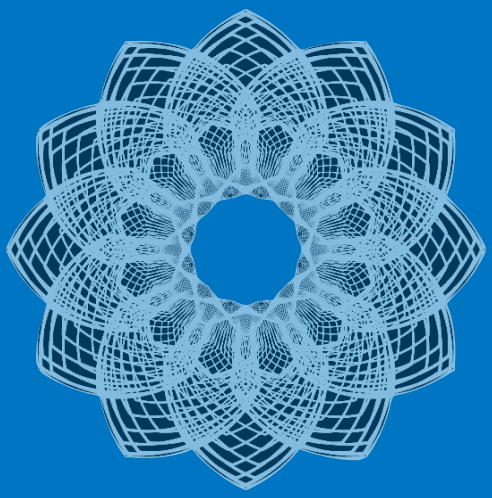
Syllabus for
M.Sc. Computer Science

(As per UGC CBCS w.e.f 2016-17)



Department of Mathematics
Osmania University
Hyderabad
Telangana

With effect from the academic year 2016-2017

Department of Mathematics

Osmania University

M.Sc. [Computer Science]

Course under Choice Based Credit System

SEMESTER – I

Paper	Code	Paper Title	HpW	Marks	Credits
I	CS101T	Advanced Java Programming	4	20+80 =100	4
II	CS102T	Operating Systems	4	20+80 =100	4
III	CS103T	Software Engineering	4	20+80 =100	4
IV	CS104T	Discrete Mathematics	4	20+80 =100	4
V	CS105P	Advanced Java Lab	6	75	3
VI	CS106P	Operating Systems Lab	6	75	3
VII	CS107P	Software Engineering Lab	4	50	2
		Total	**32**	**600**	**24**

SEMESTER – II

Paper	Code	Paper Title	HpW	Marks	Credits
I	CS201T	Programming in Python	4	20+80 =100	4
II	CS202T	Computer Networks	4	20+80 =100	4
III	CS203T	Design and Analysis of Algorithms	4	20+80 =100	4
IV	CS204T	Automata Theory	4	20+80 =100	4
V	CS205P	Python Lab	6	75	3
VI	CS206P	Computer Networks Lab	6	75	3
VII	CS207P	Design and Analysis of Algorithms Lab	4	50	2
		Total	**32**	**600**	**24**

With effect from the academic year 2017-2018

Department of Mathematics
Osmania University
M.Sc. [Computer Science]
Course under Choice Based Credit System

SEMESTER – III

Paper	Code	Paper Title	HpW	Marks	Credits
I	CS301T	Programming in C#	4	20+80 =100	4
II	CS302T	Compiler Design	4	20+80 =100	4
III	Elective CS303T(A) CS303T(B)	 Network Security Big Data Analytics	4	20+80 =100	4
IV	Elective CS304T(A) CS304T(B)	 Object Oriented Analysis and Design Data Mining	4	20+80 =100	4
V	CS305P	C# Lab	6	75	3
VI	CS306P	Compiler Design Lab	6	75	3
VII	Elective CS307P(A) CS307P(B)	 Network Security Lab Big Data Analytics Lab	4	50	2
		Total	**32**	**600**	**24**

SEMESTER – IV

Paper	Code	Paper Title	HpW	Marks	Credits
I	CS401T	Computer Organization	4	20+80 =100	4
II	CS402T	Cloud Computing	4	20+80 =100	4
III	Elective CS403T(A) CS403T(B)	 Mobile Computing Distributed Systems	4	20+80 =100	4
IV	Elective CS404T(A) CS404T(B)	 Artificial Intelligence Internet of Things	4	20+80 =100	4
V	CS405P	Project Work	16	200	8
		Total	**32**	**600**	**24**

CS101T Advanced Java Programming

Theory: 4 Hours/Week **Credits**: 4

Unit – I

Review of Core Java: Class Object, Object Oriented Concepts with respect to Java, Interfaces, Packages and Exception Handling, Applets, Overview of Collection Framework (No question to be set from above topics).
AWT: Introduction, AWT Class Hierarchy, Creating Container, Adding Components, Layout, Using Panel, Text Field, TextArea, List, Checkbox, Check Box Group, Choice, Event Handling, Dialog Boxes, ScrollBar, Menu.
Swing: Containment Hierarchy, Adding Components, JTextField, JPasswordField, JTable, JComboBox, JProgressBar, JList, JTree, JColorChooser, Dialogs.
Remote Method Invocation (RMI): Introduction, Remote Method Invocation, Java RMI Interfaces and Classes, an Application, Compiling the Program, Generating Stub Classes, Running the Program, Callback with an Application.

Unit – II

Servlet: Server-Side Java, Servlet Alternatives, Servlet Strengths, Servlet Architecture, Servlet Life Cycle, GenericServlet, HttpServlet, Servlet Example, Passing Parameters to Servlets, Retrieving Parameters, Cookies, Filters.
Java Server Pages (JSP): Introduction, JSP Engines, How JSP Works, JSP and Servlet, Anatomy of a JSP Page, JSP Syntax, JSP Components, Beans, Session Tracking, Users Passing Control and Data between Pages, Sharing Session and Application Data.

Unit – III

Java Database Connectivity (JDBC): Introduction, JDBC Drivers, JDBC Architecture, JDBC Classes and Interfaces, Loading a Driver, Making a Connection, Execute SQL Statement, SQL Statements, Retrieving Result, Getting Database Information, Scrollable and Updatable Resultset, Result Set Metadata.
Hibernate: Introduction, Writing POJO Class, Creating a Table, Writing a Hibernate Application, Compiling and Running Application, Book Application Using Annotation, Object Life Cycle, HQL, Using Native SQL Query, Named Queries, Generating DDL, Generator Class, Hibernate Tools.

Unit – IV

Java Naming and Directory Interface (JNDI): Naming Concepts, Directory Concepts, Java Naming and Directory Interface, Specifying JNDI Properties, Name Servers, Naming Operations, Working with Directory.
Overview of J2EE, Introduction to JavaBeans, Advantages of JavaBeans, Properties of JavaBeans with examples, JavaBeans API, Enterprise JavaBeans (EJB), Applications using Session Beans and Entity Beans, Introduction to Struts Framework.
Java Server Faces (JSF): Introduction, Simple Application, Request Processing Life-Cycle, Tracing Phases, Managed Bean, Basic JSF Tags, Expression Language, Event Handling with Example, Page Navigation.

Text Uttam K. Roy, *Advanced Java programming*

References

1. Herbertt Schildt, *Java Complete Reference*
2. Sharanam Shah, Vaishali Shah, *Java EE 7 for Beginners*
3. Cay S. Horstmans, Gray Coronell, *Core Java Vol. II – Advanced Features*

CS102T Operating Systems

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction: Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection- Security, Kernel Data Structures, Computing Environments, Open-Source Operating Systems.

Operating-System Structures: Operating-System Services, User Interface for Operating-System, System Calls, Types of System Calls, Operating-System Design and Implementation, Operating-System Structure, Operating-System Debugging.

Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Examples of IPC Systems, Communication in Client–Server Systems.

Threads: Overview, Multithreading Models, Threading Issues.

Process Synchronization: Concept, Critical-Section Problem, Peterson's Solution, Synchronization, Classic Problems of Synchronization, Semaphores, Monitors.

Unit – II

CPU Scheduling: Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Real-Time CPU Scheduling, Algorithm Evaluation.

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

Unit – III

Memory Management: Main Memory, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory: Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files.

Mass-Storage Structure: Overview, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure, Stable-Storage Implementation.

Unit – IV

File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, Protection. File-System Structure and Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Recovery, Network File System.

Protection and Security: Goals of Protection, Principles of Protection, Domain of Protection,

Access Matrix, Access Control, Revocation of Access Rights, The Security Problem, Program Threats, System and Network Threats, Cryptography as a Security Tool, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks, Computer-Security Classifications. Case Study: Windows 7 and Linux System.

