SCHEME AND SYLLABUS FOR

PROGRAM

B.TECH (INFORMATION TECHNOLOGY)

Offered by the Department of Computer Engineering,
National Institute of Technology, Kurukshetra



VISION AND MISSION OF THE INSTITUTE

VISION

To be a role-model in technical education and research, responsive to global challenges.

MISSION

To impart technical education that develops innovative professionals and entrepreneurs and to undertake research that generates cutting-edge technologies and futuristic knowledge, focusing on the socio-economic needs.

VISION AND MISSION OF THE DEPARTMENT

VISION

To address societal needs and global industry challenges in the field of Computer & IT with state-of-art education & research.

MISSION

M-1: To create a platform for education, research and development by providing sound theoretical knowledge and practical skills in Computer Engineering and Information Technology.

M-2: To produce motivated professional technocrats capable of generating solutions for industry and society.

M-3: To develop the ability to work ethically at individual and team level and be responsive towards socio-economic needs.

VISION AND MISSION OF THE PROGRAM

VISION

To disseminate state-of-the-art education to develop competent professionals in Computer Engineering with capability to serve the global society**.**

MISSION

To educate and train manpower engaged in cutting-edge research by offering latest in the field of Computer Engineering for sustainable development of society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

Program Educational Objectives (PEOs) of Undergraduate Programme of Information Technology are:

PEOs	Description
PEO 1	Graduates will have adequate knowledge of computer engineering domain to become employable in Industry.
PEO 2	Graduate will have strong fundamentals and problem solving skills to analyze, design and develop economically feasible solutions for technical and social problems.
PEO 3	Graduate will be aware of recent research trends, higher education and entrepreneurial opportunities, and will work ethically towards society.

PROGRAM SPECIFIC OUTCOMES (PSOs)

Program Specific Outcomes (PSOs) of B.Tech (Information Technology) are:

PEOs	Description
PSO 1	Augmented knowledge of Computer Science and technologies
PSO 2	Enhanced capability to develop computational tools and applications and improved skills to solve contemporary challenges
PSO 3	Exhibit attitude for continuous learning and deliver proactive solutions for futuristic challenges

PROGRAM OUTCOMES (POs)

Program Outcomes for Graduates of B.Tech (Information Technology) are:

PO	Description
PO 1	**Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**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 modeling to complex engineering activities with an understanding of the limitations.
PO6	**The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and Teamwork**: 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 the set to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary 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.

SEMESTER – I

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2025-26 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	-	Communication Skills in English OR Financial Education	2 0 2	3	4
2.	-	Differential Calculus and Differential Equations	3 0 0	3	3
3.	-	Engineering Physics	3 0 2	4	5
4.	ITIC 101	Problem Solving & Programming Skills I	3 0 2	4	5
6.	-	Energy & Environmental Science	2 0 2	3	4
7.	-	Engineering Practices (P)	1 0 3	2	4
8.	-	AU1	2 0 0	2	2
9.	-	Sports	0 0 2	1	2
10.	-	NCC/NSS/Yoga	0 0 2	1	2
Total				**21**	**31**

SEMESTER – II

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2025-26 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	-	Economics for Engineers OR Business Studies	3 0 0	3	3
2.	-	Integral Calculus and Difference Equations	3 0 0	3	3
3.	ITIC 100	Digital System Design	3 0 0	3	3
4.	ITIC 102	Web Design*	1 0 2	2	3
5.	ITIC 104	Programming Using Python	3 0 2	4	5
6.	ITPC 100	Data Structures	3 0 2	4	5
7.	-	AU2	2 0 0	2	2
8.	-	Sports	0 0 2	1	2
9.	-	NCC/Yoga/Clubs/NSS	0 0 2	1	2
Total				**21**	**28**

Note: The Syllabus of CSIC 11, CSIC 13, CSIC 12, CSIC 14, CSIC 16, CSPC 16 is now updated as CSIC 101, CSIC 103, CSIC 100, CSIC 102, CSIC 104 & CSPC 100 respectively.

*** Treated as a practical course (not integrated), evaluation will be as per practical course.**

SEMESTER -III

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2023-24 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	MAIC 201	Mathematics III	3 0 0	3	3
2.	ITPC 201	Design and Analysis of Algorithms	3 0 2	4	5
3.	ITPC 203	Computer Organization and Architecture	3 0 0	3	3
4.	ITPC 205	Object-Oriented Programming using Java	3 0 2	4	5
5.	ITPC 207	Software Engineering	3 0 2	4	5
6.	ITPC 209	IoT Programming #	1 0 2	2	3
7.	SWNC101	NCC/Sports/Yoga	0 0 4	1*	2*
8.	SWNC102	NSS/Club/Technical Societies	0 0 4	1*	2*
Total				**20**	**24**

 # To be treated as a practical course (not integrated), evaluation will be as per practical course.

SEMESTER -IV

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2023-24 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	ITPC 200	Operating Systems	3 0 2	4	5
2.	ITPC 202	Computer Networks	3 0 2	4	5
3.	ITPC 204	Artificial Intelligence and Soft Computing	3 0 2	4	5
4.	ITPC 206	Database Management Systems	3 0 2	4	5
5.	ITPE XXX	Department Elective – I	3 0 2	4	5
6.	SWNC101	NCC/Sports/Yoga	0 0 4	1*	2*
7.	SWNC102	NSS/Club/Technical Societies	0 0 4	1*	2*
Total				**20**	**25**

Sr. No.	Course Code	Department Elective – I Course Titles
1.	ITPE 210	Software Development using UML and Agile Methodology
2.	ITPE 212	Competitive Programming and Efficient Coding
3.	ITPE 214	Scripting Languages

SEMESTER -V

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2023-24 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	ITIC 301	Machine Learning	3 0 2	4	5
2.	ITPC 301	Theory of Computation	3 0 0	3	3
3.	ITPC 303	Data Mining and Analytics	3 0 2	4	5
4.	ITPC 305	Full Stack Development	3 0 2	4	5
5.	ITPC 307	Information Security	3 0 0	3	3
6.	-	Open Elective – I	3 0 0	3	3
7.	ITPC 309	Seminar and Group Discussion	0 0 2	1	2
Total				**22**	**26**

Open Elective – I to be started for all branches from V semester

Sr. No.	Course Code	Open Elective – I Course Titles for branches other than CO and IT
1.	ITOE 301	Data Structures
2.	ITOE 303	Machine Learning
3.	ITOE 305	Database Management Systems

SEMESTER -VI

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2023-24 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	ITPC 302	Big Data Analytics	3 0 2	4	5
2.	ITPC 304	Distributed Blockchain Technologies	3 0 0	3	3
3.	ITTC XXX	**Track Course-I**	3 0 2	4	5
4.	ITPE XXX	**Department Elective-II**	3 0 2	4	5
5.	ITPE XXX	**Department Elective-III**	3 0 0	3	3
6.	-	**Open Elective – II**	3 0 2	3	3
Total				**21**	**24**

Sr. No.	Course Code	Department Elective-II (Course Title)
1.	ITPE 302	Advanced Database Management System
2.	ITPE 304	Game Theory and Applications
3.	ITPE 306	Image Processing
4.	ITPE 308	Software Verification and Validation
5.	ITPE 310	Ethical Hacking
6.	ITPE 312	DevOps
7.	ITPE 326	Computer Vision

Sr. No.	Course Code	Department Elective-III (Course Title)
1.	ITPE 314	Bio-Informatics
2.	ITPE 316	Advanced Topics in Computer Networks
3.	ITPE 318	Software Project Management
4.	ITPE 320	Compiler Design
5.	ITPE 322	Stochastic Processes and Queuing Theory
6.	ITPE 324	Human Centered Computing

Sr. No.	Course Code	Open Elective - II (Course titles for branches other than CO and IT)
1.	ITOE 302	Design and Analysis of Algorithms (Pre-Requisite: Data Structures)
2.	ITOE 304	Operating Systems
3.	ITOE 306	Object-Oriented Programming using Java

Sr. No.	Track Course- I	Course Code	Course Title
1	Algorithms and Computing	ITTC 302	Randomized Algorithms and Probabilistic Analysis
		ITTC 304	High Performance Computing
		ITTC 306	Distributed Computing
2	Artificial Intelligence and Machine Learning	ITTC 308	Deep Learning
		ITTC 310	Knowledge Representation and Reasoning
		ITTC 318	Generative AI
3	Data Science	ITTC 312	Information Retrieval
		ITTC 314	Introduction to Statistical Modelling and R programming
		ITTC 316	Optimization Techniques for Data Science

SEMESTER -VII

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2022-23 admitted

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	IC	Entrepreneurship and Start-ups	3 0 0	3	3
2.	ITPE XXX	**Department Elective- IV**	3 0 2	4	5
3.	ITTC XXX	**Track Course II**	3 0 2	4	5
4.	-	**Open Elective- III**	3 0 0	3	3
5.	ITPC 401	Project	0 0 6	6#	6
Total				**20**	**22**

 # The credit assignment for Project is in accordance with those of contact hours of theory courses.

Sr. No.	Course Code	Department Elective- IV Course Title
1.	ITPE 407	Medical Image Computing
2.	ITPE 409	Modern Network Security
3.	ITPE 411	Number Theory and Cryptography
4.	ITPE 413	Advanced Data Structures and Algorithms

Sr. No.	Course Code	Open Elective- III (Course titles for branches other than CO and IT)
1.	ITOE 401	Advanced Data Structures and Algorithms (Pre-Requisite: Design and Analysis of Algorithms)
2.	ITOE 403	Computer Networks
3.	ITOE 405	Deep Learning (Pre-Requisite: Machine Learning)

Sr. No.	Track Course- II	Course Code	Course Title
1	Algorithms and Computing	ITTC 401	Parallel and Distributed Algorithms
		ITTC 403	Cloud Architecture and Networking
		ITTC 405	Quantum Computing
2	Artificial Intelligence and Machine Learning	ITTC 407	Cognitive applications of ML
		ITTC 409	Natural Language Processing
		ITTC 411	Semantic Web
3	Data Science	ITTC 413	Advances in Data Science
		ITTC 411	Semantic Web
		ITTC 415	Natured Inspired Optimization Techniques

SEMESTER -VIII (Internship/Course Work)

Scheme for B.Tech. (Information Technology)
w.e.f. Academic Year 2022-23 admitted

Students going for 6 months Internship are required to obtain at least 6 credits from the list of NPTEL MOOC /SWAYAM Courses approved by DAC:

Sr. No.	Course Title	Credits
1.	Internship (ITPC-400)	9
2.	XXX*	3
3.	XXX*	3
Total		**15**

*** Two courses from the list of NPTEL courses selected by DAC.**

All remaining students will study according to the following course structure.

Sr. No.	Course Code	Course Title	L_T_P	Credits	Contact Hrs
1.	ITPC 402	Cloud, IoT and Big Data Security	3 0 0	3	3
2.	ITPC 404	Mobile Application Development	3 0 2	4	5
3.	ITPC 406	Multicore and GPU Programming	3 0 0	3	3
4.	-	**Open Elective -IV**	3 0 0	3	3
5.	ITPC 408	Term Work	0 0 2	2	2
Total				**15**	**16**

Sr. No.	Course Code	Open Elective -IV Course titles for branches other than CO and IT
1.	ITOE 402	Artificial Intelligence
2.	ITOE 404	Information Security
3.	ITOE 406	Theory of Computation

Semester wise Program Credits

Semester	Credits
I	21
II	21
III	20
IV	20
V	22
VI	21
VII	20
VIII	15
Total Credits	**160**

Course Code	**:**	**ITIC 101**
Course Title	**:**	**Problems Solving and Programming Skills I (For CO, IT & AI & ML, AI & DS, M&C)**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:
The objective of this course is to provide fundamentals of problem-solving using C language programming.

Course Content:
Unit 1 Programming Fundamentals & Control Statements:
Block Diagram of Computer, Hardware vs software, concept of operating system and compiler, Introduction to C programming, basic programming using input and output operators and expressions, programming using if and if-else, Programming using looping-for, while, do-while; use of switch and break.

Unit 2 Arrays based Programming:
Defining and processing 1-D and 2-D Arrays for Problem Solving and Strings.

Unit 3 Modular programming using Functions:
Structured Programming, Defining and calling a function, modular programming using functions, passing arguments and arrays to functions, functions of void and returning values.

Unit 4 Programming using pointers, structures and unions:
Pointers in C: Pointer declaration, Passing Pointer to functions, pointers vs arrays, dynamic memory allocation. Structures and Unions, Programming Using Array of Structures and Unions, Memory Requirements for Unions.

Reference Books:
1. Byron S. Gottfried, Programming with C Language, Schaum Series, Tata McGraw Hill, 2015.
2. E Balaguruswamy, Programming with C, Tata McGraw Hill, 2015.
3. Kernighan & Richie, C Programming, Prentice Hall of India, 2002.

Course Outcomes:
1. Understand the use of software and programming for problem solving.
2. Learn programming using simple concepts of input, output and control statements.
3. Use arrays, functions, strings, structures and pointers for problem solving.

Course Code	**:**	**ITIC 100**
Course Title	**:**	**Digital System Design**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:

1. Awareness of intricate design details of components in any digital system.
2. Knowledge of number system and conceptual understanding of different codes.
3. Design fundamentals of computing machinery.
4. Introduction of computational automation process.

Course Content:

Unit 1: Number Systems and Coding Schemes:

Number Systems and Codes Introduction to the positional number system, signed magnitude numbers, floating point numbers, binary arithmetic: addition, subtraction, multiplication and division, Base conversion, conversion formulas with examples, one's and two's compliment arithmetic, Computer codes – BCD codes, gray codes, excess-3 codes, parity checks, Hamming and alphanumeric codes.

Unit 2: Combinational Logic:

Design Introduction, standard representations for logical functions, Karnaugh map representation, simplification of logical functions using K-map, minimization of logical functions specified in minterms/maxterms or Truth Table, minimization of logical functions not specified in minterms/maxterms, Don't care conditions, design examples, Ex-or and Ex-nor simplification of Kmaps, five and six-variable K-maps, QM method, MEV method, Introduction of multiplexers and their use in combinational logic design, demultiplexers/decoders and their use in combinational logic design, adders and their use as subtractors, digital comparators, parity generators/checkers, code converters, priority encoders.

Unit 3: Synchronous Sequential Circuits:

Introduction, FSM model, memory elements and their excitation functions. Synthesis of synchronous sequential circuits, capabilities and limitation of FSM, state equivalence and minimization, simplification of incompletely specified machines, registers and counters, RAM design, ROM design and programmable logic array.

Unit 4: Asynchronous Sequential Circuits:

Fundamental mode and Pulse mode Circuits Analysis and Design.

Books:

1. M. Morris Mano and Michael D. Ciletti: Digital Logic Design, Sixth Edition, Pearson Education.
2. R.P. Jain: Modern Digital Electronics, Fifth Edition, TMH.
3. Z Kohavi and Niraj. K. Jha: Switching And Finite Automata Theory, Third Edition, Cambridge University Press.
4. Kumar A. Anand: Fundamentals of Digital Circuits, Fourth Edition, PHI.
5. James Bignell and Robert Donovan: Digital Electronics, Fifth Edition, Cengage Learning.

Course Outcomes:

1. Clarity of application of different number system and coding schemes.
2. Proficiency in design and analysis of combinational and sequential circuits.
3. Circuit level understanding of computer addressing and memory layouts.
4. Application of digital circuits for design of finite automaton.

Course Code	**:**	**ITIC 110**
Course Title	**:**	**Digital System Design**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:

1. Awareness of intricate design details of components in any digital system.
2. Knowledge of number system and conceptual understanding of different codes.
3. Design fundamentals of computing machinery.
4. Introduction of computational automation process.

Course Content:

Unit 1: Number Systems and Coding Schemes:

Number Systems and Codes Introduction to the positional number system, signed magnitude numbers, floating point numbers, binary arithmetic: addition, subtraction, multiplication and division, Base conversion, conversion formulas with examples, one's and two's compliment arithmetic, Computer codes – BCD codes, gray codes, excess-3 codes, parity checks, Hamming and alphanumeric codes.

Unit 2: Combinational Logic:

Design Introduction, standard representations for logical functions, Karnaugh map representation, simplification of logical functions using K-map, minimization of logical functions specified in minterms/maxterms or Truth Table, minimization of logical functions not specified in minterms/maxterms, Don't care conditions, design examples, Ex-or and Ex-nor simplification of Kmaps, five and six-variable K-maps, QM method, MEV method, Introduction of multiplexers and their use in combinational logic design, demultiplexers/decoders and their use in combinational logic design, adders and their use as subtractors, digital comparators, parity generators/checkers, code converters, priority encoders.

Unit 3: Synchronous Sequential Circuits:

Introduction, FSM model, memory elements and their excitation functions. Synthesis of synchronous sequential circuits, capabilities and limitation of FSM, state equivalence and minimization, simplification of incompletely specified machines, registers and counters, RAM design, ROM design and programmable logic array.

Unit 4: Asynchronous Sequential Circuits:

Fundamental mode and Pulse mode Circuits Analysis and Design.

Books:

1. M. Morris Mano and Michael D. Ciletti: Digital Logic Design, Sixth Edition, Pearson Education.
2. R.P. Jain: Modern Digital Electronics, Fifth Edition, TMH.
3. Z Kohavi and Niraj. K. Jha: Switching And Finite Automata Theory, Third Edition, Cambridge University Press.
4. Kumar A. Anand: Fundamentals of Digital Circuits, Fourth Edition, PHI.
5. James Bignell and Robert Donovan: Digital Electronics, Fifth Edition, Cengage Learning.

Course Outcomes:

1. Clarity of application of different number system and coding schemes.
2. Proficiency in design and analysis of combinational and sequential circuits.
3. Circuit level understanding of computer addressing and memory layouts.
4. Application of digital circuits for design of finite automaton.

Course Code	**:**	**ITIC 102**
Course Title	**:**	**Web Design**
Number of Credits and L/T/P scheme	**:**	**2 and 1/0/2**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:

1. Introduction and brief history of World Wide Web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, client requirements and principles of web page design.
4. Introduction of Web architecture.

Course Content:

Unit 1: Introduction: Introduction to world wide web, Web Browsers, Web Servers, Hypertext Transfer Protocol, URLs, Domain Names, Internet Service Provider, Basic steps for Developing Website, Choosing the Contents, Planning and Designing Web Site, Creating a Website, Web Publishing, Hosting Site, Types of hosting packages, Five Golden rules of web designing.

Unit 2: Web essentials and standards: Clients, servers, introduction to Markup languages, scripting languages, Introduction to elements of HTML, XHTML and CSS, Introduction to Document object model (DOM), working with text, list, tables, frames, hyperlinks, Images, forms and controls. CSS properties, Id and Class, Box Model.

Unit 3: Javascript: Javascript as programming language, Data types, Values, Variables, Expressions and Operators. JavaScript Statements, loops, arrays, strings, methods, Defining and Invoking functions and their closure, random functions and maths library, representing dates, Pattern Matching and Regular Expressions, difference between server side and client side javascript, embedding javascript in HTML, hiding HTML elements, showing hidden HTML elements. DOM and event handling, error handling, mouse, text, and keyboard events and cookies.

Unit 4: XML: XML: Introduction – benefits of XML, well-formed XML documents, XML syntax, XML declaration, XML schema, XML with CSS, Document Type Definition (DTD) creating DTD – Types (internal DTD, external DTD), XSL.

Reference Books:

1. Thomas A Powell, HTML: The Complete Reference, Tata McGraw Hill Publications.
2. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e,O'Reilly
3. Doug Tidwell, James Snell, Pavel Kulchenko; Programming Web Services with SOAP, O'Reilly
4. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007.
5. Yong, XML Step by Step, PHI.
6. Chris Bales, "Web programming- Building Internet Application".
7. Deitel, Deitel, Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006.
8. Marty Hall and Larry Brown, "Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.
9. Bayross Ivan, "Web Enabled Commercial Applications Development using HTML, Javascript, DHTML & PHP", BPB Publication, 2005.

Course Outcomes

1. Knowledge of basic principles of web site design.
2. Design proficiency of websites adhering to current web standards (HTML, XML, CSS).
3. Knowledge of various scripting languages.

Course Code	**:**	**ITIC 112**
Course Title	**:**	**Web Design**
Number of Credits and L/T/P scheme	**:**	**2 and 1/0/2**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:

1. Introduction and brief history of World Wide Web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, client requirements and principles of web page design.
4. Introduction of Web architecture.

Course Content:

Unit 1: Introduction: Introduction to world wide web, Web Browsers, Web Servers, Hypertext Transfer Protocol, URLs, Domain Names, Internet Service Provider, Basic steps for Developing Website, Choosing the Contents, Planning and Designing Web Site, Creating a Website, Web Publishing, Hosting Site, Types of hosting packages, Five Golden rules of web designing.

Unit 2: Web essentials and standards: Clients, servers, introduction to Markup languages, scripting languages, Introduction to elements of HTML, XHTML and CSS, Introduction to Document object model (DOM), working with text, list, tables, frames, hyperlinks, Images, forms and controls. CSS properties, Id and Class, Box Model.

Unit 3: Javascript: Javascript as programming language, Data types, Values, Variables, Expressions and Operators. JavaScript Statements, loops, arrays, strings, methods, Defining and Invoking functions and their closure, random functions and maths library, representing dates, Pattern Matching and Regular Expressions, difference between server side and client side javascript, embedding javascript in HTML, hiding HTML elements, showing hidden HTML elements. DOM and event handling, error handling, mouse, text, and keyboard events and cookies.

Unit 4: XML: XML: Introduction – benefits of XML, well-formed XML documents, XML syntax, XML declaration, XML schema, XML with CSS, Document Type Definition (DTD) creating DTD – Types (internal DTD, external DTD), XSL.

Reference Books:

10. Thomas A Powell, HTML: The Complete Reference, Tata McGraw Hill Publications.
11. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e,O'Reilly
12. Doug Tidwell, James Snell, Pavel Kulchenko; Programming Web Services with SOAP, O'Reilly
13. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007.
14. Yong, XML Step by Step, PHI.
15. Chris Bales, "Web programming- Building Internet Application".
16. Deitel, Deitel, Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006.
17. Marty Hall and Larry Brown, "Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.
18. Bayross Ivan, "Web Enabled Commercial Applications Development using HTML, Javascript, DHTML & PHP", BPB Publication, 2005.

Course Outcomes

1. Knowledge of basic principles of web site design.
2. Design proficiency of websites adhering to current web standards (HTML, XML, CSS).
3. Knowledge of various scripting languages.

Course Code	**:**	**ITIC 104**
Course Title	**:**	**Programming using Python**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites	**:**	
Course Type	**:**	**IC**

Course Learning Objectives:

1. Building robust applications using Python programming.
2. Building multithreaded, platform-independent and GUI based python applications for business problems.

Course Content:

Unit 1: The concept of data types:

Variables, Assignments; Immutable Variables; Numerical Types; Arithmetic Operators And Expressions; comments in the program; understanding error messages; Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation; Strings and text files; manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab-separated); String manipulations: subscript operator, indexing, slicing a string.

Unit 2: Lists, tuples, and dictionaries:

Basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments.

Unit 3: Simple Graphics and Image Processing:

"turtle" module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing: Simple image manipulations with 'image' module (convert to bw, greyscale, blur, etc). Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects; inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; exception handling, try block

Unit 4: Graphical user interfaces:

Event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

Reference Books:

1. T.R. Padmanabhan, Programming with Python, Springer, 1st Ed., 2016.
2. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning,, 1st Ed., 2012.

Course Outcomes:

1. Programming ability for solving simple business problems.
2. Design of robust and multithreaded python applications.
3. Familiarity of simple GUI interfaces.

Course Code	**:**	**ITPC 100**
Course Title	**:**	**Data Structures**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites	**:**	**Problems Solving and Programming Skills**
Course Type	**:**	**PC**

Course Learning Objectives

1. This course introduces the concept of Data Structures used in various computer science applications
2. The students are introduced to understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using C programming language.

