Syllabus Structure
Third Year – Department of Information Technology
To be followed from Academic Year 2018-19

SEMESTER – V

Course Code	Course Title	Lectures (L)	Tutorials (T)	Practical (P)	Credits	
					Th	*Pr*
IT301	Theory of Computation	03	01	--	04	--
IT303	Software Engineering	03	--	02	03	01
IT305	Operating Systems	03	--	02	03	01
IT307	Data Base Management System	03	01	02	04	01
MA302	**Elective – II** MA302A: Mathematics- IV (Complex Analysis) MA302B: Mathematics- IV(Statistics and Probability) MA302C: Mathematics- IV(Numerical Analysis)	04	--	--	04	--
IT311	Computer Laboratory-III (PHP+ MySQL)	02	--	02	--	01
IT313	Seminar - I	--	--	02	--	01
	Total	**18**	02	**10**	**18**	**05**

SEMESTER -VI

Course Code	Course Title	Lectures (L)	Tutorials (T)	Practical (P)	Credits	
					Th	*Pr*
IT302	Data Mining and Data Warehousing	03	01	--	04	--
IT304	Design and Analysis of Algorithms	03	01	02	04	01
IT306	Computer Networks	03	--	02	03	01
IT308	Software Testing	03	--	02	03	01
IT310	Compiler Construction	03	01	02	04	01
IT312	Computer Laboratory-IV	02	--	02	--	01
IT314	Seminar - II	--	--	02	--	01
	Total	**17**	03	**12**	**18**	**06**

Head
Department of Information Technology

IT-301: THEORY OF COMPUTATION

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

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 and grammars.
4. Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science.
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's Theorem.

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. Turing Machine: [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 analyser, 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 to Automata Theory Languages and Computations”, Pearson Education Asia, Second Edition.
2. E V Krishnamurthy, “Introduction to Theory of Computer Science”, EWP Second 2nd Edition.

Course Outcomes (CO)

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 practical importance.

IT303: SOFTWARE ENGINEERING

(Total Credits: 3, 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 in Industry
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: [4 Hrs]

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: [2 Hrs]

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

5. Requirements Engineering [4 Hrs]

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 [4 Hrs]

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

7. Design Engineering [4 Hrs]

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

8. Testing [4 Hrs]

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 **[4 Hrs]**

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

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

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",* 6th Ed. TMH.
2. Ian Sommerville, "Software Engineering", 8th Ed. Addison-Wesley.

Reference Books:

1. Mall R., "Fundamentals of Software Engineering", Second Edition, Prentice Hall 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, ISBN 0-19-510539-7
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 (CO)

CO1: Ability to analyse 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

IT305: 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: [06 hrs]

Architecture of OS (Ex. Monolithic, Microkernel, Layered and Exokernel), Operating system objectives and functions, Interaction of O. S. and hardware architecture, Evolution of operating systems, Batch, Multiprogramming, Multitasking, Multiuser, parallel, distributed & real –time O.S., System calls, 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: [08 hrs]

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 the semester.

Course Outcomes (CO)

CO1: Understand functional architecture of an operating system

CO2: Develop algorithms for subsystem components

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

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

IT307: DATA BASE MANAGEMENT SYSTEM

(Total Credits:5, Lectures/Week:3, Practical/Week:2)

Course Objectives:

1 Demonstrate an understanding of the relational data model.
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 a database.
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: [3 Hrs]

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: [5 Hrs]

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, cost estimation, measures of query cost, selection operation, sorting, join operation and join strategies, evaluation of expressions.

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: **[4 Hrs]**

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 Database Management"
2. G. Everest, "Database Management", McGraw Hill
3. C.J. Date, "An Introduction to database Concepts".

Course Outcomes (CO)

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.

MA302: ELECTIVE - II: M-IV

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

MA302A: COMPLEX ANALYSIS

Course objectives:

1. To perform algebra with complex numbers.
2. To identify complex-differentiable functions.
3. To compute complex line integrals.
4. To use residue theorem.
5. To understand the conformal mappings and their engineering applications.

