Deccan Education Society's

FERGUSSON COLLEGE, PUNE
(AUTONOMOUS)

SYLLABUS UNDER AUTONOMY

FIRST YEAR M.Sc. (Computer Science)
SEMESTER - I

SYLLABUS M.Sc.-I (Computer Science)
Academic Year 2016-2017

Deccan Education Society's
Fergusson College (Autonomous), Pune
Faculty of Science
Post Graduate Syllabus (Computer Science)
First Year

Semester	Course Code	Title of the Course	Core / Elective	No. of Credits
I	CSC4101	Principles Of Programming Languages	CORE	4
	CSC4102	Distributed Database Concepts	CORE	4
	CSC4103	Design And Analysis Of Algorithms	CORE	4
	CSC4104	Advanced Networking and Network Programming	CORE	4
	CSC4105	Practical - I (Based on PPL & ANNP)	PCORE	4
	CSC4106	Practical - II (Project)	PCORE	4
	CSC4107	Logic and analytical skill development	CORE	1
II	CSC4201	Software Metrics & Project Management	CORE	4
	CSC4202	Data Mining and Data Warehousing	CORE	4
	CSC4203	Programming with DOT NET	CORE	4
	CSC4204	Information System and Security	CORE	4
	CSC4205	Artificial Intelligence	Elective-1	4
	CSC4206	Web Services	Elective-2	4
	CSC4207	Advanced Design And Analysis Of Algorithms	Elective-3	4
	CSC4208	Practical - I (Based DOT NET)	PCORE	4
	CSC4209	Practical - II (Project)	PCORE	4
	CSC4210	Emerging Technology	CORE	2
Note : Students should choose one Elective subject out of the given three Electives				
			TOTAL	**55**

Extra Credits

Semester	Course Code	Title of the Course	No. of Credits
I	XHR0001	Human Rights	1
	XCS0002	Introduction to Cyber Security / Information security	1
	XSD0003	Skill Development	1
II	XHR0004	Human Rights	1
	XCS0005	Introduction to Cyber Security / Information security	1
	XSD0006	Skill Development	1
		TOTAL	**06**

PAPER CODE: CSC4101

PAPER –I: Principles of Programming Language

[Credit -4: No. of Lectures 48]

Prerequisites:

It is assumed that student learning this course have the following background:

- Experience with an OOP language (such as Java or C++)
- Experience with a procedural language (such as C)
- Working knowledge of C, C++, and Java programming.
- Basic algorithms and data structure concepts.

Objectives:

- This course will prepare you to think about programming languages analytically:
  - Separate syntax from semantics
  - Compare programming language designs
  - Learn new languages more quickly
  - Use standard vocabulary when discussing languages
  - Understand basic language implementation techniques
- This course focuses on both:
  - Theory is covered by the textbook readings, lectures, and on the tests
  - Implementation is covered by the homework assignments

	Title and Contents	No. of Lectures
Unit -I	**Introduction** The Art of Language Design The Programming Language Spectrum Why Study Programming Languages? Compilation and Interpretation Programming Environments	**2**
Unit -II	**Non-Imperative Programming Models: Functional, Logic Languages** Common LISP Basic LISP Primitives (FIRST, REST, SETF, CONS, APPEND, LIST,NTHCDR, BUTLAST,LAST, LENGTH, REVERSE, ASSOC) Procedure definition and binding, DEFUN, LET Predicates and Conditional, EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM, NUMBERP, SYMBOLP, NIL, NULL, IF, WHEN, UNLESS,COND, CASE Procedure Abstraction and Recursion **Turbo Prolog** Introduction, facts, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates	**10**

Unit –III	**Names, Scopes, and Bindings** The Notion of Binding Time Object Lifetime and Storage Management: Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection Scope Rules Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping The meaning of Names in a Scope Aliases, Overloading, Polymorphism and Related Concepts The Binding of Referencing Environments Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures Macro Expansion	**5**
Unit - IV	**Control Flow** Expression Evaluation Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation Structured and Unstructured Flow Structured Alternatives to goto Sequencing Selection Short-Circuited Conditions, Case/Switch Statements Iteration Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion Iteration and Recursion, Applicative and Normal -Order Evaluation	**5**
Unit - V	**Data Types** Introduction Primitive Data Types Numeric Types Integer Floating point Complex Decimal Boolean Types Character Types Character String Types Design Issues Strings and Their Operations String Length Operations Evaluation Implementation of Character String Types User defined Ordinal types	**8**