Course Content

Unit 1: Pointers & File Handling:

Revision of Pointers and Dynamic Memory, Files and related operations.
Searching techniques: Linear and Binary, Sorting techniques: Selection, Bubble, Insertion, Merge sort, Quicksort.

Unit 2: Simple Data Structures

Arrays based Linear Data Structures: Array storage, sparse arrays; Transpose, addition, and multiplication of sparse matrices, Stacks and Queues and their applications, multiple stacks and queues in an array.

Unit 3: Linked Data Structures

Singly, Doubly & Circular Linked Lists; representation, operations and applications, linked stacks and queues. linked lists based polynomial addition.

Unit 4: Advanced Data Structures

Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, Binary tree traversal techniques, some more operations on binary trees, Heaps and heapsort.

Reference Books:

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.L. Kruse: Data Structures & Program Design in C, PHI.
3. D.F. Knuth: The art of Computer Programming Vol-1, Narosa Publications, 1985.
4. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum's Outlines Series, TMH, 2005.

Course Outcomes

1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithms and implement their applications.
2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
3. Understand the concept of trees and graphs, their implementation and applications.
4. Able to implement standard algorithms for searching and sorting.
5. Analyze efficiency of different algorithms using time and space complexity.

Course Code	**:**	**ITPC 201**
Course Title	**:**	**Design and Analysis of Algorithms**
Number of Credits	**:**	**4 & 3/0/2**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Content:

Unit 1: Introduction

Concept of Time and space complexity, analysis of algorithms, asymptotic notation, recurrence relations, design and analysis of D & C problems like quick sort etc, heap sort, priority queues, sorting in linear time, hashing, binary search trees.

Unit 2: Graph Algorithms

Graph representation & traversal (search), topological sort, strongly connected components, minimum spanning trees – Kruskal and Prim's, Single source shortest paths, relaxation, Dijkstra's algorithm, Bellman-Ford algorithm, single source shortest paths for directed acyclic graphs, all-pairs shortest path.

Unit 3: B-Trees and Dynamic programming

B-Trees: representation and operations; Elements of Dynamic Programming, structure and steps, Matrix-chain multiplication, longest common subsequence.

Unit 4: Greedy & Backtracking Approaches:

Greedy algorithms – Elements, activity-selection problem, Huffman codes, task scheduling problem, Knapsack Problem, Backtracking – Elements, 8 – Queens, Graph Coloring, Hamiltonian Cycles.

Text Books:

1. Cormen, Leiserson and Rivest: Introduction to Algorithms, 3/e, PHI.
2. Horowitz, Sahni, and Rajasekaran: Fundamentals of Computer Algorithms, Second Edition, Universities Press, Hyderabad.
3. Aho, Hopcroft, and Ullman: The Design and Analysis of Computer Algorithms, Addison Wesley.

Course Outcomes

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Code	**:**	**ITPC 203**
Course Title	**:**	**Computer Organization and Architecture**
Number of Credits and L/T/P scheme	**:**	**3 & 3 0 0**
Prerequisites	**:**	**Digital System Design**
Course Category	**:**	**PC**

Course Learning Objectives:

1. Working of Computer Systems & its basic principles.
2. Concepts of processor and control design.
3. Concepts of pipelining techniques.
4. Memory hierarchy and its organization.
5. Concept of I/O devices and working.

Course Content:

Unit 1: Introduction

Introduction to digital electronics: combinational circuits and sequential circuits. Basic Machine Principle, Structure and representation of real world data. Subroutine, Branching & Macro facility.

Unit 2: Processor Design

Processor Organization, Information representation and Number format, Instruction cycle and Instruction format, Addressing modes, Arithmetic operation, timed point addition, subtraction, multiplication and division, ALU design, Parallel processing – Performance consideration, Pipeline processor.

Unit 3: Control Design

Instruction sequencing and Interpretation, Hardware Control design method and Microprogrammed Control.

Unit 4: Memory organization

Memory device characteristic, Random access and serial access memories, Virtual memory – memory hierarchies, Page replacement policies, Segments, pages and file organization, High speed memories – cache and associative memory.

Unit 5: System Organization

Programmed I/O, DMA and interrupts, I/O processors & CPU – I/O interaction.

Text Books:

1. John L. Hennessy , David A. Patterson: Computer Architecture: A Quantitative Approach, Sixth Edition, Morgan Kaufmann.
2. Kai Hwang: Computer Architecture and Parallel Processing, McGraw Hill Education.M.M. Mano: Computer System Architecture, 3rd Ed. PHI.
3. J.P. Hayes: Computer Architecture and Organization, 3rd Ed. TMH,
4. C.W. Gear: Computer organization and Programming, TMH.
5. A.S. Tanenbaum: Computer System Organization, PHI

Course Outcomes:

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

1. Understand the Computer System concepts.
2. Understand the organization of a computer system in terms of its main components.
3. Understand the processor and control design of a computer system.
4. Understand the various types of memory.
5. Understand input/output mechanisms.

Course Code	**:**	**ITPC 205**
Course Title	**:**	**Object-Oriented Programming using Java**
Number of Credits and L/T/P scheme	**:**	**4 & 3 0 2**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. Building robust applications using Java's object-oriented features.
2. Understanding the usage of java class libraries.
3. Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Content:

Unit 1: The overview of Java's architecture and the architecture of the Java Virtual Machine (JVM).

Classes: Declaring Members (Fields and Methods), Instance Members, Static Members.

Objects: Class Instantiation, Reference Values, and References, Object Aliases. Basic Language Elements, Primitive Data Types, Variable Declarations, Initial Values for Variables, Class Declarations, Method Declarations, this reference, Method Overloading, Constructors, The Default Constructor and Constructors overloading. Arrays, Anonymous Arrays, Multidimensional Arrays, Variable Arity Methods, The main() Method, Program Arguments.

Unit 2: Packages: Defining Packages, Using Packages, Compiling Code into Packages, Running Code from Packages. Scope Rules, Accessibility Modifiers, Overview of other Modifiers for Members.Operators and Expressions**,** Overview of Control Flow Statements.

Exception Handling: The try Block, The catch Block, The finally Block, The throw Statement, The throws Clause, Checked and Unchecked Exceptions, Defining New Exceptions.

Unit 3: Object-Oriented Programming: Single Implementation Inheritance, Overriding Methods, Hiding Members, The Object Reference super, Chaining Constructors Using this() and super()

Interfaces: Defining Interfaces, Abstract Method Declarations, Implementing Interfaces, Extending Interfaces, Interface References, Constants in Interfaces, Polymorphism and Dynamic Method Lookup.

Fundamental Classes: Overview of the java.lang Package, The Object Class, The Wrapper Classes, The String Class, The StringBuilder and the StringBuffer Classes.

Unit 4: Multithreading: Overview of Threads, the Main Thread, Thread Creation, Synchronization, Thread Transitions. Basics of Event Handling, Graphics Programming using Java Fx.

Reference Books:

1. Y. Daniel Liang: Introduction to Java Programming, Comprehensive Version, Pearson Education, 12th Ed.
2. Bruce Eckel, Thinking In Java, Pearson Education, 4th Ed., 2006.
3. Dietel & Deitel, Java How to Program, Pearson Education, 10th Ed., 2015.
4. Kathy Sierra & Bert Bates, Head First Java, O'REILLY, 2nd Ed., 2005.
5. Cay s. Horstmann & Gary Cornell, Core Java. Volume I, Fundamentals, Sun Microsystems Press, 8th Ed., 2008.

Course outcomes:

1. Write Java programs that solve simple business problems.
2. Create Java applications that are robust and multithreaded.
3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Code	**:**	**ITPC 207**
Course Title	**:**	**Software Engineering**
Number of Credits and L/T/P scheme	**:**	**4 & 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives

1. To inculcate software design abilities.
2. Impart knowledge-based awareness of user requirements.
3. To enable the students to conduct feasibility analysis.
4. Knowledge of software quality testing for its sustainability, reliability and durability.

Course Content

Unit 1: Introduction - Software Crisis, Software Characteristics, and Software Processes; Software life cycle models – Build & Fix, waterfall prototype evolutionary, spiral model.

Unit 2: Problem Analysis- DFD, Data dictionaries, ER diagrams, object diagrams; approaches to problems analysis; SRS; specifying behavioral & non-behavioral requirements, software configuration management.

Unit 3: Software Design and Implementation- Modularity, strategy of design, function-oriented design, object-oriented design, User interface design, UML modelling, Software implementation.

Unit 4: Software Quality- Size metrics, data structure metrics, information flow metrics, entropy-based measures, metric analysis, Software Quality Frameworks, ISO 9001 Model, SEI-CMM Model.

Unit 5: Software Validation and Reliability - Software Testing, verification and validation, errors, failures, faults, fault tolerance, Software Maintenance, reliability models – macro, basic, logarithmic Poisson, calendar time component, micro models.

Text Books:

1. Pressman, R.S. and Maxim, B.R., Software Engineering: A Practitioner's Approach, 9th ed., McGraw Hill, 2020.
2. Sommerville, Ian, Software Engineering, 10th ed., Pearson Education 2017.
3. K.K.Aggarwal, Yogesh Singh: Software Engineering, New Age International Ltd, 2001.
4. Pankaj Jalote, An Integrated Approach to Software Engineering 3rd Ed., 2005 Narosa Publishing.

Reference Books:

1. Singh,Y.,Software Testing, Cambridge University Press, 2013.

Course Outcomes

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

1. Apply the concepts to choose appropriate software process models as per user requirements
2. Analyze requirement techniques, including Data flow diagram, Entity relationship diagram, and Object diagram
3. Understand the various Software Design strategies for software implementation and emphasize different software metrics used for analyzing the software
4. Design and test the various software reliability measures to assess software quality in case of multiple faults and failures

Course Code	**:**	**ITPC 209**

Course Title	**:**	**IoT Programming**
Number of Credits and L/T/P scheme	**:**	**2 & 1/0/2**
Prerequisites (Course code) **Programming on the Raspberry Pi**	**:**	
Course Category	**:**	**PC**

Course Learning Objectives

1. Familiarity with sensors, actuators and microcontroller- based Arduino and Raspberry Pi platform.
2. Writing C programs in Arduino IDE.
3. Understanding of the Communication between microcontroller and PC using serial communication.
4. Knowledge of the design of IoT based applications.

Course Content

Unit 1: Introduction to Internet of Things: Applications/Devices, Protocols and Communication Model, the Physical Design/Logical Design of IoT, Functional blocks of IoT and Communication Models, Development Tools used in IoT.

Unit 2: Things and Connections Working of Controlled Systems: Real-time systems with feedback loop, Connectivity models, different type of modes using wired and wireless methodology.

Unit 3: Sensors, Actuators and Microcontrollers Sensor: Types of sensors, actuator and actuators controller, Microcontroller vs microprocessor, different types of microcontrollers in embedded ecosystem.

Unit 4: Building IoT applications using Arduino: Introduction to Arduino IDE, Embedded 'C' Language basics, interfacing sensors, interfacing LED, Button, Sensors-DHT, LDR, MQ135, IR.

Unit 5: Building IoT applications using Raspberry Pi: Raspberry Pi Hardware, Versions, Accessories, Precautions, Raspberry Pi Software – Operating System, Communicating and Connecting, Programming on the Raspberry Pi.

Text Books:

1. Macro Schwartz, "Internet of Things with Arduinon – Cookbook", Packt 2016
2. Arshdeep Bajga and Vijay Madisetti, "Internet of Things- A Hands-on Approach" Universities Press, 2014.
3. Derek Molloy, "Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux", John Wiley & Sons, 2016, ISBN: 978-1-119-18868-1
4. Yogesh Mishra, "Programming and Interfacing with Arduino", CRC Press, 2022, ISBN: 978-1-032-06316-4
5. Massimo Banzi, "Getting started with Arduino", 2nd Edition, Oreilly, 2011.
6. Macro Schwartz, "Internet of Things with Arduino", Open Home Automation.
7. Michael Margolis, "Arduino Cookbook", Oreilly, 2011.

Reference Books:

1. G. R. Kanagachidambaresan, "Internet of Things Using Single Board Computers: Principles of IoT and Python Programming", Apress, 2022 ISBN-13: 978-1-4842-8107-9
2. Jeff Cicolani, "Beginning Robotics with Raspberry Pi and Arduino Using Python and OpenCV", Second Edition, Apress, 2021, ISBN-13: 978-1-4842-6890-2
3. Olivier Hersent,David Boswarthick, Omar Elloumi, "The Internet of Things key applications and protocols", Wiley.
4. Adrian McEwen, Hakin Cassimally, "Designing the Internet of Things" Wiley India.
5. Michael Miller, "The Internet of Things" by Pearson.

Course outcomes:

1. Awareness of basic concepts, principles and challenges in IoT.
2. Understanding of the functioning of hardware devices and sensors used for IoT.
3. Ability to analyze network communication aspects and protocols used in IoT.
4. Apply IoT for developing applications using Ardunio programming.

Course Code	**:**	**ITPC 200**

Course Title	**:**	**Operating System**
Number of Credits and L/T/P scheme	**:**	**4 & 3/ 0/ 2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives

1. To understand the services and design of an operating system.
2. To understand the structure and organization of file system
3. To understand the process states and various concepts such as scheduling and synchronization related with it.
4. To understand different memory management approaches.
5. Students should be able to use system calls for managing processes, memory and file system.
6. students should understand the data structures and algorithms for implementation of OS.

Course Content

Unit-I: Computer system architecture and organization, Introduction and evolution of OS, Introduction to distributed OS, Real time systems and multimedia systems. OS structures: OS services, system calls and programs, OS design and implementation. Processes: Process concept, scheduling policies, algorithms, multilevel queuing, operations on process, Inter-process communication. Threads: multithreading models and threading issues. CPU scheduling: Criteria and algorithms, multiprocessor and thread scheduling.

Unit 2: Process synchronization: critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.).

Deadlocks: modelling, resource allocation, characterization, prevention and avoidance, detection and recovery.

Unit 3: Memory management: Swapping, contiguous memory allocation, paging, multilevel paging, segmentation, demand paging, page replacement algorithms, allocation of frames, thrashing, working set model. Input/Output: I/O system and services, device controllers and device drivers, disks, scheduling algorithms and management.

Unit 4: File system interface: access methods, access control, directory structures, file organization, file sharing and protection. system performance, protection and security, , OS design considerations for security, access control lists and OS support, internet and general network security. Operating system as service provider: Access control matrix, access control list, capability matrix, encryption and access permissions in Linux and Windows.

Reference Books:

1. A. Silberschatz, Peter B. Galvin and G. Gagne, "Operating System Concepts," (9th or newer edition), Wiley.
2. H. Brinch, "Operating System Principles," Prentice Hall of India.
3. Dhamdhere, "Systems programming & Operating systems," TataMcGrawHil
4. A. N. Habermann, "Introduction to Operating System Design," Galgotia publication, New Delhi.
5. A.S. Tanenbaum, "Modern Operating Systems," Prentice Hall of India.

Course Outcomes

At the end of the course student will be able to

1. Understand functions, structures and history of operating systems
2. Able to know the design issues associated with operating systems
3. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks
4. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms
5. Be familiar with various protection and security mechanisms
6. Be familiar with virtualization and operating system components

Course Code	**:**	**ITPC 202**
Course Title	**:**	**Computer Networks**
Number of Credits and L/T/P scheme	**:**	**4 & 3 0 2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives:

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Content:

Unit 1: Introduction

History and development of computer networks, Network Functions, Network Topology, Network Services, Switching Approaches, Transmission media and systems, OSI Reference Model, Overview of TCP/IP architecture.

Unit 2: Physical, MAC and Data Link Layer

Different types of transmission media, errors in transmission: attenuation, noise, multiplexing and signalling techniques, Encoding (NRZ, NRZI, Manchester, 4B/5B, etc). Aloha, CSMA, CSMA/CD, CSMA/CA protocols, Ethernet, including Gigabit Ethernet and WiFi (802.11), Token Ring, Bluetooth and Zigbee. Error detection, Sliding Window, Stop and Wait protocols, ARQ, Sliding Window Protocols, HDLC, PPP protocols.

Unit 3: Network layer:

Internet Protocol, IPv6, ARP, DHCP, ICMP, Routing algorithms: Distance vector, Link state, Metrics, Inter-domain routing. Subnetting, Classless addressing, Network Address Translation, IPv4 to IPv6 Translation strategies, Packet network topology, Datagrams and Virtual Circuits, Structure of Switch/Router, Connectionless and Virtual Circuit packet Switching, Traffic management and QoS – FIFO, Priority Queues, Fair Queuing, MPLS.

Unit 4: Transport and Application layer:

UDP, TCP. Connection establishment and termination, sliding window revisited, flow and congestion control, timers, retransmission, TCP extensions, etc. Client Server Model, HTTP, DNS, SMTP, FTP, and TFTP.

Text Books:

1. Behrouz A. Forouzan: Data Communications and Networks, Fifth Edition, McGraw Hill, 2017.
2. Leon Garcia and IndraWidjaja: Communication Networks – Fundamental Concepts and Key Architectures, Second Edition TMH.
3. A.S. Tanenbaum: Computer Networks, Sixth Edition, Pearson Education, 2022.
4. William Stallings: Data and Computer Communications 10/e, Pearson Education.

Course Outcomes:

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

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Code	**:**	**ITPC 204**
Course Title	**:**	**Artificial Intelligence and Soft Computing**
Number of Credits and L/T/P scheme	**:**	**4 & 3/0/2**
Prerequisites	**:**	**Design and analysis of algorithm**
Course Category	**:**	**PC**

Course Learning Objectives

1. Introduction and awareness of problem-solving strategies of artificial intelligence.
2. Understanding of evolutionary computation concepts.
3. Inception of biologically inspired algorithm concepts.
4. Overview of fuzzy systems.

Course Content

Unit 1: Introduction: Introduction of Soft Computing, Soft Computing vs. Hard Computing, Various Types of Soft Computing Techniques, Applications of Soft Computing.

Unit 2: AI Problem Solving: Uninformed Search Strategies, Informed Search Strategies, Local Search Strategies, Adversarial Search, Search for Constraint Satisfaction Problems.

Unit 3: Knowledge and Reasoning: Logical Agents, First-Order Logic Inference in First-Order Logic, Knowledge Representation, Automated Planning.

Unit 4: Nature Inspired Algorithms: Genetic Algorithms, Genetic Programming, Evolutionary Programming, Swarm Intelligence: Ant Colony Optimization, Artificial Bee colony, Particle Swarm Optimization.

Unit 5: Fuzzy Logic: Fuzzy Sets, Fuzzy rules, Fuzzy Reasoning, Fuzzification and Defuzzification Methods, Fuzzy inference system.

Text Books:

1. Stuart J. Russell and Peter Norvig, Artificial Intelligence A Modern Approach, Prentice Hall, Fourth Edition, 2022.
2. Zbigniew Michalewicz, Genetic Algorithm+ Data Structures = Evolutionary Programs, Springer, Third Edition.
3. James Kennedy, Russell C. Eberhart and Yuhui Shi, Swarm Intelligence, Morgan Kaufmann, First Edition.
4. Jang J.S.R., Sun C.T. and Mizutani E , Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Pearson Education India, First Edition.

Reference Books:

1. D.E. Goldberg, Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley, N.Y.
2. Marco Dorigo and Thomas Stützle, Ant Colony Optimization, The MIT Press, First Edition.
3. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, Third Edition.

Course Outcomes

1. Design proficiency of various artificial intelligence problem solving strategies.
2. Comprehensive understanding of evolutionary computation.
3. Extensive exposure of nature inspired computational concepts.
4. Knowledge of fuzzy applications.

Course Code	**:**	**ITPC 206**
Course Title	**:**	**Database Management Systems**
Number of Credits and L/T/P scheme	**:**	**4 & 3/0/2**
Prerequisites	**:**	
Course Category	**:**	**PC**

Course Learning Objectives:

1. To understand data models and database systems.
2. To understand relational database theory,
3. To learn database design and its normalization.
4. To understand concepts of transaction, its processing, and management.
5. To learn concurrency control techniques.
6. To develop practical skill in using MySQL DBMS.

Course Content:

Unit-1 (Introduction)

Data, Database, Database management system, Historical background from file systems to Database Systems, Data Models, Relational Data Model, ER model, Schemas and Instances, Database users, DBMS architecture.

Unit-2 (Relational Model and Query Language)

Theory of Relational Database, Key Integrity constraint, Relational Algebra, Relational Calculus: Domain Relational Calculus, Tuple Relational Calculus, SQL, queries writing in SQL.

Unit-3 (Database Design)

Normalization, Database Anomalies, Functional Dependencies, Candidate and Super Key, Non-loss Decomposition, Dependency Preservation, First, Second, Third Normal, BCNF, etc.

Unit-4 (Transaction Processing)

Transaction concept, a simple transaction model, states, ACID Properties, implementation of ACID properties, Serializability.

Unit-5 (Concurrency Control)

Need for Concurrency, Lock-based protocols, Deadlock, Starvation, deadlock handling, time-stamp based protocols, validation-based protocols.

Text Books:

1. A Silberschatz, H.F. Korth & S. Sudarshan: Data Base System Concepts, Mc Graw Hill, 4^{th}, 5^{th} or 6^{th} edition.
2. Elmasri &Navathe : Fundamentals of Database Systems, 5^{th}, 6^{th}, or 7^{th} edition Pearson.

Course Outcomes

1. To Design and Implement a small database for applications using MySQL DBMS.
2. To apply the concepts of normalization to database design.

Course Code	**:**	**ITPE 210**
Course Title	**:**	**Software Development using UML and Agile Methodology**
Number of Credits and L/T/P scheme	**:**	**4 & 3/0/2**
Prerequisites (Course code)	**:**	**Software Engineering**
Course Category	**:**	**PE**

Course Learning Objectives

The objective of this course is to prepare students for software development using UML and agile methodology.

Course Content

Unit-1: Unified Modelling Language (UML) Fundamentals

Introduction, An Overview of UML, Modelling Concepts - Systems, Models, and Views, Data Types, Abstract Data Types, and Instances, Classes, Abstract Classes, and Objects, Event Classes, Events, and Messages, Object-Oriented Modelling, Falsification and Prototyping.

A Deeper View into UML: Use Case Diagrams, Object diagram, Class Diagrams, Sequence diagram, Collaboration diagram, Interaction overview diagram, State machine diagram, Activity diagram, Component diagram, Deployment diagram, Package diagram, Timing diagram.

Unit-2: Agile Fundamentals

What is Agile? Why Agile? Why not Agile? Scrum, overview of PMI-ACP certification, The Agile Manifesto, twelve principles behind the Agile Manifesto.

Agile Concepts: Scrum, Extreme Programming, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools, Burndown chart

Unit-3: Managing Project with Scrum

The Rules of Scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum Events, Scrum Roles, Scrum Artifacts, Product backlog, SprintBacklog, User story: definition, characteristics and content, Sprint planning.

Unit-4: Agile Software Development in Practice

Lean Software Development, Kanban, Jira, Asana, Trello, Slack, Version Control. Using Patterns: What is a pattern and what makes a pattern? Properties of Patterns, Pattern – A Three-part Schema,Layer Pattern, Broker Pattern, Pipe and Filter Pattern, Model-View-Controller Pattern.

Text Books:

1. Bernd Bruegge & Allen H. Dutoit, "Object-Oriented Software Engineering Using UML, Patterns, and Java" Third Edition, Pearson
2. Andrew Stellman,& and Jennifer Greene., "Head First Agile: A Brain-Friendly Guide to Agile Principles, Ideas, and Real-World Practices. " O'Reilly Media, Inc.", 2017.
3. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Pearson Education, Third edition, 2013.
4. BernhardRumpe, "Agile modeling with UML: code generation, Testing, Refactoring", Cham: Springer, 2017.

Reference Books:

1. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2001.
2. Grady Booch, James Rumbaugh, Ivar Jacobson, Unified Modeling Language User Guide, Addison Wesley, 2nd Edition, 2012.
3. Robert C. Martin, Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.

