Department of Information Technology

Third Year Syllabus

To be followed from Academic Year 2020-21

(L-T-P) indicates L-Lecture, T-Tutorial and P-Practical

Program Educational Objectives (PEOs):

PEO1	To provide students strong foundation in mathematics and engineering fundamentals to have carrier in various fields of IT such as Networks and Security, Data Analysis and Management, Web Development etc.
PEO2	To imbibe in them professional and ethical responsibilities towards their profession, society and the environment as well as the respect for diversity.
PEO3	To enable graduates apply necessary techniques, Software and Hardware tools to foster innovation, invention and entrepreneurship.
PEO4	To help students acquire effective oral and written communication and lifelong learning skills to have productive careers in IT industries.
PEO5	To provide opportunity to the students to work effectively as individuals or in teams demonstrating their skills in solving IT related problems.

Program Outcomes (POs):

PO1 (a)	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and specialization to solve complex engineering problems.
PO2 (b)	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using principles of mathematics, natural and engineering sciences.
PO3 (c)	Design/development of solutions: Design and develop solutions by considering the public health and safety, cultural, societal, and environmental considerations to complex multidisciplinary engineering problems.
PO4 (d)	Conduct investigations of complex problems: Use research-based knowledge and methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5 (e)	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 (f)	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 (g)	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 (h)	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9 (i)	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10 (j)	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 (k)	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12 (l)	Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Objectives (PSOs):

PSO1	Foundation of mathematical concepts: To apply mathematical methodologies to crack the real-world problems using appropriate mathematical analysis, data structure and efficient computer algorithms.
PSO2	Knowledge of recent trends: To provide effective and efficient knowledge of recent technologies such as Web Technologies, Data Science, Machine Learning, Artificial Intelligence, Cyber Security, Internet of Things, etc.
PSO3	Project based learning: To provide platform to the students to develop a new and innovative multidisciplinary software development to cater industry needs.

Table of Correlation:

PO/PSO➡ ⬇ PEO	a	b	c	d	e	f	g	h	i	j	k	l	PSO 1	PSO 2	PSO 3
I	✓	✓		✓			✓			✓		✓		✓	
II			✓		✓	✓	✓		✓	✓	✓		✓		✓
III		✓		✓		✓				✓		✓	✓		✓
IV					✓	✓		✓		✓		✓		✓	✓
V	✓	✓	✓	✓	✓				✓	✓				✓	✓

Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded

DEPARTMENT OF INFORMATION TECHNOLOGY

Curriculum Structure of T.Y. B. Tech.
(With effective from 2020-2021)

Semester I						
Course Code	**Name of the course**	**L**	**T**	**P**	**Credits** **Th**	**Pr**
PCC-IT301	Theory of Computation	03	01	--	04	--
PCC-IT302	Software Engineering	03	--	02	03	01
PCC-IT303	Operating Systems	03	--	02	03	01
PCC-IT304	Data Base Management System	03	--	02	03	01
PCC-IT305	Python Programming	03	--	02	03	01
SEM-IT306	Seminar - I	--	--	02	--	01
	Total	**15**	**01**	**10**	**21**	
Semester II						
Course Code	**Name of the course**	**L**	**T**	**P**	**Credits** **Th**	**Pr**
PCC-IT307	Machine Learning	03	--	02	03	01
PCC-IT308	Design and Analysis of Algorithms	03	--	02	03	01
PCC-IT309	Computer Networks	03	--	02	03	01
PEC-IT3**	Elective - II	03	--	--	03	--
PCC-IT313	Compiler Design	03	--	02	03	01
LAB-IT314	Computer Laboratory – III: Full Stack Development*	--	--	02	--	01
	Total	**15**	**00**	**10**	**20**	

L – No. of Lecture Hours/week, T – No. of Tutorial Hours/week, P – No. of Practical Hours/week

* In Computer Laboratory-III along with Lab Practical, Students are required to complete Minimum 40 hrs./ 8 week Course from Swayam /MOOC/Udemy/Coursera, etc. The course can be a programming course or one related to the syllabus.

Elective – II	
Course Code	**Name of the course**
PEC-IT310	Cloud Computing
PEC-IT311	Big Data Analytics
PEC-IT312	Blockchain Technology

PCC-IT301: THEORY OF COMPUTATION

(Total Credits: 4, Lectures/Week: 3, Tutorial/Week: 1)

Course Objectives:

1. Master regular languages and finite automata.
2. Master context-free languages, push-down automata, and Turing recognizable languages.
3. Ability to describe and transform regular expressions andgrammars.
4. Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computerscience.
5. Be exposed to a broad overview of the theoretical foundations of computer science.

Course Contents:

1. Basic Concept: [6 Hrs]

Representation for Formal Languages: Functions, Relations. Set theory - Definition, Finite and infinite set, Countability of a set, Cardinality of a set, Closure of a set, Closure properties of relations. Symbols, Strings, Language. Finite State Machine.

Regular Expressions and Languages: Recursive definition of regular expression, regular set, identities of regular expressions, regular expressions, examples, identity rules, regular sets properties, pumping lemma for regular sets.

