Subject Name	L	T	P	Credit
Theory of Computation	3	0	0	3

Unit 1: Introduction to Theory of Computation and Finite Automata
Mathematical preliminaries : Alphabet, Strings, Languages, States, Transitions, Grammar, Languages and Automata, Chomsky Hierarchy, Designing Deterministic and Nondeterministic Finite Automata, Regular Expressions and Languages: Recursive definition of regular expression, regular set, identities of regular expressions, examples and FA. Equivalence of DFA and NDFA, Mealy & Moore machines, Minimization of finite automata. Two way finite automata.

Unit 2: Context Free Grammars (CFG) and Languages
Finite automata and regular expression, Pumping lemma and regular sets, Application of pumping lemma, Myhill-Nerode Theorem, Linear Grammar, Right linear Grammar and Left Linear grammar , Regular Grammar Context Free Grammar- Definition, Derivation, sentential form, parse tree, Derivation, Parse tree, Ambiguity in Grammar and Language, Simplification of CFG-Elimination of Useless Symbol, Null Production and Unit Production, Normal Forms- Chomsky Normal form, Greibach normal form.

Unit3: Push Down Automata (PDA)
Definition, The Language of PDA, Equivalence of PDA 's and CFG- CFG to PDA, PDA to CFG, Deterministic Push Down Automata (DPDA) – Regular language and DPDA, DPDA and
CFL, DPDA and ambiguous grammar, Non-deterministic Push Down Automata (NPDA).
The pumping lemma for CFL's, Closure properties of CFL's, Decision problems involving CFL's.

Unit 4: Turing machines
Turing machines (TMs): TM Model and conventions, Formal Definition, TM Instantaneous Description (ID), Transition Function, Languages of TM, Types of TM: Deterministic Turing Machines (DTM) and Non-deterministic Turing Machines (NTM), Extension to Basic TM: TM with Multiple tracks, Multi tape TMs, Universal TM (UTM), Church-Turing hypothesis , Post Machines: Introduction to Post Machines (PMs), Comparison between FA, PDA,and ,TM Undecidable problems about Turing Machines, Properties of recursive & recursively enumerable languages, Universal Turing Machine.

Unit 5: Tractable & Intractable
Problems Classes: P, NP, NP complete and NP Hard, Examples of these Problems like Satisfiability problems, Vertex Cover Problem, Hamiltonian Path Problem, Traveling sales man problem, Partition problem etc.

Text Book :

1. K.L.P Mishra & N.Chandrasekaran, "Theory of Computer Science", PHI Learning

References:

1. John Martin "Introduction to Languages and Theory of Computation", McGraw-Hill .
2. John E. Hopcroft, Jeffery Ullman Introduction to Automata theory, Languages & computation" , Narosa Publishers
3. Daniel I.A. Cohen, "Introduction to Computer Theory", Wiley India.
4. Michael Sipser, "Introduction to the Theory of Computation", Books/Cole Thomson
Learning, 2001.

Subject Name	L	T	P	Credit
Analysis and design of algorithms	3	0	0	3

Unit 1:

Introduction: Algorithm, performance evaluation of algorithms, space & time complexity, asymptotic notations, Master's Theorem. Divide and Conquer: General Concept, Finding the maximum and minimum, Quick Sort, Merge Sort, Max and Min Heap, Heap Sort, Binary Search, Strassen's matrix multiplication.

Unit 2:

Greedy Algorithm : General Concept, Knapsack Problem (Fractional Knapsack), Job Sequencing with Deadline, Huffman's Codes, Minimum Cost Spanning Tree Kruskal's Algorithm, Prim's Algorithm, Single Source Shortest Path Dijkstra's Algorithm. Optimal merge Pattern.

Unit3:

Dynamic Programming: General Concept, Multistage graph, Reliability design, Matrix Chain Multiplication,0/1 Knapsack Problem DP solution, Activity selection problem DP solution, Single Source Shortest Path ,Bellman Ford Algorithm, All pairs shortest paths, Traveling salesman problem.