Course Outcomes

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

1. Understand the concepts of Unified Modeling Language (UML) and UML diagrams during object-oriented analysis and design.
2. Software development and process control using agile and scrum methodology.
3. Use tools and techniques such as Kanban, Jira, Asana, Trello, Slack for Agile Software Development.
4. Justify designs based on design principles, patterns, and heuristics.

Course Code	**:**	**ITPE 212**
Course Title	**:**	**Competitive Programming and Efficient Coding For Problem Solving**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0 /2**
Prerequisites(Course code)	**:**	**Data Structures Design and Analysis of Algorithms**
Course Category	**:**	**PE**

Course Learning Objectives:

1. Identifying multiple alternate coding solutions of problems discussed in programming tests and Competitive Problems (CP) and developing understanding of their efficiency issues.
2. Designing and developing brute-force solutions followed by time/space efficient solutions for CP problems.
3. Learning commonly used problem solving approaches and applying them for coding the solutions for new problems.

Course Content:

Unit 1: Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL

Unit 2: Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding for Permutation Combination; XORing based and pattern based solutions,

Unit 3: Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matching algorithms; Kadane's algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.

Unit 4: Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path algos for CP problems with reduced time/space complexity

Unit 5: CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

Text Books:

1. Karumanchi: Data Structures and Algorithms Made Easy, CareerMonk Publications, 2017
2. G L McDowell: Cracking the Coding Interviews, CareerCup Publications, 2016
3. geekforgeeks.org
4. codechef.com, spoj.com, codeforces.com, interviewbit.com

Course Outcomes:

1. Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems.
2. Analyzing the space and time complexity of brute force solutions and designing efficient solutions.
3. Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems.
4. Creating efficient solutions of CP problems using the learnt algorithmic approaches.

Course Code	**:**	**ITPE 214**
Course Title	**:**	**Scripting Languages**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives:

(The relevant objectives of the course are to be listed)

1. Understanding the Basics of scripting history of scripting and use of scripting languages to webpages.
2. Understand the use of HTML5 Elements and differentiate with HTML for authoring web pages.
3. Able to apply Cascading style sheet for applying stylistic information to web pages.
4. Able to apply JavaScript logic on webpage for creating interactive web pages.
5. Understands and applies PHP and the basis of database connectivity.

Course Content:

Unit 1: Introduction

Introduction to Scripting Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Shell Scripts, Web Scripting, and the universe of Scripting Languages

Unit 2: HTML: HTML, HTML5, and XHTML basics, Elements, Attributes and Tags, Basic and Advanced Tags, Frames, Images, Relative URLs, Forms, Creating HTML Documents.

Unit 3: Style Sheet: CSS-Introduction to Cascading Style Sheets, Features, Core Syntax, Style Sheets and HTML Box Model.

Unit 4: JAVA SCRIPT&AJAX:

JavaScript: Introduction, Variables, Literals, Operators, Control structure, Conditional statements, Arrays, Functions, Objects, Predefined objects, Object hierarchy, Accessing objects, Reactive web pages elements: Events, Event handlers, multiple windows and Frames, Form object and Element, Advanced JavaScript and HTML, Data entry and Validation, Tables and Forms

AJAX Introduction, XML Http Request Object, Call back Methods, DHTML with JavaScript.

Unit 5: PHP Essential PHP - Operators and Flow control - Strings and Arrays - Creating functions - Reading data in web pages - PHP Browser Handling Power - File Handling -Session Handling in PHP – Cookies – Connection with MySQL.

Text Books:

1. Jeffrey C. Jackson, "Web Technologies - A Computer Science Perspective", Pearson Education.
2. Rasmus Lerdorf and Levin Tatroe, “Programming PHP”, O’Reilly, Third Edition.
3. Chris Bates, Web Programming – Building Intranet Applications, 3rd Edition, Wiley Publications, 2009.
4. The World of Scripting Languages, David Barron, Wiley Publications.
5. Web Programming, building internet applications, Chris Bates 2nd Edition, WILEY.
6. Beginning JavaScript with Dom scripting and AJAX, Russ Ferguson, Christian Heilmann, Apress.

Course Outcomes:

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

1. Understanding the Basics of scripting history of scripting and use of scripting languages to webpages.
2. Able to use HTML5 Elements and differentiate with HTML.
3. Able to apply style sheet and JavaScript logic on webpage.
4. Build a dynamic web page using JavaScript & Ajax.
5. Able to write a script for webpage using PHP.

Course Code	**:**	**ITIC 301**
Course Title	**:**	**Machine Learning**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course Code)	**:**	**CSPC204, MAIC201**
Course Type	**:**	**IC**

Course Learning Objectives:
1. Be able to understand and apply supervised and unsupervised learning algorithms.
2. Understand the fundamental concepts in machine learning and popular machine learning algorithms.
3. Interpretation of data and results using statistics and analysis
4. Be able to solve real-life problems using machine learning algorithms with programming in Python.

Course Contents:

Unit-1 Introduction

Basic definitions, types of learning, hypothesis space and inductive bias, probability distributions and hypothesis testing using statistical tests, data pre-processing.

Unit-2 Regression Techniques and Analysis

Supervised learning using regression, types of regression, linear regression: simple and multiple, logistic regression, overview of other types of regression; model diagnostics and analysis using residual and outlier, handling of multi-collinearity.

Unit-3 Classification

Supervised learning using classification, types of classification, linear models: regression & SVM; non-linear models: decision tree induction, attribute selection measures: information gain, gain ration, Ginni index; tree pruning and scalability; Naive Bayes classifier; kNN; classification model evaluation metrics; improving accuracy using ensemble, random forest, bagging, boosting.

Unit-4 Clustering

Basic Clustering concepts, Standard k-Means clustering, finding optimal number of clusters, Modifications to the k-Means; hierarchical clustering,

Unit-5 Advances in Machine Learning

Gradient descent algorithm, bias-variance, cross-validation, recommender systems: association rules, collaborative filtering, user of similarity, overview of use of neural networks for learning

Reference Books:

1. M. Pradhan, U. Dinesh Kumar, Machine Learning using Python, Wiley India, Latest Edition.
2. J. Han, M. Kamber, J. Pei, Data Mining Concepts and Techniques, Latest Edition
3. Christopher Bishop. Pattern Recognition and Machine Learning. 2e.
4. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
5. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.

Course Outcome:

At the end of this course, a student will be able to:
1. Compare and differentiate among different machine learning techniques
2. Implement the studied regression and classification techniques and analyze the results
3. Apply the suitable supervised and unsupervised techniques and models for real-life problems, and predict the outcome
4. Explain the applicability of advanced concepts for recommender systems.
5. Hands-on problem solving with programming in Python.

Course Code	**:**	**ITPC 301**

Course Title	**:**	**Theory of Computation**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**MAIC201**
Course Category	**:**	**PC**

Course Learning Objectives

The objective of this course is to give students a clear understanding of concepts formal languages, grammars and formal methods of computation.

Course Content

Unit-1 Introduction to Finite State Machine, Moore and Mealy FSMs, Equivalence, Minimal Finite Automata, unions, Intersections and complement of FA, NFA, NFA with null transitions, Equivalence.

Unit-2 Regular Languages, Regular expressions, memory required to recognize a language, Distinguishing one string from another, Criterion for Regularity, The pumping lemma, decision problems, Equivalence of FA and regular expressions, Arden's Theorem, Non-determinism and Kleen's Theorem.

Unit-3 Context – Free Grammars, Definition of CFG, example of familiar languages, unions, concatenations and closures of CFLs, Derivation Tree, Ambiguity, unambiguous CFG for algebraic expressions, CNF, GNF, removal anomalies in grammars, Push down automata, definition, deterministic PDA, PDA to CFG and Vice Versa, Context Free and Non Context Free Languages, Pumping lemma for CFG, Intersection and complements of CFL.

Unit-4 Turing Machine Definition, Turing Machining as Language acceptors, combining TM, computing Partial Function with TM. Recursively Enumerable and Recursive Languages, Regular Grammars, context Sensitive grammars, Chomsky Hierarchy. Concept of insolvability & reducibility, Halting Problem, Post correspondence Problem, Rice theorem, P and NP. Polynomial-Time reductions and NP-Completeness.

Reference Books:

1. John C. Martin: Introduction to Languages and the Theory of Computation, MGH.
2. Lewis & Papadimitriou: Elements of the Theory of Computation, PHI.
3. Daniel I.A. Cohen: Introduction to Computer Theory: John Wiley.
4. J.E. Hopcroft and J.D. Ullman: Introduction to Automata Theory Languages and Computation, Narosa.

Course outcomes:

1. To understand the foundations finite automata (deterministic or non-deterministic), FSM and master regular languages.
2. Be exposed to types of languages (type-0, type-1, type-2, type-3), context-free languages (type-2) and using pushdown automata (PDA) as its acceptor.
3. To design the Turing machines for various mathematical function, string function etc
4. To demonstrate concept of decidability, unsolvability and recursive languages.

Course Code	**:**	**ITPC 303**
Course Title	**:**	**Data Mining and Analytics**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives:

1. The course explores the concepts and techniques of data mining, a promising and flourishing frontier in data science. Analyzing large amounts of data has become a necessity and the problem therefore arises as to "how to analyze the data".
2. Data Mining is the subject dealing automated extraction of useful information or patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive data repositories. It is a technology that blends traditional data analysis methods with sophisticated algorithms for association rule mining, clustering, classification and outlier analysis.
3. The course is designed to provide students with a broad understanding in the design and use of data mining algorithms.
4. The course will provide algorithmic as well as application perspectives of data mining.

Course Content

Unit-1 Data Mining and Data Pre-processing: Introduction to Data Mining, Motivation behind Data Mining, Data Mining Tasks, Issues in Data Mining Applications. Types of data, Data Quality, Data pre-processing, Similarity and Dissimilarity, Outlier Analysis.

Unit-2 Clustering: Introduction, Applications, K-Means Algorithm, Hierarchical Algorithms, Density based Algorithms, Fuzzy c-Mean Algorithm, Silhouette Coefficient, DB-Index.

Unit-3 Association Rule Mining: Introduction, Applications, Market-Basket Analysis, Frequent Itemsets, Apriori Algorithm, Alternative Methods, FP Growth algorithm.

Unit-4 Dimension Reduction and Introduction to Analytics: Feature Selection Techniques, Principal Component Analysis, Classification overview.

Unit-5 Web Data Mining: Social Network Analysis: Preliminaries and Properties, Homophily, Triadic Closure, Clustering Coefficient, Centrality, Community Detection in Graphs and Networks.

Text Books:

1. T1. Tan,Pang-Ning & others. "Introduction to Data Mining" Pearson Education, 2006.

Reference Books:

1. R1. Han J & Kamber M, "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, Second Edition, 2006.
2. R2. Christopher Bishop: "Pattern Recognition and Machine Learning", Springer International Edition
3. R3. Tom M. Mitchell: "Machine Learning", The McGraw-Hill Companies, Inc.
4. R4. Charu C. Aggarwal "Outlier Analysis" Springer International Publishing (2017).
5. R5. Christopher D.M., Prabhakar R. & Hinrich S. "Introduction to Information Retrieval" Cambridge UP Online edition, 2009.

Course outcomes

At the end of the course the student should be able to

1. Choose an appropriate data pre-processing techniques based on the given data.
2. Identify and design an appropriate data mining analysis technique given a problem.
3. Gain practical hands-on experience in implementing data mining algorithms.

Course Code	**:**	**ITPC 305**

Course Title	**:**	**Full Stack Development**
Number of Credits and L/T/P scheme	**:**	**4 and 3 0 2**
Prerequisites (Course code)	**:**	**Software Engineering**
Course Category	**:**	**PC**

Course Learning Objectives

The main objective of the course is to enable the students to develop a complete web application that includes Frontend, Backend and Data-exchange technologies.

Course Content

Unit 1. Planning Your Work

Rise of the Web, Mobile Web, The State of HTML, Applications vs. Web Sites, Identifying Requirements, Defining the Work, Tracking the Work, Scrum, Kanban, Continuous Improvement, Prioritization & Estimation, Managing Bugs, Continuous Delivery

Unit 2. Designing Systems

System Architectures, Identifying Concepts, Identifying User Interactions, Handling Commonalities, Working with Legacy and External Dependencies, Component Interactions, Applications vs. Modules, Cross-Functional Requirements, Caching, Designing for Failure, Designing Modules, Refactoring,

Unit 3: Front End

HTML, From Server to Browser, Styling, Components, Responsive Design, Progressive Enhancement, Search Engine Optimization, Build Tools, Test-Driven Development

JavaScript, JavaScript in the Browser, Offline-First Development, Document Object Model, Server-Side JavaScript, JavaScript Modules, Structuring Your JavaScript, JavaScript Types

Unit 4. Back-End Development and Storing Data

Server-side programming languages (e.g., Node.js, Python, Ruby), APIs, API Responsibilities, Designing a REST API, Securing Your API, Using APIs, Storing DataTypes of Databases, To SQL or NoSQ?, Where to Store Your Data, Accessing Data from Your App, Managing Your Data, Protecting Your Data.

Text Books:

1. Chris Northwood, “The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer”, Apress, 2018.

Reference Books:

1. Jon Duckett, “Web Design with HTML, CSS, JavaScript and JQuery Set”, Wiley, 2014.
2. Nicholas C. Zakas, “Professional JavaScript for Web Developers”, Wiley, 2012

Course outcomes

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

1. Learn about the most recent web development technologies.
2. Understand the basics of HTML, CSS, and Bootstrap to structure and style web pages.
3. Acquire awareness about the front end and back-end Tools.

Course Code	**:**	**ITPC 307**
Course Title	**:**	**Information Security**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**Computer Networks**
Course Category	**:**	**PC**

Course Learning Objectives

1. To introduce the concept of information security, and number theory.
2. To make students familiar with different aspects of access control, system security and its applications.
3. To make students familiar with various cryptographic algorithms.
4. To design security solutions for different operation systems and Networks

Course Content

Unit 1. Introduction to Information Security: Meaning of Information security, privacy, vulnerability, threat, attack, CIA Triad, CNSS Security Model, Components of an Information System, OSI Security Architecture, Model for Network Security, Network Access Security Model, Access Control

Unit 2 Introduction to Number theory: Divisibility, Division Algorithm, Euclidean Algorithm, Extended Euclidean Algorithm, Fast exponentiation Algorithm; Modular Arithmetic, Congruences, Chinese Remainder Theorem, Algebraic structures, Galois Fields

Unit 3. Introduction to Cryptography: Private Key Cryptography: Traditional Symmetric-key ciphers, Modern Symmetric Key ciphers, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Block Cipher modes of Operation; Public Key Cryptography: Trapdoor one-way Function, RSA, Diffie-Hellman Key exchange, Cryptographic Hash Function, Digital Signature

Unit 4. System and Operating Security: Overview, Security policy, File Protection Mechanisms, User Authentication, Security models, Trusted Operating System Design, Host based Intrusion detection system, Network based Intrusion detection system, Host based Intrusion prevention system, Network based Intrusion prevention system.

Unit 5. Network Security: Network security Concepts, Network threats, Security controls, Firewalls, Protecting Programs and Data, Secure Program, sniffing, spoofing, Non Malicious Program errors, Malware, Viruses and other malicious code, Control against Program, Honeypot, SDN Security, IoT Security

Text Books:

1. Charles P. Pfleeger, Share Lawrence Pfleeger, Security in Computing, Pearson Education, 2/e.
2. William Stallings, Cryptography and Network Security, PHI, 7/e
3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer 2006.
4. B.A. Forouzan, Cryptography and Network Security, McGraw-Hill, 3/e

Reference Books:

1. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006)
2. Charlie Kaufman, Perlman & S Peeciner, Network Security, Pearson Education, 2/e.

Course outcomes:

On successful completion of this, course students will:

1. Able to understand the information security and number theory concepts.
2. Able to analyses different aspects of access control, system security and its applications.
3. Able to understand and analyse various cryptographic algorithms.
4. Able to design and develop security solutions for different operation systems and networks

Course Code	**:**	**ITPC 302**
Course Title	**:**	**Big Data Analytics**
Number of Credits and L/T/P scheme	**:**	**4 and 3 0 2**
Prerequisites (Course code)	**:**	**Data Mining and Analytics**
Course Category	**:**	**PC**

Course Learning Objectives

1. To understand the concepts, challenges and applications of big data
2. To gain the knowledge of various concepts and techniques in data stream mining
3. To learn about big data storage technologies and algorithms for big data processing

Course Content

Unit 1 Introduction To Big Data: Why Big Data? , The Four Dimensions of Big Data: Volume, Velocity, Variety, Veracity, Drivers for Big Data, Data Streams: Data Stream Models, Basic Streaming Methods, Data Synopsis, Sampling, Histograms, Wavelets

Unit 2 Clustering and Frequent Pattern Mining from Data Streams: Basic Concepts, Leader Algorithm, Clustering Algorithms, Search Space, Landmark Windows, Mining Recent Frequent Item Sets

Unit 3 Classification from Data Streams: Decision Trees, VFDT- Base Algorithm, Extensions to Basic Algorithm, Exhaustive Search, Functional Tree Leaves, Detecting Changes

Unit 4 Big Data Enabling Technologies: Hadoop Stack, Hadoop Distributed File System, Hadoop MapReduce, Introduction To Spark, Spark Streaming And Sliding Window Analytics, CAP Theorem, Key-Value Storage Model (Amazon's Dynamo), Document Storage Model (Facebook's Cassandra)

Text Books:

1. Baesens, Bart. Analytics in a big data world: The essential guide to data science and its applications. John Wiley & Sons, 2014.
2. Gama, Joao. Knowledge discovery from data streams. CRC Press, 2010.
3. Aggarwal, Charu C., ed. Data streams: models and algorithms. Vol. 31. New York: Springer, 2007.
4. Dirk Deroos et al., Hadoop for Dummies, Dreamtech Press, 2014.
5. Lam, Chuck. Hadoop in action. Simon and Schuster, 2010.

Reference Books:

1. Rutkowski, Leszek, Maciej Jaworski, and Piotr Duda. Stream data mining: algorithms and their probabilistic properties. Springer International Publishing, 2020.
2. Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. Mining of massive data sets. Cambridge university press, 2020.

Course outcomes

At the end of the course student will be able to

1. To analyze the attributes and associated issues of big data as well as will be able to propose solution for them.
2. To design efficient algorithms for mining different patterns from data streams.
3. To develop big data processing pipelines using different big data technologies.

Course Code	**:**	**ITPC 304**
Course Title	**:**	**Distributed Blockchain Technologies**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**Data Structures and Algorithms, Cryptographic Techniques, Programming Fundamentals**
Course Category	**:**	**PC**

Course Learning Objectives

The objective of the course is:

1. To provide conceptual understanding of blockchain technology and its working along with fundamental knowledge of cryptocurrency.
2. To give students the understanding of emerging abstract models for blockchain technology and to familiarise with the functional/operational aspects of cryptocurrency eco-system.
3. To study the security issues and safeguards related to blockchain network attacks.
4. To familiarise students to real-time application development using blockchain with different tools.

Course Content

Unit 1: Introduction**:** Need for distributed record keeping, Modeling faults and adversaries, Fault-tolerant distributed computing, Paxos, Byzantine Generals Problem, Consensus algorithms and their scalability problems, Nakamoto's concept with Blockchain based cryptocurrency (Bitcoin). Technical concepts of Blockchain systems – Block, Hash pointers and Merkle tree, Digital signature, Public-key cryptography, Digital cash, Zero-knowledge systems, Public vs Private Blockchain, Permissioned model of Blockchain, Concept of Smart Contract.

Unit 2: Bitcoin and Blockchain**:** Creation of coins, Payments and double spending problem, Bitcoin scripts, Bitcoin P2P network, Transaction in Bitcoin network, Block mining, Block propagation and Block relay. Consensus in Bitcoin – Distributed consensus in open environments, Consensus in a Bitcoin network, Proof of Work (PoW): Basic introduction, Hashcash PoW, Bitcoin PoW, Attacks on PoW and the monopoly problem, Proof of Stake (PoS), The life of a Bitcoin Miner, Mining Difficulty, Mining Pool.

Ethereum Basics – Ethereum and Smart Contracts, Using smart contracts to enforce legal contracts, comparing Bitcoin scripting vs. Ethereum smart contracts, Writing smart contracts using Solidity & JavaScript.

Unit 3: Security and Privacy Issues**:** Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation, Scalability problems, Stopping adoption, Network attacks to destroy bitcoin. Attacks on Blockchains: Sybil attacks, selfish mining, 51% attacks, advent of Algorand, Sharding-based consensus algorithms to prevent these attacks

Unit 4: Blockchain Application Development and Case Studies**:** Hyperledger Fabric- Architecture, Identities and Policies, Membership and Access Control, Channels, Transaction Validation, Writing smart contract using Hyperledger Fabric, Writing smart contract using Ethereum, Overview of Ripple and Corda. Case Studies – Block chain in Financial Service, Supply Chain Management and Government Services

Text Books:

1. Josh Thompsons, "Block Chain: The Block Chain for Beginners- Guide to Block chain Technology and Leveraging Block Chain Programming", Createspace Independent Pub, 2017
2. Imran Bashir, "Mastering Block Chain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained", Packt Publishing, 2nd ed., 2018
3. Narayanan, Bonneau, Felten, Miller and Goldfeder, "Bitcoin and Cryptocurrency Technologies– A Comprehensive Introduction", Princeton University Press, 2016
4. Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, "Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum and Hyperledger", Packt Publishing Limited, 2018.
5. Anshul Kaushik, "Block Chain and Crypto Currencies", Khanna Publishing House, 2018.

Reference Books:

1. Antony Lewis, "The Basics of Bitcoins and Blockchains", Mango Media, 2018
2. Daniel Drescher, "Block Chain Basics", Apress, 1st ed., 2017
3. Jay Isaac, "Blockchain: The Complete Step-by-step Guide to Understanding Blockchain and the Technology Behind It", Createspace Independent Pub, 2017
4. Merunas Grincalaitis, "Mastering Ethereum: Implement Advanced Blockchain Applications Using Ethereum-supported Tools, Services, and Protocols", Packt Publishing, 2019
5. [MOOC Course] Prof. Sandip Chakraborty, Dr. Praveen Jayachandran, "Blockchain Architecture Design and Use Cases"

Course Outcomes

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

1. Describe the blockchain technology, its distributed computing, cryptographic and cryptocurrency primitives.
2. Develop blockchain-based solutions and write smart contract using Hyperledger Fabric and Ethereum frameworks.
3. Illustrate the concepts of Bitcoin and their usage.
4. Implement Ethereum blockchain contract for real-world project development.
5. Apply security features in blockchain technologies.

Course Code	**:**	**ITPE 302**
Course Title	**:**	**Advanced Database Management System (Distributed Database Systems)**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

1. To develop understanding of distributed database design and query processing.

Course Content

Unit-1 (Introduction and Distributed Database Design)

Introduction: What is a Distributed Database System? Data Delivery Alternatives, Promises of DDBSs, Design Issues, Distributed DBMS Architecture.

Distributed Database Design: Data fragmentation, allocation, combined approaches, adaptive approaches, data directory.

Unit-2 (Distributed Data Control)

View management, access control, and semantic integrity control.

Unit-3 (Distributed Query Processing)

Overview, data localization, join ordering in distributed queries, distributed cost model, distributed query optimization, adaptive query processing.

Unit-4 (Distributed Transaction Processing)

Background and terminology, distributed query control, distributed concurrency control using snapshot isolation, distributed DBMS reliability, modern approaches to scaling out transaction management.

Unit-5 (Data Replication)

Consistency of replicated databases, update management strategies, replication protocols, group communication, replication and failures.

Text Books:

1. Principles of Distributed Database Systems by M. Tamer Özsu and Patrick Valduriez Fourth Edition, Springer.

Reference Books:

Research papers.