Text Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts* (9e)

References

1. Thomas W. Doeppner, *Operating systems in depth*
2. Andrew S. Tanenbaum, *Modern Operating Systems*
3. William Stallings, *Operating Systems – Internals and Design Principles*
4. Dhananjay M. Dhandhere, *Operating Systems-A Concept Based Approach*

CS103T Software Engineering

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Software Engineering: The Nature of Software, Changing Nature of Software, Defining the Discipline, Software Process, Software Engineering Practice. The Software Process: A Generic Process Model, Defining a Framework Activity, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, Unified Process, Personal and Team Process Models. Defining Agility, Agile Process, Extreme Programming, Psychology of Software Engineering, Software Team Structures, Software Engineering Using the Cloud, Global Teams.

Unit – II

Requirements: Core Principles of Modeling, Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Analysis Model, Requirements Analysis, UML Models That Supplement the Use Case, Identifying Analysis Classes, Specifying Attributes, Defining Operations, Class-Responsibility-Collaborator Modeling, Associations and Dependencies, Analysis Packages.
Design Concepts: Design within the Context of SE, Design Process, Design Concepts, Design Model, Software Architecture, Architectural Styles, Architectural Considerations, Architectural Design, Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Based Development, User Interface Design Rules.

Unit – III

Quality Management: Quality, Software Quality, Software Quality Dilemma, Achieving Software Quality, Defect Amplification and Removal, Reviews, Informal Reviews, Formal Technical Reviews, Elements of Software Quality Assurance, SQA Tasks, Goals, and Metrics, Software Reliability, A Strategic Approach to Software Testing, Test Validation Testing, System Testing, Debugging, Software Testing Fundamentals, White-Box Testing, Black-Box Testing, Path Testing, Control Structure Testing, Object-Oriented Testing Strategies& Methods, Security Engineering Analysis, Security Assurance, Security Risk Analysis.

Unit – IV

Software Configuration Management, SCM Process, Product Metrics for Requirements Model, Design Model, Source Code, Testing and Maintenance. Managing Software Projects: The Project Management Spectrum, W^5HH Principle, Metrics in the Process and Project Domains, Software Measurement, Metrics for Software Quality, Integrating Metrics within the Software Process, Software Project Estimation, Decomposition Techniques, Project Scheduling – basics, scheduling, Software Risks, Risk Mitigation, Monitoring, and Management, Software Maintenance, Software Reengineering, Reverse Engineering, Forward Engineering.

Text Roger S Pressman, B R Maxim, *Software Engineering – A Practitioner's Approach* (8e)

References

1. Ian Sommerville, *Software Engineering*
2. Hans Van Vliet, *Software Engineering*
3. D. Bell, *Software Engineering for Students*
4. K.K. Aggarwal, Y. Singh, *Software Engineering*
5. R. Mall, *Fundamentals of Software Engineering*
6. Pankaj Jalote, An Integrated *Approach to Software Engineering*

CS104T Discrete Mathematics

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Mathematical Logic: propositional logic, propositional equivalences, predicates & quantifiers, rule of inference, direct proofs, proof by contraposition, proof by contradiction.
Boolean Algebra: Boolean functions and its representation, logic gates, minimizations of circuits by using Boolean identities and K-map.

Unit – II

Basic Structures: Sets representations, set operations, functions, sequences and summations.
Division algorithm, modular arithmetic, solving congruences, applications of congruences.
Recursion: Proofs by mathematical induction, recursive definitions, structural induction, generalized induction, recursive algorithms.

Unit – III

Counting: Basic counting principle, inclusion-exclusion for two-sets, pigeonhole principle, permutations and combinations, Binomial coefficient and identities, generalized permutations and combinations.
Recurrence Relations: introduction, solving linear recurrence relations, generating functions, principle of inclusion-exclusion, applications of inclusion-exclusion.
Relations: relations and their properties, representing relations, closures of relations, equivalence relations, partial orderings.

Unit – IV

Graphs: Graphs definitions, graph terminology, types of graphs, representing graphs, graph isomorphism, connectivity of graphs, Euler and Hamilton paths and circuits, Dijkstra's algorithm to find shortest path, planar graphs–Euler's formula and its applications, graph coloring and its applications
Trees: Trees definitions–properties of trees, applications of trees–BST, Haffman Coding, tree traversals: pre-order, in-order, post-order, prefix, infix, postfix notations, spanning tress–DFS, BFS, Prim's, Kruskal's algorithms.

Text Kenneth H. Rosen, *Discrete Mathematics and its Applications (7e)*

References

1. Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics*
2. Stein, Drysdale, Bogart, *Discrete Mathematics for Computer Scientists*
3. J.P. Tremblay, R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*
4. Joe L. Mott, Abraham Kandel, Theoder P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*

CS105P Advanced Java Lab

Practical: 6 Hours/Week **Credits:** 3

1. Create GUI to present a set of choices for a user to select stationary products and display the price of Product after selection from the list.
2. Create GUI to demonstrate typical Editable Table which describing Employee for a software company.
3. Create GUI to demonstrate swing components using student registration form.
4. Create a Remote Object for simple arithmetic operators. Use AWT/SWING to create user interface.
5. Write an RMI application using call back mechanism
6. Develop Servlet Question-Answer Application using HttpServletRequest and HttpServletRequest interfaces.
7. Develop Servlet application to accept HTNO of a student from client and display the memorandum of marks from the server
8. JSP Programs
   a. Create a JSP page that prints temperature conversion (from Celsius to Fahrenheit) chart
   b. Create a JSP page to print current date and time
   c. Create a JSP page to print number of times page is referred after the page is loaded.
9. Write a simple JSP application to demonstrate the use of implicit object (at least 5).
10. Develop a Hibernate application to Store Feedback of Website Visitors in MySQL Database.
11. Develop a JSP Application to accept Registration Details from the user and store database table.
12. Develop a JSP Application to Authenticate User Login as per the Registration Details. If Login Success then forward User to Index Page otherwise show Login failure Message.
13. Develop a web Application to add items in the inventory using JSF.
14. Write EJB applications using stateless session beans and state-full session beans.
15. Develop a Room Reservation System Application using Entity Beans.
16. Create Three-tire application using Servlets, JSP, EJB.

CS106P Operating Systems Lab

Practical: 6 Hours/Week **Credits:** 3

1. Write shell programs using 'case', 'then' and 'if' & 'else' statements.
2. Write shell programs using while, do-while and for loop statements.
3. Write a program to create a child process using fork(), exec() system calls and use other system calls.
4. Write a program to convert upper case to lower case letters of a given ASCII file.
5. Write a program to program to search the given pattern in a file.
6. Write a program to implementation of Signals in UNIX.
7. Write a program to simulate UNIX commands like ls, grep, cp.
8. Write a program to demonstrate FCFS and SJF process schedules on the given data.
9. Write a program to demonstrate CPU Priority and Round Robin Scheduling on the given burst time and arrival times.
10. Write a program to simulate Inter Process Communication using pipes.
11. Write a program to implementing Producer and Consumer problem using Semaphores.
12. Write a program to simulate Bankers Algorithm for Dead Lock Avoidance
13. Write a program to simulate Bankers Algorithm Dead Lock Prevention.
14. Write a program to simulate Paging Techniques of memory management.
15. Write a program to simulate FIFO, LRU, LFU Page replacement algorithms.
16. Write a program to simulate Sequential, Indexed, and Linked file allocation strategies.

Note:

- Recommended to use Open Source Software like Fedora, Ubuntu, CentOS, etc...
- Recommended to write programs using C/C++ on Linux systems.

CS107P Software Engineering Lab

Practical: 4 Hours/Week **Credits:** 2

1. Study of case tool

   Requirements

2. Implementation of requirements engineering activities such as elicitation, validation, management using case tools

   Analysis and Design

3. Implementation of Analysis and design using case tools
4. Study and usage of software project management tools such cost estimates and scheduling
5. Documentation generators –Study and practice of Documentation generators
6. Data Modeling using automated tools
7. Practice reverse engineering and re-engineering using tools
8. Exposure towards test plan generators, test case generators, test coverage and software metrics.
9. Meta modeling and software life cycle management.

   Case Studies:

10. Structure charts, Data Flow Diagrams, Decision tables and ER diagrams for
    a. Banking System
    b. Railway Reservation System
    c. Hotel management system
    d. Inventory Control System
    e. Library management system

Note:

- To draw dataflow diagrams, structured charts etc… using Microsoft Visio Software, SmartDraw, etc…
- To draw UML diagrams using Rational Rose Software, StarUML, etc...
- The teacher should define the boundaries for the above case study problems and make the practice of problems mentioned.