2. Finite Automata: [8 Hrs]

Finite automata (FA) - Definition of Deterministic Finite Automaton (DFA) and Non - deterministic Finite Automaton (NFA), Language acceptance by FA, Conversion of NFA to DFA, Minimization of a DFA, Conversion of NFA with Є to NFA without Є, Construction of RE equivalent to FA using Arden'sTheorem.

FA with output: Moore and Mealy machines - Definition, models, inter conversion between Moore and Mealy machine.

3. Grammars: [9Hrs]

Context Free Grammar (CFG) - Definition, Production rules, Derivation trees, Ambiguous grammar. Reduced form grammar-removal of unit productions, useless symbols, null production. Simplification of CFG, Chomsky hierarchy, Context Free Language (CFL).

Normal Forms - Chomsky Normal Form (CNF), Griebach Normal Form (GNF).

Regular grammar - definition, Left linear & right linear Regular Grammar. Inter conversion between left linear and right linear regular grammar. Regular grammar and Finite Automata.

4. Push Down Automata: [7Hrs]

Formal definition, Pushdown Automata (PDA), Deterministic Push Down Automata (DPDA), Non-deterministic Push Down Automata (NPDA), Inter conversion between PDA and CFG, closure properties of CFLs.

5. TuringMachine: [6Hrs]

Definition and Construction of Turing Machines. Languages of TM. Types of TM. Halting Problem, Church's Turing Hypothesis, recursively enumerable sets, decidability, undecidability Post Correspondence Problem.

6. Applications: [4Hrs]

Application of RE and FA - Lexical analyzer, Text editor and searching using RE. Application of CFG - syntax analysis, Language definition.

Textbooks:

1. “Theory Of Computer Science “ By K.L.P. Mishra &Chandrasekharan
2. "Introduction to Automata Theory Languages And Computation” By John E. Hopcoroft, Rajeev Motwani, Jeffrey D-Ullman,LPE

Reference Books:

1. Daniel I.A. Cohen, "Introduction toAutomata Theory Languages and Computations”, Pearson Education Asia, SecondEdition.
2. E V Krishnamurthy, “Introduction to Theory of Computer Science”, EWPSecond 2nd Edition.

Course Outcomes:

CO1: Understand the essence of computing through simple models of computational devices

CO2: Be able to apply these models in practice to solving problems in diverse areas such as string searching, pattern matching, cryptography and language design.

CO3: Understand the limitations of computing, the relative power of formal languages and the inherent complexity of many computational problems of practicalimportance.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	1	2		2	3	1	2		1		2	3	2	2	3
CO2	3	1	3	2	1			1	1		3	3	3	1	2
CO3	2		3	1		2	2	3		3			3	1	1

PCC-IT302: SOFTWARE ENGINEERING

(Total Credits: 4, Lectures/Week: 3, Practical/Week: 2)

Course Objectives:

1. To study the theories, processes, methods, and techniques of software engineering.
2. To help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain
3. To train the students on Software Engineering approaches used inIndustry
4. To understand the basic principles of project management and application of these to real world project.

Course Contents:

Part I: Software Engineering

1. Introduction to Software Engineering: [3Hrs]

Evolving Role of software, Software, The changing nature of software, Software Myths.

2. A generic View of process: [4Hrs]

Software Engineering – Layered Technology, A process framework, CMMI, Personal and Team software process, process technology, product and process.

3. Process Models: [4Hrs]

Prescriptive Models, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process models, The unified process

4. An agile view of process: [2Hrs]

What is agility, What is an agile process?, Extreme programming (XP), scrum.

5. Requirements Engineering [4Hrs]

A bridge to design and construction, Requirements Engineering Tasks, Initiating the requirements engineering process, Eliciting requirements, Developing use cases, Building the analysis model, Negotiating requirements, validating requirements.

6. Building the analysis model [4Hrs]

Requirements Analysis, Analysis modelling approaches, data modelling concepts, object-oriented analysis, scenario-based modelling, class-based modelling.

7. Design Engineering [4Hrs]

Design within the context of software engineering, Design process and design quality, Design Concepts, The design model.

8. Testing [4Hrs]

A strategic approach to software engineering, Unit testing, integration testing, regression testing, smoke testing, validation testing, System testing, the art of debugging.

Part II: Software Project Management

9. Project Management [4 Hrs]

The Management spectrum, The people, product, process, project.

10. Estimation **[4Hrs]**

Resource estimation, project estimation, Empirical estimation model, make/buy decision.

11. Project Scheduling **[4Hrs]**

Basic Concepts, Project scheduling, defining task set for software project, defining a task network, Scheduling, Earned value analysis.

Term work:

Students will have to perform at least 10-15 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical during the semester, marks should be given at the end of the semester.

Textbooks:

1. Roger S. Pressman, "*Software Engineering: A practitioners approach",* 6thEd.TMH.
2. Ian Sommerville, "Software Engineering", 8thEd.Addison-Wesley.