Course Outcomes:

1. Student will be able to compute sum, product, division, polar form, and nth roots of complex numbers and will also be able to understand their geometry.
2. Student will able identify the analytic functions and will be able to express these functions as a power series.
3. Student will be able to compute line integrals of complex functions through parameterization of curves.
4. Student will be able to use Cauchy's theorem, residue theorem to calculate line integrals and improper integrals.
5. Student will be able to use analytic functions as conformal mappings.

Course Contents:

1. Introduction:

Introduction to Complex Variables.

2. Function of complex variables:

Limit, continuity, differentiability, analytic functions and their properties, Cauchy-Riemann equation, harmonic functions, elementary complex functions and their properties.

3. Line Integral:

Cauchy's theorem, Cauchy's Integral formula, and their applications.

4. Series:

Taylor and Laurent theorems, classification of singularities, residues, Cauchy's residue theorem, improper Integrals, conformal mappings.

Reference Books:

1. Anant R. Shastri, *An Introduction to Complex Analysis*, Macmillan Publishers India (2000).
2. James W. Brown, Ruel V. Churchill, *Complex Variables and Applications (Seventh Edition)*, Mc Graw Hill (2003).
3. Erwin Kreyszig, *Advanced Engineering Mathematics (10th Edition), Wiley Publication*

MA302B: Probability Theory and Statistics

Course Objectives:

1. To equip the students with the broad perspective of probability theory.
2. To develop the understanding of various discrete and continuous distributions along with their properties.
3. To understand and differentiate among various statistical and random processes techniques.

Course Outcomes:

1. Student will demonstrate the ability of data analysis and will be able to describe sample space for various random experiments
2. Student will identify the random variables as discrete and continuous random variables and will be able to apply appropriate distribution methods.
3. Student will be able to interpret the mean of a random variable in terms of the Law of Large Numbers.
4. Student will be able to use the Normal distribution, including the preservation of Normality under linear transformation.
5. Student will apply the Central Limit Theorem to problems involving sums and averages of variables from arbitrary distributions.
6. Student will be able to apply the tests of goodness of fit.

Course Contents:

1. Introduction:

Sample Space and Events, Classical Probability, Conditional Probability, Independent Events, Bayes Theorem, Random Variable, Probability Measure, Sigma Field, Distribution Function.

2. Functions of Random Variables:

Expectation, Moment Generation Function and Its Properties, Characteristic Function.

3. Distributions:

Special Discrete Distributions – Binomial, Poisson, Geometric, Special Continuous Distributions – Uniform, Exponential, Beta, Gamma. Normal Distribution.

4. Chebychev's Inequality, Transformation of Variables, Joint and Marginal Distributions, Conditional Distribution.

5. Covariance:

Correlation, Transformation of Variables, Independence of Random Variables, Random Vector, Weak Law of Large Numbers, Central Limit Theorem, Bivariate Normal Distribution.

6. Regression:

Regression, Least Square Method, Sampling Distributions of Parameters, ChiSquare, t and F Distribution.

7. Theory of Estimation:

Theory of point estimation, Properties of Point Estimator, Maximum Likelihood Estimator, Interval Estimation, Confidence Interval, Testing of Hypotheses, Likelihood Ratio Test, Goodness of Fit test, Stochastic Processes.

Reference Books:

1. Sheldon M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, Elsevier (2012)
2. Vijay K. Rohatgi and A. K. Md. Ehsanes Saleh, *An introduction to probability and Statistics*, Wiley (2011)
3. Richard A. Johnson, Miller and Freund, *Probability and Statistics for Engineers*, PHI Learning (2010)
4. Athanasios Papoulis and S. Unnikrishna, *Probability, Random Variables, and Stochastic Processes*, Tata McGraw Hill (2002)
5. Erwin Kreyszig, *Advanced Engineering Mathematics (Tenth Edition)*, Wiley Publication.

IT311: COMPUTER LABORATORY - III

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

Course Contents:

PHP

1. PHP Basics, Functions, PHP Forms, PHP Filters, PHP session handling
2. Working with files in PHP
3. Exception Handling
4. PHP XML

MySQL

1. A brief history of databases, MYSQL Databases. MySQL datatypes
2. Introduction to PHP Myadmin
3. Creating and Checking Tables
4. MYSQL Connect
5. MYSQL Limit data

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.

