machine translation, Speech recognition. PERCEPTION – Image formation, Early image processing operations, Object recognition by appearance, Reconstructing the 3D world, Object recognition from structural information, Using vision. ROBOTICS – Introduction, Robot hardware, Robotic perception, Planning to move, Planning uncertain movements, Moving, Robotic software architectures, Application domains.

Text Books / Reference Books / Online Resources:

1. Stuart Russell, Peter Norvig, "*Artificial Intelligence -A Modern Approach*", 2/e, Pearson, 2003.
2. Nils J Nilsson, "*Artificial Intelligence: A New Synthesis*", Morgan Kaufmann Publications, 2000.

CS467	VIRTUAL REALITY AND AUGMENTED REALITY	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Apply concepts of Virtual Reality
CO2	Apply concepts of Augmented Reality
CO3	Integrate sensors with AR/VR system
CO4	Design AR/VR application for a given task
CO5	Analyze existing AR/VR systems

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	2	2	-	-	-	-	-	-	-	1	1	-	-
CO2	2	2	2	2	2	-	-	-	-	-	-	-	1	1	-	-
CO3	2	2	2	1	2	-	-	-	-	-	-	-	-	2	-	-
CO4	3	2	2	3	3	-	-	-	-	1	-	-	2	2	2	2
CO5	2	3	-	3	-	-	-	-	-	-	-	-	-	-	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Virtual Reality, History, Overview of Various Realities, Immersion. Perception: Objective & Subjective Reality, Perceptual Modalities, Perception of Space & Time. Content Creation: Environmental design, Affecting Behavior, VR Content Creation. Interaction: VR Interaction Concepts, Input Devices, Interaction Patterns & Techniques. Iterative Design of VR: Philosophy of Iterative Design, The Define stage, Make stage and Learn stage.

Software Development in Virtual Reality.

Introduction to Augmented Reality, Displays, Tracking, Computer Vision for Augmented Reality, Calibration & Registration, Visualization, Interaction, Modeling & Annotation, Authoring, Navigation, Collaboration.

Software Development in Augmented Reality.

Applications of VR/AR in Entertainment, Medical, Manufacturing, Education, etc., Future of VR/AR.

Text Books / Reference Books / Online Resources:

1. Jason Jerald, "*The VR Book: Human-Centered Design for Virtual Reality*", ACM and Morgan & Claypool Publishers, 2015.
2. Dieter Schmalstieg, Tobias Hollerer, "*Augmented Reality, Principles and Practice*", Addison Wesley (Pearson Education), 2016.
3. Jesse Glover and Jonathan Linowes, "*Complete Virtual Reality and Augmented Reality Development with Unity*", Packt Publishers, 2019.

CS468	SOCIAL NETWORKS	Credits 3-0-0: 3

Pre-requisites: Data Warehousing and Data Mining, Data Structures and Algorithms
Course Outcomes:
At the end of the course, the student will be able to

CO1	Understand the importance of social networks and social graphs
CO2	Enhance analytical skills for analyzing social networking data
CO3	Develop skills to leverage extended enterprise data
CO4	Create real-life case studies using social networks

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	-	-	1	-	1	-	-	1	-	-	1	-	1	-	-	1
CO2	1	-	-	2	2	3	2	-	-	1	1	-	-	-	2	-
CO3	1	2	2	-	1	-	2	-	-	2	-	2	-	2	-	2
CO4	-	1	1	2	-	2	-	1	2	-	-	1	-	3	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to social network analysis: Graphs – nodes, edges, direct and indirect friends/neighbors, degree and degree distribution, shortest path, cycle, tree, complete graph, bipartite graphs, directed graphs, weighted graphs, adjacency matrix, social interactions and connected components. Technological networks (internet, telephone network, power grids, transportation networks), social networks (facebook, movie collaboration, paper collaboration), information networks (web), biological networks (neural networks, ecological networks).

Network Centrality Measures and Models: Properties of real-world network – degree distribution, clustering coefficient, average path length; Random Graphs – Evolution of random graphs, properties of random graphs, modeling real-world networks with random graphs, Erdos-Renyi model of random graph; Small-world Model – Properties of the Small-world model, modeling real-world networks with the small-world model; Preferential attachment model – Properties of the preferential attachment model, modeling real-world networks with the preferential attachment model.

Random walk-based proximity measures, other graph-based proximity measures. Clustering with random-walk based measures.

Influence and Homophily: Measuring Assortativity, Measuring and modeling Influence, Measuring and modeling Homophily, Distinguishing influence and homophily – shuffle test, edge-reversal test, randomization test; Spread of influence through a network, influence maximization in networks, spread of disease on networks.

Games on networks, game theory strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, multiple equilibria-coordination games, multiple equilibria-the Hawk-Dove game, mixed strategies, Modeling social network traffic using game theory.

Text Books / Reference Books / Online Resources:

1. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, "*Social Media Mining – An Introduction*", Cambridge University Press, 2014.
2. David Easley and Jon Kleinberg, "*Networks, Crowds, and Markets: Reasoning About a Highly Connected World*", Cambridge University Press,2010.
3. Mark Newman, "*Networks: An Introduction*", Oxford University Press,2010.
4. Hansen, Derek, Ben Sheiderman, Marc Smith, "*Analyzing Social Media Networks with NodeXL: Insights from a Connected World*", Morgan Kaufmann,2011.
5. Avinash Kaushik, "*Web Analytics 2.0: The Art of Online Accountability*", Sybex,2009.

CS469	FOG AND EDGE COMPUTING	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

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

CO1	Understand the basic requirements of fog and edge computing.
CO2	Understand the key architectures and applications in fog and edge computing.
CO3	Perform fog and edge computing services.
CO4	Implement software using standard open-source fog and edge computing software for data analytics.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO2	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO3	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO4	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Fog Computing, Limitation of Cloud Computing, Differences between Cloud and Fog Computing, Advantages, Business Models, Architecture, Opportunities and Challenges, Challenges in Fog Resources: Taxonomy and Characteristics, Resource Management Challenge, Optimization Challenges, Miscellaneous Challenges, IoT and Fog: Programming Paradigms, Research Challenges and Research Directions, Fog Protocols, Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds, Data Management and Analysis in Fog Computing, Case Studies. Introduction to Edge Computing, Origins of Edge, Edge Helping Low-End IoT Nodes, Architecture, Edge Helping Higher-Capability Mobile Devices: Mobile Offloading, Edge Helping the Cloud, Edge for Augmented Reality, Data Processing on the Edge, Dispersed Learning with Edge/Fog Computing, Video Analytics on the Edge, Edge Computing Applications.

Text Books / Reference Books / Online Resources:

1. Rajkumar Buyya, Satish Narayana Srirama, "*Fog and Edge Computing*", Wiley Publications, 2019.
2. Wei Change and Jie Wu, "*Fog/Edge Computing for Security, Privacy and Applications*", Springer, 2021.