Course outcomes

1. To comprehend issues and solutions of distributed databases.
2. To develop understanding of design and implementation of distributed databases.

Course Code	**:**	**ITPE 304**
Course Title	**:**	**Game Theory and Applications**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

Objective of the course is to make student understand the concepts of game theory and its applications to various discipline Engineering.

Course Content

Unit 1: Introduction and Outline of The Course: Definitions, Utilities, Rationality, Intelligence, Common Knowledge, Classification of Games and Real-world Scenarios.

Unit 2: Non-Cooperative Game Theory: Extensive Form Game, Strategic Form Games With Illustrative Examples, Dominant Strategy Equilibria, Pure Strategy Nash Equilibrium With Illustrative Examples And Key Results, Mixed Strategy Nash Equilibrium With Illustrative Examples And Key Results Such As The Nash Theorem, Computation Of Nash Equilibria And Introduction To Algorithmic Theory, Matrix Games: Saddle Points, Minimax Theorem, Bayesian Games, Bayesian Nash Equilibrium, Evolutionary Game Theory (Ess Strategies), Repeated Game.

Unit 3: Mechanism Design: The Mechanism Design Environment, Social Choice Functions with Illustrative Examples, Implementation of Social Choice Functions, Incentive Compatibility And Revelation Theorem, Gibbard-Satterthwaite And Arrow Impossibility Theorem, Vickrey-Clarke-Groves (Vcg) Mechanisms, Bayesian Mechanisms (Dagva), Revenue Equivalence Theorem, Myerson Optimal Auction, Further Topics In Mechanism Design.

Unit 4: Cooperative Game Theory: Correlated Strategies and Correlated Equilibrium, The Nash Bargaining Problem, Coalitional Games (Transferable Utility Games), The Core, The Shapley Value, Other Solution Concepts: Kernel, Nucleolus.

Text Books, and/or reference material

1. Martin J. Osborne. An Introduction to Game Theory. Oxford University Press. Indian Edition, 2003.
2. Roger B. Myerson. Game Theory: Analysis of Conflict. Harvard University Press, 1991.
3. Y. Narahari, Dinesh Garg, Ramasuri Narayanam, Hastagiri Prakash. Game Theoretic Problems in Network Economics and Mechanism Design Solutions. Springer, London, 2009.
4. NPTEL website, http://nptel.ac.in/syllabus/106108053/.

Course Outcomes

1. To understand the concepts of game theory and its mathematical foundation
2. To identify fields of application of game theory in real-life
3. To design and implement game theoretic algorithms for automation
4. To apply game theoretic strategies in industrial engineering problems.

Course Code	**:**	**ITPE-306**
Course Title	**:**	**Image Processing**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	**Basic knowledge of mathematics and programming**
Course Category	**:**	**PE**

Course Learning Objectives

1. Introduction to image processing
2. The understanding of simple mathematics applied for image processing
3. Implement image processing algorithms
4. Learning the simple filters and transformation methods
5. Introduction to color image processing

Course Content

Unit-1: Introduction to digital image processing, Applications of digital image processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships between Pixels, An Introduction to the Mathematical Tools Used in Digital Image Processing, Array versus Matrix Operations, Linear versus Nonlinear Operations, Arithmetic Operations.

Unit-2: Basics of Intensity Transformations and Spatial Filtering, Image Negatives, Log Transformations, Power-Law (Gamma) Transformations, Piecewise-Linear Transformation Functions, Histogram Equalization, Histogram Matching (Specification), Local Histogram Processing, Using Histogram Statistics for Image Enhancement, Spatial Filtering, Spatial Correlation and Convolution, Vector Representation of Linear Filtering, Spatial Filter Masks, Smoothing Linear Filters, Laplacian, The Gradient.

Unit-3: Complex Numbers, Fourier Series, Impulses and Their Sifting Property, The Fourier Transform of Functions of One Continuous Variable, Convolution, Sampling, The Fourier Transform of Sampled Functions, The Sampling Theorem, Discrete Fourier Transform (DFT) of One Variable, Functions of Two Variables, Properties of 2-D Discrete Fourier Transform.

Unit-4: Ideal Lowpass Filters, Butterworth Lowpass Filters, Gaussian Lowpass Filters, Additional Examples of Lowpass Filtering, Ideal Highpass Filters, Butterworth Highpass Filters, Gaussian Highpass Filters, The Laplacian in the Frequency Domain, Unsharp Masking, Highboost Filtering, and High-Frequency- Emphasis Filtering, Homomorphic Filtering, The Fast Fourier Transform (FFT)

Unit-5: The RGB Color Model, The CMY and CMYK Color Models, The HSI Color Model, Intensity Slicing, Intensity to Color Transformations, Color Complements, Color Slicing, Tone and Color Corrections, Histogram Processing, Color Image Smoothing, Color Image Sharpening.

Text Books:

1. Gonzalez and Woods, Digital Image Processing, 3/e, Prentice Hall, 2007.
2. M Sonka et. Al., Image Processing: Analysis and Machine Vision, 3/e, Cole Pub. Co., 2008

Reference Books:

1. Wilhelm Burger and Mark J. Burge, Principles of Digital Image Processing, springer, 2009

Course Outcomes

1. Students will be able to understand the basics of image processing.
2. They will understand the applications of the matrix, Fourier transformation and other mathematics.
3. Hand on practice in the lab will provide a deep knowledge and connectivity for the students.

Course Code	**:**	**ITPE 308**
Course Title	**:**	**Software Verification and Validation**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Software Engineering, Software Development using UML and Agile Methodology**
Course Category	**:**	**PE**

Course Learning Objectives

The objectives of the course is to prepare students for learning different techniques of verification and validation of object oriented and web based software.

Course Content

Unit-1: Preliminaries [1, 2]

Fundamentals of Software Testing, errors and testing, Software quality, Requirements, behavior, and correctness, Correctness versus reliability, Testing and debugging, Software and hardware testing, verification and validation, Types of testing, testing process

Unit-2: Introduction to Testing & Software Verification [1]

Functional Testing: Boundary Value analysis, Equivalence Class Testing, Decision Table Based Testing, Cause-Effect Graphing Technique.

Structural Testing: Control Flow Testing, Data Flow Testing, Slice Based testing, Mutation Based testing

Software Verification: Verification methods, SRS document verification, SDD document verification, source code reviews, User document verification, Software project Audit.

Unit-3: Test Case Generation & Regression Testing [1]

Test Case Generation from Requirements and Use Cases: Introduction to Use case diagram and use cases, test case generation from Use cases, validity checking and its guidelines, Data validation Strategies, Database Testing.

Test Generation from Finite State Models [2]: Overview of Finite State Machines, What is a Finite State Machine?. Properties of FSM, The Fault Model for FSM based testing, Transition Tour (TT) Method, Testing with State Verification

Introduction to regression testing: regression test cases selection methods and their management, risk analysis, code coverage prioritization techniques, Test selection using Execution Trace [2], Test selection using Dynamic slicing [2].

Unit-4: Object Oriented Testing and Test Generation from Combinatorial Designs [1, 2]

Combinatorial designs, A combinatorial test design process, Fault model, Latin squares, Mutually orthogonal latin squares, Pairwise design: binary factors, Pairwise design: multi-valued factors.

Object Oriented Testing: Introduction to object orientation and testing, issues in object oriented testing, path testing, state based testing, class testing.

Text Books:

1. Yogesh Singh, Software Testing, Cambridge Publication, 2013
2. Aditya P. Mathur, “Foundations of Software Testing”, Second edition, Pearson Education, 2013.

Reference Books:

1. Kshirasagar Naik, Priyadarshi Tripathy, “Software Testing and Quality Assurance: Theory and Practice”, Wiley.
2. M.G. Limaye, “Software Testing: Principle, Techniques and Tools”, McGrawHill, 2009.
3. A. Basu, “Software Quality Assurance, Testing and Metrics”, PHI, 2015.
4. Paul Jorgensen, “Software Testing A Craftsman's Approach”, Second Edition, 2002
5. R. Patton, “Software Testing”, SAMS, 2005

Course outcomes

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

1. Describe techniques in the areas of software inspection, software testing and software validation.
2. Apply different techniques for verification of the software.
3. Generate test cases for validation object-oriented software.

Course Code	**:**	**ITPE 310**
Course Title	**:**	**Ethical Hacking**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives:

1. To impart students with basic information of prevalent vulnerabilities and threats.
2. Awareness of required safety practices with regards to internet usage and safety protocols.
3. Accurate overview of the hacking methodologies and solutions therein.

Course Content

1. Introduction: Ethical hacking introduction, Networking fundamentals: TCP/IP stack, IP addressing and routing. TCP and UDP, IP subnets, routing protocols, IP version 6, Network Scanning.
2. Introduction to cryptography: private-key encryption, public-key encryption, Cryptographic hash functions, digital signature and certificate, applications, steganography, biometric authentication, network-based attacks, DNS and Email security.
3. Attacks: Packet sniffing, password attack, social engineering attacks and Denial of service attacks, password cracking, privilege escalation, remote code execution, elements of hardware security, side-channel attacks, physical inclinable functions, hardware Trojans, Malware analysis.
4. Session hijacking, Attack on web servers: password attack, SQL injection, cross site scripting, web applications hacking, wireless networks hacking mobile platforms hacking, evading IDS, firewalls and honeypots, IoT hacking.
5. Case Studies: Threats and remedies

Text Books:

1. Dr. Daniel Graham, Ethical Hacking, 1st edition., No Starch Press.
2. Stallings, W. Data and computer communications. 9th Edition, Pearson Education India.
3. Forouzan, B. A. Data Communications and Networking Global Edition 5th edition. McGraw Hill.
4. Forouzan, B. A.TCP/IP protocol suite (4th edition) McGraw-Hill Higher Education.

Reference Books:

1. Richard Stevens, W., Fenner, B., & Rudoff, A. M. UNIX network program-ming, Volume 1: the sockets networking 3rd edition API.
2. Stallings, W., & Brown, L. Computer security: principles and practice. 4th edition Pearson.

Course outcomes

1. Assess various vulnerabilities of the system.
2. Aptly judge the security credentials of any software or hardware applications.
3. Be able to decide upon the communication and computation models for a reasonably secure system.

Course Code	**:**	**ITPE 312**
Course Title	**:**	**DevOps**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

1. **Understand the principles and objectives of DevOps**: Gain a clear understanding of the core principles, values, and goals of DevOps, including collaboration, automation, continuous integration, continuous delivery, and continuous deployment.
2. **Familiarity with DevOps tools and technologies**: Explore a range of DevOps tools and technologies commonly used in the industry, such as version control systems (e.g., Git), build automation tools (e.g., Jenkins), configuration management tools (e.g., Ansible), containerization (e.g., Docker), and infrastructure orchestration (e.g., Kubernetes).
3. **Gain proficiency in configuration management**: Learn how to effectively manage and automate software configurations using configuration management tools. Understand the concepts of infrastructure as code (IaC) and learn how to provision and manage infrastructure using tools like Chef, Puppet, or Ansible.
4. **Understand Continuous Integration/Continuous Delivery (CI/CD) pipelines**: Learn how to design and implement CI/CD pipelines to enable the rapid and automated delivery of software. Explore concepts such as code versioning, automated builds, testing, and deployment.
5. **Gain knowledge of cloud computing and virtualization**: Understand how cloud platforms and virtualization technologies support DevOps practices. Learn about Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) models.
6. **Explore monitoring and logging practices**: Understand the importance of monitoring and logging in DevOps. Learn how to implement effective monitoring solutions to ensure system performance, detect issues, and facilitate proactive problem resolution.

Course Content

Unit-1: Introduction to DevOps, Definition, principles, and benefits of DevOps, DevOps culture and mindset, DevOps vs. traditional software development and operations models, Continuous Integration and Continuous Delivery (CI/CD), Introduction to CI/CD pipelines, Version control systems (e.g., Git), Automated build and deployment processes, Testing and quality assurance in CI/CD, Release management and versioning, Infrastructure as Code (IaC).

Unit-2: Introduction to infrastructure automation, Configuration management tools (e.g., Ansible, Chef, Puppet), Provisioning and managing infrastructure using IaC, Managing environments and infrastructure scalability, Containerization and Orchestration, Introduction to containerization (e.g., Docker), Container orchestration platforms (e.g., Kubernetes), Containerizing applications and microservices, Scaling and managing containers in production, Monitoring, Logging, and Performance Optimization.

Unit-3: Importance of monitoring and logging in DevOps, Monitoring tools and techniques, Log management and analysis, Performance monitoring and optimization strategies, Cloud Computing and DevOps, Introduction to cloud computing models (IaaS, PaaS, SaaS), Cloud platforms (e.g., Amazon Web Services, Microsoft Azure), Deploying and managing applications in the cloud, Cloud automation and resource management, Security and Compliance in DevOps, DevOps security principles and practices, Securing CI/CD pipelines and infrastructure, Compliance and regulatory considerations, Vulnerability scanning and remediation, Collaboration and Communication in DevOps.

Unit-4: Cross-functional collaboration and teamwork, Communication and feedback loops, Agile methodologies and DevOps integration, Continuous learning and improvement in DevOps, Case Studies and Real-world Examples, Examining successful DevOps implementations, Case studies of organizations adopting DevOps, Lessons learned and best practices from real-world scenarios, Emerging Trends and Future of DevOps, Exploring emerging technologies and trends in DevOps, DevOps in the context of emerging fields (e.g., AI, IoT), Industry standards and certifications in DevOps.

Text Books:

1. Kim G, Humble J, Debois P, Willis J, Forsgren N. The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations. IT Revolution; 2021 Nov 30.
2. Humble J, Farley D. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education; 2010 Jul 27.

Reference Books:

1. Davis J, Daniels R. Effective DevOps: building a culture of collaboration, affinity, and tooling at scale. " O'Reilly Media, Inc."; 2016 May 30.
2. Farcic V. The DevOps 2.4 Toolkit: Continuous Deployment to Kubernetes: Continuously Deploying Applications With Jenkins to a Kubernetes Cluster. Packt Publishing Ltd; 2019 Nov 28.

Course outcomes

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

1. Apply DevOps practices to improve collaboration, communication, and efficiency between development and operations teams.
2. Demonstrate proficiency in setting up and managing CI/CD pipelines.
3. Implement infrastructure automation using configuration management tools and infrastructure-as-code (IaC) practices.
4. Apply security practices and compliance considerations in the DevOps process.
5. Analyze and troubleshoot issues in the DevOps pipeline and identify areas for improvement.

Course Code	**:**	**ITPE 326**
Course Title	**:**	**Computer Vision**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Objective:

1. Able to recognize and describe both the theoretical and practical aspects of computing with images.
2. Able to describe the foundation of image formation and image analysis.
3. Understand the geometric relationships between 2D images and the 3D world.
4. Have gained exposure to motion, matching, recognition, extraction, and categorization from images.

Course Outcomes:

1. Understand the basics of 2D and 3D Computer Vision.
2. Design and describe various methods used for motion, registration, alignment, detection, recognition and matching in images.
3. Build computer vision applications.

Syllabus:

Unit-1: Introduction And Digital Image Formation: Overview, Fundamentals of Image Formation, What is computer vision?, A brief history. Image Formation And Image Models: Geometric Camera Models, Geometric Camera Calibration, Radiometry, Measuring Light, Shadows and shading, Color.

Unit-2: Early Vision - Multiple Images: The Geometry of Multiple Views, Stereopsis, Affine Structure from Motion, Projective Structure from Motion. Feature Detection and Matching: Points and patches, Edges, Lines

Unit-3: Mid Level Vision: Segmentation by Clustering, Segmentation by Fitting a Model, Segmentation and Fitting using Probabilistic Methods, Tracking with Linear Dynamic Models. Feature-Based Alignment: 2D and 3D feature-based alignment, Pose estimation, Geometric intrinsic calibration.

Unit-4: High Level Vision – Geometric Methods: Model-Based Vision, Smooth Surfaces and their Outlines, Aspect Graphs, Range Data. Structure From Motion: Triangulation, Two-frame structure from motion, Factorization, Bundle adjustment, Constrained structure and motion. Probabilistic and Inferential Methods: Recognition by Relations between Templates, Geometric Templates from Spatial Relations, Application, Image Based Rendering.

Unit-5: Dense Motion Estimation: Translational alignment, Parametric motion, Spline-based motion, Optical flow, Layered motion. 3D Reconstruction: Shape from X, Active range finding, Surface representations, Point-based representations - Volumetric representations - Model-based reconstruction - Recovering texture maps.
Recognition: Object detection, Face recognition, Instance recognition, Category recognition, Context and scene understanding.

References:

1. Digital Image Processing, Rafael C. Gonza Lez and Richard E. Woods, Fourth Edition, Pearson, 2018.
2. Computer Vision: A Modern Approach, Forsyth and Ponce, Second Edition, Pearson, 2015.
3. Computer Vision: Algorithms and Applications, Richard Szeliski, Springer-Verlag, 2010.
4. Computer Vision: Models, Learning, and Inference, Dr Simon J. D. Prince, Cambridge University Press, 2012.
5. Multiple View Geometry in Computer Vision, Richard Hartley and Andrew Zisserman, Cambridge University Press, 2004.

Course Code	**:**	**ITPE 314**
Course Title	**:**	**Bio-Informatics**
Number of Credits and L/T/P scheme	**:**	**3 and 3 0 0**
Prerequisites (Course code)	**:**	**Design and Analysis of Algorithms**
Course Category	**:**	**PE**

Course Learning Objectives

This course aims to introduce the bioinformatics concepts, techniques and tools related to biological databases, sequence analysis, phylogenetic analysis, genomics and protein analysis to students with a focus on drug discovery application.

Course Content

Unit 1: Introduction: Bioinformatics an Interdisciplinary Approach, Genome, DNA Sequence, Central Dogma, Transcription, Translation, DNA Structure and Modifications.

Unit 2: Biological Databases: Types of Biological Information, Biological Database Search and Data Retrieval, Techniques for Extracting DNA Sequence, Protein Sequence and Protein Structure Information, Conventions for Database Indexing and Specification of Search Terms

Unit 3: Sequence Alignment: Pair wise Alignment of Sequences, Scoring Matrices, and Tools for Similarity Search and Sequence Alignment, Dynamic Programming Algorithms

Unit 4: Phylogenetic Analysis and Genomics: Introduction to Phylogenetics, Comparative Genomics, Orthologs, Paralogs, Methods of Phylogenetic Analysis. Whole Genome Analysis, Gene Prediction, Protein Structure Overview, Search and Visualization, Protein Structure Prediction, Protein Function Prediction

Unit 5: Drug Discovery: Technologies and Strategies, Key to Drug Discovery, Structural Biology and Virtual Screening for Drug Discovery, Emerging Role of Biomarkers in Drug Development, Potential Drug Targets

Text Books:

1. Letovsky, Stanley I., ed. Bioinformatics: databases and systems. Springer Science & Business Media, 2006.
2. Jones, Neil C., and Pavel A. Pevzner. An introduction to bioinformatics algorithms. MIT press, 2004.
3. Baxevanis, Andreas D., Gary D. Bader, and David S. Wishart, eds. Bioinformatics. John Wiley & Sons, 2020.

Reference Books:

1. Sorin Draghici, Bioinformatics Databases: Design, Implementation, and Usage, CRC, 2013
2. Rastogi, S. C., Parag Rastogi, and Namita Mendiratta. Bioinformatics: Methods and Applications-Genomics, Proteomics and Drug Discovery. PHI Learning Pvt. Ltd., 2022

Course outcomes

At the end of the course student will be able

1. To search and extract information from different biological databases.
2. To apply different alignment techniques to biological sequences.
3. To perform phylogenetic analysis and genome analysis using different tools.
4. To analyze proteins structure and functions.

Course Code	**:**	**ITPE 316**
Course Title	**:**	**Advanced Topics in Computer Networks**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

1. To give the students an understanding of the principles behind the latest advances in network technology from the physical to application layer

Course Content

Unit 1: Introduction to IPv6 IPv6 protocol architecture, IPv6 addressing, IPv6 headers, Static & automatic address configuration, ICMPv6, Neighbor discovery, IPsec, Authentication header, Encapsulation security payload header, transport and tunnel Mode, Interoperation concepts, Dual stack servers, Tunnelling and header translation.

Unit 2: Mobile IP Introduction to mobile IPv4, mobile IPv6, protocol over view of mobile IPv6, enhanced handover schemes, fast handover scheme, network mobility (NEMO), hierarchal mobile IPv6, security in mobile IP, VPN problems & solutions, bidirectional tunneling & route optimization, DHCPv6.

Unit 3: Software Defined Network SDN: Background and Motivation, SDN Data Plane and OpenFlow, Control Plane, Application Plane , Network Function Virtualization Network Function Virtualization (NFV) - Architecture and Concepts

Unit 4: Congestion Control in Wired and Wireless Networks TCP Congestion Control – fairness, scheduling and Delay modeling, QoS issues, differentiated services, Transport layer in Wireless Networks, IEEE 802.11a/b/n/g/p, 802.15, standards for Wireless.

Unit 5: IoT Architecture, Working with IoT Technologies - Bluetooth, Zibgee, BLE and LoRA, CoAP and MQTT, IoT Security, Cloud System Architectures and Working with Virtualization, Cloud Security.

Reference Books:

1. Youngsong Mun and Hyewon Keren Lee, Understanding IPv6, Springer, 2010
2. William Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, 2015
3. Rajeev S. Koodli and Charles E. Perkins, Mobile Inter-networking with IPv6: Concepts, Principles and Practices, Wiley, 2007
4. Michael Welzl, Network Congestion Control: Managing Internet Traffic, Wiley, 2005 5. Marc Blanchet, Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks, Wiley, 2005

Course Outcomes:

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

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Code	**:**	**ITPE 318**
Course Title	**:**	**Software Project Management**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

1. Define the scope of 'Software Project Management (SPM)'
2. Understanding a few problems related to SPM
3. Need for careful planning, monitoring, and control

Course Content

Unit-1 (Introduction to SPM)

Software Project Vs Other Types of Project, Activities covered by SPM, Ways of Categorising Software Projects, Problems with Software Projects, Management Control, Stakeholders, Information and Control in Organizations.

Introduction to Step Wise Project Planning (An Overview of Project Planning)

Unit-2 (Project Evaluation & Software Efforts Estimation)

Strategic Assessment, Technical Assessment, Cost-benefit Analysis, Cash Flow Forecasting, Cost-benefit Evaluation Techniques, Risk Evaluation

Problems with over- and under-estimates, The basis for software estimating, Software Effort Estimation Techniques, Function Point Analysis, COCOMO Model.

Unit-3: The Objectives of Activity Planning, Project Schedules, Projects & Activities, Sequencing & Scheduling Activities, and Network Planning Models.

Risk Management: The nature of risk, Managing Risk, Risk Identification, Risk Analysis, Evaluating Risks to the Schedule

Resource Allocation: The Nature of Resources, Identifying resource requirements, and Scheduling Resources.

Unit: 4 (Project Execution And Closure)

Project Monitoring and Control: Project Tracking, Milestone Analysis, Activity-Level Analysis, Defect Analysis And Prevention, Process Monitoring And Audit. Project Closure: Project Closure Analysis,

Text Books:

1. Software Project Management, Bob Hughes & Mike Cotterell, McGraw-Hill Education, 3rd Edition
2. Software Project Management In Practice, Pankaj Jalote, Pearson Education India; First Edition
3. Software Project Management: A Unified Framework, By W. Royce, Pearson India; 1st edition .

Course Outcomes

1. Able to identify project planning objectives, along with various cost/effort estimation models.
2. Organize & schedule various project activities
3. Capable to monitor and control project activities.

Course Code	**:**	**ITPE 320**
Course Title	**:**	**Compiler Design**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	**Automata Theory**
Course Category	**:**	**PE**

Course Learning Objectives

The objective of this course is to give students a clear understanding of the major concept areas of language translation and to enrich the knowledge in various phases of compiler and its use, parsing techniques, code optimization techniques, machine code generation, and use of symbol table

Course Content

Unit 1: Introduction and types of translators, the structure of a compiler, design of lexical analyzer, specification & implementation of lexical analyzer. Parsers, shift-reduce parsing, operator precedence parsing, top-down parsing, recursive descent parsing, predictive parsers.