Unit 4:

Backtracking: Basic idea, 4 queen's problem ,8 Queens problem, Graph Coloring, Hamiltonian Cycles.
Branch And Bound: Basic idea, LC search, the 15 puzzle problem, LC Branch and Bound, 0/1 Knapsack Problem.

Unit 5:

Advanced Algorithms : Fibonacci Heaps, Network flows, Maximum Flow; Minimum cost circulation, Approximation Algorithms.
Introduction to NP Completeness: Basic concepts on NP hard and NP Complete Problems, Discussion on one NP hard graph problem.

Text Book :

1. 1. E.Horowitz and S. Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press; Second edition(2008)

References:

2. Thomas H. Cormen, Charles E. Leiserson , Ronald L. Rivest, and Clifford Stein "Introduction to Algorithm" , PHI.

3. Simen Harris, James Ross, "Beginning Algorithms ",Wiley India.

4. Robert Sedgewick, Kevin Wayne, " Algorithms ", Addison Wesley

5. S.K. Basu, "Design methods and analysis of algorithms, PHI

Suggested List of Experiment

1. Implementation and Time analysis of linear and binary search algorithm
2. Implementation of Max Heap Sort Algorithm.
3. Implementation and time analysis of factorial program using iterative and recursive method.
4. Implementation of a knapsack problem using dynamic programming.
5. Implementation of chain matrix multiplication using dynamic programming.
6. Implementation of a knapsack problem using greedy algorithm.
7. Implementation of Prim's Algorithm.
8. Implementation of Kruskal's Algorithms.
9. Implementation of Dijkstra's algorithms.
10. Implementation of Bellman Ford algorithm and analyze its time complexity.
11. Implementation of Hamiltonian Cycle Problem.

Subject Name	L	T	P	Credit
Relational database management systems	2	0	4	4

Unit 1 :
Introduction to DBMS : Introduction to DBMS concepts and architecture, file system organization, advantages of DBMS, Data models, schemas and instances, Data dependency, functions of DBA, Entities and attributes, entity types, Key attributes, Relationships, ER data model: Entitles and attributes, Entity types, Defining the E-R diagram, Concept of Generalization, Aggregation and Specialization.transforming ER diagram into the tables., Various data models, hierarchical data model, Network data model, Relational data model, comparison between three models.

Unit 2 :
Relational data models: Relational data models, Domains, tuples, attributes, relations, characteristics of relations, keys, key attributes of a relation, Relational database, schemas, Integrity constraints, intension and extension.
Relational Query languages: SQL-DDL, DML, integrity constraints, Complex queries, various joins, indexing, triggers, Relational algebra and relational calculus, Relational algebra operations like select, Project, Join, Division, outer union. Tuple oriented and domain oriented relational calculus and its operations.

Unit 3 :
Data Base Design concepts, introduction to normalization, various normal forms, functional dependency, Decomposition, Dependency preservation and lossless join, problems with null valued and dangling tuples, multivalued dependencies. Introduction to query optimization, steps of optimization, various algorithms to implement select, project and join operations of relational algebra, heuristic based, cost estimation based optimization methods

Unit 4 :
Transaction Processing Concepts: Transaction System, Testing of Serilizability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures. Log based recovery. Checkpoints deadlock handling. Concurrency. Control Techniques: - Concurrency Control, locking Techniques for concurrency control, time stamping protocols, validation based protocol, multiple granularity. Multi version schemes, Recovery with concurrent transaction. Introduction to Distributed databases, data mining, data warehousing, Object Technology and DBMS, Comparative study of OODBMS Vs DBMS . Temporal, Deductive, Multimedia, Web & Mobile database.