IT313: SEMINAR - I

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

Seminars are included with an aim to help students improve their presentation skills and to help them gain confidence. Students are required to deliver two seminars through the semester (individually). Out of the two seminars, one is to be delivered on technical topic which will highlight some recent technology or invention. The second seminar should be on some non-technical topic. They have to write report of both the seminars, spiral bound them and submit to the concern faculty in - charge.

SEMESTER -VI

IT-302: DATA MINING & DATA WAREHOUSING

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

Course Objectives

1. To understand the principles of Data warehousing and Data Mining.
2. To be familiar with the Data warehouse architecture and its Implementation.
3. To understand the various Data pre-processing Methods.
4. To perform classification and prediction of data

Course Contents:

Introduction **[8 Hrs]**

Introduction to Data Mining, Importance of Data Mining, Data Mining functionalities, Classification of Data mining systems, Data mining architecture, Major Issues in Data Mining, Data mining metrics, Applications of Data Mining, Social impacts of data, Data Mining from a Database Perspective, Online Analytical Processing (OLAP) – OLAP and Multidimensional Data Analysis.

Data Pre-processing **[6 Hrs]**

Introduction, Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization.

Classification and Prediction **[8 Hrs]**

Basic issues regarding classification and predication, Classification by Decision Tree, Bayesian classification, classification by back propagation, Associative classification, Prediction, Statistical-Based Algorithms, Decision Tree -Based Algorithms, Neural Network Based Algorithms, Rule-Based Algorithms, Other Classification Methods, Combining Techniques, Classifier Accuracy and Error Measures

Clustering **[6 Hrs]**

Similarity and Distance Measures, Hierarchical Algorithms, Partitioned Algorithms, Clustering Large Databases, Clustering with Categorical Attributes

Association Rules **[6 Hrs]**

Basic Algorithms, Advanced Association Rule Techniques, Measuring the Quality of Rules

Applications and other Data mining techniques **[6 Hrs]**

Data Mining Applications, Mining Event Sequences, Visual DM Text Mining, Web Mining, The WEKA data mining Workbench.

Course Outcomes (CO)

CO1: Understanding of the Data Mining principles and techniques for real time applications.
CO2: Store voluminous data for online processing
CO3: Pre-process the data for mining applications and association rules for mining the data
CO4: Deign and deploy appropriate classification techniques
CO5: Cluster the high dimensional data for better organization of the data
CO6: Discover the knowledge imbibed in the high dimensional system

IT304: DESIGN AND ANALYSIS OF ALGORITHM

(Total Credits: 5, Lectures/Week: 3, Practical/Week:2)

Course Objectives:

1. To study the theories, processes, methods, and techniques of Computer Algorithm.
2. To train the students on performance analysis of algorithms.
3. To understand the basic principles of problem solving.
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 Flow Chart.

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 recursive Algorithm.

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: [08 Hrs]

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: [08 Hrs]

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", Galgotia Publications.
4. Venugopal & Prasad, “Mastering C”, TMH, 2nd Edition

Course Outcomes (CO)

CO1: Understand and compare various data structure.
CO2: Arrange the correctness of algorithm using inductive proofs and invariants.
CO3: Understand the design of efficient algorithm.
CO4: Describe, apply and analyze the complexity of certain divide and conquer, greedy and dynamic programming algorithm.
CO5: Explain the ways to analyze the algorithms.
CO6: Choose the proper suitable algorithmic design technique for solving any problem.

IT306: COMPUTER NETWORKS

(Total Credits: 3, 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/IP model.
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, IEEE 802.11.
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: [07 Hrs]

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

3. Transport Layer: [10 Hrs]

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: [10 Hrs]

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: [06 Hrs]

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 the semester.

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 FirouzMosharraf, Computer Networks, A Top-Down Approach, Tata McGraw-Hill, 2012

Reference Books:

1. Larry L Peterson and B S Davie, Computer Networks: A Systems Approach, 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, Pearson Education,
5. A S Tanenbaum, "Computer Networks", 4th Edition, Pearson Education
6. Alberto Leon Garcia and IndraWidjaja, "Communication Networks, Fundamental Concepts and Key Architectures", 2nd Edition, Tata McGraw-Hill, 2004

Course Outcomes (CO)

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.

IT308: SOFTWARE TESTING

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

Course Objectives:

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
5. To gain software testing experience by applying software testing knowledge and methods to

practice-oriented software testing projects.