Unit 2: LR Parsers, the canonical collection of L R (O) items, construction of SLR parsing tables, constructing canonical L.R. Parsing tables, Constructing LALR parsing tables, implementation of L R Parsing Tables. Syntax-directed translation schemes, Parse trees and syntax trees, three address code, quadruples, and triples, translation of assignment statements, Boolean expressions, control statements.

Unit 3: Symbol table, data structures for symbol tables, representing scope information. Implementation of a simple stack allocation scheme, implementation of block structured languages, storage allocation in block structured language. Error -lexical phase errors, syntactic phase errors, semantic errors.

Unit 4: The principal sources of optimization, loop optimization, the DAG representation of basic blocks, problems in code generation, a machine model, a simple code generator, code generation techniques: DAGs, Loop optimizations (loop unrolling, loop fusion, loop interchange), peephole optimization.

Reference Books:

1. Aho A.V. and Ullman J.D. Principles of Compiler Design, Addison Wesley.
2. Donovan, J, System Programming, TMH.
3. D.M. Dhamdhere: Compiler Construction-Principles and Practice McMillan India.
4. David Gries: Compiler Construction for Digital Computer.

Course Outcomes:

1. Learn the fundamentals of language translator and compiler design.
2. Ability to design algorithms for Parsers (Top-down and Bottom-Up).
3. Classify and discuss intermediate code generation and optimization techniques to improve the performance of a program
4. Gain knowledge of techniques of symbol table organization, fundamentals of runtime environment and Code generation.

Course Code	**:**	**ITPE 322**
Course Title	**:**	**Stochastic Processes and Queuing Theory**

Number of Credits	**:**	**3 and 3/0/0**
Prerequisites	**:**	**MAIR13**
Course Type	**:**	**PE**

Course Learning Objectives:

1. It provides an essential base for mathematical modelling which is normally used to solve the problems of pattern recognition and machine learning.
2. It is used in the research of various science and engineering problem.

Course Content:

Unit 1: Random variables, Functions of random variables, joint1y distributed random variables. Expectation, Expectation of functions of more than one random variable, Covariance and Correlation, Auto and Cross Correlation, Correlation Co-efficient, Parameter Estimation – Maximum Likelihood Estimation and Maximum a Posteriori.

Unit 2: Stochastic Processes, Classification of stochastic processes, Stationary Random Processes- First order, second order and nth order, Strict sense stationary process, Ergodic Process, the Bernoulli process, The Poisson process, Renewal process, Advanced Renewal Theory, Renewal Function, Alternating Renewal Processes .

Unit 3: Markov Chains, Computation of n-step transition probabilities, Chapman-Kolmogorov theorem, Classification of states of a Markov Chain, Distribution of times between state changes, Irreducible finite chains with aperiodic states, Birth and Death processes, Pure Birth Process, Pure Death Process, Non-Birth-Death Processes, Analysis of program execution time.

Unit 4: Introduction to Queuing Theory, General Queuing, Components of a queuing system, Deterministic Queues, (M/M/1):(∞/FIFO)-Single server with infinite capacity, Extension to (M/M/2): (∞/FIFO), (M/M/k) : (∞/FIFO)- Multiple server with infinite capacity, Engset Loss Formula, M/G/1 queuing system, M/G/1 with non-FIFO disciplines, M/G/∞, Queues with time-varying arrival rate.

Reference Books:

1. A. Populis and S.U. Pillai, "Probability, Random Variables, and Stochastic Process", 4th Edition, McGraw-Hill.
2. S. Palaniammal, "Probability and Queueing theory." Prentice Hall of India.
3. V. Sundarapandian, "Probability, Statistics and Queueing Theory." Prentice Hall of India.

Course outcomes

1. Develop an understanding to the basic concepts of Random Processes, Expectations and Parameter Estimation.
2. Able to understand and apply the Stochastic processes to Science and Engineering problems and applications.
3. Able to calculate the n-step transition probabilities for any Markov chain and understand about the birth and death of processes.
4. Able to apply Markov chain & Birth Death process to real life problems.
5. Develop an understanding of various Queuing Systems.

Course Code	**:**	**ITPE 324**
Course Title	**:**	**Human Centered Computing**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives

1. **Understand the principles and concepts of Human-Centered Computing**: Gain a foundational understanding of the theories, models, and concepts related to Human-Centered Computing, including human-computer interaction, user-centered design, usability, and user experience.
2. **Apply user-centered design principles**: Learn how to apply design principles and methodologies that focus on creating interfaces and systems that meet users' needs and preferences. Understand the importance of iterative design, prototyping, and usability testing in the design process.
3. **Develop skills in interaction design**: Acquire skills in designing effective user interfaces and interactive systems. Learn about information architecture, visual design, interaction patterns, and designing for different platforms and devices.
4. **Understand the social and ethical implications of computing**: Explore the social, cultural, and ethical aspects of Human-Centered Computing. Consider the impact of technology on individuals, communities, and society as a whole. Examine issues such as privacy, accessibility, inclusivity, and digital divide.

Course Content

Unit-1: Introduction to Human-Centered Computing: Overview of human-centered computing, Historical perspectives and key milestones in HCI, Human factors and cognitive psychology in computing, Ethical considerations in human-centered design.

Unit-2: User-Centered Design, Principles of user-centered design, User research methods (interviews, surveys, observations), Task analysis and user modelling, Conceptual design and prototyping, Iterative design and evaluation.

Unit-3: Module 3: Usability Engineering, Usability principles and guidelines, Usability testing methods (think-aloud protocol, heuristic evaluation), Quantitative and qualitative data analysis, Usability metrics and evaluation frameworks, Accessibility and inclusive design.

Unit-4: User Experience Design, Introduction to user experience (UX) design, User interface design principles, Information architecture and interaction design, Visual design and aesthetics, UX evaluation techniques (user testing, user feedback), Human-Centered Computing in Practice, Human-centered computing in different domains (web, mobile, wearable), Cross-platform design considerations, Mobile and responsive design, Emerging technologies and human-centered computing, Case studies and real-world applications

Unit-5: Social Implications and Future Directions, Ethical considerations in human-centered computing, Privacy, security, and trust in interactive systems, Human-centered AI and machine learning, Designing for diverse user populations, Emerging trends and future directions in HCI

Text Books:

1. Judith S. Olson and Wendy A. Kellogg, eds. *Ways of Knowing in HCI*. Vol. 2. New York, NY, USA:: Springer, 2014.
2. Don Norman, The psychology of everyday things. Basic books; 1988.
3. Jennifer Preece, Yvonne Rogers, and Helen Sharp, Interaction design: beyond human-computer interaction. John Wiley & Sons, 2015.
4. Julie A. Jacko and Andrew Sears, Human-computer interaction fundamentals. CRC press; 2009 Mar 2.

Reference Books:

1. Vredenberg K, Isensee S, Righi C. User-Centered Design: An Integrated Approach with Cdrom. Prentice Hall PTR, 2001
2. Te'eni D, Carey J, Zhang P. Human-computer interaction: Developing effective organizational information systems. John Wiley & Sons, Inc.; 2007.

Course outcomes

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

1. Apply user-centered design methodologies to the development of interactive systems.
2. Evaluate and critique the usability and user experience of interactive systems.
3. Demonstrate proficiency in designing and prototyping user interfaces.
4. Apply human-centered computing principles to different domains, such as web, mobile, and wearable technologies.
5. Identify ethical considerations and social implications related to human-centered computing.

Course Code	**:**	**ITTC 302**
Course Title	**:**	**Randomized Algorithm and Probabilistic Analysis**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives:

1. To develop the understanding of advanced algorithms.
2. To study the complexity of advanced algorithms.
3. Design new algorithms or modify existing ones for new applications and able to analyze the space & time efficiency of most algorithms

Course Content:

Unit 1: Introduction: Models of computation, randomized polynomial identity testing algorithm (Schwartz-Zippel), Karger's randomized min-cut algorithm. Randomized Primality Testing, arkov's inequality, Chebyshev's inequality, moment generating functions and Chernoff bounds. Randomized routing.

Unit 2:. Moments and Deviations, Second-Moment Method, Numerical Probabilistic algorithms, Lovasz local lemma with applications, Poissonization technique, Power of two choices, Metric embeddings,

Unit 3: Intro to Markov Chains, Mixing Times. Strong Stationary Times, and Coupling. Martingales, stopping times. The Martingale Stopping Theorem, and applications.

Unit 4:. Approximation Algorithms: Greedy Approximation Algorithms, Dynamic Programming and Weakly Polynomial-Time Algorithms, Linear Programming Relaxations, Randomized Rounding, Load balancing.

Text Books:

1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Third Edition. MIT Press and PHI, 2010.
2. Algorithm Design and Applications by Michael T. Goodrich and Roberto Tamassia, John wiley publication.
3. Randomized Algorithms by Rajeev Motwani, Prabhakar Raghavan, published by Cambridge University Press, 2014.
4. Probability and Computing: Randomized Algorithms and Probabilistic Analysis, by Mitzenmacher and Upfal, Cambridge University Press, 2nd edition, 2017
5. The Design of Approximation Algorithms by David P. Williamson and David B. Shmoys, Cambridge University Press. 6. Algorithm Design by Jon Kleinberg, Eva Tardos by Pearson publications

Course Outcomes:

1. Apply basic concepts of approximation, randomization and distributed computing in algorithmic context.
2. Designs randomized parallel algorithms, approximation and distributed algorithms that run fast or that return the correct output with high probability
3. Derives good upper bounds for the expected running time of advanced algorithms.
4. Can apply the probabilistic method to show the existence of certain combinatorial objects design and analyse.

Course Code	**:**	**ITTC 304**
Course Title	**:**	**High Performance Computing**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course Code)	**:**	**CSPE315**
Course Type	**:**	**TC**

Course Learning Objectives:
The objective of course, is to provide a comprehensive introduction to parallel algorithms and parallel programming, with strong emphasis on the design of parallel algorithms and their rigorous analysis. Exposure to parallel programming is provided through programming assignments using MPI. Throughout the course, the design of algorithms is interlaced with the programming techniques useful in coding them.

Course Content:
Unit 1: Performance measures: Speedup, efficiency and scalability. Model of parallel computation and basic communication primitives. Parallel prefix and applications, Parallel sorting, Embeddings, Parallel matrix algorithms.

Unit 2: Communication networks for parallel computers and parallel models of computation, Parallel fast Fourier transforms.

Unit 3: Parallel Programming with MPI: Writing and executing MPI programs, collective communication, grouping data for communication, communicators and topologies.

Unit 4: Parallel random number generation, Parallel Octrees, Parallel N-body methods, Parallel Bayesian network construction.

Reference Books:
1. Introduction to Parallel Computing: Design and Analysis of Algorithms by A. Grama, A. Gupta, G. Karypis and V. Kumar.
2. Parallel Programming with MPI by Peter S. Pacheco.
3. Introduction to High Performance Scientific Computing Victor Eijkhout, Edmond Chow, Robert van de Geijn.

Course Outcomes
1. Understand in a general sense the architecture of high-performance computers.
2. Understand how the architecture of high-performance computers affects the speed of programs run on HPCs.
3. Understand how memory access affects the speed of HPC programs.
4. Understand Amdahl's law for parallel and serial computing.
5. Understand the importance of communication overhead in high performance computing.
6. Understand some of the general types of parallel computers.
7. Understand how different types of problems are best suited for different types of parallel computers.
8. Understand some of the practical aspects of message passing on MIMD machines.

Course Code	**:**	**ITTC 306**
Course Title	**:**	**Distributed Computing**
Number of Credits	**:**	**4 and 3/0/2**
Prerequisites (Course Code)	**:**	
Course Type	**:**	**TC**

Course Learning Objectives:
1. To understand fundamental concepts of distributed computing.
2. To acquire knowledge to design and develop protocols for distributed systems and networks.

Course content

Unit 1: Fundamental issues in Distributed Systems, Models of Distributed Computations, Handling time through clocks, Physical Time and Hardware Clock Synchronization, Logical Time and Software Clocks, Message Ordering Protocols, Interprocess Communication, Shared Memory Vs Message Passing Models, Distributed and Localized Algorithms, Evaluation of Correctness and Performance.

Unit 2: Protocols for Routing and Designing Virtual Constructs, MST, MIS, and CDS, Distributed Mutual Exclusion, Leader Election,

Unit 3: Distributed Deadlock Detection, Termination Detection, Global State Computation and Snapshot Recording.

Unit 4: Global Predicate Detection, Fault-Tolerance Issues, Basic Techniques for handling Crash Failure in Distributed Systems, Checkpointing and Rollback Recovery, Z-path and Z-cycles, Other Types of Failures in Distributed Computing Systems, Consensus and Agreement Protocols, FLP impossibility, Byzantine Generals Problem, Failure Detectors, Wait Freedom, Self-Stabilization, Peer-to-Peer Computing, Overlay Structures, Distributed Scheduling and Load Balancing, Naming, Security.

Reference Books:
1. A D Kshemkalyani and MukeshSinghal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press 2008.
2. G Colouris, J Dollimore, and T Kindberg, Distributed Systems: Concepts and Design, 3/e Pearson Ed. 2002.
3. A S Tanenbaum and M van Steen, Distributed Systems: Principles and Paradigms, 3/e Pearson Ed. 2002.
4. V K Garg, Principles of Distributed Systems, Kluwer Academic Publishers, 1996.
5. Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, 2/e Chapman& Hall.
6. V K Garg, Principles of Distributed Computing, IEEE Press.
7. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann Press.
8. Gerard Tel, Introduction to Distributed Algorithms, Cambridge University Press.

Course Outcome
1. Able to apply knowledge of operating systems and networks to distributed systems
2. Design distributed algorithms for complex computing problems
3. Analyze the performance of algorithms and evaluate the scalability of solutions
4. Adapt existing centralized algorithms to distributed set up
5. Simulate the algorithms in distributed environment

Course Code	**:**	**ITTC 308**
Course Title	**:**	**Deep Learning**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

The objective Deep Learning course is to cover the fundamentals of neural networks as well as some advanced topics such as recurrent neural networks, long short-term memory cells and convolutional neural networks. This course also covers the basic applications of the deep learning.

Course Content

Unit-1: History of Deep Learning, Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm.

Unit-2: Feedforward Networks: Multilayer Perceptron, Gradient Descent, Backpropagation, Empirical Risk Minimization, regularization, autoencoders. Deep Neural Networks: Difficulty of training deep neural networks, Greedy layerwise training.

Unit-3: Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization), Recurrent Neural Networks: Back propagation through time, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs.

Unit-4: Convolutional Neural Networks: LeNet, AlexNet, ZF-Net, VGGNet,Visualizing Convolutional Neural Networks, Guided Backpropagation, Deep Dream, Deep Art, Fooling Convolutional Neural Networks, Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

Unit-5: Recent Trends: Variational Autoencoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning, Applications: Image based classification, Vision, NLP.

Text Books:

1. Ian J. Goodfellow, Bengio Yoshua, and Aaron Courville. "Deep Learning." An MIT Press book in preparation, 2016.
2. Charu C. Aggarwal “Neural Networks and Deep Learning: A Textbook” Springer Nature, 2018.
3. Seth Weidman “Deep Learning from Scratch: Building with Python from First Principles” O’REILLY, 2019
4. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola “Dive into Deep Learning: Tools for Engagement” CORWIN, 2019

Reference Books:

1. Ian J. Goodfellow, Bengio Yoshua, and Aaron Courville. "Deep Learning." An MIT Press book in preparation, 2016.

Course outcomes

At the end of the course, student will have understanding of following concepts/ theories/ application:

1. Fundamental Understanding of Neural network.
2. Basic and Advance concept of Deep Learning.
3. Application of Deep Learning.

Course Code	**:**	**ITTC 310**
Course Title	**:**	**Knowledge Representation and Reasoning**
Number of Credits and L/T/P scheme	**:**	**4 and 3 0 2**
Prerequisites (Course code)	**:**	**Some exposure to formal languages, logic and programming**
Course Category	**:**	**TC**

Course Learning Objectives

1. An understanding of formal logic.
2. Familiarity with representations for several core domains.
3. Hands-on experience creating and using knowledge representations.

Course Content

Unit-1: Introduction, Propositional Logic, Syntax and Semantics, Proof Systems, Natural Deduction, Tableau Method, Resolution Method, First Order Logic (FOL), Syntax and Semantics, Unification, Forward Chaining.

Unit-2: The Rete Algorithm, Rete example, Programming Rule Based Systems, Representation in FOL, Categories and Properties, Reification, Event Calculus, Deductive Retrieval, Backward Chaining, Logic Programming with Prolog

Unit-3: Resolution Refutation in FOL, FOL with Equality, Complexity of Theorem Proving , Description Logic (DL), Structure Matching, Classification, Extensions of DL, The ALC Language, Inheritance in Taxonomies

Unit-4: Default Reasoning, Circumscription, The Event Calculus Revisited, Default Logic, Autoepistemic Logic, Epistemic Logic, Multi Agent Scenarios.

Text Books:

1. Ronald J. Brachman, Hector J. Levesque: Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.
2. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.

Reference Books:

1. Schank, Roger C., Robert P. Abelson: Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures. Hillsdale, NJ: Lawrence Erlbaum, 1977.

Course outcomes

1. Describe, apply and demonstrate syntactic and semantic formal models for natural language processing
2. Apply and demonstrate knowledge-based learning methods.
3. Write Prolog programs to solve automated reasoning tasks and explain how they will execute
4. Apply reasoning techniques (transformation to clausal form, resolution, saturation) to establish properties of first-order problems.

Course Code	**:**	**ITTC-318**
Course Title	**:**	**Generative AI**
Number of Credits and L/T/P scheme	**:**	**4 & 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

1. To learn about working principles of different GANs
2. To learn about VAEs and Diffusion Models
3. To implement and apply different generative models in different application domains
4. To To understand the ethical concerns of Generative AI

Course Content

Unit 1 Foundations of Generative AI and

Deep learning and neural networks, Key concepts: activation functions, back propagation, gradient descent, Different types of neural networks - CNNs, RNNs, transformers, Generative vs. discriminative models

Unit 2 Generative Adversarial Networks (GANs)

Core concepts of GANs: generator and discriminator, Training process and challenges, Advanced GAN Variants: DCGAN, WGAN, StyleGAN, Applications of GANs: image generation, data augmentation

Unit 3 Variational Autoencoders (VAEs)

Concepts of VAEs and their architecture, Difference between VAEs and GANs, Mathematical Foundation: Latent Space Representation, Encoding and decoding process, Applications of VAEs: anomaly detection, dimensionality reduction

Unit 4 Diffusion Models

Understanding the diffusion process, Forward and reverse diffusion, Applications of diffusion models: text generation, audio synthesis

Unit 5 Generative AI and Ethical Concerns

Bias, Copyright Issues, Adversarial Attacks, Responsible AI development and deployment, Future trends and research directions in Generative AI

Text Books:

1. Altaf Rehmani , Generative AI for everyone: Understanding the essentials and applications of this breakthrough technology, Kindle Edition, 2024
2. Dhamani, Numa, Introduction to generative AI, Simon and Schuster, 2024
3. Charu C, Aggarwal, Neural networks and deep learning: a textbook, Springer, 2018.

Reference Books and Papers:

1. Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144
2. Zhi-Han, Yang. "Training latent variable models with auto-encoding variational Bayes: A tutorial." arXiv preprint arXiv:2208.07818 (2022)

Course outcomes

1. Students will be equipped with the knowledge of different types of GANs, VAEs and Diffusion Models
2. Students will be able to implement and apply the concept of generative AI to different applications
3. Students will be well aware of ethical concerns of Generative AI and future directions

Course Code	**:**	**ITTC 312**
Course Title	**:**	**Information Retrieval**
Number of Credits and L/T/P scheme	**:**	**4 and 3-0-2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

1. **Understand the fundamentals of Information Retrieval**: Gain a foundational understanding of the key concepts, models, and theories in Information Retrieval. Learn about the retrieval process, including indexing, query formulation, relevance ranking, and result presentation.
2. **Familiarity with Information Retrieval models**: Learn about different retrieval models such as Boolean, vector space, probabilistic, and language models. Understand how these models represent and measure the relevance of documents to user queries.
3. **Study retrieval evaluation**: Learn about evaluation metrics and methodologies used to assess the effectiveness of information retrieval systems. Understand how to measure retrieval performance, relevance judgments, and precision, recall, and effectiveness measures like average precision and mean average precision.
4. **Study advanced retrieval techniques and applications**: Explore advanced topics in information retrieval, such as relevance feedback, query expansion, clustering, filtering, and recommendation systems. Understand how these techniques improve the effectiveness and efficiency of retrieval systems. Understand the application of information retrieval techniques in different domains, such as text mining, information extraction, question answering, and recommender systems.
5. **Understand web search and ranking algorithms**: Learn about the unique challenges and techniques involved in web search. Explore web crawling, link analysis, PageRank, and other ranking algorithms used in search engines.

Course Content

Unit-1: Fundamental notions: Data vs Information Processing, DIKW model, data vs information retrieval, search system vs search engine, components of search system, type of search system, role of IR process in search system. design challenges and issues, IR Versus Web search, components of IR process/system, search system computing modules(overview): Text Acquisition, Transformation, Indexing, Relevance Estimation, etc.

Unit 2- Building Blocks of IR (Indexing Process): Text acquisition: Crawling & Feed, type of Crawling, meta-crawlers, Design challenges of crawling process, Text Transformation**:** Stopping, Stemming, Stemming Strategies & Algorithms etc, Link Analysis, Strategies and Algorithms, Indexing: Details of Indexing process & elements, Inverted indices, web indexes, near-duplicate detection, Index Compression, efficient processing with sparse vectors.

Unit-3: Internals of Retrieval Process: Introduction of Retrieval Model, Role of retrieval model in IR, History of model, Types of retrieval strategy and Evolution: Boolean and Vector-space retrieval models, Language Model based IR, Probabilistic IR, Advanced Models, Relevance Estimation: Term weighting scheme, TF-IDF weighting, cosine similarity, etc. Relevance feedback and query expansion. User Interface, Visual Elements: Visualization Accomplishment & Visualization Implications.

Unit-4: Web Search and IR Paradigms: Web Search Overview, Evolution of WEB (ARPANET, Internet, WWW), Characterizing the Web, Meta-search, Web Search Architectures, Search engine optimization (SEO), Link Analysis and Specialized Search: Link Analysis, Hubs and Authorities, Page Rank and HITS algorithms, Searching and Ranking: Relevance Scoring and Ranking for Web Similarity, Advanced Search Paradigms: Recommendation System: Collaborative filtering, Content-based recommendation of documents and products, Hidden Web: handling “invisible”, Web-Snippet generation, Summarization, Question Answering System, Cross-Lingual Retrieval & Social Media Search.

Text Books:

1. Ricardo Baeza -Yates and Berthier Ribeiro Neto, Modern Information Retrieval: The Concepts and Technology behind Search 2 nd Edition, ACM Press Books 2011.
2. C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

Reference Books:

1. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1 st Edition Addison Wesley, 2009.
2. Mark Levene, An Introduction to Search Engines and Web Navigation, 2 nd Edition Wiley, 2010.

Course outcomes

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

1. Demonstrate knowledge of various retrieval models, such as Boolean, vector space, probabilistic, and language models.
2. Develop proficiency in implementing indexing techniques to efficiently organize and retrieve information.
3. Evaluate the effectiveness of retrieval systems using appropriate evaluation metrics and methodologies.
4. Understand the challenges and techniques involved in web search, including web crawling, link analysis, and ranking algorithms.
5. Explore the application of information retrieval techniques in areas such as text mining, information extraction, and question answering.

Course Code	**:**	**ITTC 314**
Course Title	**:**	**Introduction to Statistical Modelling and R-Programming**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

To develop an ability to learn how to solve the problems by statistical methods and design the efficient algorithms to solve the real time problems by R- programming.