Reference Books:

1. MallR.,"Fundamentals of Software Engineering",Second Edition,Prentice Hall of India, 2004, ISBN 81-203-2445-52.
2. Vliet H., "Software Engineering Principles and Practices", Second Edition, John Wiley And Sons, ISBN 9971-51-357-9
3. Sommerville "Software Engineering" 8th Edition, Person Education
4. Behfarooz A:, Hudson F., "Software Engineering Fundamentals", Oxford University Press, 2002, ISBN0-19-510539-7
5. "An Integrated Approach to Software Engineering", Third Edition, Pankaj Jalote
6. Fredrick P Brooks,"The Mythical Man-Month: Essays on Software Engineering", Addison-Wesley Professional

***Course Outcomes*:**

CO1: Ability to analyze and specify software requirements.
CO2: Ability to apply software engineering principles and techniques to develop large-scale software systems.
CO3: Ability to plan and work effectively in a team.
CO4: Understanding and develop the necessary skills to handle software projects in a principled way

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	3	3	3	2	1	3	3	3	2	1	3	3
CO2	2	3	3	2	2	3	1	2	3	2	3	3	2	3	3
CO3	1	2	3	2	3	2	1	2	3	3	3	2	1	3	3
CO4	1	2	1	2	3	1		2	1	3	2	2	1	3	2

PCC-IT303: OPERATING SYSTEMS

(Total Credits:4, Lectures/Week:3, Practical/Week:2)

Course Objectives:

1. Understanding & Remembering Basic Concepts and Problems in Operating Systems.
2. Applying Solution for solving problems in Operating System for demonstration purpose.
3. Analyzing different aspects of OS like Process Management, Memory Management, File Management and Disk Management.
4. Evaluating performance of different management aspects using mathematical problems as well as programs.
5. To create applications using different terminal Commands, C programs and Shell Scripting related to OS.

Course Contents:

1. Introduction: [06hrs]

ArchitectureofOS(Ex.Monolithic,Microkernel,LayeredandExokernel),Operatingsystemobjectives and functions, Interaction of O. S. and hardware architecture, Evolution of operating systems, Batch, Multiprogramming,Multitasking,Multiuser,parallel,distributed&real–timeO.S.,Systemcalls,O.S. Shell, Linux Shell commands, Shell programming. Examples of O. S.: Ubuntu, Linux, MS-Windows & Mobile Phone O.S. (Android, Windows Phone, iOS and etc)

2. Process Management, Process Communication and Synchronization: [14 hrs]

Process Management : Process, Process description, Process states, Process control, Threads, Processes and Threads, Uniprocessor Scheduling: Types of scheduling, CPU scheduling algorithms: FCFS, SJF, Priority, Round Robin, UNIX Multi-level feedback queue scheduling, Thread Scheduling, Multiprocessor Scheduling concept, Real Time Scheduling concept.

Process Communication and Synchronization:

(a) Concurrency: Principles of Concurrency, Mutual Exclusion H/W Support, software approaches, Semaphores and Mutex, Message Passing, Monitors, Classical Problems of Synchronization: Readers-Writers Problem, Producer Consumer Problem, Dining Philosopher problem.

(b) Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategies.

3. Memory Management: [08 hrs]

MMU, Memory Management requirements, Memory partitioning: Fixed & Dynamic partitioning, Memory allocation Strategies (First Fit, Best Fit, Worst Fit, Next Fit), Fragmentation, Paging, Segmentation, Virtual Memory, Demand paging, Page Replacement Policies (FIFO, LRU, MRU and Optimal), Thrashing.

4. I/O and File Management: [08hrs]

I/O Management and Disk Scheduling: I/O Devices, Organization of I/O functions, Operating System Design issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, C-SCAN, SSTF), Disk Caches.

File Management: Overview, File Organization and access, File Directories, File Sharing, Security issues, Record Blocking, Secondary Storage Management. Comparative study of Windows and UNIX file system.

5. Protections and Security [08 hrs]

Computer security & protection: Security Threats, Attacks and assets, Intruders, Malicious software.

Protection: Protection Policy and Mechanisms, Authentications: Internal Access Authorizations,

Implementations.

Term work:

Students will have to perform at least 10-15 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical during the semester, marks should be given at the end of thesemester.

Reference Books:

1. Operating Systems, Andrew S. Tenenbaum, PHI
2. Silberschatz, Galvin, Gagne: Operating System Concepts, 7th Edition, Wiley
3. Operating Systems: Internals and Design Principles, William Stalling, PHI.

***Course Outcomes*:**

CO1: Understand functional architecture of an operating system

CO2: Develop algorithms for subsystem components

CO:3 Design device drivers and multithreading libraries for a tiny OS

CO4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1		2	1		1				2	1	2	3		2	
CO2	3	2	3		2		1	2	2		2	3	1	3	2
CO3		2	3	1	2	1		2	2	1	3			3	2
CO4	2	1	2		2				1	2		3		2	1

PCC-IT304: DATA BASE MANAGEMENT

(Total Credits:4, Lectures/Week:3, Practical/Week:2)

Course Objectives:

1 Demonstrate an understanding of the relational datamodel.
2 Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS. Formulate, using relational algebra, solutions to a broad range of query problems.
3 Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of adatabase.
4 Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.

Course Contents:

1. Introduction: [3Hrs]

Purpose of database systems, view of data, data models, database languages, transaction management, storage management, database administrator, database users, overall system structure.

2. Entity-Relationship Model: [4 Hrs]

Basic concepts, design issues, mapping constraints, keys, E-R diagram, weak entity sets, extended E-R features, design of an E-R database schema, reduction of an E-R schema to tables.