Unit 5 :
Study of Relational Database Management Systems through Oracle/Postgres.
SQL/MySQL: Architecture, physical files, memory structures, background process. Concept of table spaces, segments, extents and block. Dedicated server, multi threaded server. Distributed database, database links, and snapshot. Data dictionary, dynamic performance view. Security, role management, privilege management, profiles, invoker defined security model. SQL queries, Data extraction from single and multiple tables Hierarchical queries, inline queries, flashback queries. Introduction of ANSI SQL branching and looping constructs in ANSI SQL. Cursor management: nested and parameterized cursors, Oracle exception handling mechanism. Stored procedures, in, out, in out type parameters, usage of parameters in procedures. User defined functions and their limitations. Triggers

Text Books :

1. Korth, Silbertz,Sudarshan, “Fundamental of Database System”, McGraw Hill.
2. Elmasri, Navathe, “Fundamentals Of Database Systems”, Pearson Educations.
3. Atul Kahate , “ Introduction to Database Management System”, Pearson Educations.

Reference Books :

1. Date C J, “An Introduction To Database System”, Pearson Educations
2. Rob, “ Data Base System:Design Implementation & Management”, Cengage Learninig
3. Oracle 9i Database Administration Fundamental-I, Volume I, Oracle Press,TMH.

Suggested list of experiments:

1. Delete duplicate row from the table.
2. Display the alternate row from table.
3. Delete alternate row from table.
4. Update multiple rows in using single update statement.
5. Find the third highest paid and third lowest paid salary.
6. Display the 3rd, 4th, 9th rows from table.
7. Display the ename, which is start with j, k, l or m.
8. Show all employees who were hired the first half of the month.
9. Write a sql statements for rollback commit and save points.
10. Write a pl/sql for select, insert, update and delete statements.
11. Write a pl/sql block to delete a record. If delete operation is successful return 1 else return 0.
12. Display name, hire date of all employees using cursors.
13. Display details of first 5 highly paid employees using cursors.
14. Write a database trigger which fires if you try to insert, update, or delete after 7 ’O’ clock.
15. Write a data base trigger, which acts just like primary key and does not allow duplicated values.
16. Create a data base trigger, which performs the action of the on delete cascade.
17. Write a data base trigger, which should not delete from emp table if the day is Sunday.

Note: Number of experiments may be extended to make the better understanding of subject.

Subject Name	L	T	P	Credit
Computer Networks	2	1	2	4

UNIT :- I

Introduction to Computer Networks: need for networking, service description, components, architecture, classifications & types, circuit and packet switching, access Networks and Physical Media.
Layered Architecture: Protocol hierarchy, Design Issues, Services and Interfaces Connection Oriented & Connectionless Services, Service primitives, Design issues & its functionality.
ISO-OSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP. Network standardization. Internet Basics: Naming and Internet addressing, subnets, DNS.

UNIT :- II

Physical Layer: Direct Link Networks, encodings, modulation, Error detection and correction, CRC, Internet Checksum.
Data Link Layer & Protocol : Need, Services Provided, Framing , Flow Control, Error control, Elementary & Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Bit oriented protocols: SDLC, HDLC, BISYNC, LAP and LAPB.

UNIT :- III

MAC Sublayer: MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed
Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted-ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA).
Collision Free Protocols: Basic Bit Map, BRAP, Binary Count Down, MLMA Limited Contention Protocols: Adaptive Tree Walk, URN Protocol, High Speed LAN: Fast Ethernet, Gigabit Ethernet, FDDI, Performance Measuring Metrics. IEEE Standards 802 series & their variant.

UNIT :- IV

Network Layer: Need, Services Provided , Design issues, Routing algorithms: Least Cost Routing algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing, Broadcast Routing, Multicast Routing,
Congestion Control Algorithms: General Principles of Congestion control, Prevention Policies, Congestion Control in Virtual-Circuit Subnets, Congestion Control in Datagram subnets. IP protocol, IP Addresses, Comparative study of IPv4 & IPv6, Mobile IP, Switching vs routing. Switching architectures.