CS201T Programming in Python

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction to Python Programming: How a Program Works, Using Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations (Operators. Type conversions, Expressions), More about Data Output.
Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables.
Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops.

Unit – II

Functions: Introduction, Defining and Calling a Void Function, Designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions-Generating Random Numbers, Writing Our Own Value-Returning Functions, The math Module, Storing Functions in Modules.
File and Exceptions: Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions.

Unit – III

Lists and Tuples: Sequences, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples.
Strings: Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings.
Dictionaries and Sets: Dictionaries, Sets, Serializing Objects.
Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

Unit – IV

Object-Oriented Programming: Procedural and Object-Oriented Programming, Classes, Working with Instances, Techniques for Designing Classes, Inheritance, Polymorphism.
GUI Programming: Graphical User Interfaces, Using the tkinter Module, Display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels as Output Fields, Radio Buttons, Check Buttons.

Text

Tony Gaddis, *Starting Out With Python (3e)*

References

1. Kenneth A. Lambert, *Fundamentals of Python*
2. Clinton W. Brownley, *Foundations for Analytics with Python*
3. James Payne, *Beginning Python using Python 2.6 and Python 3*
4. Charles Dierach, *Introduction to Computer Science using Python*
5. Paul Gries, *Practical Programming: An Introduction to Computer Science using Python 3*

CS202T Computer Networks

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Computer Networks Fundamentals: Overview, Network Hardware, Network Software, Reference models– OSI Model, TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference Model, Example Networks, Network Standardization.

Physical Layer: Guided Transmission Media, Wireless Transmission, Multiplexing, Switching.

Data Link Layer: Design Issues, Error Detection and Correction, Data Link Layer Protocols, Sliding Window Protocol

Unit – II

Multiple Access Sublayer: ALOHA, CSMA, Collision Free Protocols, Ethernet, Wireless LAN-802.11, Data Link Layer Switching–Repeaters, Hubs, Bridges, Switches, Routers, Gateways.

Network Layer: Design Issues, Routing Algorithms – Shortest path, Flooding, Distance Vector Routing, Link state Routing, Hierarchical, Broadcast Routing, Multicast Routing; Congestion Control Algorithms.

Unit – III

Internetworking: Tunneling, Internetwork Routing, Fragmentation, IPv4 Vs IPv6Protocol, IP Addresses, CIDR, Internet Control Protocols–IMCP, ARP, RARP, DHCP.

Transport Layer: Services provided to the upper layers, Transport Protocols, Overview of Congestion Control.

Unit – IV

The Internet Transport Protocols: Introduction to UDP&RPC, Real Time Transport Protocols, The Internet Transport Protocols–TCP, TCP Service Model, TCP protocol, TCP Segment Header, TCP Connection Establishment, TCP Connection Release, Modeling TCP Connection Management, TCP Sliding Window, TCP Time Management, TCP Congestion Control.

Application Layer: DNS, TELNET, E-Mail, FTP, HTTP, SSH, Overview of WWW.

Text Andrew S. Tanenbaum, David J Wetherall, *Computer Networks (5e)*

References

1. S S Shinde, *Computer Network*
2. William Stallings, *Data and Computer Communications*
3. Behrouz A. Forouzan, *Data Communication and Networking*
4. James F Kurose, Keith W Ross, *Computer Networking – A Top-Down Approach*
5. Behrouz A Forouzan, Firouz Mosharraf, *Computer Networks A Top-Down Approach*

CS203T Design and Analysis of Algorithms

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction: Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types. Fundamentals of the Analysis of Algorithm: The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive & Recursive Algorithms. Brute Force Search: Selection Sort, Bubble Sort, Sequential Search, Brute-Force String Matching, Exhaustive Search, Depth-First Search, Breadth-First Search.

Unit – II

Decrease–&–Conquer: Insertion Sort, Topological Sorting, Binary Search, Interpolation Search
Divide-and-Conquer: Merge Sort, Quick Sort, Multiplication of Large Integers, Strassen's Matrix Multiplication, Transform-and-Conquer: Presorting, Balanced Search Trees, Heaps and Heap Sort, Problem Reduction. Space and Time Trade-Offs: Hashing, B-Trees.

Unit – III

Dynamic Programming: Knapsack Problem, Optimal Binary Search Trees, Warshall's and Floyd's Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes. Iterative Improvement: Simplex Method, Maximum-Flow Problem.

Unit – IV

Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete Problems. Backtracking: n-Queens Problem, Hamiltonian Circuit Problem, Subset-Sum Problem, Branch-and-Bound: Assignment Problem, Knapsack Problem, Traveling Salesman Problem, Approximation Algorithms for the Knapsack Problem.

Text Anany Levitin, *Introduction to the Design and Analysis of Algorithms (3e)*

References

1. Richard Neapolitan, *Foundations of Algorithms*
2. Thomas H. Cormen, *Introduction to Algorithms*
3. E. Horowitz, S. Sahni, *Fundamentals of Computer Algorithms*
4. A.V. Aho, J.V. Hopcroft, J.D. Ullmann, *The Design and Analysis of Computer Algorithms*
5. Donald E Knuth, *The Art of Programming_ Volumes-1, 2, 3, 4*

CS204T Automata Theory

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Fundamentals – alphabets, strings, languages, problems, graphs, trees, Finite State Systems, definitions, Finite Automaton model, acceptance of strings, and languages, Deterministic finite automaton and Nondeterministic finite automaton, transition diagrams, transition tables, proliferation trees and language recognizers, equivalence of DFA's and NFA's.

Finite Automata with ε-moves, significance, acceptance of languages, ε-closure, Equivalence of NFA's with and without ε-moves, Minimization of finite automata, Two-way finite automata, Finite Automata with output–Moore and Melay machines.

Unit – II

Regular Languages: regular sets, regular expressions, identity rules, constructing finite automata for a given regular expressions, conversion of finite automata to regular expressions. Pumping lemma of regular sets and its applications, closure properties of regular sets.

Grammar Formalism: Regular grammars–right linear and left linear grammars, equivalence between regular linear grammar and finite automata, inter conversion, Context free grammar, derivation trees, sentential forms, right most and leftmost derivation of strings, ambiguity.

Unit – III

Context Free Grammars: Simplification of Context Free Grammars, Chomsky normal form, Greiback normal form, Pumping lemma for context free languages and its applications, closure of properties of CFL (proofs omitted).

Push Down Automata: PDA definition, model, acceptance of CFL, acceptance by final state and acceptance by empty state and its equivalence. Equivalence of PDA's and CFL's, inter-conversion. (Proofs not required).

Unit – IV

Membership Algorithm (CYK Algorithm) for Context Free Grammars.

Turing Machine: TM definition, model, design of TM, computable functions, unrestricted grammars, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs omitted). Linear bounded automata and Context sensitive language.

Computability Theory: Chomsky hierarchy of languages, Introduction to DCFL, DPDA, LR(0) grammar, decidability and undecidable problems. Definitions of P and NP problems, NP complete and NP hard problems.

Text J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*

References

1. John C. Martin, *Introduction to Languages and the Theory of Computation*
2. Mishra, Chandrashekaran, *Theory of Computer Science*
3. Perter Linz, *An Introduction to Formal Languages and Automata*
4. ZviKohav, Niraj K Jha, *Switching and Finite Automata Theory*

CS205P Python Lab

Practical: 6 Hours/Week **Credits:** 3

1. Write a program that displays the following information: Your name, Full address, Mobile number, College name, Course subjects.
2. Write a program to find the largest three integers using if-else and conditional operator.
3. Write a program that asks the user to enter a series of positive numbers (The user should enter a negative number to signal the end of the series) and the program should display the numbers in order and their sum.
4. Write a program to find the product of two matrices [A]mxp and [B]pxr
5. Write recursive and non-recursive functions for the following:
    a. To find GCD of two integers.
    b. To find the factorial of positive integer
    c. To print Fibonacci Sequence up to given number n
6. Write a program to display two random numbers that are to be added, such as: 247 + 129, the program should allow the student to enter the answer. If the answer is correct, a message of congratulations should be displayed. If the answer is incorrect, a message showing the correct answer should be displayed.
7. Write recursive and non-recursive functions to display prime number from 2 to n.
8. Write a program that writes a series of random numbers to a file from 1 to n and display.
9. Write a program to create file, write the content and display the contents of the file with each line preceded with a line number (start with 1) followed by a colon.
10. In a program, write a function that accepts two arguments: a list and a number n. The function displays all of the numbers in the list that are greater than the number n.
11. Write a program with a function that accepts a string as an argument and returns the no. of vowels that the string contains. Another function to return number of consonants.
12. Write a program that opens a specified text file and then displays a list of all the unique words found in the file. (Store each word as an element of a set.)
13. Write a program to analyze the contents of two text files using set operations.
14. Write a program to implement the inheritance and dynamic polymorphism.
15. Write a GUI program that converts Celsius temperatures to Fahrenheit temperatures.
16. Write a GUI program that displays your details when a button is clicked.