3. Relational Model: [5Hrs]

Structure of relational databases, the relational algebra, the tuple relational calculus, the domain relational calculus, extended relational algebra operations, modifications of the database, views. Study of SQL, embedded SQL, and other SQL features.

4. Relational Database Design: [6 Hrs]

Integrity Constraints, Domain constraints, referential integrity, assertions, triggers, unconditional dependencies. Pitfalls in relational database design, decomposition, normalization using functional dependencies, multi valued dependencies, join dependencies, domain key normal form, alternative approaches to database design.

5. Storage and File Structure: [4 Hrs]

Magnetic disks, RAID, Tertiary storage, File organization, organization of records in files, data dictionary storage, storage structures for object oriented databases.

6. Indexing and Hashing: [5 Hrs]

Basic concepts, ordered indices, B+ tree index files, B tree index files, static hashing, dynamic hashing, comparison of ordered indexing and hashing, index definition.

7. Query Processing: [4 Hrs]

Overviews,costestimation,measuresofquerycost,selectionoperation,sorting,joinoperationandjoin strategies, evaluation ofexpressions.

8. Transaction and Concurrency Control: [5 Hrs]

Transaction concept, transaction state, atomicity and durability, concurrent executions, serializability, recoverability, isolation, transaction definition in SQL, testing for serializability. Concurrency control, lock based protocols, time stamp based protocols, validation based protocols, multiple granularity, multi version schemes, deadlock handling, insert and delete operations, concurrency in index structures.

9. Recovery System: [4Hrs]

Failure classification, storage structure, recovery and atomicity. Log based recovery, shadow paging,

recovery with concurrent transactions, buffer management.

Term work: Students will have to perform at least 10 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical in the semester marks should be given at the end of the semester.

Textbooks:

1 Silberschatz, Korth and Sudarshan, "Database system Concepts", McGraw Hill.

Reference Books:

1. Aho Ullman, "Principles of DatabaseManagement"
2. G. Everest, "Database Management", McGrawHill
3. C.J. Date, "An Introduction to databaseConcepts".

Course Outcomes:

CO1: Understanding of database management system and its architecture

CO2: Define the terminology, features, and characteristics embodied in database systems.

CO3: Demonstrate an understanding of the relational model and entity relationship model.

CO4: Formulate, using SQL, solutions to a broad range of query and data update problems.

CO5: Be familiar with the relational database theory, design and be able to write relational algebra expressions for queries.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1		2	2	1	2			2		2	3	3		3	3
CO2		2	3		3		1	1			3	3	2	3	3
CO3		3	3	1	3				3	3	3	3		3	2
CO4	3	2	3	3	3	1		2	2		3	3	2	3	3
CO5	3	2	2	3	2	1		1	2		3	3	3	1	

PCC-IT305: Python Programming

(Total Credits: 4, Lectures/Week: 3, Practical/Week: 2)

Course Objectives:

1. Master the principles of object-oriented programming and the interplay of algorithms and data structures in well-written modular code;
2. Solve problems requiring the writing of well-documented programs in the Python language, including use of the logical constructs of that language;
3. Demonstrate significant experience with the Python program development environment.

Course Contents:

UNIT – I:
Introduction: History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input/output, Indentation. Types - Integers, Strings, Booleans;

UNIT – II:
Operators and Expressions: Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations
Data Structures Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

UNIT – III:
Control Flow - if, if-else-else, for, while, break, continue, pass
Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments,
Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

UNIT – IV:
Modules: Creating modules, import statement, from import statement, name spacing, Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages
Error and Exceptions: Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions
Object Oriented Programming OOP in Python: Classes, 'self-variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data Hiding,

UNIT – V:
Brief Tour of the Standard Library - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multi-Threading, GUI Programming, Turtle Graphics

Textbooks:
1. Think Python: How to Think Like a Computer Scientist by Allen B. Downey. Green Tea Press. (available online under the GNU Free Documentation License) Ian Sommerville, “Software Engineering”, 8th Ed. Addison-Wesley.

Reference Books:

1. “Beginning Python®: Using Python 2.6 and Python 3.1” by James Payne. 2010. Wrox.
2. “Python Programming: An Introduction to Computer Science” 2ndEd. by John M. Zelle. 2010. Franklin, Beedle and Associates Inc.
3. “Non-Programmer's Tutorial for Python 3”. (Wikibooks)
4. “A Byte of Python” by Swaroop C.H. (availabe free online)
5. Python Programming: A Modern Approach, Vamsi Kurama, Pearson 2.
6. Learning Python, Mark Lutz, Orielly

***Course Outcomes*:**

CO1: Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
CO2: Express different Decision Making statements and Functions
CO3: Interpret Object oriented programming in Python
CO4: To design GUI Applications in Python and evaluate different database operations

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	1		3			1	2		3	3	2	3	3
CO2		3			3	1				2	2	2		2	2
CO3	1	3		1	3	1		1		2	2	3		3	3
CO4		2	3		3	2	1	1	2	3	3	3	2	3	3

SEM-IT306: SEMINAR

(Total Credits: 1, Practical/Week: 2)

Seminars are included with an aim to help students improve their presentation skills and to help them gainconfidence.Studentsarerequiredtodelivertwoseminarsthroughthesemester(individually).Out of the two seminars, one is to be delivered on technical topic which will highlight some recent technologyorinvention.Thesecondseminarshouldbeonsomenon-technicaltopic.Theyhavetowrite report of both the seminars, spiral bound them and submit to the concern faculty in -charge.