	Enumeration types Designs Evaluation Subrange types Ada’s design Evaluation Implementation of user defined ordinal types Array types Design issues Arrays and indices Subscript bindings and array categories Heterogeneous arrays Array initialization Array operations Rectangular and Jagged arrays Slices Evaluation Implementation of Array Types Associative Arrays Structure and operations Implementing associative arrays Record types Definitions of records References to record fields Operations on records Evaluation Implementation of Record types Union Types Design issues Discriminated versus Free unions Evaluation Implementation of Union types Pointer and Reference Types Design issues Pointer operations Pointer problems Dangling pointers Lost heap dynamic variables Pointers in C and C++ Reference types Evaluation Implementation of pointer and reference types Representation of pointers and references Solution to dangling pointer problem Heap management	
Unit - VI	**Subroutines and Control Abstraction** Fundamentals of Subprograms Design Issues for subprograms	**5**

	Local Referencing Environments Parameter-Passing Methods Parameters That are Subprograms Overloaded Subprograms Generic Subroutines Generic Functions in C++ Generic Methods in Java Design Issues for Functions User-Defined Overloaded Operators Coroutines The General Semantics of Calls and Returns Implementing “Simple” Subprograms Implementing Subprograms with Stack-Dynamic Local Variables Nested Subprograms Blocks Implementing Dynamic Scoping	
Unit - VII	**Data Abstraction and Object Orientation** Object-Oriented Programming Encapsulation and Inheritance Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance Initialization and Finalization Choosing a Constructor, References and Values, Execution Order, Garbage Collection Dynamic Method Binding Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures Multiple Inheritance Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance	**8**
Unit - VIII	**Concurrency** Introduction Multiprocessor Architecture Categories of concurrency Motivations for studying concurrency Introduction to Subprogram-level concurrency Fundamental concepts Language Design for concurrency. Design Issues Semaphores Introduction Cooperation synchronization Competition Synchronization Evaluation Monitors	**5**

	Introduction Cooperation synchronization Competition Synchronization Evaluation Message Passing Introduction The concept of Synchronous Message Passing Java Threads The Thread class Priorities Competition Synchronization Cooperation Synchronization	

References:

1. Scott, Programming Language Pragmatics, 3e(With CD) ISBN 9788131222560 Kaufmann Publishers, An Imprint of Elsevier, USA
2. Robert W. Sebesta, Concepts of Programming Languages, Eighth Edition, Pearson Education
3. Carl Townsend, Introduction to Turbo Prolog
4. Patrick Henry Winston & Berthold Klaus Paul Horn ,LISP 3rd edition –BPB
5. M. Gabbrielli, S. Martini, , Programming Languages: Principles and Paradigms, Springer ISBN: 9781848829138

PAPER CODE: CSC4102

PAPER –II: Distributed Database Concepts

[Credit -4: No. of Lectures 48]

Prerequisites:

- Students should be well-versed with the basic and advanced concepts of RDBMS

Objectives:

- Main objective is to understand the principles and foundations of distributed databases. This course addresses architecture, design issues, integrity control, query processing and optimization, transactions, and concurrency control & distributed transaction reliability.

	Title and Contents	No. of Lectures
Unit -I	**Distributed databases: An overview** Features of distributed Vs centralized databases Why DDB? DDBMS Promises / problem areas in implementing a DDB	**2**
Unit -II	**DDBMS Architecture** DBMS Standardization Architectural models for DDBMS DDBMS architecture Distributed catalog management	**4**
Unit –III	**Distributed database design** Alternative design strategies Distributed design issues Concepts of join graphs Fragmentation and allocation	**10**
Unit - IV	**Overview of Query processing** Query processing problems Objectives of query processing Complexity of relational algebra operators Characterization of query processors Layers of query processing	**4**
Unit -V	**Query decomposition & data localization** Query decomposition Localization of distributed data	**2**
Unit -VI	**Optimization of distributed queries** Query optimization Centralized query optimization Join ordering in fragment queries. Distributed query optimization algorithms Centralized query optimization	**10**