Note: Handle the exceptions raised from file operations.

CS206P Computer Networks Lab

Practical: 6 Hours/Week **Credits:** 3

1. Program to identify the category of the IP address for the given IP address
2. Program to implement sliding window protocol
3. Program for Socket pair system call usage in IPC
4. Program for Socket options using signals
5. Program to implement Echo concurrent Stream Server
6. Program to implement Echo concurrent stream client
7. Program to implement Listener and Talker
8. Program to implement TCP time service
9. Program to implement UDP time service
10. Program to implement Ping service
11. Program to implement Route tracing program
12. Program to implement File Transfer Protocol
13. Program to implement any Shortest path routing Algorithm
14. Program to implement Distance Vector Routing Implementation
15. Program to implement ICMP Error Message simulations
16. Program to implement Reverse Address Resolution Protocol

Note:

- Recommended to use Open Source Software like Fedora, Ubuntu, CentOS, etc...
- Strictly recommended to write above programs using C language on Linux systems.

CS207P Design and Analysis of Algorithms Lab

Practical: 4 Hours/Week **Credits:** 2

1. Write a program recursive and non-recursive function for the following:
   a) Factorial of an integer b) GCD of two integers c) Fibonacci Sequence
2. Write a program for sorting the given list using Insertion Sort, Topological Sort.
3. Write a program for sorting the given list using Selection Sort, BubbleSort.
4. Write a program for sorting the given list using Merge Sort.
5. Write a program for sorting the given list using Quick Sort.
6. Write a program for sorting the given list using Heap Sort.
7. Write a program to find the given number in a list using Sequential Search, Binary Search.
8. Write a program to find product of two matrices [A]mxp and [B]pxr
9. Write a program to create AVL tree.
10. Write a program to create B-tree.
11. Write a program to find the Euler circuit and the Hamiltonian circuit for a weighted graph.
12. Write a program to find the shortest path in a weighted graph using Dijkstra's Algorithm.
13. Write a program to solve travelling sales man problem.
14. Write a program to solve knapsack problem.
15. Write a program to find the minimum spanning tree for a weighted graph using Kruskal's Algorithm.
16. Write a program to find the minimum spanning tree for a weighted graph using Prim's Algorithm.

Note:

- Recommended to analyze all the above problems with respect to Time Complexity.
- Recommended to write above programs using C/C++/Java/Python language on Linux systems.

CS301T Programming in C#

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction: architecture and components of .NET framework 4.5, managed code, intermediate language, meta data and JIT compiler, common language runtime, automatic memory management, private and shared assemblies, exploring Visual Studio .NET IDE. Introduction to C#: identifiers, keywords, data types, variables, constants, operators, precedence, associativity, type conversion, decision and loop statements, generics, enumerations, namespaces. Object Oriented Programming: encapsulation, inheritance, polymorphism, abstraction, interfaces.

Unit – II

Introduction to Windows Programming: creating windows forms, windows controls, menus and dialogue boxes, overview of xml. Window programming vs. Window presentation foundation, main features of WPF 4.5, WPF 4.5 architecture, WPF 4.5 class hierarchy, types of WPF applications, WPF 4.5 designer interface, Using XAML in WPF 4.5 applications, WPF properties, WPF events, working with dialog boxes in WPF application, compiling and running WPF 4.5 applications, WPF 4.5 controls, resources, styles, templates, commands.

Unit – III

Introduction to SQL, architecture of ADO.NET, creating a connection to a database, OLEDB database, using OLEDB adapter for excel file, ODBC data source, ADO.NET commands, data adapters, creating data view, data reader, stored procedures.

Introduction to ASP.NET: ASP.NET life cycle, exploring ASP.NET 4.5 web application, creating a sample ASP.NET 4.5 website, application structure and state, global.asax application file, web forms – standard controls, validation controls, master pages, web services.

Unit – IV

Introduction to ASP.NET MVC: ASP.NET MVC project templates, understanding the structure of an ASP.NET MVC project, naming conventions, creating views, defining controllers, defining a data model, execution flow of MVC.

Deploying Websites and Services, deploying to internet information server, preparing a web application using IIS manager, web deploying IIS.

Text Kogent Learning Solutions Inc., *.NET 4.5 Programming – Black Book (dreamtech)*

References

1. Joseph Albahari, Ben Albahari, *C# 6.0 in a Nutshell*
2. Christian Nagel, *Professional C# 6 and .NET Core 1.0*
3. Andrew Troelsen, Philip Japikse, *C# 6.0 and the .NET 4.*6
4. Paul Deitel, Harvey Deitel, *Visual C# 2012 – How to Program*
5. Bill Sempf, Chuck Sphar, Stephen R Davis, *C# 5.0 all-in-one for Dummies*
6. Mark J Price, *C# 6 and .NET Core 1.0 – Modern Cross-platform Development*
7. Benjamin Perkins, Jacob Vibe Hammer, Jon D. Reid, *Beginning C# 6 Programming with Visual Studio 2015*

CS302T Compiler Design

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction: language processors, phases of a compiler, a model for a compiler front end, syntax-directed translation, parsing, a translator for simple expressions, Lexical Analysis: role of lexical analyzer, input buffering, specification of tokens, Lex lexical analyzer generator, data structures in compilation. Top-Down Parsing: Introduction, Context free grammars, writing a grammar, recursive-descent parsing, LL(1) grammars, predictive parsing, preprocessing steps required for predictive parsing.

Unit – II

Bottom-Up Parsing: shift reduce parsing, SLR parsing, CLR parsing and LALR parsing, error recovery in parsing, handling ambiguous grammar, parser generator – YACC. Semantic Analysis: syntax-directed definitions, evaluation order for SDD's, application of SDT.

Unit – III

Intermediate-Code Generation: syntax trees, three-address code, types and declarations, translation of expressions, type checking. Runtime Environment: storage organization, stack allocation of space, heap management, storage allocation for arrays, strings and records, introduction to garbage collection and trace-based collection,

Unit – IV

Code Generation: issues in the design of code generator, target language, addresses in the target code, basic blocks and flow graphs, optimization of basic blocks, peephole optimization, register allocation and assignment.
Code Optimization: principal sources of optimization, data flow analysis, constant propagation, partial-redundancy elimination, loops in flow graphs.

Text A. V. Aho, Monica S. Lam, Ravi Sethi, J. D. Ullman, *Compilers Principles, Techniques, & Tools, (2e)*

References

1. Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, *Modern Compiler Design*
2. Kenneth C. Louden, *Compiler Construction Principles and Practice*
3. Thomas w. Parsons, *Introduction to Compiler Construction*
4. Andrew N. Appel, *Modern Compiler Implementation in C*
5. John R. Levin, Tony Mason, Doug Brown, *LEX & YACC*
6. Cooper, Linda, *Engineering a Compiler*

CS303T(A) Network Security

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Overview of Network Security: Computer Security Concepts, the OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, a Model for Network Security. Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography.
Block Ciphers and the Data Encryption Standard: Traditional Block Cipher Structure, the Data Encryption Standard (DES), A DES Example, Strength of DES. Block Cipher Operation: Double DES, Triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode.

Unit – II

Advanced Encryption Standard (AES): The Origins AES, AES Structure, AES Round Functions, AES Key Expansion, an AES Example AES Implementation. Pseudorandom Number Generation and Stream Ciphers: Principles of Pseudorandom Number Generation, Pseudorandom Number Generators, Pseudorandom Number Generation using Block Cipher, StreamCiphers-RC4. Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm. Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption and Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Diffie-Hellman Key Exchange.