SEMESTER -VI

PCC-IT307: Machine Learning

(Total Credits: 4, Lectures/Week: 3, Practical/Week:2)

Course Objectives:

1. To understand the principles of Machine Learning and perform classification and prediction ofdata
2. To introduce students to the basic concepts and techniques of Machine Learning.
3. To develop skills of using recent machine learning software for solving practical problems.

Course Contents:

Unit-1 **[8Hrs]**

Introduction – History of Machine Learning, Programs vs learning algorithms, Machine Learning definition, Components of a learning, Different Types of Learning

Supervised Learning (Regression/Classification): Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Nave Bayes, Linear models: Linear Regression, Logistic Regression, Generalized Linear Mode, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary Classification: Multi-class/Structured Outputs, Ranking

Unit-2 **[6Hrs]**

Unsupervised Learning: Clustering: K-means/Kernel K-means, Dimensionality Reduction: PCA and kernel PCA, Matrix Factorization and Matrix Completion, Generative Models (mixture models and latent factor models).

Unit-3 **[6Hrs]**

Evaluating Machine Learning algorithms and Model Selection, Introduction to StatisticalLearning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)

Unit-4 **[6Hrs]**

Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning andFeature Representation Learning

Unit-5 **[8Hrs]**

Scalable Machine Learning (Online and Distributed Learning)- A selection from some otheradvanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference.

Unit 6 **[6 Hrs]**

Neural Networks: Representation, Recent trends in various learning techniques of machine learning and classification methodsfor IOT applications. Various models for IOT applications.

Reference Books:

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
2. Machine Learning For Dummies by John Mueller and Luca Massaron
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online).
4. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007

***Course Outcomes*:**

CO1: Understanding of the Machine Learning principles and techniques for real time applications.

CO2: Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning

CO3: Understand the concept behind neural networks for learning non-linear functions.

CO4: Be able to design and implement various machine learning algorithms in a range of real-world applications.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	2		2		2		1	2	3	3	3	3	2
CO2	3	3	3	1	2		1	1			3	3	3	3	
CO3	2	2	3		2	1					2	3	2	3	
CO4	3	2	3	3	3	3			3	2	3	3	3	3	3

PCC-IT308: DESIGN AND ANALYSIS OF ALGORITHM

(Total Credits: 4, Lectures/Week: 3, Practical/Week: 2)

Course Objectives:

1. To study the theories, processes, methods, and techniques of ComputerAlgorithm.
2. To train the students on performance analysis ofalgorithms.
3. To understand the basic principles of problemsolving.
4. To study about different algorithmic design techniques and their realization using programming language.

Course Contents:

1. Prerequisite, Introduction & Fundamental of the Analysis of Algorithm Efficiency: [08 Hrs]
Prerequisite and Introduction: Basics of Programming Language (Recursion, Array, Functions and Pointers) and Data Structures (Stack, Queue, Linked List, Graph & Tree and Searching & Sorting), what is mean by Algorithm? Pseudo code and FlowChart.
Fundamental of the Analysis of Algorithm Efficiency: Analysis framework, asymptotic notations and Basic Efficiency Classes, Properties of Asymptotic Notations, Solving recurrence relations, Master Theorem and Mathematical Analysis of Recursive & Non recursiveAlgorithm.

2. Brute Force, Divide-&-Conquer, Decrease-&-Conquer and Transform-&-Conquer:[08 Hrs]
Brute Force: Selection Sort, Bubble Sort, Sequential Search, Exhaustive Search and BF String matching algorithm
Divide-&-Conquer: Merge Sort, Quick Sort, Binary Search and Strassan's Matrix Multiplication
Decrease-&-Conquer: Insertion Sort, DFS, BFS and Topological Sorting
Transform-&-Conquer: Matrix Inverse, AVL Trees, Heap and Heap sort

3. Dynamic Programming and Greedy Technique: [08 Hrs]
Dynamic Programming: Warshall's and Floyd's Algorithms (All pair shortest path), Knapsack problem
Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Huffman Tree and Dijkstra's Algorithm

4. Space and Time Tradeoffs, Limitation of Algorithm Power and Coping with limitations of Algorithm power: [08Hrs]
Space and Time Tradeoffs: Sorting by Counting and hashing
Limitation of Algorithm Power: P, NP, NP-Complete and NP-Hard Problems
Coping with limitations of Algorithm power**:** Backtracking and Branch & Bound

5. Advanced in Computer Algorithm: [08Hrs]
Recursion with memorization, Randomized Algorithm, Monte-Carlo Algorithm, Las Vegas Algorithm, Atlantic Algorithm, Probabilistic Analysis & Amortized Analysis, Probabilistic Algorithm and Parallel Algorithms and Advanced trends in Computer Algorithms.

Reference Books:

1. Anany Levitin, "Introduction to the design & analysis of algorithms", Pearson Publication, 2nd Edition.
2. T. H. Cormen, Leiserson and Rivest, "Introduction to Algorithms",PHI.
3. Horowitz, Sahni and Rajasekaran, "Fundamentals of Computer Algorithms", GalgotiaPublications.
4. Venugopal & Prasad, "Mastering C", TMH, 2ndEdition

Course Outcomes:

CO1: Arrange the correctness of algorithm using inductive proofs and invariants.