	Join ordering in fragment queries Distributed query optimization algorithms	
Unit - VII	**Management of distributed transactions** Framework for transaction management Supporting atomicity of distributed transactions Concurrency control of distributed transactions Architectural aspects of distributed transactions	**2**
Unit - VIII	**Concurrency control** Foundations of distributed concurrency control Distributed deadlocks Concurrency control based on timestamps Optimistic methods for distributed concurrency control	**6**
Unit - IX	**Distributed DBMS reliability** Reliability concepts & measures Failures & fault tolerance in distributed systems Failures in DDBMS Local reliability protocols Distributed reliability protocols Dealing with site failures Network partitioning	**8**

References:

1. M. Tamer Ozsu and Patrick Valduriez , Principles of Distributed Database Systems; 2nd Edition Publishers: Pearson Education Asia.
2. Stefano Ceri and Giuseppo Distributed Database; Principles & Systems Pelagatti Publications: McGraw-Hill International Editions.
3. Raghuramakrishnan and Johannes , Database systems (2nd edition)

PAPER CODE: CSC4103

PAPER –III: Design and Analysis of Algorithms

[Credit - 4: No. of Lectures 48]

Prerequisites:

- Basic algorithms and data structure concepts.
- Basic programming concepts

Objectives:

This course will prepare students in

- Basic Algorithm Analysis techniques and understand the use o asymptotic notation
- Understand different design strategies
- Understand the use of data structures in improving algorithm performance
- Understand classical problem and solutions
- Learn a variety of useful algorithms
- Understand classification of problems

	Title and Contents	No. of Lectures
Unit -I	**Analysis** Algorithm definition, space complexity, time complexity, worst case –best case –average case complexity, asymptotic notation, sorting algorithms (insertion sort, heap sort) , sorting in linear time, searching algorithms, recursive algorithms (Tower of Hanoi , Permutations).	**6**
Unit -II	**Decrease and conquer** DFS and BFS, Topological sorting, Exponentiation by squaring Euclid’s algorithm for greatest common divisor	**6**
Unit –III	**Divide and conquer** Control abstraction, binary search, merge sort, Quick sort, Stassen’s matrix multiplication	**6**
Unit - IV	**Greedy method** Knapsack problem, job sequencing with deadlines, minimum-cost spanning trees, Kruskal and Prim’s algorithm, optimal storage on tapes, optimal merge patterns, Huffman coding	**8**
Unit - V	**Dynamic programming** Matrix chain multiplication, . single source shortest paths, Dijkstra’s algorithm, Bellman- ford algorithm , all pairs shortest path, longest common subsequence, string editing, 0/1 knapsack problem, Travelling salesperson problem.	**8**
Unit - VI	**Backtracking** General method, 8 Queen’s problem, Sum of subsets problem, graph coloring problem, Hamiltonian cycle	**5**

Unit - VII	**Branch and Bound Technique** FIFO, LIFO, LCBB, TSP problem	**3**
Unit - VIII	**Transform and conquer** Horner’s Rule and Binary Exponentiation – Problem Reduction	**4**
Unit - IX	**Problem classification** Nondeterministic algorithm, The class of P, NP, NP-hard and NP- Complete problems, significance of Cook’s theorem	**2**

References:

1. Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, Computer Algorithms, Galgotia.
2. T. Cormen, C. Leiserson, & R. Rivest, Algorithms, MIT Press, 1990 1
3. A. Aho, J. Hopcroft, & J. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974
4. Donald Knuth, The Art of Computer Programming (3 vols., various editions, 1973-81), Addison Wesley
5. Steven Skiena, The Algorithm Manual, Springer ISBN:9788184898651
6. Jungnickel, Graphs, Networks and Algorithms, Springer, ISBN: 3540219056

PAPER CODE: CSC4104

PAPER –IV: Advanced Networking and Network Programming

[Credit -4: No. of Lectures 48]

Prerequisites:

- Working Knowledge of C
- Basic Understanding of Networking Concepts
- User Level Knowledge of Linux