UNIT :- V

Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, Carrying Unicast /Multicast Real-Time Traffic, TCP: Connection Management, Reliability of Data Transfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP Timer Management. Session layer: Authentication, Authorization, Session layer protocol (PAP, SCP, H.24).
Presentation layer: Data conversion, Character code translation, Compresion, Encryption and Decryption, Presentation layer protocol (LPP, Telnet, X.25 packet Assembler/Disassembler). Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).

Text Books:

1. A.S.Tanenbaum, "Computer Networks", Prentice Hall India.
2. Forouzan B. A., "Data Communication and Networking", McGrawHill.

References Books:

1. Larry L. Peterson and Bruce S. Davie, “Computer Networks: A Systems Approach”, Elsevier, Fourth edition, 2007.
2. William Stallings, “Data and Computer Communications,” 5th edition, Prentice Hall India.
3. Uyless Black, “Computer Networks”, Prentice Hall India.
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, “Computer Networks: An Open Source
Approach”, McGraw Hill.

List of Experiments:

1. Study of Different Type of LAN& Network Equipments.
2. Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.
3. LAN installations and Configurations.
4. Study of basic network command and Network configuration commands.
5. Study of network IP.
6. Study of Network simulator (NS).
7. Write a socket Program for Echo/Ping/Talk commands.
8. Create a socket (TCP) between two computers and enable file transfer between them.
9. Create a socket (UDP) between two computers and enable file transfer between them.
10. Implement & Simulate various types of routing algorithm.
11. Study and Simulation of MAC Protocols like Aloha, CSMA, CSMA/CD and CSMA/CA using Standard Network Simulators.
12. Write a program to implement RPC (Remote Procedure Call).

Subject Name	L	T	P	Credit
NAND2TETRIS: Building Computers from First Principles	2	1	2	4

Unit 1 :
Introduction :
Introduction, Boolean function and logic Gates, General information about the course.
Boolean arithmetic, ALU and memory ,Binary numbers, Specification Adders, Arithmetic Logic Unit, Sequential Logic, Flip-flops, Register, memory, Counter. Combinational logic, culminating in the construction of a simple ALU, Sequential logic, from elementary flip-flop gates to registers and RAM units of arbitrary sizes. Machine language, computer architecture and assembler
Overview & Elements of Machine Language, Hack computer and Machine Language, Input/output.

Unit 2 :
Introduction to computer Architecture, Von Neumann Architecture, Fetch-Execute Cycle, Central Processing Unit, Hack Hardware, Integrating the chip-sets into a computer platform capable of running programs written in the machine language.
Assembly Languages and Assemblers, Hack Assembly Language, The Assembly Process introducing an instruction set in both binary and assembly (symbolic) versions, Writing some low-level assembly programs, handling Instructions & Symbols.

Unit 3 :
Virtual machine
VM-I: Stack Arithmetic – Background, VM Specification part I, VM Programming.
VM-II: Program Control – Program flow, Subroutine Calling, VM Specification part II, VM Implementation.
High-level language
Overview & Introduction to High-Level Language, Introducing Jack, a simple high-level object-based language with a Javalike syntax Jack Language Specification, discussing various trade-offs related to the language design and implementation, Using Jack to write a simple interactive game and running it on the computer built.

Unit 4 :
Compiler
Compiler I: Syntax Analysis – Overview, Specification, Implementation. Context-free grammars and recursive parsing algorithms; Building a syntax analyzer (tokenizer and parser) for the jack language.
Compiler II: Code Generation – Overview, Specification, Implementation. Code generation, low-level handling of arrays and objects.

Unit 5:
Operating system
Overview & Introduction, Jack OS Specification, Design and implementation of some classical arithmetic and geometric algorithms that come to play in the implementation of OS, as well as classical mathematical, memory management, string processing, and I/O handling algorithms, needed for completing the OS implementation.

Text Books
1. Noam Nisan & Shimon Schocken, “The Element of Computing Systems”, The MIT Press, Massachusetts.

References
1. http://www.nand2tetris.org/course.php

Suggested list of experiments.