CO2: Understand the design of efficient algorithm and analyze the algorithms

CO3: Describe, apply and analyze the complexity of certain divide and conquer, greedy and dynamic programming algorithm.

CO4: Choose the proper suitable algorithmic design technique for solving any problem.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO CO	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	1	3	3	1	3	1			1			3	2	2	3
CO2	2		3	2	1			1	1		2	3	3	3	
CO3	2	3	2		3	2	2			3			3	1	1
CO4	1	2	1	3		2	1	3	2		2				3

PCC-IT309: COMPUTER NETWORKS

(Total Credits: 4, Lectures/Week: 3, Practical/Week:2)

Course Objectives:

1. Study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IPmodel.
2. Solve problems in Transport Layer: connection management, flow control, congestion control and QoS.
3. Familiar with wireless networking concepts and network management solutions and related protocol: SNMP, CSMA, IEEE802.11.
4. Identify, describe and give examples of the networking applications used in everyday tasks such as email, WWW, HTTP, FTP.

Course Contents:

1. Introduction to computer networks and Internet [07 Hrs]

Understanding of network and Internet, The network edge, The network core, Understanding of Delay, Loss and Throughput in the packet-switching network, protocols layers and their service model, History of the computer network.

2. Application Layer: [07Hrs]

Principles of computer applications, Web and HTTP, E-mail, DNS, Socket programming with TCP and UDP.

3. Transport Layer: [10Hrs]

Introduction and transport layer services, Multiplexing and Demultiplexing, Connection less transport (UDP), Principles of reliable data transfer, Connection oriented transport (TCP), Congestion control.

4. Network Layer: [10Hrs]

Introduction, Virtual and Datagram networks, study of router, IP protocol and addressing in the Internet, Routing algorithms, Broadcast and Multicast routing.

5. The Link layer and Local area networks: [06Hrs]

Introduction and link layer services, error-detection and correction techniques, Multiple access protocols, addressing, Ethernet, switches.

Term work:

Students will have to perform at least 10-15 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical during the semester, marks should be given at the end of thesemester.

Textbooks:

1. James F. Kurose and Keith W. Ross*, " Computer Networking: A Top-Down Approach",* Pearson Education India, 5/e,2012
2. B. A. Forouzan and Firouz Mosharraf, Computer Networks, A Top-Down Approach, Tata McGraw- Hill,2012

Reference Books:

1. LarryLPetersonandBSDavie,Computer Networks:ASystemsApproach,Elsevier,2012

2. W. Richard Stevens, TCP/IP Illustrated, Vol. 1: The Protocols, 2nd Edition,Pearson
3. B. A. Forouzan, “Data Communications and Networking”, 4th Edition, Tata McGraw-Hill,2010
4. William Stallings, “Data and computer Communication”, 7th Edition, PearsonEducation,
5. A S Tanenbaum, “Computer Networks”, 4th Edition, PearsonEducation
6. Alberto Leon Garcia and Indra Widjaja, “Communication Networks, Fundamental Conceptsand Key Architectures”, 2nd Edition, Tata McGraw-Hill,2004

Course Outcomes:

CO1: Understand TCP-IP Network stack, the interplay between multiple layers, issues and functionality and socket programming functions.

CO2: Describe the Session layer design issues and Transport layer services.

CO3: Design networks with ipV4 classful addressing, realize issues, design with CIDR, IPV6 fields and benefit.

CO4: Describe packet forwarding and routing in internet using appropriate protocols and usage of other protocols in network layer.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**D**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	2	2	3		2	1	2	2		2		3	1	3	
CO2		2			2			2	2			2		3	
CO3	3		3		2	1					2	3	2	3	
CO4	2	1	2	2	1					2	2	2		2	1

PEC-IT310 -Cloud Computing

(Total Credits: 3, Lectures/Week: 3)

Course Objectives:

1. To provide an in-depth and comprehensive knowledge of the Cloud Computing fundamental issues, technologies, applications and implementations.
2. To understand the underlying principle of cloud virtualization, cloud storage, data management and data visualization
3. To understand different cloud programming platforms and tools to develop and deploy applications on cloud

Course Contents:

1. Unit-I **[8Hrs]**

History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing.Introduction to Cloud Computing. Technologies for Network based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.case Studies of cloud (Apache VCL, Amazon, IBM, Azure)

2. Unit-II [8 Hrs]

Introduction to Cloud Computing- Cloud issues and challenges - Properties - Characteristics - Service models, Deployment models. Cloud resources: Network and API - Virtual and Physical computational resources - Data-storage. Virtualization concepts - Types of Virtualization- Introduction to Various Hypervisors - High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs .

3. Unit-III [8 Hrs]

Service models - Infrastructure as a Service (IaaS) - Resource Virtualization: Server, Storage, Network - Case studies. Platform as a Service (PaaS) - Cloud platform & Management: Computation, Storage - Case studies. Software as a Service (SaaS) - Web services - Web 2.0 - Web OS - Case studies – Anything as a service (XaaS).