Objectives:

- Students will get aware of advanced networking concepts.
- Course discusses the concepts underlying in the design and implementation of Networking Concepts
- Most of the Units start with the theory and then switches focus on how the concepts are implemented in a C program

	Title and Contents	No. of Lectures
Unit -I	**Review of Basic Concepts** TCP/IP Protocol Suite , Underlying Technologies : LAN (802.3), Wireless Lans (802.11) , Point-to-point WANS, Switched WANS	**3**
Unit -II	**Routing Protocols** Forwarding, Structure of a Router , Routing Tables, Intra – And Inter-Domain Routing, Distance Vector Routing, RIP, OSPF, BGP ,Multicast Routing	**5**
Unit –III	**The Internet Layer Protocols** IPv4, IPv6, Transition from IPv4 to IPv6 , ICMPv4, ICMPv6 Socket Address Structures (IPv4 & IPv6), Value-Result Arguments, Byte Ordering Functions, Byte Manipulation Functions, inet_aton, inet_addr, and inet_ntoa Functions, inet_pton and inet_ntop Functions, sock_ntop and Related Functions, readn, writen, and readline Functions, isfdtype Function getsockopt and setsockopt Functions, Checking If an Option Is Supported and Obtaining the Default, Socket States, Generic Socket Options, IPv4 Socket Options, ICMPv6 Socket Option, IPv6Socket Options, TCP Socket Options	**10**
Unit - IV	**The Transport Layer** The Transport Service, Elements of Transport Protocols, TCP, UDP	**6**

Unit - V	**Elementary TCP Socket** socket Function, connect Function, bind Function, listen Function, accept Function, fork and exec Functions, Concurrent Servers, close Function, getsockname and getpeername Functions TCP Echo Server: main Function, TCP Echo Server: str_echo Function, TCP Echo Client: main Function, TCP Echo Client: str_cli Function, Normal Startup, Normal Termination, Connection Abort before accept Returns, Termination of Server Process, SIGPIPE Signal, Crashing of Server Host, Crashing and Rebooting of Server Host, Shutdown of Server Host I/O multiplexing, select Function, str_cli Function (Revisited), Batch Input, shutdown Function, str_cli Function (Revisited), TCP Echo Server (Revisited), pselect Function, poll Function, TCP Echo Server (Revisited)	**10**
Unit - VI	**Elementary UDP Socket** recvfrom and sendto Functions, UDP Echo Server: main Function, UDP Echo Server: dg_echo Function, UDP Echo Client: main Function, UDP Echo Client: dg_cli Function, Lost Datagrams, Verifying Received Response, Server Not Running, Summary of UDP example, connect Function with UDP, dg_cli Function (Revisited), Lack of Flow Control with UDP, Determining Outgoing Interface with UDP, TCP and UDP Echo Server Using select User Datagram Protocol, File Transfer, Error Handling	**6**
Unit - VII	**Protocols, Sessions, State, and Implementing Custom Protocols** State vs. Stateless, Methods for Maintaining State, What Is a Protocol?, Designing a Custom Protocol, Our Chat Protocol, Protocol Registration	**4**
Unit - VIII	**Elementary Name, Address Conversions and design decisions** Domain Name System, gethostbyname Function, RES_USE_INET6 Resolver Option, gethostbyname2 Function and IPv6 Support, gethostbyaddr Function, uname Function, gethostname Function, getservbyname and getservbyport Functions	**4**

References:

1. Behrouz A. Forouzan, TCP / IP Protocol Suite Fourth Edition
2. Andrew Tanenbaum, Computer Networks Fourth Edition
3. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, Volume 1: The Sockets Networking API, 3/E PHI
4. KEIR DAVIS, JOHN W. TURNER, AND NATHAN YOCOM, The Definitive Guide to Linux Network Programming, Apress.