Unit – III

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Secure Hash Algorithm (SHA) & MD5 Algorithm. Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MACs, MACs Based on Hash Functions: HMAC, MACs Based on Block Ciphers: DAA and CMAC. Digital Signatures: Digital Signatures, NIST Digital Signatures Algorithm.

Unit – IV

Transport-Level Security: Web Security Considerations, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS, Secure Shell (SSH), E-Mail Security: Pretty Good Privacy, S/MIME. IP Security: IP Security Overview, IP Security Architecture, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange. Intruders, Virus and Firewalls: Intruders, Intrusion Detection, Password Management, Virus and Related Threats, Countermeasures, Firewall Design Principles, Types of Firewalls.

Text William Stallings, *Cryptography and Network Security – Principles and Practice (6e)*

References

1. Zhenfu Cao, *New Directions of Modern Cryptography*
2. Douglas R. Stinson, *Cryptography Theory and Practices*
3. Tom St Denis, Simon Johnson, *Cryptography for Developers*
4. Joseph Migga Kizza, *A Guide to Computer Network Security*
5. A. Menezes, P. Van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*
6. Henk C.A. van Tilborg, Sushil Jajodia, *Encyclopedia of Cryptography and Security*
7. Keith M. Martin, *Everyday Cryptography–Fundamental Principles and Applications*
8. Chwan-Hwa Wu, J. David Irwin, *Introduction to Computer Networks and Cyber Security*
9. Saiful Azad, Al-Sakib Khan Pathan, *Practical Cryptography-Algorithms and Implementations Using C++*

CS303T(B) Big Data Analytics

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Overview of Big Data: What is Big Data? Evolution of Big Data, Structuring Big Data, Elements of Big Data, Big Data Analytics. Exploring the Use of Big Data in Business Context: Use of Big Data in Social Networking, Use of Big Data in Preventing Fraudulent Activities, Use of Big Data in Detecting Fraudulent Activities in Insurance Sector, Use of Big Data in Retail Industry. Introducing Technologies for Handling Big Data: Distributed and Parallel Computing for Big Data, Introducing Hadoop. Understanding Hadoop Ecosystem: Hadoop Ecosystem, HDFS, MapReduce, Hadoop YARN, HBase, Hive, Pig and Pig Latin, Sqoop, ZooKeeper, Flume, Oozie.

Unit – II

Understanding MapReduce Fundamentals and HBase: The MapReduce Framework, Techniques to Optimize MapReduce Jobs, Role of HBase in Big Data Processing. Exploring the Big Data Stack, Virtualization and Big Data, Virtualization Approaches. Storing Data in Databases and Data Warehouses: RDBMS and Big Data, Non-Relational Database, Integrating Big Data with Traditional Data Warehouses, Big Data Analysis and Data Warehouse, Changing Deployment Models in Big Data Era. Processing Your Data with MapReduce: Developing Simple MapReduce Application, Points to Consider while Designing MapReduce.

Customizing MapReduce Execution: Controlling MapReduce Execution with InputFormat, Reading Data with Custom RecordReader, Organizing Output Data with OutputFormats, Customizing Data with RecordWriter, Optimizing MapReduce Execution with Combiner, Implementing a MapReduce Program for Sorting Text Data.

Unit – III

Understanding Hadoop YARN Architecture: Introduction YARN, Advantages of YARN, YARN Architecture, Working of YARN. Exploring Hive: Introducing Hive, Getting Started with Hive, Hive Services, Data Types in Hive, Built-In Functions in Hive, Hive DDL, Data Manipulation in Hive, Data Retrieval Queries, Using JOINS in Hive. Analyzing Data with Pig: Introducing Pig, Running Pig, Getting Started with Pig Latin, Working with Operators in Pig, Working with Functions in Pig, Debugging Pig, Error Handling in Pig.

Unit – IV

Using Oozie: Introducing Oozie, Installing and Configuring Oozie, Understanding the Oozie Workflow, Simple Application. NoSQL Data Management: Introduction to NoSQL, Types of NoSQL Data Models, Schema-Less Databases, Materialized Views, Distributed Models, Sharding, MapReduce Partitioning and Combining, Composing MapReduce Calculations. Understanding Analytics and Big Data: Comparing Reporting and Analysis, Types of Analytics, Developing an Analytic Team. Analytical Approaches and Tools to Analyze Data: Analytical Approaches, History of Analytical Tools, Introducing Analytical Tools, Comparing Various Analytical Tools.

Text DT Editorial Services, *Big Data – Black Book (dreamtech)*

References

1. Radha S, M. Vijayalakshmi, *Big Data Analytics*
2. Arshdeep B and Vijay M, *Big Data Science & Analytics – A Hands-On Approach*.
3. Frank Ohlhorst, *Big Data Fundamentals – Concepts, Drivers, Techniques*
4. Kuan-Ching Li, H Jiang, L T Yang, A Cuzzocrea, *Big Data Algorithms, Analysis and Applications*
5. Tom White, *Hadoop: The Definitive Guide*
6. Shiva Achari, *Hadoop Essentials*
7. Alex Holmes, *Hadoop in Practice*

CS304T(A) Object Oriented Analysis and Design

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Complexity: the structure of complex systems, the inherent complexity of software, the five attributes of a complex system, organized and disorganized complexity, bringing order to chaos, on designing complex systems The object model: the evolution of the object model, foundations of the object model, elements of the object model, applying the object model.

Unit – II

Classes and Objects: the nature of an object, relationships among objects, the nature of a class, relationships among classes, the interplay of classes and objects, on building quality classes and objects. Classification: the importance of proper classification, identifying classes and objects, key abstractions and mechanisms. Notation: unified modeling language (UML), package diagrams, component diagrams, deployment diagrams.

Unit – III

Notation: use case diagrams, activity diagrams, class diagrams, sequence diagrams, interaction overview diagrams, composite structure diagrams, state machine diagrams, timing diagrams, object diagrams, communication diagrams. Process: first principles, the macro process: SDLC, the micro process: the analysis and design process. Pragmatics: management and planning, staffing, release management, reuse, quality assurance and metrics, documentation, tools, special topics, the benefits and risks of object-oriented development.

Unit – IV

System Architecture – Satellite-Based Navigation: inception, elaboration, construction, post-transition. Control System – Traffic Management: inception, elaboration, construction, post-transition, Web Application – Vacation Tracking System: inception, elaboration, construction, transition and post-transition,

Text Grady Booch, *Object-Oriented Analysis and Design with Applications*

References

1. Ali Bahrami, Object Oriented Systems Development
2. Grady Booch, *The Unified Modeling Language User Guide*
3. Berd Oestereich, *Developing software with UML – OOAD in practice*
4. Sarnath Ramnath, Brahma Dathan, *Object-Oriented Analysis and Design*
5. B. D. McLaughlin, Gary Pollics, David West, *Head First – Object Oriented Analysis & Design*

CS304T(B) Data Mining

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction to Data Mining: Why data mining? What is data mining? What kinds of data can be mined? What kinds of patterns can be mined? Which technologies are used? Major issues in data mining. Getting to Know Your Data: data objects and attribute types, basic statistical description of data, data visualization, measuring data similarity and dissimilarity. Data Processing: an overview, data cleaning, data integration, data reduction, data transformation and data discretization.

Unit – II

Data Warehousing and Online Analytical Processing (OLAP): basic concepts of data warehouse, data warehouse modelling–data cube and OLAP, data warehouse design and usage, data warehouse implementation, data generalization by attribute–oriented. Data Cube Technology: data cube computation preliminary concepts, data cube computation methods, processing advanced kinds of queries by exploring cube technology, multidimensional data analysis in cube space. Mining Frequent Patterns, Associations, and Correlations: basic concepts, frequent itemset mining methods, mining various kinds of association rules, from association mining to correlation analysis, constraint-based association mining.

Unit – III

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, classification by backpropagation, support vector machines, associative classification, lazy learners, other classification methods. Cluster Analysis: basic concepts of cluster analysis, partitioning methods, hierarchical methods, density-based methods, evaluation of clustering.