4. Unit-IV [6Hrs]

Cloud Access: authentication, authorization and accounting - Cloud Provenance and meta-data - Cloud Reliability and fault-tolerance - Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards

5. Unit-V **[10 Hrs]**

Serving the Business with SOA and Cloud Computing, Query API - User Authentication Connecting to the Cloud - OpenSSH Keys - Tunneling / Port Forwarding - Simple Storage Service - S3, EC2 - EC2 Compute Units, Platforms and storage, EC2 pricing, EC2 customers Amazon Elastic Block Storage - EBS - Ubuntu in the Cloud - Apache Instances in EC2 – Amazon Cloud Services- Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple DB, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon Simple Queue Service (Amazon SQS), Amazon Relational Database Service (Amazon RDS) , EC2 Applications - Web application design - AWS EC2 Capacity Planning – Apache Servers - Mysql Servers - Amazon Cloud Watch - Monitoring

Textbooks:

1. Barrie Sosinsky, " Cloud Computing Bible" John Wiley & Sons, 2010
2. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, "Distributed and cloud computing from Parallel Processing to the Internet of Things", Morgan Kaufmann, Elsevier – 2012.
3. Judith S. Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper "Cloud Computing For Dummies"

Reference Books:

1. Amazon Web Services For Dummies By Bernard Golden
2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, Cloud Security and Privacy An Enterprise Perspective on Risks and Compliance, O'Reilly 2009

Course Outcomes:

CO1: Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing

CO2: Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.

CO3: Identify and understand the fundamental concepts in Amazon cloud, including benefits process, methods

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**D**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1		2	2	3	2		2	2	1		2	2		3	3
CO2		2	3		3				2		3	3	1	3	3
CO3	1		3	1	3			1	2	1	3	3		3	3

PEC-IT311 -Big Data Analytics

(Total Credits: 3, Lectures/Week: 3)

Course Objectives:

1. To optimize business decisions and create competitive advantage with Big Data analytics.
2. To explore the fundamental concepts of big data analytics.
3. To learn to analyze the big data using intelligent techniques.
4. To understand the various search methods and visualization techniques and various techniques for mining data stream.
5. To understand the applications using Map Reduce Concepts.
6. To introduce programming tools PIG & HIVE in Hadoop echo system.

Course Contents:

Unit 1 Introduction to Big data: Introduction to Big Data Platform, Challenges of Conventional Systems, Intelligent data analysis, Nature of Data, Analytic Processes and Tools, Analysis vs Reporting.
Mining data streams : Introduction To Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams , Counting Distinct Elements in a Stream, Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform(RTAP) Applications , Case Studies: Real Time Sentiment Analysis, Stock Market Predictions.

Unit 2 Hadoop: History of Hadoop, the Hadoop Distributed File System, Components of Hadoop Analysing the Data with Hadoop- Scaling Out, Hadoop Streaming, Design of HDFS-Java interfaces to HDFS Basics, Developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run-Failures-Job Scheduling-Shuffle and Sort, Task execution , Map Reduce Types and Formats- Map Reduce FeaturesHadoop environment.

Unit 3 Frameworks: Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services – HiveQL, Querying Data in Hive, fundamentals of HBase and ZooKeeper - IBM InfoSphere BigInsights and Streams.

Unit 4 Predictive Analytics: Simple linear regression, Multiple linear regression, Interpretationof regression coefficients. Visualizations - Visual data analysis techniques, interaction techniques, Systems and applications.

Course Outcomes:

CO1: Work with big data platform and explore the big data analytics techniques business applications.

CO2: Design efficient algorithms for mining the data from large volumes.

CO3: Analyse the HADOOP and Map Reduce technologies associated with big data analytics.

CO4: Explore on Big Data applications Using Pig and Hive.

CO5: Understand the fundamentals of various big data analytics techniques.

.

Reference Books:

1. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer, 2007.
2. Tom White “Hadoop: The Definitive Guide” Third Edition, O’reilly Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”, McGrawHill Publishing, 2012.
4. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics”, John Wiley& sons, 2012.
5. Glenn J. Myatt, “Making Sense of Data”, John Wiley & Sons, 2007.

6. Pete Warden, “Big Data Glossary”, O’Reilly, 2011. 8. Jiawei Han, Micheline Kamber “Data Mining Concepts and Techniques”, 2 nd Edition, Elsevier, Reprinted 2008.
7. Jiawei Han, Micheline Kamber “Data Mining Concepts and Techniques”, 2 nd Edition, Elsevier, Reprinted 2008.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**C**	**D**	**e**	**f**	**g**	**h**	**i**	**j**	**K**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1		2	2	3	2		2	2	1		1	2		3	3
CO2		2	3		3				2			3	2	2	3
CO3	2		2	1	2			1	2	1	3	3		3	3
CO4	1		2	1	3			1	2	1	3	2		3	3
CO3	1		3	1	3			1	2	1	2	3	1	3	1

PEC-IT312- Blockchain Technology

(Total Credits: 3, Lectures/Week: 3)

Course Objectives:

1. Understand how blockchain systems (mainly Bitcoin and Ethereum) work
2. The student will be able to assess Blockchain applications in a structured manner.
3. The student will be able to present Blockchain concepts clearly and persuasively.
4. The student will be able to create an own Crypto token.