CSC4105: Lab Course –I (Based on PPL & ANNP)

Practical Assignment based on

1. Principles of Programming Languages
2. Advanced Networking with Network programming

Continuous Internal Assessment

1	Journal	10Marks
2	Viva	
	Principles of Programming Languages	10 Marks
	Advanced Networking with Network programming	10 Marks
3	Internal test – for skill assessment	10 Marks
4	Attendance + Active participation	10 Marks
	Total	**50 Marks**

End Semester Assessment

1	Principles of Programming Languages	25 Marks
2	Advanced Networking with Network programming	25 Marks
	Total	**50 Marks**

	PAPER CODE: CSC4105 **PAPER–V:Lab Course –I (Based on PPL &ANNP)** **[Credit -4: No. of Practicals 12]**
	Title of Experiment/ Practical
	Principles of Programming Language Assignments **LISP: Set of Assignments**
1	Define a LISP function to compute sum of squares.
2	Define a LISP function to compute difference of squares. (if x > y return x2-y2, otherwise y2-x2)
3	Define a Recursive LISP function to compute factorial of a given number.
4	Define a Recursive LISP function which takes one argument as a list and returns last element of the list. (do not use last predicate)
5	Define a Recursive LISP function which takes one argument as a list and returns a list except last element of the list. (do not use butlast predicate)
6	Define a Recursive LISP function which takes two arguments first, an atom, second, a list, returns a list after removing first occurrence of that atom within the list.
7	Define a recursive LISP function which takes 2 lists as arguments and returns a list containing alternate elements from each list e.g. if L1 = (1 5 7) and L2 = (2 4 9 3) output should be (1 2 5 4 7 9 3)
	Prolog: Set of Assignments
8	Write a tail recursive prolog to print numbers from n to 1
9	Prolog programs doing formal reasoning and resolutions proofs. e.g. Consider the following statements: “John likes all kinds of food. Apples are food. Chicken is food. Anything anyone eats, and is still alive, means whatever he ate was a food. Sue eats everything Bill eats. Bill eats Peanuts and is still alive.” Write a Prolog program to prove that John likes Peanuts, and to answer the question “What food does Sue eat?”
10	Write a prolog program to prove that "A table supports a bottle". - If x is on the top of y, y supports x. - If x is above y and they are touching each other then x is on the top of y. - A bottle is above the table. - A bottle is touching the table.
11	To find factorial of a given number.
12	Consider the following sentences: Mammals have 4 legs and no arms, or 2 arms and no legs. A cow is a mammal. A cow has no arms. Write a prolog program to represent these sentences. Can we prove that “cow has 4 legs”?
	Advanced Networking and Network Programming Assignments
1	Explore different network related commands on LINUX and interpret the result.
2	Write a program to calculate the checksum.
3	Write a program to simulate Distance vector routing protocol.
4	Write a program to simulate Link state routing protocol.

5	Write a program to simulate Boarder Gateway routing protocol (BGP)
6	Write a program to catch a signal and print a message.
7	Write a program to print network byte order of a computer.
8	Write a program to create a socket and display its file descriptor.
9	Write program to print the host name using gethostbyname ( ) system call.
10	Write a program to create a socket and bind it to port address and listen to a particular connection and accept it.
11	Write a program to create a day-time client and day-time server.
12	Write a program to implement a client - server user-level application using sockets API. Server accepts strings from clients (even multiple strings from each client) and replies with reverse strings.

CSC4106: Lab Course –II (Project)

Objective:

The objective of project is to make the students understand Requirement analysis, design and implementation cycle. Any open problem statement can be taken for implementation. The system can be designed in any programming language implemented in any platform.

The Project can be platform, Language and technology independent. Project will be evaluated by project guide. Assessment will be done weekly in the respective batch. Evaluation will be on the basis of weekly progress of project work, progress report, oral, results and documentation and demonstration.

You should fill your status of the project work on the progress report and get the Signature of project guide regularly. Progress report should sharply focus how much time you have spent on specific task. (The format of progress report is given as follow.) You should keep all signed progress report. Project will not be accepted if progress report is not submitted and all responsibility remains with student.

Project Progress Report

Roll No & Name of the student	
Title of the Project	
Project guide Name	

Sr.	From Date	To Date	Details of Project work	Project guide sign (with date)

Head,
Dept. of Computer Science

PAPER CODE: CSC4107

PAPER – VII: LOGIC and ANALYTICAL SKILLS DEVELOPMENT

[Credit -1: No. of Lectures 15]

	Title and Contents	No. of Lectures
Unit -I	Logic building	4
Unit -II	Analytical Skills	7
Unit –III	Domain Expertise	4

Note: The evaluation is for 10 Marks based on above concepts.