Course Content

Unit 1. Random Variables- Engineering Applications to Random variables, Discrete random vectors, Probability Mass Function (PMF), Probability Density Function (PDF), Cumulative Distribution function, Probability Generating Function, Discrete distributions: Bernoulli, Binomial, Geometric, Negative Binomial, Poisson, Hyper geometric and Uniform; Continuous distributions: Exponential, Hyperexponential, Erlang, Gamma, Normal; Functions of random variables, jointly distributed random variables.

Unit 2. Expectation- Moments, Expectation of functions of more than one random variable, Conditional PMF, PDF and expectation, Moments and transforms of some distributions: Uniform, Bernoulli, Binomial, Geometric, Poisson, Exponential, Gamma, Normal; Computation of mean time to failure. Stochastic Processes: Bernoulli process, Poisson process, renewal process and their classification.

Unit 3. Markov Chains and Regression - Computation of n-step transition probabilities, Distribution of times between state changes, M/G/1 queuing system, Discrete parameter Birth-Death processes, Analysis of program execution time. Continuous parameter Markov Chains, Birth-Death process with special cases, Non-Birth-Death Processes, Maximum likelihood estimation, Bayesian Estimation Techniques, Expectation Maximization, Linear Regression, Application of Regression in pattern recognition

Unit 4. Programming- R interpreter, major R data structures like vectors, matrices, arrays, list and data frames, Control Structures, vectorized if and multiple selection, functions. Read/write data from/in files, extracting data from web-sites, clean data, Transform data by sorting, adding/removing new/existing columns, centring, scaling and normalizing the data values, converting types of values, using string in-built functions

Unit 5. Visualizing Data and Designing GUI: Statistical analysis of data for summarizing and understanding data, Visualizing data using scatter plot, line plot, bar chart, histogram and box plot, Building interactive application and connecting it with database and Building Packages.

Text Books:

1. Trivedi, Kishor S. Probability, Statistics with Reliability, Queuing and Computer Science Applications, 2nd Ed. Wiley-India Ed., 2008.
2. Ross, Sheldon M., Introduction to Probability Models. 11th Ed. Academic Press Inc, 2014.
3. Corol Ash, The Probability Tutoring Book: An Intuitive Course for Engineers and Scientists (And Everone Else!) 1st Ed., Wiley. 1993
4. Feller, W. An Introduction to Probability Theory and its applications, 2nd Vols., Wiley Eastern, 1975.
5. Cotton, R., Learning R: a step by step function guide to data analysis. 1st Ed. O'reilly Media Inc.

Reference Books:

1. Hayes, J.F. Modeling of Computer Communication Networks. Khanna Publishing, Delhi.
2. William Mendenhall and Terry T. Sincich, Statistics for the Engineering and Computer Sciences, Dellen Publishing Company, 1988.
3. Kleinrock, L. Queuing Systems, 2 Vols, John Wiley, 1976.
4. Gardener, M. Beginning R: The statistical programming language, Wiley, 2017.
5. Lawrence, M., & Verzani, J, Programming Graphical User Interfaces in R. CRC press, 2016.

Course Outcomes

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

1. Understand to the basic concepts of Probability and expectation
2. Apply Probability theory to formulate discrete and continuous distributions
3. Apply Markov chain, queuing theory, birth Death process and regression techniques in computation algorithms
4. Develop the real-world applications using R-Programming.

Course Code	**:**	**ITTC 316**
Course Title	**:**	**Optimization Techniques for Data Science**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites Course code)	**:**	
Course Category	**:**	**TC**

Course Objectives

Optimization algorithms has been an essential tool in many scientific and engineering fields, particularly so in data science where large amounts of data need to be processed quickly.

This has caused a paradigm shift from conventional optimization algorithms such as interior-point methods to the old and classic first-order gradient methods, with many new extensions and analyses. The main goal of this course is to introduce the students to these new/old algorithms

Aim is to understand which algorithm is good for solving what kind of data science problems.

Prerequisites

Knowledge of linear algebra (vector space, eigen value, matrix multiplication, etc.), multivariate calculus (derivative, gradient, etc.), basic analysis (convergence, limits, etc.), and basic probability (common distributions, means, etc.).

Exposure to numerical computing or machine learning is a plus but not required. Knowledge of programming in either Python or Julia (can learn in a few hours).

Course Contents

Unit 1: Introduction: Data Analysis and Optimization, Least Squares, Matrix Factorization Problems, Support Vector Machines, Logistic Regression, Deep Learning

Unit 2: Foundations of Smooth Optimization: A Taxonomy of Solutions to Optimization Problems, Taylor's Theorem, Characterizing Minima of Smooth Functions, Convex Sets and Functions, Strongly Convex Functions

Unit 3: Descent Methods: Descent Directions, Line search. Acceptability of step sizes,General minimization algorithm; Steepest-Descent Method: Convergence, Line-Search Methods; Mirror Descent, The KL and PL Properties

Unit 4: Gradient Methods, Stochastic Gradient, Coordinate Descent

Unit 5: First-Order Methods for Constrained Optimization: Optimality Conditions, Euclidean Projection, The Projected Gradient Algorithm, The Conditional Gradient (Frank–Wolfe) Method

Unit 6: Duality and Algorithms: Quadratic Penalty Function, Lagrangians and Duality, First-Order Optimality Conditions, Strong Duality, Dual Algorithms, Dual Subgradient, Augmented Lagrangian Method, Alternating Direction Method of Multipliers, Some Applications of Dual Algorithms

Textbooks

1. Optimization for Data Analysis. Stephen J. Wright and Benjamin Recht. Cambridge University Press, 2022.
2. First-Order Methods in Optimization. Amir Beck. SIAM, 2017.
3. Introduction to Optimization. Boris Polyak. Optimization software, 1987.
4. Lectures on Convex Optimization (2nd). Yurii Nesterov. Springer, 2018.

Course Outcome: The course will start from basic definitions and theorems in convex analysis, emphasizing the fundamental role of convex duality. Then, we are going to analyze an arsenal of gradient algorithms, each motivated using an appropriate machine learning application. Towards the end of the course, the students will be able to

1. Recognize and formulate convex, nonconvex, and structured optimization problems;
2. Understand the basic theory and algorithms for solving large-scale problems;
3. Gain some programming experience on numerical implementations;
4. Use the material in current or future research.

Course Code	**:**	**ITPE 407**
Course Title	**:**	**Medical Image Computing**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives
The goal of the course is to give students the background and skills of medical image computing.

Course Content
Unit-I: Introduction to Medical Imaging, Basic image processing techniques, Noise and error propagation and methods to handle it, Medical image Enhancement, Introduction to medical Imaging Modalities and image analysis softwares.

Unit-2: Image registration: Rigid models, Non-Rigid models, Image registration Application and demonstration, Texture in Medical Images, Medical Image segmentation: Region Growing and Clustering, Random Walks for Segmentation, Active Contours for Segmentation, Systematic Evaluation and Validation, Decision Trees for Segmentation and Classification, Random Forests for Segmentation and Classification, Neural Networks for Segmentation and Classification

Unit-3: Image Analysis, Analysis of Speckle Images: Ultrasound, Doppler and tissue harmonic, Speckle reduction, beam steering, image compounding and filtering, Cardiovascular and Ophthalmic imaging, Learning based Image Analysis, Deep Learning for Medical Image Analysis,

Unit-4: Retinal Vessel Segmentation and Classification, Vessel Segmentation in CT Scan of Lungs, Lesion Segmentation in Brain MRI, Tissue Characterization in Ultrasound, Learning based diagnosis, Application: Classification and Diagnosis of Brain Tumer, Lung Cancer, retinal disease, cardiovascular disease, other diseases.

Text Books:
1. G. R. Sinha, Bhagwati Charan Patel, “Medical Image Processing Concepts and Applications” PHI, 2014.
2. Geoff Dougherty “Digital Image Processing for Medical Applications” Cambridge University Press, 2010.
3. Rasmus R. Paulsen, Thomas B. Moeslund, “Introduction to Medical Image Analysis”, Springer, 2020.
4. S. Kevin Zhou, Hayit Greenspan, Dinggang Shen, “Deep Learning for Medical Image Analysis” Academic Press, 2017

Reference Books:
1. Geoff Dougherty “Digital Image Processing for Medical Applications” Cambridge University Press, 2010.

Course Outcomes
1. At the end of the course the students will also develop the skills that are vital to medical image-based diagnosis.
2. Students should have fundamental skills and understanding of medical image modalities and basic processing approach.
3. Students should have segmentation, analysis and learning based diagnosis approach.

Course Code	**:**	**ITPE 409**
Course Title	**:**	**Modern Network Security**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives:

1. To comprehend the importance of network security and recognize various network security threats.
2. To apply knowledge of IP security protocols, transport-level security mechanisms, and email security standards to secure network communications.
3. To understand the security concerns in wireless networks and implement security measures for wireless LANs and mobile devices.
4. To deploy and configure firewalls and intrusion detection/prevention systems to protect network infrastructure.
5. To understand and secure modern network architectures.

Course Content:

Unit 1: Fundamentals of Network Security: Need for Network Security, Challenges for a Secure Network, OSI Security Architecture, Recent Developments in Network Security, Introduction to Cryptography, Network Security Threats: Brute Force, Distributed Denial of Service (DDoS), Botnets, ARP poisoning, IP Spoofing, Malware, Structured Query Language (SQL) injection, Cross Site Scripting (XSS) Vulnerability.

Unit 2: End-to-End Security: IP Security: the IPSec Protocol, Authentication Header, Encapsulating Security Payload, Internet Key Exchange, IPv6 and Mobile IPv6 Security, Transport-Level Security: Secure Socket Layers (SSL) and Transport Layer Security (TLS) protocols, Electronic Mail Security: S/MIME, DNS Security Domain Keys Identified Mail.

Unit 3: Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security.

Unit 4: Firewalls and Intrusion Detection/Prevention System: Need for Firewalls, Types of Firewalls, Firewall Location and Configurations, Virtual Private Networks (VPNs), Honeypots, Intrusion Prevention Systems, Applications of Machine and Deep Learning for Intrusion Detection.

Unit 5: Modern Network Architectures Security: Software-Defined Networking (SDN) Architecture and Components, Traditional Networks vs. SDN, Security Requirements, Threats to SDN, Software-Defined Security, Network Functions Virtualization (NFV) Security: Attack Surfaces, Security Techniques.

Text Books:

1. William Stallings and Brown Lowrie, “Computer Security: Principles and Practice”, Pearson, Fourth Edition, 2018.
2. William Stallings, “Cryptography and Network Security: Principles and Practice”, Seventh Edition, Pearson, 2017.
3. William Stallings, "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud" Pearson, 2016.

Reference Books:

1. Atul Kahate, “Cryptography and Network Security”, Tata McGraw-Hill, Third Edition, 2013.

Course Outcomes:

1. Ability to articulate the fundamental principles of network security, and analyze different network security threats.
2. Ability to implement end-to-end security, including IPSec, SSL/TLS, and S/MIME, to ensure secure communication.
3. Ability to apply security measures to protect wireless networks and mobile devices against potential threats.
4. Ability to configure and manage firewalls and intrusion detection/prevention systems to enhance network security.
5. Ability to understand and secure modern network architectures such as SDN and NFV.

Course Code	**:**	**ITPE 411**
Course Title	**:**	**Number Theory and Cryptography**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Computer Networks**
Course Category	**:**	**PE**

Course Objectives:

1. To emphasize the application of the number theory in the design of cryptographic algorithms.
2. To understand the strength and weakness of cryptosystems.
3. To introduce the elliptic curve cryptography.

Course Content:

Unit 1: Elementary Number Theory: Divisibility, Division Algorithm, Euclidean Algorithm, Extended Euclidean Algorithm, Fast exponentiation Algorithm; Modular Arithmetic, Congruences, Chinese Remainder Theorem, Algebraic structures, Galois Fields

Unit 2: Primality Testing and Factorization: Prime Numbers, Fermat's Theorem, Euler's Totient Function, Euler's Theorem, Testing of Primality, Miller-Rabin Algorithm, Deterministic Primality Algorithm, Fundamental Theorem of Arithmetic, Checking for Primeness, Factorization Methods: Trial Division Method, Fermat Method, Pollard's Rho method;

Unit 3: Private Key Cryptosystem: Traditional Symmetric-key ciphers, Modern Symmetric Key ciphers, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Block Cipher modes of Operation

Unit 4: Public Key Cryptosystems: Traditional Cryptosystem, limitations; Public Key Cryptography; Discrete Logarithm problem; One-way functions, Trapdoor functions; Diffie-Hellmann key exchange; RSA cryptosystem; Cryptographic Hash Function, Digital signature schemes; RSA signature scheme.

Unit 5: Elliptic Curve Cryptography: Introduction to Elliptic Curves, Geometry of elliptic curves over reals; point at infinity; Addition of two points; Elliptic Curves over finite fields, Discrete Log problem for Elliptic curves; Elliptic Curve Cryptography; ElGamal Public Key Cryptosystem for elliptic curves.

Text Books:

1. A Course in Number Theory and Cryptography, Neal Koblitz, (Springer 2006)
2. William Stallings, Cryptography and Network Security, PHI, 7/e
3. B.A. Forouzan, Cryptography and Network Security, McGraw-Hill, 3/e
4. Elliptic curves: Number theory and cryptography, Lawrence C. Washington, (Chapman & Hall/CRC 2003)

Reference books:

1. An Introduction to Cryptography, R.A. Mollin (Chapman & Hall, 2001)
2. Rational Points on Elliptic Curves, Silverman and Tate (Springer 2005)
3. Guide to elliptic curve cryptography Hankerson, Menezes, Vanstone (Springer, 2004)
4. Elementary Number Theory, Jones and Jones (Springer, 1998)
5. An Introduction to Mathematical Cryptography, Jill Pipher, Jeffrey Hoffstein, Joseph H. Silverman (Springer, 2008)
6. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006)

Course Outcomes:

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

1. Understand the significance of cryptography to the modern world
2. Able to learn basic elements of number theory and its applications in cryptography
3. Understand the mathematical foundations of Cryptographic algorithms
4. Understand Public Key Cryptography, Discrete Logarithm problem, RSA Cryptosystem, ECC and various attacks
5. Solve elementary problems in number theory relating to cryptography.

Course Code	**:**	**ITPE 413**

Course Title	**:**	**Advanced Data Structures and Algorithms**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PE**

Course Learning Objectives:
1. To develop the understanding of advanced data structures.
2. To develop the understanding of algorithmic design paradigms.

Course Content

Unit 1: Complexity: Algorithms, Revision of Performance analysis- time complexity and space complexity using asymptotic notation, Amortized complexity

Unit 2: Advanced Data Structures: Threaded binary trees, Binomial heaps, Fibonacci heaps, Balanced trees- AVL, Multi-way search trees, red-black trees, splay trees, segment trees ; Suffix tries

Unit 3: String Matching: String-matching algorithm, Rabin-Karp algorithm, String matching with automata, Knuth-Morris-Pratt algorithm.

Unit 4: NP – Hard & NP – Complete problems: Concepts of NP-hard & NP complete problems, Clique Decision problem, Node Cover decision problem, Travelling Salesperson decision problem.

Suggested Books:
1. E Horowitz and S. Sahni, Fundamentals of Data Structures, Universities Press.
2. Cormen, Leiserson and Rivest, Introduction to Algorithms, 3/e, PHI.
3. Horowitz, Ellis and Sahni, Fundamentals of Computer Algorithms, Universities Press.

Reference Books:
1. Oded Goldreich. P, NP, and NP-Completeness: The Basics of Computational Complexity, Cambridge University Press, 2010.
2. Anne Benoit, Yves Robert, Frédéric Vivien. A Guide to Algorithm Design: Paradigms, Methods, and Complexity Analysis, Taylor & Francis, 2013.
3. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, Data Structures and Algorithms, Addison Wesley, Reading Massachusetts, USA, 1983.
4. Donald Knuth. The Art of Computer Programming: Fundamental Algorithms, Third Edition. AddisonWesley, 1997.
5. Donald Knuth. The Art of Computer Programming Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0.

Course Outcome:
1. Compare, contrast, and apply key data structures: trees, heaps, string algorithms etc.
2. Compare, contrast, and apply algorithmic tradeoffs: time vs. space, deterministic vs. randomized, and exact vs. approximate.
3. Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

Course Code	**:**	**ITTC 401**
Course Title	**:**	**Parallel and Distributed Algorithms**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Operating systems, Data Structure and Algorithms, Distributed Systems**
Course Category	**:**	**TC**

Course Learning Objectives

1. To study basics of parallel and distributed computing systems.
2. To learn several techniques and methods for the creation of algorithms as well as familiar with the design constraints.
3. To explore various application areas of the parallel and distributed algorithms.

Course Content

Unit- I: Introduction

Fundamentals: Models of parallel and distributed computation, complexity measures;
Parallel Programming Models: Shared-memory model (PRAM, MIMD, SIMD), network model (line, ring, mesh, hypercube), performance measurement of parallel algorithms;

Unit- 2: Parallel algorithm design technique for PRAM model

The PRAM Model: balancing, divide and conquer, parallel prefix computation, pointer jumping, symmetry breaking, pipelining, accelerated cascading;

Parallel algorithm design model for PRAM model

list ranking, Euler Tour, sorting, searching and Merging, graph algorithms, string algorithms; parallel complexity and complexity classes NC, P-completeness, lower bounds;

Unit- 3: Parallel algorithm for network models

Interconnection Networks: topologies (arrays and mesh networks, trees, systolic networks, hypercubes, butterfly) and fundamental algorithms, matrix algorithms, *sorting, graph algorithms*, routing, relationship with PRAM models; Asynchronous Parallel Computation; Multiprocessor Algorithms;

Unit- 4: Distributed Algorithms

Distributed Algorithms: models and complexity measures, safety, liveness, termination, logical time and event ordering, global state and snapshot algorithms, mutual exclusion, clock synchronization, election, termination detection, routing, Distributed graph algorithms; Applications of Distributed algorithms.

Text Books:

1. Michael J Quinn, Parallel Computing, TMH
2. Joseph Jaja, An Introduction to Parallel Algorithms, Addison Wesley
3. Mukesh Singhal and Niranjan G. Shivaratri, Advanced Concepts in Operating Systems, TMH
4. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Pearson

Reference Books:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition, Addison Wesley, 2003.
2. Michael J. Quinn, Designing efficient algorithms for parallel computers, McGraw-Hill Ryerson, Limited, 1987.
3. Selim G. Akl , The Design and Analysis of Parallel Algorithms, Prentice Hall, Englewood Cliffs, New Jersey, 1989.
4. Mukesh Singhal and Niranjan Shivaratri, Advanced Concepts in Operating Systems, McGraw-Hill.
5. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
6. Andrew S. Tanenbaum, Distributed Operating Systems, ACM Press.
7. Jie Wu, Distributed Systems, CRC Press.
8. Hagit Attiya, Jennifer Welch, Distributed Computing: Fundamentals, Simulations and Advanced Topics, McGraw-Hill.

9. Sape Mullender (ed.), Distributed Systems, Addison-Wesley.
10. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press.
11. Allan Borodin, Ran El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press.
12. Robert Endre Tarjan, Data Structures and Network Algorithms, SIAM.
13. L. Grotchel, L. Lovasz, and A. Schrijver, Geometric algorithms and Combinatorial Optimization, Springer.
14. M. Kearns and U. Vazirani, An Introduction to Computational Learning Theory. MIT Press.
15. N. Alon and J. H. Spencer, The Probabilistic Method, John Wiley.
16. Vijay Vazirani, Approximation Algorithms, Springer.
17. Fan Chung, Spectral Graph Theory, American Mathematical Society
18. Joseph F Jájá, An Introduction to Parallel Algorithms, Addison-Wesley, 1992.
19. Michael J Quinn, Parallel Computing: Theory and Practice, second edition, McGraw Hill, 1994/2002.
20. Russ Miller and Laurence Boxer, Algorithms: Sequential and Parallel — A Unified Approach, third edition, Cengage, 2013.
21. Fayez Gebali, Algorithms and Parallel Computing, Wiley, 2011.
22. Seyed H Roosta, Parallel Processing and Parallel Algorithms: Theory and Computation, Springer, 2000.

Course Outcomes

1. The students will be able to contructs parallel and distributed algorithms.
2. They will be able to execute several instructions simultaneously on different processing devices and then combine all the individual outputs to produce the common result.

Course Code	**:**	**ITTC 403**
Course Title	**:**	**Cloud Architecture and Networking**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Objectives:

1. To understand the concepts of Cloud Computing.
2. To learn Taxonomy of Virtualization Techniques.
3. To learn Cloud Computing Architecture.
4. To acquire knowledge on Aneka Cloud Application Platform.
5. To cloud networking issues.

Course Content

Unit 1: Introduction to Cloud Computing: Cloud Computing (NIST Model), Cloud service providers Properties, Characteristics & Disadvantages, Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Role of Open Standards

Unit 2: Cloud Computing Architecture: Cloud computing architecture, Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services

Unit 3: Service Models: Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Interoperability and Standards, Scalability and Fault Tolerance

Unit 4: Introduction to virtualization: Need of virtualization – cost, administration, fast deployment, reduce infrastructure cost – limitations Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization Desktop virtualization: Software virtualization – Memory virtualization - Storage virtualization – Data virtualization – Network virtualization,

Unit 5: Service Management in Cloud Computing: Service Level Agreements (SLAs), Billing & Accounting Managing Data, Looking at Data, Scalability & Cloud Services, Database & Data Stores in Cloud Large Scale Data Processing

Unit 6: Cloud Security: Infrastructure Security, Network level security, Host level security, Application-level security, Data security and Storage, Data privacy and security Issues, Jurisdictional issues raised by Data location, Identity & Access Management, Access Control Trust, Reputation, Risk, Authentication in cloud computing

Case Study on Open Source & Commercial Clouds: Eucalyptus, Microsoft Azure, Amazon EC2

Reference Books

1. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
2. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011
3. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
4. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010

Course Outcomes:

At the end of this course student will:

1. Understand the concept of virtualization and how this has enabled the development of Cloud Computing
2. Know the fundamentals of cloud, cloud Architectures and types of services in cloud
3. Understand scaling, cloud security and network management
4. Design different Applications in cloud
5. Explore some important cloud computing driven commercial systems.

Course Code	**:**	**ITTC 405**

Course Title	**:**	**Quantum Computing**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

1. be familiar with basic quantum algorithms and their analysis.
2. be familiar with basic quantum protocols such as teleportation and super dense coding.
3. see how the quantum model relates to classical models of deterministic and probabilistic computation.
4. appreciate the importance of efficient error-suppression if quantum computation is to yield an advantage over classical computation.
5. gain a general understanding of the important topics in near-term quantum computing, including adiabatic quantum computing.

Course Content

Unit-1:

Bits and qubits: Introduction to quantum states and measurements with motivating examples, Comparison with discrete classical states.

Linear algebra: Review of linear algebra: vector spaces, linear operators, Dirac notation, the tensor product.

The postulates of quantum mechanics: Postulates of quantum mechanics, incl. evolution and measurement.

Unit-2:

The quantum circuit model: The circuit model of quantum computation, Quantum gates and circuits, Universality of the quantum circuit model, and efficient simulation of arbitrary two-qubit gates with a standard universal set of gates.

Quantum search: Grover's search algorithm: analysis and lower bounds.

Unit-3:

Quantum Fourier Transform and Quantum Phase Estimation: Definition of the Quantum Fourier Transform (QFT), and efficient representation thereof as a quantum circuit., Application of the QFT to enable Quantum Phase Estimation (QPE).

Application of QFT / QPE Factoring: Shor's algorithm: reduction of factoring to period finding and then using the QFT for period finding.

Unit-4:

Quantum complexity: Quantum complexity classes and their relationship to classical complexity, Comparison with probabilistic computation.

Quantum error correction: Introducing the concept of quantum error correction.