6. To understand software test automation problems and solutions.

Course Contents:

1. Introduction [8 Hrs]

Faults, Errors, and Failures, Basics of software testing, Testing objectives, Principles of testing, Requirements, behavior and correctness, Testing and debugging, Test metrics and measurements, Verification, Validation and Testing, Types of testing, Software Quality and Reliability, Software defect tracking. Test Cases,Test case Design,Building test cases Manual vs Automation testing

2. White Box and Black Box Testing [8 Hrs]

White box testing, static testing, static analysis tools, Structural testing: Unit/Code functional testing, Code coverage testing, Code complexity testing, Black Box testing, Requirements based testing, Boundary value analysis, Equivalence partitioning, state/graph based testing, Model based testing and model checking, Differences between white box and Black box testing.

3. Integration, System, and Acceptance Testing [8 Hrs]

Top down and Bottom up integration, Bi-directional integration, System integration, Scenario Testing, Defect Bash, Functional versus Non-functional testing, Design/Architecture verification, Deployment testing, Beta testing, Scalability testing, Reliability testing, Stress testing, Acceptance testing: Acceptance criteria, test cases selection and execution.

4. Test Selection & Minimization for Regression Testing [8 Hrs]

Regression testing, Regression test process, Initial Smoke or Sanity test, Selection of regression tests, Execution Trace, Dynamic Slicing, Test Minimization, Tools for regression testing, Ad hoc Testing: Pair testing, Exploratory testing, Iterative testing, Defect seeding.

5. Test Management and Automation [8 Hrs]

Test Planning, Management, Execution and Reporting, Basics of automation testing – why, when and how to perform automation testing, Factors for choosing a particular tool Software Test Automation: Scope of automation, Design & Architecture for automation, Generic requirements for test tool framework, Test tool selection, Testing in Object Oriented Systems, Explanation of Selenium and its advantages, Selenium IDE, Selenium RC, Selenium Grid, Selenium Webdriver/version 2.0Differences between Selenium and QTP

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. Aditya P Mathur, “Foundations of Software Testing”, 2nd Ed, Pearson Ed.
2. S. Desikan and G. Ramesh, “Software Testing: Principles and Practices”, Pearson Education.

Reference Books:

1. Louise Tamres, "Introducing Software Testing", Pearson Education, ISBN:9788177582437
2. Ron Patton, “Software Testing”, Pearson Ed.
3. Edward Kit, “Software Testing in the Real World: Improving the process”, AW Professional.
4. Naik and Tripathy, “Software Testing and Quality Assurance”, Wiley
5. K. K. Aggarwal and Yogesh Singh, “Software Engineering”, New Age International Publication.

Course Outcomes (CO)

CO1: Identify and understand the fundamental concepts in software testing, including software testing objectives, process, criteria strategies, and methods.

CO2: Analysis of various software testing issues and solutions in software, unit test, integration, regression, and system testing

CO3: Understanding of planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report

CO4: To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions

CO5: To gain software testing experience by applying software testing knowledge and methods to practice-oriented Software testing projects and Understanding of Automation Testing.

IT310: COMPILER CONSTRUCTION

(Total Credits: 5, 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-up parsers.
3. Construction of LL, SLR, CLR and LALR parse table.
4. Syntax directed translation, synthesized and inherited attributes.
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 [4 Hrs]

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

2. Lexical analysis [7 Hrs]

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. Syntax analysis [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. Syntax Directed Translation [5Hrs]

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

5. Intermediate code generation **[5Hrs]**

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

6. Code Generation **[7Hrs]**

Issues in design of code generator, The target language, Addresses in target code, Basic blocks of flow 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 the semester.

Reference Books:

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

Course Outcomes (CO)

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

IT312: COMPUTER LABORATORY - IV

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

Course Objectives (COs):

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" 2nd Ed. 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

IT314: SEMINAR - II

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

Seminars are included with an aim to help students improve their presentation skills and to help them gain confidence. Students are required to deliver two seminars through the semester (individually). Out of the two seminars, one is to be delivered on technical topic which will highlight some recent technology or invention. The second seminar should be on some non technical topic. They have to write report of both the seminars, spiral bound them and submit to the concern faculty in - charge.