Deccan Education Society's

FERGUSSON COLLEGE, PUNE
(AUTONOMOUS)

SYLLABUS UNDER AUTONOMY

FIRST YEAR M.Sc. (Computer Science)
SEMESTER - II

SYLLABUS M.Sc. (Computer science)
Academic Year 2016-2017

PAPER CODE: CSC4201
PAPER–I: Software Metrics and Project Management
[Credit -4: No. of Lectures 48]

Prerequisites:

- Knowledge of Software Engineering
- Basics of Software Testing

Objectives:

- Project Management covers skills that are required to ensure successful medium and large scale software projects.
- Learn to select and apply project management techniques for process modelling, planning, estimation, process metrics and risk management.
- To learn process of software project management, cost estimation, use of project management tools.

	Title and Contents	No. of Lectures
Unit -I	**Introduction to Project Management** What is Project? What is Project management? Project phases and project life cycle, organizational structure, Qualities of Project Manager.	**5**
Unit -II	**Project Management Components** Project Integration Management-Project plan development and execution, change controls, configuration management.	**6**
Unit –III	**Scope Management** Strategic planning scope planning, definition ,verification and control.	**4**
Unit –IV	**Time management** Activity planning, schedule development and control.	**4**
Unit –V	**Cost Management** Cost estimation and Control	**4**
Unit –VI	**Quality Management** Quality planning and assurance.	**2**
Unit –VII	**Human Resource Management** Organizational planning , staff acquisition	**2**
Unit –VIII	**Communication Management** Information distribution , reporting.	**2**
Unit –IX	**Risk Management** Risk identification, Quantification and control	**3**

Unit –X	**Procurement Management** Solicitation, contract administration	**2**
Unit –XI	**Software Metrics** The scope of software metrics, software metrics data collection, analyzing software data, measuring size, structure, external attributes.	**6**
Unit –XII	**Planning a measurement program** What is metrics plan: Developing goals, questions and metrics Where and When: Mapping measures to activities. How: Measurement tools. Who: Measurers, analyst, tools revision plans.	**5**
Unit –XIII	**Quality Standards** CMM, PSP/TSP	**3**

References:

1. Kathy Schwalbe, Information Technology Project Management
2. Norman Fenton, Shari Lawrence Pfleeger, Software Metrics A rigorous and practical approach
3. Roger Pressman, Software Engineering

PAPER CODE: CSC4202

PAPER –II: Data Mining and Data Warehousing

[Credit -4: No. of Lectures 48]

Prerequisites:

- Knowledge of databases.
- Ease of adaptability related to techniques.

Objectives:

- Course will introduce various applications of mining and warehouse.
- To understand the various representation of the data.
- To make the students aware of possible architectures of data.
- Students will be aware of implementing tools and techniques in data mining.
- Making students to take lead in analytics.

	Title and Contents	No. of Lectures
Unit -I	**Introduction to Data Mining** Basic Data Mining Tasks, DM versus Knowledge Discovery in Databases, Data Mining Issues, Data Mining Metrics, Social Implications of Data Mining, Overview of Applications of Data Mining	**4**
Unit -II	**Introduction to Data Warehousing** Architecture of DW, OLAP and Data Cubes, Dimensional Data Modelling-star, snowflake schemas, Data Pre-processing – Need, Data Cleaning, Data Integration & Transformation, Data Reduction, Machine Learning, Pattern Matching	**4**
Unit –III	**Data Mining Techniques** Frequent item-sets and Association rule mining: A priori algorithm, Use of sampling for frequent item-set, FP tree algorithm, Graph Mining: Frequent sub-graph mining, Tree mining, Sequence Mining	**8**
Unit –IV	**Classification & Prediction** Decision tree learning, Construction, performance, attribute selection, Issues: Over-fitting, tree pruning methods, missing values, continuous classes, Classification and Regression Trees (CART), Bayesian Classification, Bayes Theorem, Naïve Bayes classifier, Inference, Parameter and structure learning, Linear classifiers, Prediction, Linear regression, Non-linear regression	**16**