Unit – IV

Outlier Detection: outliers and outlier analysis, outlier detection methods, statistical approaches, proximity-based approaches, clustering-based approaches, classification-based approaches. Data Mining Trends and Research Frontiers: mining complex data types, other methodologies of data mining, data mining applications, data mining and society, data mining trends.

Text Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining – Concepts and Techniques (3e)*

References

1. Arun K Pujari, *Data Mining Techniques*
2. Paweł Cichosz, *Data Mining Algorithms: Explained Using R*
3. Nong Ye, *Data Mining – Theories, Algorithms, and Examples*
4. Pang-Ning Tan, M Steinbach, V Kumar, *Introduction to Data Mining*
5. S. Sumathi, S.N. Sivanandam, *Introduction to Data Mining and its Applications*

CS305P C# Lab

Practical: 6 Hours/Week **Credits:** 3

1. Using Windows Forms Application, collect the user details like First Name, Middle Name, Last Name, Address, Gender, User Photo, Course name, Course Timing with Submit and Clear buttons. Display the User Details in another form.
2. Create a Windows Forms Application containing different genre of movies and two buttons to display the genre liked by user and favorite genre selected by user.
3. Create a Windows Forms Application to demonstrate any 3 mouse and keyboard events each.
4. Create a Windows Forms Application to load an image into a picture box using file open dialogue box.
5. Create a WPF Application for auto complete text box with suitable windows icon.
6. Crate a WPF Application to validate user data like name (cannot be more than 10 characters), age (must be a number), phone number (xxxx-xxxx-xx).
7. Create a WPF Application for a bank customer. If user name and password is correct then display a welcome screen with options "balance enquiry", "deposit", "with draw" and "transfer money". Display Successful or failure messages accordingly.
8. Create a WPF calendar control with the following features.
   - Change the calendar background and foreground
   - Select a date
   - Blackout your vacation dates
   - Change the first day of week and highlight today's date
   - Select 2nd ,4th Saturdays and Sundays from the calendar
   - Display a selected date, month and year in separate textboxes
9. Develop a database application to store the details of an employee using ADO.NET using windows Forms.
10. Develop a database application using ADO.NET to insert, modify, update and delete operations on and display the result of the following queries.
    - Find all clerks who earn between 1000 and 2000
    - List of employees who are working in dept. 20 or getting salary is more than 3000
    - Only names of employees, which have TH or LL in their names
    - Display salary, dept. and names of employees whose name contains the letter 's' and ends with 's'
11. Create a web form to collect user details like Name (required field validator), password (Range Validator), confirm password (compare validator), email (regular expression validator), country name and phone number (range validator) with suitable validation controls. Display validation summary. If validation is successful display the user details.
12. Create a web application using master page for which displays course list on left pane. On selecting the course its content.
13. Create a web application using master page with a title "DATA CENTER". In which left pane contains "Home" and "Databases". Display the table data selected from "Databases" in a GridView control with add, edit, delete records and Page size options and also DetailsView control.
14. a. Create a web service application to call web service to calculate compound interest for the given data.
    b. Crate a database web service to display the list of employees in the decreasing order of their Job and salary.
15. a. Create a simple MVC application to display a message "MVC is Amezing…!".
    b. Crate a simple MVC application which accepts user name and password with suitable validation controls. If the details provided are correct then say Hello to the user.
16. Create movie database using MVC with the following specifications
    - List a set of movie database records
    - Create a new movie database record
    - Edit an existing movie database record

Note:

- Recommended to use the C/C++/Java (without using collection framework) language on Linux systems.

CS306P Compiler Design Lab

Practical: 6 Hours/Week **Credits:** 3

1. Write a program to design token separator for the given expression.
2. Write a program to implement a symbol table.
3. Write a program to develop a lexical analyzer to recognize a few patterns.
4. Write a program to develop a lexical analyzer using Lex tool.
5. Write a program to recognize a valid arithmetic expression using YACC.
6. Write a program to recognize a valid variable name using YACC.
7. Write a program to implement calculator using Lex and YACC.
8. Write a program for implementing type checking for given expression.
9. Write a program to convert the BNF rules into YACC.
10. Write a program to implement data flow and control flow analysis.
11. Write a program to implement stack storage allocation strategies.
12. Write a program to implement heap storage allocation strategies.
13. Write a program to construct a directed acyclic graph (DAG).
14. Write a program to implement the back end of the compiler.
15. Write a program to implement simple code optimization technique.
16. Write a program to implement simple code optimization technique using do-while.

Note:

- Recommended to use the C/LEX/YACC on Linux systems.

CS307P(A) Network Security Lab

Practical: 4 Hours/Week **Credits:** 2

17. Write a program to generate cipher text and recover the plaintext using
    a) Caesar-Cipher text algorithm b) Product Cipher
18. Write a program to generate cipher text and recover the plaintext using
    a) Play fair cipher b) Hill cipher
19. Write a program to generate random numbers using pseudo random number generation algorithm
20. Write a program to implement Poly-Alphabetic Cipher
21. Write a program to implement Transposition Cipher and Rail fence Technique
22. Write a program to implement DES Algorithm
23. Write a program to implement AES Algorithm
24. Write a program to implement RSA public key encryption algorithm
25. Write a program to implement Diffie- Helman Key Exchange Algorithm
26. Write a program to implement SHA-1 algorithm
27. Write a program to implement MD5 algorithm
28. Write a program to Implement the Signature Scheme Digital Signature Standard
29. Write a program to retrieve the data from the database and encrypt them using any encryption algorithm
30. Generation of public key and private keys
31. Write a program to write the contents in the file in encrypted manner and read them in decrypted manner using any algorithm
32. Write a program to implement FEM (Fast Exponentiation Method) for find congruential values used in RSA algorithms Ex: 577mod 45.

Note:

- Recommended to use the C/C++/Java (without using collection framework) language on Linux systems.

CS307P(B) Big Data Analytics Lab

Practical: 4 Hours/Week **Credits:** 2

1. Perform setting up and installing Hadoop in its three operating modes: stand alone, Pseudo distributed.
2. Perform some tasks by using web based tools of Hadoop system.
3. Implement the following file management tasks in Hadoop:
   - Adding file and directories
   - Creating file, Retrieving file and deleting files
4. Write a Map Reduce program for basic word count.
5. Write a Map Reduce program for sorting text data.
6. Write a Map Reduce program for analyzing student report.
7. Write a Map Reduce program for mining weather data.
8. Installing and running Hive, practice some Hive commands.
9. Using Hive; create, insert, update, alter, delete, and drop the tables
10. Using Hive; query the data from the data base tables.
11. Using Hive; create views, use functions, create indexes for the data base tables.
12. Installing and running Pig, practice some Pig commands.
13. Write Pig Latin scripts using eval functions to analyze your data.
14. Write Pig Latin scripts using math functions to analyze your data.
15. Write Pig Latin scripts using string functions to analyze your data.
16. Write simple scripts to understand the using NoSQL in Hadoop systems.

Note:

- Recommended to install Hadoop on Linux systems.
- Recommended to use the Java/Python/R language on Linux systems.

CS401T Computer Organization

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Digital Logic Circuits: Digital Computers, Logic Gates, Boolean Algebra, Map Simplification, Combinational Circuits, Flip-Flops, Sequential Circuits. Digital Components: Integrated Circuits, Decoders, Multiplexers, Registers, Shift Registers, Binary Counters, Memory Unit. Data Representation: Data Types, Complements, Fixed Point Representations, Floating Point Representation, Binary Codes, and Error Detection Codes.

Unit – II

Register Transfer and Microoperations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Microoperations, Logic Microoperations, and Shift Microoperations. Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory Reference Instructions, Input-Output and Interrupt, Design of Accumulator Logic. Programming the Basic Computer: Machine Language, Assembly Language, The Assembler Program Loops, Programming Arithmetic and Logic Operations, Subroutines, Input-Output Programming.

Unit – III

Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC). Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipelines, Instruction Pipelines and RISC Pipelines, Vector Processing. Computer Arithmetic: Addition and Subtraction, Multiplication Algorithms, Division Algorithms, and Floating Point Arithmetic Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations.