Fault tolerant quantum computing: Elements of fault tolerant computing; the threshold theorem for efficient suppression of errors.

Unit-5:

Adiabatic quantum computing and quantum optimisation: The quantum adiabatic theorem, and adiabatic optimisation. Quantum annealing and D-Wave.

Case studies in near-term quantum computation: Examples of state-of-the-art quantum algorithms and computers, including superconducting and networked quantum computers.

Text Books:

1. Nielsen M.A., Chuang I.L. (2010). Quantum Computation and Quantum Information. Cambridge University Press.
2. Mermin N.D. (2007). Quantum Computer Science: An Introduction. Cambridge University Press.
3. McGeoch, C. (2014). Adiabatic Quantum Computation and Quantum Annealing Theory and Practice. Morgan and Claypool.

Reference Books:

1. Quantum Computing, A Gentle Introduction , Eleanor G. Rieffel and Wolfgang H. Polak MIT press (2014)

Course outcomes:

On successful completion, students will gain understanding of:
The basic principles of quantum computing.

1. The fundamental differences between conventional computing and quantum computing.
2. Several basic quantum computing algorithms.
3. The classes of problems that can be expected to be solved well by quantum computers.

Course Code	**:**	**ITTC 407**
Course Title	**:**	**Cognitive Applications of ML**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	
Course Category	**:**	**TC**

Course Learning Objectives

1. Understand the fundamental concepts and principles of machine learning and cognitive computing.
2. Identify and describe the various cognitive applications of machine learning in different domains.
3. Critically evaluate the challenges and limitations of cognitive applications of machine learning.

Course Content

Unit-I: Introduction to Cognitive Applications of Machine Learning: Overview of Machine Learning and its Application, Introduction to Cognitive Computing and its Significance, Ethical Considerations in Cognitive Applications, Neural Network and Deep Learning.

Unit-2: Cognitive Computing and Natural Language Processing: Introduction to Natural Language Processing, Text Classification and Sentiment Analysis, Named Entity Recognition and Entity Linking, question Answering System. Computer Vision and Cognitive Applications: Introduction to Computer Vision and Image Understanding, Image Classification and Object Detection, Image segmentation and image-to-text generation, Cognitive Applications in Healthcare and Autonomous vehicles.

Unit-3: Deep Learning and Cognitive Applications: Overview of Deep Learning and Neural Networks, CNN for Image Analysis, RNN for Sequence Data, Generative Models and their Cognitive Applications. Reinforcement Learning and Cognitive System, Cognitive Computing and AI Systems: Introduction to Cognitive Architecture, Knowledge Representation and Reasoning, Cognitive Agent and Multi-agent Systems, Explainable AI and interpretability in Cognitive System.

Unit-4: Applications of Cognitive Machine Learning: Intelligent virtual assistants and Chatbots, Recommendation systems, Cognitive analytics and decision support systems, Healthcare applications and medical diagnosis. Ethical and Social Implications of Cognitive Machine Learning, Analysis of real-world cognitive machine learning applications.

Text Books:

1. Judith Hurwitz, Marcia Kaufman, and Adrian Bowles, "Cognitive Computing and Big Data Analytics", Wiley, 2015.
2. Vinit Kumar Gunjan and Jacek M. Zurada, “Modern Approaches in Machine Learning & Cognitive Science: A Walkthrough” Springer, 2022.
3. Sasikumar Gurumoorthy, Bangole Narendra Kumar Rao and Xiao-Zhi Gao “Cognitive Science and Artificial Intelligence Advances and Applications” Springer, 2018.

Reference Books:

1. Dagmar Monett Diaz and Mario Arias Oliva, “Cognitive Computing: Theory and Applications”, Elsevier, 2016.

Course Outcomes

1. Understand the principles and concepts of cognitive applications of machine learning.
2. Identify and describe various cognitive applications of machine learning in different domains, such as natural language processing, computer vision, and human-robot interaction.
3. Apply machine learning techniques to develop cognitive applications for tasks such as text classification, image recognition, and decision-making.
4. Evaluate the ethical and societal implications of cognitive applications of machine learning, considering factors such as bias, fairness, privacy, and human-AI collaboration.

Course Code		**ITTC 409**
Course Title	**:**	**Natural Language Processing**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Artificial Intelligence, Data structures and algorithms, programming languages**
Course Category	**:**	**TC**

Course Learning Objectives

1. To learn the fundamentals of Natural language Processing and its algorithm.
2. To understand machine translation and applications of NLP.
3. Basic understanding of deep learning models for NLP.

Course Contents

UNIT -1: Introduction to NLP: Characteristics of Natural Language, Language structure, Sentence Structure, Language analyzer, Lexicon, word formation, Morphology, syntax analysis (parsing), semantics, ambiguity, pragmatics and discourse.

UNIT- 2: NLP Algorithms: Understanding Corpus and data attributes, Corpus Formats CSV, JSON, XML, LibSVM, Operations on Text Corpus, Tokenisation, stop words, Term Frequency Inverse Document Frequency (TF-IDF), Text Analysis and word embedding using word2vec, doc2vec, GLoVe, Bag-of-words (BoW).

UNIT-3: Machine Translation and Applications of NLP: Introduction to Machine Translation (MT), Approaches, Structure of Anusaraka: an Interlingua based MT system, Example/Analogy based MT, Word/phrase based MT, Neural MT. Applications of NLP: Sentiment analysis, chatbots, conversational models (Question Answering system) for Digital Assistants

UNIT- 4: Deep learning models for NLP: Neural Net based NLP models: Study of Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit(GRU) using Natural Language Toolkit (NLTK)

Text Books

1 Thanaki, Jalaj, “Python natural language processing”. Packet Publishing Ltd, 2017.
2 Cole Howard, Hannes Max Hapke, and Hobson Lane,” Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python”, Manning, 2019.
3 Daniel Jurafsky, James H. Martin,”Speech and Language Processing: An Introduction to Natural Language Processing”, Computational Linguistics and Speech, Pearson Publication, 2014.

Reference Books

1 Lawrence Rabiner And Biing-Hwang Juang, “Fundamentals of Speech Recognition”, Pearson Education, 2003.
2 Samuel Burns, “Natural Language Processing: A Quick Introduction to NLP with Python and NLTK” Independently Published, 2019
3 Bird, Steven, Ewan Klein, and Edward Loper. “Natural language processing with Python: analyzing text with the natural language toolkit." O'Reilly Media, Inc.", 2009.

Course Outcomes:

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

1. Understand the algorithmic approach to NLP.
2. Understand the sentence structure and the Computational model of Natural Languages.
3. Understand the Data Science approach to Machine Translation and its applications.
4. Understand Machine Learning approach to NLP.

Course Code	**:**	**ITTC 411**
Course Title	**:**	**Semantic Web**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Knowledge representation and reasoning, artificial intelligence, data structures and algorithms, Web technologies, such as URL, http, HTML, and XML-based technologies, Databases.**
Course Category	**:**	**TC**

Course Learning Objectives

1. To offer an introduction to knowledge and logic-based information technologies, using logic programming as the primary example of knowledge-based reasoning, and the Semantic Web as the primary example of a knowledge-based application area.
2. To introduce the W3C standard Web Ontology Language, OWL, and its underlying Description Logics
3. To provide experience using a set of established patterns for developing OWL ontologies
4. To understand linked data technologies and applications

Course Contents:

UNIT-I: Knowledge Engineering and the Web of Data, Semantic Web standards:Uniform Resource Identifier (URI) RDF (Resource Description Framework); Ontology Engineering; OWL (Web Ontology Language), SPARQL, Semantic Web mission; concepts of semantic interoperability, integration and automation; concept of metadata and ontology; description logics.

UNIT-2: Methods for developing and evaluating ontologies. Application development using the OWL API, Tableaux Algorithm, DL Reasoning Problems, Canonical forms, Resolution (PL/FOL), OWL and RDF(S) Semantics Basics, Open and Closed world assumptions, Rules for inferring knowledge, First order Logic, RDF-S semantics, Web Ontology Language(OWL), Semantic Web Rule Language(SWRL), Friend-of-a-Friend(FOAF).

UNIT-3: Query languages SPARQL, SWRL (Semantic Web Rules Language); Semantic Technology;Rules, Protége, Ontology Alignment, Ontology Evaluation, More Ontology Design Methodologies, Metadata, Fundamentals of Ontology and its types, monolithic vs. modular ontology, ontology design methodology, ontology learning, ontology learning from text, automated ontology learning process.

UNIT-4: Linked Data Engineering,Semantic (Web) infrastructure, applications and Services; quality of semantic web applications, Relation to Big Data and Industry 4.0, Linked Data Programming, Semantic Annotation, Named Entity Resolution, Semantic Search, Exploratory Search, Linked Data Analytics, Semantic Recommendations.

Text Books

1. A Semantic Web Primer, by Grigoris Antoniou, Paul Groth, Frank van Harmelen and Rinke Hoekstra, Publisher: MIT Press; 3rd edition (September, 2012).
2. Linked Data: Structured Data on the Web, by David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. Publisher: Manning Publications; 1 edition (January 24, 2014).
3. Learning SPARQL: Querying and Updating with SPARQL 1.1, by Bob DuCharme Publisher: O'Reilly Media; 2 edition (July 18, 2013).

Reference Books:

1. A Developer's Guide to the Semantic Web" by Liyang Yu (Springer Science & Business Media, 2011)
2. Handbook on ontologies" by SteffanStaab and Rudi Studer (Eds.) (Springer Science & Business Media, 2010)
3. Kumar, S. and Baliyan, N., 2018. Semantic Web-based Systems: Quality Assessment Models. Springer.

Course Outcomes

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

1. Apply RDF, OWL, and SWRL syntax for semantic annotations and rule specification for web resources.
2. Build and visualize rich ontologies using editors such as Protégé.
3. Describe Linked Data principles and architecture, as in dbpedia, Wiki, FOAF etc.
4. Perform Linked data analysis and visualization using SPARQL with R/Python.
5. Develop a deep insight to the various state-of-the-art technologies of semantic search engine, semantic web browser and semantic recommender systems.

Course Code	**:**	**ITTC 413**
Course Title	**:**	**Advances in Data Science**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Data Mining and Analytics**
Course Category	**:**	**TC**

Course Learning Objectives

This course aims to introduce the students to different pattern mining tools and techniques for complex data considering different real time constraints on data analysis pertaining in different application domains.

Course Content

Unit 1: Pattern Mining: Maximal and Closed Patterns, Pattern Mining in Multilevel, Multidimensional Space, Constraint-Based Pattern Generation: Pruning Pattern Space and Pruning Data Space, Mining High-Dimensional Data and Colossal Patterns, Mining Compressed Patterns by Pattern Clustering, Sequential Pattern Mining

Unit 2: Advance Clustering: Clustering High-Dimensional Data, Challenges Subspace Clustering, CLIQUE, PROCLUS, Clustering Graph and Network Data, Similarity Measures, Clustering with Constraints

Unit 3: Anomaly Detection: Types of Anomalies, Statistical Approaches, Proximity Based Approached, Supervised Approaches, Mining Contextual and Collective Anomalies

Unit 4: Analysis of Complex Data and Case Studies: Algorithms for Analysis of Spatial Data, Graph Data, Text Data, Time Series Data, Case Studies in Different Domains: Medical Domain, Finance, Environment, and Transportation Systems

Text Books:

1. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.
2. Han, Jiawei, et al. Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, 2012.
3. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

Reference Books:

1. Ruey S. Tsay "Analysis of Time-series data," Third Edition, Wiley 2014
2. Jonathan Gross and Jay Yellen, Graph Theory and its Applications, Second Edition, CRC Press. 2018
3. Sara Moridpour, Alireza Toran Pour, Tayebeh Saghapour "Big Data Analytics in Traffic and Transportation Engineering: Emerging Research and Opportunities" IGI Global 2019
4. Jørgensen, Sven Erik, ed. Handbook of Environmental Data and Ecological Parameters: Environmental Sciences and Applications. Vol. 6. Elsevier, 2013.

Course outcomes

At the end of the course student will be able

1. To design efficient algorithms for finding patterns and clusters in complex data
2. To design techniques for detecting anomalies in data
3. To analyse real data coming from different domains such as medical, transport, environment etc.

Course Code	**:**	**ITTC 415**
Course Title	**:**	**Natured Inspired Optimization Techniques**
Number of Credits and L/T/P scheme	**:**	**4 and 3/0/2**
Prerequisites (Course code)	**:**	**Design and Analysis of Algorithms**
Course Category	**:**	**TC**

Course Learning Objectives

Provide the mathematical background for carrying out the optimization associated with natured inspired techniques. The primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of natural inspired optimization techniques.

Course Content

Unit 1: Introduction to Algorithms: Optimization, Search for Optimality, Nature-Inspired Metaheuristics, Analysis of Algorithms: Analysis of Optimization Algorithms, Nature-Inspired Algorithms, Parameter Tuning and Parameter Control.

Unit 2: Random Walks and Optimization: Random Variables, Optimization as Markov Chains, Step Sizes and Search Efficiency, Modality and Intermittent Search Strategy, Importance of Randomization, Eagle Strategy, Simulated Annealing: Annealing and Boltzmann Distribution, SA Algorithm, Unconstrained Optimization, Basic Convergence Properties, Stochastic Tunneling.

Unit 3: Genetic Algorithms: Role of Genetic Operators, Choice of Parameters, GA Variants, Convergence Analysis, Particle Swarm Optimization: Swarm Intelligence, PSO Algorithm, Firefly Algorithms: Standard Firefly Algorithm, Variants of the Firefly Algorithm, Cuckoo Search: Cuckoo Breeding Behavior, Special Cases of Cuckoo Search.

Unit 4: A Framework for Self-Tuning Algorithms, How to Deal with Constraints, Multi-Objective Optimization, Other Algorithms and Hybrid Algorithms: Ant Algorithms, Bee-Inspired Algorithms, Differential Evolution.

Text Books:

1. Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier publications, First Edition, 2014.

Reference Books:

2. Nazmul Siddique, Hojjat Adeli, Nature-Inspired Computing Physics- and Chemistry-based Algorithms, First Edition, 2017.

Course Outcomes

1. Learn mathematical basis as well as the general principles of various Natured-Inspired optimization techniques and algorithms.
2. Understand and implement the concept of optimization techniques.
3. Develop some familiarity with current research problems and research methods in this field.

Course Code	**:**	**ITPC 402**
Course Title	**:**	**Cloud, IoT and Big Data Security**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives:

1. To understand the fundamental concepts of Cloud Computing, IoT, and Big Data analytics.
2. To identify and understand various security threats and vulnerabilities in Cloud, IoT, and Big Data environments.
3. To implement security technologies and practices in the Cloud, including identity and access management, end-to-end access control, and security-as-a-service.
4. To apply IoT security frameworks, including IoT network design, protocols, and device security.
5. To utilize tools and technologies for securing big data environments.

Course Content:

Unit 1: Overview of Cloud Computing, IoT, and Big Data: Cloud Services and Deployment Models, Benefits and Challenges of Cloud Computing, Architecture and Applications of IoT and Big Data Analytics, Threats and Vulnerabilities in the Cloud, IoT, and Big Data Environments.

Unit 2: Cloud Security Technologies: Identity and Access Management (IAM): Authentication and Authorization Mechanisms, Identity Federation and Single Sign-On (SSO), Security Management in the Cloud, End-to-End Access Control, Security Vulnerability, Patch, and Configuration Management, Security-as-a-[cloud] Service.

Unit 3: IoT Security Framework: Introduction to IoT Security Framework, IoT Network Design, IoT Protocols, IoT Stack, Utilizing IPv6 Security Features, Trusting the IoT Device.

Unit 4: Big Data Security Techniques: Tools and Technologies for the Implementation of Big Data, Secure Data Storage and Processing, Mining of Social Networks for Crime, Practical Big Data Security Solutions, Security Frameworks and Standards for Big Data.

Text Books:

1. Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud security and privacy: an enterprise perspective on risks and compliance", O'Reilly Media, Inc.", First Edition, 2009.
2. Alasdair Gilchrist, "IoT Security Issues", de Gruyter, First Edition, 2017.
3. Babak Akhgar, Gregory B. Saathoff, Hamid R Arabnia, Richard Hill, Andrew Staniforth, Petra Saskia Bayerl, "Application of Big Data for National Security", Butterworth-Heinemann, 2015.

Reference Books:

1. William Stallings, "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud" Pearson, 2016.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", Pearson, 2013.
3. Adrian McEwen and Hakim Cassimally, "Designing the Internet of Things", Wiley, 2014.
4. Judith S. Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman, "Big Data For Dummies", Wiley, 2013.

Course Outcomes:

1. Ability to demonstrate a comprehensive understanding of cloud computing, IoT, and big data.
2. Ability to identify and analyze various threats and vulnerabilities in Cloud, IoT, and Big Data environments.
3. Ability to apply identity and access management, end-to-end access control, and other cloud security technologies to secure the Cloud environments effectively.
4. Ability to design and implement secure IoT networks using appropriate frameworks and protocols.
5. Ability to employ tools and technologies to implement secure data storage and processing in big data environments.

Course Code	**:**	**ITPC 404**
Course Title	**:**	**Mobile Application Development**
Number of Credits	**:**	**4 and 3/0/2**
Prerequisites (Course Code)	**:**	**CSIR12/CEIR12*, CSIR11, CSPC23**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Introduction and characteristics of mobile applications.
2. Application models of mobile application frameworks.
3. Managing application data and User-interface design for mobile applications.
4. Integrating with cloud services.
5. Integrating networking, the OS and hardware into mobile-applications.
6. Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.
7. Testing methodologies for mobile applications– Publishing, deployment, maintenance and management.

Course content:

Unit 1: Introduction to mobile devices: Introduction to Mobile Computing, Introduction to Android Development Environment, Mobile devices vs. desktop devices, ARM and intel architectures, Power Management, Screen resolution, Touch interfaces, Application deployment, App Store, Google Play, Windows Store, Development environments: XCode, Eclipse, VS2012, PhoneGAP, etc.; Native vs. web applications. Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User; Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia.

Unit 2: Mobile OS Architectures: Comparing and contrasting architectures of all three – Android, iOS and Windows, Underlying OS, Kernel structure and native level programming. Approaches to power management, Security. Android/iOS/Win 8 Survival and basic apps: Building a simple "Hello World" App in all three applications, App-structure, built-in Controls, file access, basic graphics. Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Mobile App Development Hurdles, Testing.

Unit 3: Android/iOS/Win8 Apps: DB access, network access, contacts/photos/etc. Underneath the frameworks: Native level programming on Android, Low-level programming on (jailbroken) iOS, Windows low level APIs. Intents and Services: Android Intents and Services, Characteristics of Mobile Applications, Successful Mobile Development; Storing and Retrieving Data: Synchronization and Replication of Mobile Data, Getting the Model Right, Android Storing and Retrieving Data, working with a Content Provider; Putting It All Together: Packaging and Deploying, Performance Best Practices, Android Field Service App.

Unit 4: Advanced Topics: Power Management: Wake locks and assertions, Low-level OS support, Writing power-smart applications. Augmented Reality via GPS and other sensors**:** GPS, Accelerometer, Camera. Mobile device security, in depth: Mobile malware, Device protections, iOS "Jailbreaking", Android "rooting" and Windows' "defenestration"; Security and Hacking: Active Transactions, More on Security, Hacking Android.

Reference Books:

1. Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 2nd edition, 2015.
2. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design and Development, Prentice Hall, 2004.
3. Tomasz Nurkiewicz and Ben Christensen, Reactive Programming with RxJava, O'Reilly Media, 2016.
4. Brian Fling, Mobile Design and Development, O'Reilly Media, Inc., 2009.
5. Maximiliano Firtman, Programming the Mobile Web, O'Reilly Media, Inc., 2nd ed., 2013.
6. Cristian Crumlish and Erin Malone, Designing Social Interfaces, 2nd ed., O'Reilly Media, Inc., 2014.
7. Suzanne Ginsburg, Designing the iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps, Addison-Wesley Professional, 2010.

Course Outcomes:

1. Understand technology and business trends impacting mobile applications
2. Be competent with the characterization and architecture of mobile applications.
3. Understand enterprise scale requirements of mobile applications.

Design and develop mobile applications using one application development framework.

Course Code	**:**	**ITPC 406**
Course Title	**:**	**Multicore and GPU Programming**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**PC**

Course Learning Objectives:

At the end of this course, you should be able to accomplish the objectives given below**:**

1. Describe different multicore architectures and programming models.
2. Given a problem, develop an efficient parallel algorithm to solve it.
3. Given a parallel algorithm, analyze its time complexity as a function of the problem size and number of processors.
4. Given a parallel algorithm, implement it using OpenMP, CUDA OpenCL/ OpenACC.
5. Given a parallel code, debug it and fix the errors.

Course content:

Unit 1: Introduction to Multi-core Architecture

Introduction to Multi-core Architecture Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper-Threading Technology, Multi-threading on Single-Core versus Multi-Core Platforms Understanding Performance. System Overview of Threading: Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application Programming Models and Threading

Unit 2: Fundamental Concepts of Parallel Programming

Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Parallel Programming Patterns. Threading and Parallel Programming Constructs: Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Barrier.

Unit 3: OpenMP

Structure of OpenMP programs, parallellism in OpenMP programs, Compiling OpenMP programs, Execution of an OpenMP program, OpenMP Support function, Syncrhonization Primitives, Managing Shared and Private Data, Loop Scheduling and Portioning, Using Barrier and No wait, Data Copy-in and Copy-out, Protecting Updates of Shared Variables

Unit 4: GPU Programming

History, GPU Architecture, Clock speeds, CPU/ GPU comparisons, Heterogeneity, Accelerators, Parallel Programming, CUDA OpenCL/ OpenACC, Kernels Launch parameters, Thread hierarchy, Warps/Wavefronts, Threadblocks/Workgroups, Streaming multiprocessors, 1D/2D/3D thread mapping, Device properties, penCL for Heterogeneous Computing , Simple Programs

Books

1. Multicore Programming , Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts , Intel Press , 2006.
2. Yan Solihin, "Fundamentals of Parallel Multicore Architecture", 1st Edition, CRC Press/Taylor and Francis, 2015.

Resources:

1. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
2. https://www.openmp.org/

Course Outcomes:

1. Identify the limitations and the need for multicore architectures
2. Define fundamental concepts of parallel programming and its design issues
3. Make out the salient features of different multicore architectures and how they exploit parallelism
4. Demonstrate the role of OpenMP and programming concept
5. Demonstrate the role of CUDA OpenCL/ OpenACC concept

Course Code	**:**	**ITOE 301**
Course Title	**:**	**Data Structures**
Number of Credits	**:**	**3 and 3/0/0**
Prerequisites	**:**	**Problems Solving and Programming Skills**
Course Type	**:**	**OE**

Course Learning Objectives

1. This course introduces the concept of Data Structures used in various computer science applications
2. The students are introduced to understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using C programming language.

Course Content

Unit 1: Pointers & File Handling:

Revision of Pointers and Dynamic Memory, Files and related operations.
Searching techniques: Linear and Binary, Sorting techniques: Selection, Bubble, Insertion, Merge sort, Quicksort.

Unit 2: Simple Data Structures

Arrays based Linear Data Structures: Array storage, sparse arrays; Transpose, addition, and multiplication of sparse matrices, Stacks and Queues and their applications, multiple stacks and queues in an array.

Unit 3: Linked Data Structures

Singly, Doubly & Circular Linked Lists; representation, operations and applications, linked stacks and queues. linked lists based polynomial addition.

Unit 4: Advanced Data Structures

Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, Binary tree traversal techniques, some more operations on binary trees, Heaps and heapsort.

Reference Books:

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.L. Kruse: Data Structures & Program Design in C, PHI.
3. D.F. Knuth: The art of Computer Programming Vol-1, Narosa Publications, 1985.
4. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum's Outlines Series, TMH, 2005.