Course Contents:

Unit1 Basics: Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete. Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

Unit2 Blockchain: Introduction, Advantage over conventional distributed database, Blockchain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Blockchain application, Soft & Hard Fork, Private and Public blockchain.

Unit3 Distributed Consensus: Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate.

Unit4 Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin.

Unit 5 Cryptocurrency Regulation: Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy. Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Blockchain.

Course Outcomes:

CO1: Understand the structure of a blockchain and why/when it is better than a simple distributed database.

CO2: Analyze the incentive structure in a blockchain based system and critically assess its functions, benefits and vulnerabilities.

CO3: Evaluate the setting where a blockchain based structure may be applied, its potential and its limitations.

CO4: Attain awareness of the new challenges that exist in monetizing businesses around blockchains and smart contracts.

Reference Books:

1. John Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).
2. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies.
3. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System.
4. DR. Gavin Wood, "ETHEREUM: A Secure Decentralized Transaction Ledger,"Yellow paper.2014. 4.

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**A**	**b**	**c**	**D**	**e**	**f**	**g**	**h**	**i**	**J**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1		2		1		2	1	2				3	1	3	2
CO2		1	2		2	3	2	2	2	3	1	2		2	
CO3	2		3	1	1			1	2	1	2	2		2	2
CO4	1	1	3	2	2				2	1	2	2		2	2

PCC-IT313: COMPILER DESIGN

(Total Credits: 4, Lectures/Week: 3, Practical/Week: 2)

Course Objectives:

1. Acquire knowledge in different phases and passes of compiler and specifying different types of tokens by lexical analyzer. Also, able to use the compiler tools like LEX, YACC,etc.
2. Parser and its types i.e. top-down and bottom-upparsers.
3. Construction of LL, SLR, CLR and LALR parsetable.
4. Syntax directed translation, synthesized and inheritedattributes.
5. Acquire knowledge in different phases and passes of Compiler, and specifying different types of tokens by lexical analyzer, and also able to use the Compiler tools like LEX, YACC,etc.

Course Contents:

1. Introduction [4Hrs]

Language processors, The structure of a compiler, The science of building a compiler, Applications of compiler technology.

2. Lexical analysis [7Hrs]

Role of a lexical analyzer, Input buffering, Specification of tokens, Recognition of tokens, Finite automata, From regular expression to automata, Minimizing the number of states of a DFA, Design of a lexical analyzer generator, Lexical Analyzer generator Lex.

3. Syntaxanalysis [12Hrs]

Introduction, CFG, Writing a grammar, Top-down parsing, Recursive descent and predictive parsers (LL), Non recursive predictive parsing, Bottom-up parsing, Simple LR, More powerful LR parsers.

4. SyntaxDirectedTranslation [5Hrs]

Syntax directed definitions, Evaluation orders of SDDs, Applications of SDTs, SDT schemes.

5. Intermediatecodegeneration [5Hrs]

Variants of syntax trees, Three address codes, Types and declarations, Type checking, Control flow, Backpatching.

6. CodeGeneration [7Hrs]

Issues in design of code generator, The target language, Addresses in target code, Basic blocks offlow graphs, Optimizations of basic blocks, peephole optimization, Register allocation and assignment, Optimal code generation for expressions.

Term work:

Students will have to perform at least 10-15 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical during the semester, marks should be given at the end of thesemester.

Reference Books:

1. V. Aho, R. Sethi, & J. P. Ullman: Compilers: Principles, Techniques & Tools, Second Edition, Addision Wesley.

***Course Outcomes*:**

CO1: Understanding of compiler design and lexical analyzer. Also, able to use the compiler tools like LEX, YACC, etc.

CO2: Understanding of Parser and its types i.e. top-down and bottom-up parsers.

CO3: Formulation of LL, SLR, CLR and LALR parse table.

CO4: To know about storage allocation and optimize, design code generator

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**D**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	2		1			1	2		2	3	2	2	3
CO2	3	2		2	2	1			2		2	2	3	2	2
CO3	3	3	3		1				2		3	3	3	2	
CO4		2	3						2	1		2		1	

LAB-IT314 Computer Laboratory-III (Full Stack Development)

(Total Credits: 1, Lectures/Week: 0, Practical/Week: 2)

Selected Language/Frameworks: HMTL5, Django, React, Redux, vue js, Node js, Express & MongoDB, PostgreSQL

Term work:

Students will have to perform at least 10-15 practical in the subject. The practical will be designed by the concerned teacher and should be based on the current trends and technology in the relevant subject. Teacher should also give some home assignment to the students. The evaluation of the student should be done continuously and based on his/her performance and attendance for the practical during the semester, marks should be given at the end of the semester

Course Outcomes:

CO1: Understanding of basic concepts of HMTL5, Django, React, Redux, vue js, Node js, Express & MongoDB, PostgreSQL, etc.

CO2: Apply the knowledge in projects assigned.

CO3: Evaluating the performance of the assignments.

CO4: Develop a product using learned technologies as mini-project

Articulation Matrix:

(3) High, **(2)** Medium, **(1)** Low															
PO / CO	**a**	**b**	**c**	**D**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**PSO1**	**PSO2**	**PSO3**
CO1	2											1	2		
CO2			3				2			2		2		2	
CO3				2			2		1					2	
CO4						3		3	2	3	2	2			3