1. Implement all logic Gates with the only building block available is NAND Gates.
2. Implement Half Adder & Full Adder Chips. The only building blocks that can be use are the chips that gradually build in experiment 1.
3. Implement Register and RAM using DFFs.
4. Implement Multiplication Program and I/O-Handling Program.
5. Build the Hack computer platform.
6. Develop an assembler that translates programs written in Hack assembly language into the binary code understood by the hack hardware platform.
7. Build the VM translator focusing on implementation of the Stack arithmetic memory access commands of the VM language.
8. Extend the basic VM translator built in experiment 7 into a full-scale VM translator.
9. Implement Syntax analyzer and extend it into a full-scale Jack compiler.
10. Implement the operating system using Jack Language.
11. Complete Implementation of OS using Pong game.

Subject Name	L	T	P	Credit
Mobile Application Development with Android Laboratory	2	0	4	4

UNIT I :
Introduction :

History of android, introduction to android operating systems, android development tools,
Android architectural overview, standard development environment for android applications, anatomy of an android application, creating a new android application, the android project structure. Frameworks and Tools, Programming in the mobile environment, Development Tools : Installing and using Eclipse with ADT plug-in

UNIT II :

Emulator - android virtual device, launching emulator, editing emulator settings. Installing Virtual machine for Android , configuring the installed tools, dalvik virtual machine & .apk file extension, fundamentals: basic building blocks, activities, services, broadcast receivers & content providers, Application structure, androidmanifest.xml, uses-permission & uses-sdk, resources & R.java, sdk layouts & drawable resources, activities and activity lifecycle, first sample application.

UNIT-III :

Basic UI design : form widgets, text fields, layouts, shared preferences preferences from xml, menu : option menu, context menu, sub menu, menu from xml, menu via code, explicit intents, implicit intents,
UI design : time and date, images and media, composite, alert, dialogs & toast popup, tabs and tab activity, styles & themes, applying themes via code and manifest file

UNIT-IV :

Event driven programming in android (text edit, button clicked), SQLite programming, sqliteopenhelper, sqlitedatabse, cursor, reading and updating contacts, reading bookmarks, Android debug bridge (adb) tool, adapters:- arrayadapters, baseadapters, listview and listactivity, custom listview, gridview using adapters, gallery using adapters

UNIT-V :

Graphics: Performance and Multithreading, Graphics and UI Performance, Android Graphics and Multimedia, Performance, Memory Management, Android Notifications and Alarms. Audio playback and Media Player, SoundPool, Mobile Agents and Peer-to-Peer Architecture, using System Services and Web Services, Mobility and Location Based Services. Using Location based services Telephony and SMS services Bluetooth, network and Wi-Fi, Camera, Accessing Internet and web services from android applications.

Text Books

1. James C. Sheusi, " Android application development for java programmers", Cengage Learning
2. Reto Meier, " Professional Android 2 Application Development", Wiley India

Suggested List of experiments

1. Installation and setup of Java development kit(JDK), setup android SDK, setup eclipse IDE, setup android development tools (ADT) plugins, create android virtual device.
2. Create “Hello World” application. That will display “Hello World” in the middle of the screen using TextView Widget in the red color.
3. Create application for demonstration of android activity life cycle.
4. Create Registration page to demonstration of Basic widgets available in android.
5. Create sample application with login module.(Check username and password) On successful login, Chnage TextView “Login Sucessful”. And on failing login, alert user using Toast “Login fail”.
6. Create login application where you will have to validate usename and passwords Till the username and password is not validated , login button should remain disabled.
7. Create and Login application as above. Validate login data and display Error to user using setError() method.
8. Implement an application that creates an alert upon receiving a message.
9. Create an application for demonstration of Relative and Table Layout in android.
10. Create an application for demonstration of Explicitly Starting New Activity using Intent.
11. Create an application that will Demonstrate Button onClick() Event and change the TextView Color based on button Clicked.
12. Create an UI such that, one screen have list of all the types of cars. On selecting of any car name, next screen should show Car details like: name, launched date, company name.
13. Develop a native application that uses GPS location information.