Course Outcomes

1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithms and implement their applications.
2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
3. Understand the concept of trees and graphs, their implementation and applications.
4. Able to implement standard algorithms for searching and sorting.
5. Analyze efficiency of different algorithms using time and space complexity.

Course Code	**:**	**ITOE 303**
Course Title	**:**	**Machine Learning**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course Code)	**:**	**CSPC204, MAIC201**
Course Type	**:**	**OE**

Course Learning Objectives:
1. Be able to understand and apply supervised and unsupervised learning algorithms.
2. Understand the fundamental concepts in machine learning and popular machine learning algorithms.
3. Interpretation of data and results using statistics and analysis
4. Be able to solve real-life problems using machine learning algorithms with programming in Python.

Course Contents:

Unit-1 Introduction

Basic definitions, types of learning, hypothesis space and inductive bias, probability distributions and hypothesis testing using statistical tests, data pre-processing.

Unit-2 Regression Techniques and Analysis

Supervised learning using regression, types of regression, linear regression: simple and multiple, logistic regression, overview of other types of regression; model diagnostics and analysis using residual and outlier, handling of multi-collinearity.

Unit-3 Classification

Supervised learning using classification, types of classification, linear models: regression & SVM; non-linear models: decision tree induction, attribute selection measures: information gain, gain ration, Ginni index; tree pruning and scalability; Naive Bayes classifier; kNN; classification model evaluation metrics; improving accuracy using ensemble, random forest, bagging, boosting.

Unit-4 Clustering

Basic Clustering concepts, Standard k-Means clustering, finding optimal number of clusters, Modifications to the k-Means; hierarchical clustering,

Unit-5 Advances in Machine Learning

Gradient descent algorithm, bias-variance, cross-validation, recommender systems: association rules, collaborative filtering, user of similarity, overview of use of neural networks for learning

Reference Books:

1. M. Pradhan, U. Dinesh Kumar, Machine Learning using Python, Wiley India, Latest Edition.
2. J. Han, M. Kamber, J. Pei, Data Mining Concepts and Techniques, Latest Edition
3. Christopher Bishop. Pattern Recognition and Machine Learning. 2e.
4. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
5. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.

Course Outcome:

At the end of this course, a student will be able to:
1. Compare and differentiate among different machine learning techniques
2. Implement the studied regression and classification techniques and analyze the results
3. Apply the suitable supervised and unsupervised techniques and models for real-life problems, and predict the outcome
4. Explain the applicability of advanced concepts for recommender systems.
5. Hands-on problem solving with programming in Python.

Course Code	**:**	**ITOE 305**
Course Title	**:**	**Database Management Systems**
Number of Credits and L/T/P scheme	**:**	**3 & 3/0/0**
Prerequisites	**:**	
Course Category	**:**	**OE**

Course Learning Objectives:

1. To understand data models and database systems.
2. To understand relational database theory,
3. To learn database design and its normalization.
4. To understand concepts of transaction, its processing, and management.
5. To learn concurrency control techniques.
6. To develop practical skill in using MySQL DBMS.

Course Content:

Unit-1 (Introduction)

Data, Database, Database management system, Historical background from file systems to Database Systems, Data Models, Relational Data Model, ER model, Schemas and Instances, Database users, DBMS architecture.

Unit-2 (Relational Model and Query Language)

Theory of Relational Database, Key Integrity constraint, Relational Algebra, Relational Calculus: Domain Relational Calculus, Tuple Relational Calculus, SQL, queries writing in SQL.

Unit-3 (Database Design)

Normalization, Database Anomalies, Functional Dependencies, Candidate and Super Key, Non-loss Decomposition, Dependency Preservation, First, Second, Third Normal, BCNF, etc.

Unit-4 (Transaction Processing)

Transaction concept, a simple transaction model, states, ACID Properties, implementation of ACID properties, Serializability.

Unit-5 (Concurrency Control)

Need for Concurrency, Lock-based protocols, Deadlock, Starvation, deadlock handling, time-stamp based protocols, validation-based protocols.

Text Books:

1. A Silberschatz, H.F. Korth & S. Sudarshan: Data Base System Concepts, Mc Graw Hill, 4^{th}, 5^{th} or 6^{th} edition.
2. Elmasri &Navathe : Fundamentals of Database Systems, 5^{th}, 6^{th}, or 7^{th} edition Pearson.

Course Outcomes

1. To Design and Implement a small database for applications using MySQL DBMS.
2. To apply the concepts of normalization to database design.

Course Code	**:**	**ITOE 302**
Course Title	**:**	**Design and Analysis of Algorithms**
Number of Credits and L/T/P scheme	**:**	**3 & 3/0/0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**OE**

Course Learning Objectives:

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Content:

Unit 1: Introduction

Concept of Time and space complexity, analysis of algorithms, asymptotic notation, recurrence relations, design and analysis of D & C problems like quick sort etc, heap sort, priority queues, sorting in linear time, hashing, binary search trees.

Unit 2: Graph Algorithms

Graph representation & traversal (search), topological sort, strongly connected components, minimum spanning trees – Kruskal and Prim's, Single source shortest paths, relaxation, Dijkstra's algorithm, Bellman-Ford algorithm, single source shortest paths for directed acyclic graphs, all-pairs shortest path.

Unit 3: B-Trees and Dynamic programming

B-Trees: representation and operations; Elements of Dynamic Programming, structure and steps, Matrix-chain multiplication, longest common subsequence.

Unit 4: Greedy & Backtracking Approaches:

Greedy algorithms – Elements, activity-selection problem, Huffman codes, task scheduling problem, Knapsack Problem, Backtracking – Elements, 8 – Queens, Graph Coloring, Hamiltonian Cycles.

Text Books:

1. Cormen, Leiserson and Rivest: Introduction to Algorithms, 3/e, PHI.
2. Horowitz, Sahni, and Rajasekaran: Fundamentals of Computer Algorithms, Second Edition, Universities Press, Hyderabad.
3. Aho, Hopcroft, and Ullman: The Design and Analysis of Computer Algorithms, Addison Wesley.

Course Outcomes

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Code	**:**	**ITOE 304**
Course Title	**:**	**Operating System**
Number of Credits and L/T/P scheme	**:**	**3 & 3/ 0/ 0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**OE**

Course Learning Objectives

1. To understand the services and design of an operating system.
2. To understand the structure and organization of file system
3. To understand the process states and various concepts such as scheduling and synchronization related with it.
4. To understand different memory management approaches.
5. Students should be able to use system calls for managing processes, memory and file system.
6. students should understand the data structures and algorithms for implementation of OS.

Course Content

Unit-I: Computer system architecture and organization, Introduction and evolution of OS, Introduction to distributed OS, Real time systems and multimedia systems. OS structures: OS services, system calls and programs, OS design and implementation. Processes: Process concept, scheduling policies, algorithms, multilevel queuing, operations on process, Inter-process communication. Threads: multithreading models and threading issues. CPU scheduling: Criteria and algorithms, multiprocessor and thread scheduling.

Unit II: Process synchronization: critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.).

Deadlocks: modelling, resource allocation, characterization, prevention and avoidance, detection and recovery.

Unit III: Memory management: Swapping, contiguous memory allocation, paging, multilevel paging, segmentation, demand paging, page replacement algorithms, allocation of frames, thrashing, working set model. Input/Output: I/O system and services, device controllers and device drivers, disks, scheduling algorithms and management.

Unit IV: File system interface: access methods, access control, directory structures, file organization, file sharing and protection. system performance, protection and security, , OS design considerations for security, access control lists and OS support, internet and general network security. Operating system as service provider: Access control matrix, access control list, capability matrix, encryption and access permissions in Linux and Windows.

Reference Books:

1. A. Silberschatz, Peter B. Galvin and G. Gagne, "Operating System Concepts," (9th or newer edition), Wiley.
2. H. Brinch, "Operating System Principles," Prentice Hall of India.
3. Dhamdhere, "Systems programming & Operating systems," TataMcGrawHil
4. A. N. Habermann, "Introduction to Operating System Design," Galgotia publication, New Delhi.
5. A.S. Tanenbaum, "Modern Operating Systems," Prentice Hall of India.

Course Outcomes

At the end of the course student will be able to

1. Understand functions, structures and history of operating systems
2. Able to know the design issues associated with operating systems
3. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks
4. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms
5. Be familiar with various protection and security mechanisms
6. Be familiar with virtualization and operating system components

Course Code	**:**	**ITOE 306**
Course Title	**:**	**Object-Oriented Programming using Java**
Number of Credits and L/T/P scheme	**:**	**3 & 3 0 0**
Prerequisites	**:**	
Course Type	**:**	**OE**

Course Learning Objectives:

1. Building robust applications using Java's object-oriented features.
2. Understanding the usage of java class libraries.
3. Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Content:

Unit 1: The overview of Java's architecture and the architecture of the Java Virtual Machine (JVM).

Classes: Declaring Members (Fields and Methods), Instance Members, Static Members.

Objects: Class Instantiation, Reference Values, and References, Object Aliases. Basic Language Elements, Primitive Data Types, Variable Declarations, Initial Values for Variables, Class Declarations, Method Declarations, this reference, Method Overloading, Constructors, The Default Constructor and Constructors overloading. Arrays, Anonymous Arrays, Multidimensional Arrays, Variable Arity Methods, The main() Method, Program Arguments.

Unit 2: Packages: Defining Packages, Using Packages, Compiling Code into Packages, Running Code from Packages. Scope Rules, Accessibility Modifiers, Overview of other Modifiers for Members.Operators and Expressions**,** Overview of Control Flow Statements.

Exception Handling: The try Block, The catch Block, The finally Block, The throw Statement, The throws Clause, Checked and Unchecked Exceptions, Defining New Exceptions.

Unit 3: Object-Oriented Programming: Single Implementation Inheritance, Overriding Methods, Hiding Members, The Object Reference super, Chaining Constructors Using this() and super()

Interfaces: Defining Interfaces, Abstract Method Declarations, Implementing Interfaces, Extending Interfaces, Interface References, Constants in Interfaces, Polymorphism and Dynamic Method Lookup.

Fundamental Classes: Overview of the java.lang Package, The Object Class, The Wrapper Classes, The String Class, The StringBuilder and the StringBuffer Classes.

Unit 4: Multithreading: Overview of Threads, the Main Thread, Thread Creation, Synchronization, Thread Transitions. Basics of Event Handling, Graphics Programming using Java Fx.

Reference Books:

1. Y. Daniel Liang: Introduction to Java Programming, Comprehensive Version, Pearson Education, 12th Ed.
2. Bruce Eckel, Thinking In Java, Pearson Education, 4th Ed., 2006.
3. Dietel & Deitel, Java How to Program, Pearson Education, 10th Ed., 2015.
4. Kathy Sierra & Bert Bates, Head First Java, O'REILLY, 2nd Ed., 2005.
5. Cay s. Horstmann & Gary Cornell, Core Java. Volume I, Fundamentals, Sun Microsystems Press, 8th Ed., 2008.

Course outcomes:

1. Write Java programs that solve simple business problems.
2. Create Java applications that are robust and multithreaded.
3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Code	**:**	**ITOE 401**
Course Title	**:**	**Advanced Data Structures and Algorithms**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	**Design and Analysis of Algorithms**
Course Category	**:**	**PC**

Course Learning Objectives:
1. To develop the understanding of advanced data structures.
2. To develop the understanding of algorithmic design paradigms.

Course Content
Unit-1 Complexity: Algorithms, Revision of Performance analysis- time complexity and space complexity using asymptotic notation, Amortized complexity

Unit-2 Advanced Data Structures: Threaded binary trees, Binomial heaps, Fibonacci heaps, Balanced trees- AVL, Multi-way search trees, red-black trees, splay trees, segment trees; Suffix tries

Unit-3. String Matching: String-matching algorithm, Rabin-Karp algorithm, String matching with automata, Knuth-Morris-Pratt algorithm.

Unit-4 NP – Hard & NP – Complete problems: Concepts of NP-hard & NP complete problems, Clique Decision problem, Node Cover decision problem, Travelling Salesperson decision problem.

Suggested Books:
1. E Horowitz and S. Sahni, Fundamentals of Data Structures, Universities Press.
2. Cormen, Leiserson and Rivest, Introduction to Algorithms, 3/e, PHI.
3. Horowitz, Ellis and Sahni, Fundamentals of Computer Algorithms, Universities Press.

Reference Books:
1. Oded Goldreich. P, NP, and NP-Completeness: The Basics of Computational Complexity, Cambridge
   University Press, 2010.
2. Anne Benoit, Yves Robert, Frédéric Vivien. A Guide to Algorithm Design: Paradigms, Methods, and
   Complexity Analysis, Taylor & Francis, 2013.
3. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, Data Structures and Algorithms, Addison Wesley, Reading
   Massachusetts, USA, 1983.
4. Donald Knuth. The Art of Computer Programming: Fundamental Algorithms, Third Edition. AddisonWesley, 1997.
5. Donald Knuth. The Art of Computer Programming Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0.

Course Outcome:
1. Compare, contrast, and apply key data structures: trees, heaps, string algorithms etc.
2. Compare, contrast, and apply algorithmic tradeoffs: time vs. space, deterministic vs. randomized, and exact vs. approximate.
3. Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

Course Code	**:**	**ITOE 403**
Course Title	**:**	**Computer Networks**
Number of Credits and L/T/P scheme	**:**	**3 & 3 0 0**
Prerequisites (Course code)	**:**	
Course Category	**:**	**OE**

Course Learning Objectives:

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Content:

Unit 1: Introduction

History and development of computer networks, Network Functions, Network Topology, Network Services, Switching Approaches, Transmission media and systems, OSI Reference Model, Overview of TCP/IP architecture,

Unit 2: Physical, MAC and Data Link Layer

Different types of transmission media, errors in transmission: attenuation, noise, multiplexing and signalling techniques, Encoding (NRZ, NRZI, Manchester, 4B/5B, etc). Aloha, CSMA, CSMA/CD, CSMA/CA protocols, Ethernet, including Gigabit Ethernet and WiFi (802.11), Token Ring, Bluetooth and Zigbee. Error detection, Sliding Window, Stop and Wait protocols, ARQ, Sliding Window Protocols, HDLC, PPP protocols.

Unit 3: Network layer:

Internet Protocol, IPv6, ARP, DHCP, ICMP, Routing algorithms: Distance vector, Link state, Metrics, Inter-domain routing. Subnetting, Classless addressing, Network Address Translation, IPv4 to IPv6 Translation strategies, Packet network topology, Datagrams and Virtual Circuits, Structure of Switch/Router, Connectionless and Virtual Circuit packet Switching, Traffic management and QoS – FIFO, Priority Queues, Fair Queuing, MPLS.

Unit 4: Transport and Application layer:

UDP, TCP. Connection establishment and termination, sliding window revisited, flow and congestion control, timers, retransmission, TCP extensions, etc. Client Server Model, HTTP, DNS, SMTP, FTP, and TFTP.

Text Books:

1. Behrouz A. Forouzan: Data Communications and Networks, Fifth Edition, McGraw Hill, 2017.
2. Leon Garcia and IndraWidjaja: Communication Networks – Fundamental Concepts and Key Architectures, Second Edition TMH.
3. A.S. Tanenbaum: Computer Networks, Sixth Edition, Pearson Education, 2022.
4. William Stallings: Data and Computer Communications 10/e, Pearson Education.

Course Outcomes:

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

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.

3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Code	**:**	**ITOE 405**
Course Title	**:**	**Deep Learning**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**Machine Learning**
Course Category	**:**	**OE**

Course Learning Objectives

The objective Deep Learning course is to cover the fundamentals of neural networks as well as some advanced topics such as recurrent neural networks, long short-term memory cells and convolutional neural networks. This course also covers the basic applications of the deep learning.

Course Content

Unit-1: History of Deep Learning, Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm.

Unit-2: Feedforward Networks: Multilayer Perceptron, Gradient Descent, Backpropagation, Empirical Risk Minimization, regularization, autoencoders. Deep Neural Networks: Difficulty of training deep neural networks, Greedy layerwise training.

Unit-3: Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization), Recurrent Neural Networks: Back propagation through time, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs.

Unit-4: Convolutional Neural Networks: LeNet, AlexNet, ZF-Net, VGGNet,Visualizing Convolutional Neural Networks, Guided Backpropagation, Deep Dream, Deep Art, Fooling Convolutional Neural Networks, Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

Unit-5: Recent Trends: Variational Autoencoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning, Applications: Image based classification, Vision, NLP.

Text Books:

1. Ian J. Goodfellow, Bengio Yoshua, and Aaron Courville. "Deep Learning." An MIT Press book in preparation, 2016.
2. Charu C. Aggarwal “Neural Networks and Deep Learning: A Textbook” Springer Nature, 2018.
3. Seth Weidman “Deep Learning from Scratch: Building with Python from First Principles” O’REILLY, 2019
4. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola “Dive into Deep Learning: Tools for Engagement” CORWIN, 2019

Reference Books:

1. Ian J. Goodfellow, Bengio Yoshua, and Aaron Courville. "Deep Learning." An MIT Press book in preparation, 2016.

Course outcomes

At the end of the course, student will have understanding of following concepts/ theories/ application:

1. Fundamental Understanding of Neural network.
2. Basic and Advance concept of Deep Learning.
3. Application of Deep Learning.

Course Code	**:**	**ITOE 402**
Course Title	**:**	**Artificial Intelligence**
Number of Credits and L/T/P scheme	**:**	**3 and 3/0/0**
Prerequisites (Course code)	**:**	**Data Structures**
Course Category	**:**	**OE**

Course Learning Objectives

1. It aims to give an introduction to the fundamentals of artificial intelligence.
2. The course is designed to develop a basic understanding of problem solving, knowledge representation, reasoning and learning methods of AI.
3. To learn about major topics of artificial intelligence, their fundamental differences and applicability.

Course Content

Unit 1 **Introduction:** Introduction **to** AI, Foundations and history, problems, Nature and Structure of Intelligent Agents.

Unit 2 **Problem-Solving:** Solving Problems by Searching, Heuristic Search techniques, Metaheuristic Search, Beyond Classical Search, Adversarial Search, Constraint Satisfaction Problems.

Unit 3 **Knowledge, Reasoning, and Planning:** Logical Agents, First-Order Logic, Inference in First-Order Logic, Classical Planning, Knowledge Representation, Uncertain Knowledge and Reasoning.

Unit 4 **Learning and Agents:** Learning from Examples, Knowledge in Learning, Definition of agents, Agent architectures, multi-agent systems.

Text Books:

1. Russell, S. J. and Norvig, P., Artificial Intelligence: A Modern Approach, Pearson Education, 3rd Edition, 2014.
2. Elaine Rich, Kevin Knight and Shivashankar B Nair, Artificial Intelligence, Tata McGraw Hill Education Private Limited, 3rd Edition, 2009.

Reference Books:

1. Patterson, D. W., Introduction to Artificial Intelligence and Expert Systems, Prentice Hall of India, 2012.

Course Outcomes

1. Understand basic concepts of AI and students will be able to apply those concepts in the field of Artificial Intelligence.
2. Identify problems where artificial intelligence techniques are applicable.
3. Apply various AI search algorithms, knowledge representation, reasoning to real-world problems.
4. Design of systems that act intelligently and learn from experience.

Course Code	**:**	**ITOE 404**
Course Title	**:**	**Information Security**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**Computer Networks**
Course Category	**:**	**OE**

Course Learning Objectives

1. To introduce the concept of information security, and number theory.
2. To make students familiar with different aspects of access control, system security and its applications.
3. To make students familiar with various cryptographic algorithms.
4. To design security solutions for different operation systems and Networks

Course Content

Unit 1. Introduction to Information Security: Meaning of Information security, privacy, vulnerability, threat, attack, CIA Triad, CNSS Security Model, Components of an Information System, OSI Security Architecture, Model for Network Security, Network Access Security Model, Access Control

Unit 2 Introduction to Number theory: Divisibility, Division Algorithm, Euclidean Algorithm, Extended Euclidean Algorithm, Fast exponentiation Algorithm; Modular Arithmetic, Congruences, Chinese Remainder Theorem, Algebraic structures, Galois Fields

Unit 3. Introduction to Cryptography: Private Key Cryptography: Traditional Symmetric-key ciphers, Modern Symmetric Key ciphers, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Block Cipher modes of Operation; Public Key Cryptography: Trapdoor one-way Function, RSA, Diffie-Hellman Key exchange, Cryptographic Hash Function, Digital Signature

Unit 4. System and Operating Security: Overview, Security policy, File Protection Mechanisms, User Authentication, Security models, Trusted Operating System Design, Host based Intrusion detection system, Network based Intrusion detection system, Host based Intrusion prevention system, Network based Intrusion prevention system.

Unit 5. Network Security: Network security Concepts, Network threats, Security controls, Firewalls, Protecting Programs and Data, Secure Program, sniffing, spoofing, Non Malicious Program errors, Malware, Viruses and other malicious code, Control against Program, Honeypot, SDN Security, IoT Security

Text Books:

1. Charles P. Pfleeger, Share Lawrence Pfleeger, Security in Computing, Pearson Education, 2/e.
2. William Stallings, Cryptography and Network Security, PHI, 7/e
3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer 2006.
4. B.A. Forouzan, Cryptography and Network Security, McGraw-Hill, 3/e

Reference Books:

1. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006)
2. Charlie Kaufman, Perlman & S Peeciner, Network Security, Pearson Education, 2/e.

Course outcomes:

On successful completion of this, course students will:

1. Able to understand the information security and number theory concepts.
2. Able to analyses different aspects of access control, system security and its applications.
3. Able to understand and analyse various cryptographic algorithms.
4. Able to design and develop security solutions for different operation systems and networks

Course Code	**:**	**ITOE 406**

Course Title	**:**	**Theory of Computation**
Number of Credits and L/T/P scheme	**:**	**3 and 3-0-0**
Prerequisites (Course code)	**:**	**MAIC201**
Course Category	**:**	**OE**

Course Learning Objectives

The objective of this course is to give students a clear understanding of concepts formal languages, grammars and formal methods of computation.

Course Content

Unit-1 Introduction to Finite State Machine, Moore and Mealy FSMs, Equivalence, Minimal Finite Automata, unions, Intersections and complement of FA, NFA, NFA with null transitions, Equivalence.

Unit-2 Regular Languages, Regular expressions, memory required to recognize a language, Distinguishing one string from another, Criterion for Regularity, The pumping lemma, decision problems, Equivalence of FA and regular expressions, Arden's Theorem, Non-determinism and Kleen's Theorem.

Unit-3 Context – Free Grammars, Definition of CFG, example of familiar languages, unions, concatenations and closures of CFLs, Derivation Tree, Ambiguity, unambiguous CFG for algebraic expressions, CNF, GNF, removal anomalies in grammars, Push down automata, definition, deterministic PDA, PDA to CFG and Vice Versa, Context Free and Non Context Free Languages, Pumping lemma for CFG, Intersection and complements of CFL.

Unit-4 Turing Machine Definition, Turing Machining as Language acceptors, combining TM, computing Partial Function with TM. Recursively Enumerable and Recursive Languages, Regular Grammars, context Sensitive grammars, Chomsky Hierarchy. Concept of insolvability & reducibility, Halting Problem, Post correspondence Problem, Rice theorem, P and NP. Polynomial-Time reductions and NP-Completeness.

Reference Books:

1. John C. Martin: Introduction to Languages and the Theory of Computation, MGH.
2. Lewis & Papadimitriou: Elements of the Theory of Computation, PHI.
3. Daniel I.A. Cohen: Introduction to Computer Theory: John Wiley.
4. J.E. Hopcroft and J.D. Ullman: Introduction to Automata Theory Languages and Computation, Narosa.

Course outcomes:

1. To understand the foundations finite automata (deterministic or non-deterministic), FSM and master regular languages.
2. Be exposed to types of languages (type-0, type-1, type-2, type-3), context-free languages (type-2) and using pushdown automata (PDA) as its acceptor.
3. To design the Turing machines for various mathematical function, string function etc
4. To demonstrate concept of decidability, unsolvability and recursive languages.