Unit – IV

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), Input-Output Processor, Serial Communication. Memory Organization: Memory Hierarchy, Main Memory, RAM and ROM, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

Text M. Morris Mano, *Computer System Architecture (3e)*

References

1. Andrew S. Tanenbaum, *Structured Computer Organization*
2. William Stallings, *Computer Organization and Architecture*
3. Zvi Kohavi, Niraj K. Jha, *Switching and Finite Automata Theory*
4. Sajjan G. Shiva, *Computer Organization, Design and Architecture*
5. David A. Patterson, John L. Hennessy, *Computer Organization Design*
6. Sivarama P. Dandamudi, *Fundamentals of Computer Organization and Design*
7. David Money Harris, Sarah L. Harris, *Digital Design and Computer Architecture*
8. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, *Computer Organization and Embedded Systems*

CS402T Cloud Computing

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Era of Cloud Computing (CC): introduction, cloud and other similar configurations, CC vs. peer-to-peer architecture, CC vs. client-server architecture, CC vs. GC, components of CC, impact of CC on businesses.
Introduction Virtualization: Introduction, virtualization benefits, implementation levels of virtualization, virtualization at the OS level, virtualization structure, open source virtualization technology, Xen virtualization architecture, binary translation with full virtualization, para-virtualization with compiler support, virtualization of CPU, memory, I/O devices, hardware support for virtualization, virtualization in multicore processors.
Cloud Computing Services: IaaS, PaaS, leveraging PaaS for productivity, guidelines for selecting a PaaS provider, concerns with PaaS, languages and PaaS, SaaS, DBaaS. Cloud Computing and Business Value: key drivers for CC, CC and outsourcing, types of scalability, use of load balancers to enhance scalability, variable operating costs using CC, time-to-market benefits of CC, distribution over the internet, levels of business values from CC.
Cloud Types and Models: private cloud, public cloud, hybrid cloud.

Unit – II

Open Source Cloud Implementation and Administration: Eucalyptus & OpenSatck cloud architectures, CSB (158)
Recent Trends in Cloud Computing and Standards: conflicts of interest for public cloud and IT product providers, BYOD and encryption exposures, cloud standards, cloud ratings, CC trends that are accelerating adoption.
Host Security in the Cloud: security for virtualization products, host security for SaaS, PaaS, IaaS. Data Security in the Cloud: challenges with cloud data and data security, data confidentiality and encryption, data availability, data integrity, CSGs. Cloud application requirements, SOA for cloud applications.

Unit – III

Adoption and Use of Cloud by Small and Medium Businesses: place of adoption, benefits, adoption phases, vendor roles and responsibilities, selection phases, provider liability, provider capabilities, success factors for CC
Adoption process of public clouds by enterprises. Cloud migration techniques, Phases during the migration of an application to the cloud. IT Service Management for Cloud Computing: ITIL based service management, service strategy, service design, service transition, service operations, continual service improvement.
SLA with Cloud Service Providers: concept, aspects and requirements of SLA, credit calculation, samples 1 and 3.
Risks, Consequences, and Costs for Cloud Computing: introduction, risk assessment and management, risk of vendor lock-in, loss of control, not meeting regulatory compliances, resource scarcity, multitenant environment, failure, inadequate SLA, malware and internet attacks, management of cloud resources, network outages, in fracture, legal, licensing, TCO, cloud costs, cost allocations, chargeback models and methodology, billable items.

Unit – IV

AAA Administration for Cloud: AAA model, single sign0on for clouds, industry implementation for AAA, authentication management in the cloud, SAML, authentication for resource utilization.
Security as a Service: benefits of security as a service, concerns with security as a service, security service providers, IdMaaS, attributes of IdMaaS providers. Cloud Certifications and Audits: certifications, cloud audit framework, cloud auditing requirements. Application Security in the Cloud: cloud application SDLC, cloud service reports by providers, application security in IaaS, PaaS and SaaS environments. Mobile Cloud Computing (MCC): architecture of MCC, benefits of MCC, MCC challenges.

Text Kailash J, Jagannath K, Donald J H, Deven Shah, *Cloud Computing – Black Book*

References

1. Rajkumar Buyya, *Cloud Computing: Principles and Paradigms*
2. Arshdeep Bahga, Vijay Madisetti, *Cloud Computing – A Hands-On Approach*
3. David E.Y. Sarna, *Implementing and Developing Cloud Computing Applications*
4. Kai Hwang, *Distributed and Cloud Computing From Parallel Processing to Internet of Things*

CS403T(A) Mobile Computing

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction to Mobile Computing: applications, a simplified reference model, Wireless Transmission: frequencies of radio transmission, signals, antennas, signal propagation, multiplexing, modulation, spread spectrum, cellular system. Media Access Control: motivation for a specialized MAC, SDMA, FDMA, TDMA, CDMA, and Comparisons.

Unit – II

GSM, DECT, Wireless LAN: Infrared vs. radio transmission, Infrastructure and ad-hoc networks, IEEE 802.11, HPERLAN, Bluetooth.

Unit – III

Mobile Network Layer: mobile IP, dynamic host configuration protocol, ad-hoc networks. Mobile Transport Layer: Traditional TCP, classical TCP improvements, TCP over 2.5/3G wireless networks.

Unit – IV

File Systems, World Wide Web, Wireless Application Protocol (WAP) and WAP 2.0.

Text Jochen H. Schiller, *Mobile Communications (2e)*

Reference

1. Raj Kamal, Mobile Computing
2. Asoke K. Talukder, Roopa R. Yavagal, *Mobile Computing*
3. Mazliza Othman, *Principles of Mobile Computing and Communications*
4. Prasant Kumar Pattnaik, Rajib Mall, *Fundamentals of Mobile Computing*
5. Ivan Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*
6. David Taniar, *Mobile Computing Concepts, Methodologies, Tools, and Applications*

CS403T(B) Distributed Systems

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction: definition of s distributed system, goals, types of distributed systems. Architectures: architectural styles, system architectures, architectures versus middleware, self-management in distributed systems. Processes: threads, virtualization, clients, servers, code migration.

Unit – II

Communication: Remote Procedure Call, Message-Oriented Communication, Stream-Oriented Communication, Multicast Communication. Naming: names, identifiers, and addresses, flat naming, structured naming, attribute based naming. Synchronization: clock synchronization, logical clocks, mutual exclusion, global positioning of nodes, election algorithms.

Unit – III

Consistency and Replication: introduction, data-centric consistency models, client-centric consistency models, replica management, consistency protocols. Fault Tolerance: introduction, process resilience, reliable client-server communication, reliable group communication, distributed commit, recovery. Security: introduction, secure channels, access control, security management.

Unit – IV

Distributed Object–Based Systems: architecture, processes, communication, naming, synchronization, consistency and replication, fault tolerance, security. Distributed File Systems: architecture, process, communication, naming, synchronization, consistency and replication, fault tolerance, security. Distributed Web-based Systems: architecture, process, communication, naming, synchronization, consistency and replication, fault tolerance, security.

Text Andrew S. Tanenbaum, Maarten Van Steen, *Distributed Systems – Principles and Paradigms (2e)*

References

1. Sukumar Ghosh, *Distributed Systems An Algorithmic Approach*
2. Joel M. Crichlow, *Distributed Systems Computing Over Networks*
3. Kai Hwang, *Distributed and Cloud Computing From Parallel Processing to Internet of Things*
4. Ajay D. Kshemkalyani, Mukesh Singhal, *Distributed Computing Principles, Algorithms, and Systems*
5. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, *Distributed Systems Concepts and Design*

CS404T(A) Artificial Intelligence

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction to Artificial Intelligence: introduction, AI techniques, problem solving with AI, AI models, data acquisition and learning aspects in AI.

Problem Solving: problem-solving process, formulating problem, problem types and characteristics, problem analysis and representation, problem space and search, toy problems, real-world problems, problem reduction methods.

Uniformed Search: general search algorithm, uniformed search methods – BFS, uniform cost search, DFS, DLS, IS, bi-directional search, comparison of the uniformed techniques.

Unit – II

Informed Search: generate and test, best first search, greedy search, A* search, memory bounded heuristic search, heuristic function, AO* search, local search algorithms and optimization problems, adversarial search methods (game theory), online search algorithms.

What is an intelligent agent? Types of agent, what is constraint satisfaction problem (CSP), CSP as search problem, local search for CSP, formulating problem structure.

Knowledge and Reasoning: knowledge representation, knowledge-based agents, the wumpus world, logic, propositional logic, predicate logic, unification and lifting: inference in FOL, representing knowledge using rules, semantic networks, frame systems, inference, types of reasoning.

Unit – III

Uncertain Knowledge and Reasoning: uncertainty and methods, Bayesian probability and belief network, probabilistic reasoning, probabilistic reasoning over time, forward and backward reasoning, perception, making simple decisions, making complex decisions, other techniques in uncertainty and reasoning process.

Planning problem, simple planning agent, planning languages, blocks world, goal stack planning, means-ends analysis, planning as a state-space search.

Learning: what is machine learning? Learning paradigms, learning concepts, methods and models, statistical learning methods, artificial neural networks–based learning, support vector machines, reinforcement learning.

Unit – IV

Expert Systems: architecture of expert system, confidence factors, existing expert systems, knowledge acquisition, shell and explanations, self-explaining system, rule-based expert systems, forward and backward chaining, frame-based expert systems, uncertainty management in expert systems, expert system and DSS, pros and cons of expert systems, case study.

Pattern Recognition: machine perception and pattern recognition, feature extraction, classification, object recognition, speech recognition, pattern mining. Game Playing: important concepts of game theory, game playing and knowledge structure, game as search problem, alpha-beta pruning, game theory problems, robotics.

Concepts and terminology of ANN, feed-forward NN, feedback networks, pattern associative networks, competitive learning, fuzzy sets, fuzzy inference process, neuro-fuzzy systems, range of AI applications, AI applications and examples, case study: agricultural domain – farmer's intelligent assistant.

Text Parag kulkarni, Prachi Joshi, *Artificial Intelligence: Building Intelligent* Systems

References

1. Nils J Nilsson, *Artificial Intelligence: A New Synthesis*
2. Kevin Knight, Elaine Rich, B Nair, *Artificial Intelligence*
3. Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach*
4. Eugene Charniak, Drew McDermott, *Introduction to Artificial Intelligence*
5. Vinod Chandra S S, Anand Hareendran S, *Artificial Intelligence and Machine Learning*

CS404T(B) Internet of Things

Theory: 4 Hours/Week **Credits:** 4

Unit – I

Introduction to Internet of Things: Introduction, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels & Deployment Templates, Domain Specific IoTs: Home Automation, Cities, Environment, Energy, Retail, Agriculture, Health & Lifestyle.

Unit – II

IoT and M2M: Introduction to M2M, Difference between IoT and M2M, SDN and NFV for IoT. IoT System Management with NETCONF-YANG: Need for IoT Systems Management, SNMP, Network Operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG. IoT Platforms Design Methodology: Introduction, IoT Design Methodology, Case Study on IoT system for weather Monitoring. Motivation for Using Python. Python Packages for IoT.

Unit – III

IoT Physical Devices & Endpoints: What is an IoT Device, Exemplary Device: Raspberry Pi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, programming Raspberry Pi with Python, Other IoT Devices.
IoT Physical Servers & Cloud Offerings: Introduction to Cloud Storage Models & Communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework-Django, Designing a RESTful Web API, Amazon Web Services for IoT, SkyNet IoT Messaging Platform.

Unit – IV

Case Studies of IoT Design: Home Automation, Cities, Environment, Agriculture, Productivity Applications. Introduction to Data Analytics for IoT, Apache Hadoop, YARN, Oozie, Spark, Storm, Health Monitoring Case study. An IoT Tool: chef, Chef Case Studies.

Text Arshdeep Bahga, Vijay Madisetti, *Internet of Things – A Hands on Approach*

References

1. Graham Meikle, Mercedes Bunz, *The Internet of Things*
2. Rajkumar Buyya, Amir Vahid Dastjerdi, *Internet of Things*
3. Adrian McEwen, Hakim Cassimally, *Designing the Internet of Things*
4. Olivier H, David B, Omar E, *The Internet of Things: Key Applications and Protocols*
5. Jean Philippe V, Adam Dunkel, *Interconnecting Smart Objects with IP: The Next Internet*
6. Daniel Minoli John, *Building the Internet of Things with IPv6 and MIPv6 – The Evolving World of M2M Comminications*
7. Ovidiu Vermesan, Peter Friess, *Internet of Thinks Converging Technologies for Smart Environments and Integrated Ecosystem.*

CS405P Project work

Theory: 16 Hours/Week **Credits:** 8

- The total allotted marks 200 are divided in to the following way
  - Internal Assessment (100 marks)
    - First seminar (50 marks – in between 25 to 30 days after commencement of class work)
      This seminar include the study of existing system, literature survey, problem definition.
    - Second seminar (50 marks – in between 55 to 60 days after commencement of class work)
      This seminar include the requirements specification, analysis, design and partial implementation.
  - External Assessment (100 marks)
- The students should submit one page of synopsis on the project work for display on the notice board.
- The project presentation is for 10 minutes followed by 05 minutes for discussion.
- The student should submit a technical write-up on the project.
- At least two teachers will be associated with the project seminar to evaluate students for the award of sessional marks which will be on the basis of performance in all the 3 items (synopsis, presentation, technical write-up).
- The project seminar presentation should include the following components of the project
  - Problem definition and specification.
  - Literature survey (familiarity with research journals).
  - Broad knowledge of available techniques to solve a particular problem.
  - Planning of the work, preparation of bar (activity) charts
  - Presentation-oral and written.
- The project report should be in the following format

I – Algorithm based Projects	II – System-Based Projects
Title Page Certificates by HoD, guide(s) and Declaration Acknowledgements Abstract Contents List of Figures Chapters: 1. Introduction 2. Literature Survey /Related Work 3. Outline the Solution 4. Results and Discussion 5. Conclusion and Future Work 6. References	Title Page Certificates by HoD, guide(s) and Declaration Acknowledgements Abstract Contents List of Figures Chapters: 1. Introduction 2. Requirements and Specifications 3. Analysis and Design 4. Implementation 5. Results and Discussion 6. Conclusion and Future Work 7. References

- References and Bibliography should be written in the format given below:
  - Author(s) Title of the Paper, Publisher, Volume No., Issue No., Year
  - Example:
    - Ganesh S., Vijayalakshmi M. and Kannan A., "Intelligent Agent based Approach for transaction Processing in mobile Database Systems", The IAJIT, Vol. 4, No. 2, pp, 97-102, 2007.
- Text format: Font Type: Time New Roman; Font Size: 12; Line Space: 1 ½
- Pages, Figures, Tables and Algorithms should be titled and numbered, Students should be discouraged writing about languages, platforms, operating systems and packages used for the purpose of project in the project report.
- The report should be organized into chapters, chapter into sections, sections into subsections etc. Hierarchical numbering should be followed in numbering the chapters, sections, subsections etc. (1, 1.1, 1.1.1), three (3 copies of the project report hard bound should be submitted to the department).

MOOCs [Massive Online Open Courses] Free Resources

E-Learning:

- NPTEL :nptel.ac.in [Core Subjects Certification]
- C++ INSTITUTE :cppinstitute.org [C++ Certification]
- ORACLEEDUCATION :education.oracle.com [Java, DBMS Certification]
- BIG DATA UNIVERSITY :bigdatauniversity.com [Big Data Certification]
- COURSERA :coursera.org [Core Subjects Certification]
- CODEACADEMY :codecademy.com [Coding Certification]
- KHANACADEMY :khanacademy.org [Core Subjects Certification]
- PIXAR IN A BOX :khanacademy.org/partner-content/pixar
- VIDEOLECTURES :videolectures.net
- YOUTUBEEDU :plus.google.com/+YouTubeEDU/posts
- DISNEY RESEARCH :disneyresearch.com
- ALISON :alison.com [Core Subjects Certification]
- INTERNET ARCHIVE :archive.org

Freeware:

- SCILAB : scilab.org [MatLab Equivalent]
- GEOGEBRA :geogebra.org [Software for Class Room Teaching]

Search Engine:

- WOLFRAM ALPHA :wolframalpha.com [Computing Engine]
- CITESEER :citeseerx.ist.psu.edu [Searching Research Articles]
- DOAJ :doaj.org [Open Access to Journals]