Unit –V	**Accuracy Measures** Precision, recall, F-measure, confusion matrix, cross-validation, bootstrap	**4**
Unit –VI	**Software for data mining and applications of data mining** R Software: Introduction, Elementary Statistics with R and Computing with R. Programming with R	**6**
Unit –VII	**Clustering** k-means, Expectation Maximization (EM) algorithm, Hierarchical clustering, Correlation clustering	**4**
Unit –VIII	**Brief overview of advanced techniques** Text mining, Web Mining	**2**

References:

1. Han, Data Mining: Concepts and Techniques, Elsevier
2. Margaret H. Dunham, S. Sridhar, Data Mining – Introductory and Advanced Topics, Pearson Education
3. Tom Mitchell, ―Machine Learning‖, McGraw-Hill, 1997
4. R.O. Duda, P.E. Hart, D.G. Stork. Pattern Classification. Second edition. John Wiley and Sons, 2000.
5. Christopher M. Bishop, ―Pattern Recognition and Machine Learning‖, Springer 2006
6. Raghu Ramkrishnan, Johannes Gehrke, Database Management Systems, Second Edition, McGraw Hill International
7. Ian H.Witten, Eibe Frank Data Mining: Practical Machine Learning Tools and Techniques, Elsevier/(Morgan Kauffman)
8. [Research-Papers]: Some of the relevant research papers that contain recent results and developments in data mining field

PAPER CODE: CSC4203
PAPER–III: Programming with DOT NET
[Credit -4: No. of Lectures 48]

Prerequisites:

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.
- Knowledge of web development

Objectives:

- To understand the DOTNET framework, C# language features and Web development using ASP.NET
- Evaluation will be as below

	Title and Contents	**No. of Lectures**
	Part I: C#	
Unit -I	**DOTNET Framework** a. Introduction to DOTNET b. DOT NET class framework c. Common Language Runtime i. Overview ii. Elements of .NET application iii. Memory Management iv. Garbage Collection d. Common Language Integration i. Common type system ii. Reflection API	**4**
Unit -II	**Introduction to C#** a. Language features i. Variables and Expressions, type conversion ii. Flow Control iii. Functions, Delegates iv. Debugging and error handling, exception handling (System Defined and User Defined) b. Object Oriented Concepts i. Defining classes, class members, Interfaces, properties ii. Access modifiers, Implementation of class, interface and properties iii. Concept of hiding base class methods, Overriding iv. Event Handling c. Collections, Comparisons and Conversions i. Defining and using collections, Indexers,	**10**

	iterators ii. Type comparison, Value Comparison iii. Overloading Conversion operators, as operator d. Generics i. Using generics ii. Defining Generics, generic Interfaces, Generic methods, Generic Delegate	
Unit –III	**Window Programming** a. Window Controls i. Common Controls ii. Container Controls (Group box and Tab controls) iii. Menus and Toolbars iv. Printing v. Dialogs b. Deploying Window Application i. Click Once deployment	**6**
Unit –IV	**Data Access** a. File System Data b. XML c. Databases and ADO.NET d. Data Binding	**5**
Unit –V	**.NET Assemblies** a. Components b. .NET Assembly features c. Structure of Assemblies d. Calling assemblies, private and shared assemblies	**2**
	Part II: ASP.NET	
Unit –I	**Introduction to ASP.NET**	**2**
Unit –II	**Server Controls and Variables, control Structures & Functions** a. Forms, webpages, HTML forms, Webforms b. Request & Response in Non-ASP.NET pages c. Using ASP.NET Server Controls d. Overview of Control structures e. Functions : web controls as parameters	**4**
Unit –III	**Even Driven Programming and PostBack** a. HTML events b. ASP.NET page events c. ASP.NET Web control events d. Event driven programming and postback	**3**
Unit –IV	**Reading from Databases** a. Data Source and Data binding controls b. ADO.NET	**3**
Unit –V	**ASP.NET Server Controls**	**4**

	a. ASP.NET Web Controls b. HTML Server Controls c. Web Controls	
Unit –VI	**DOTNET State Management** a. Introduction to Cookies, Sessions b. Session events c. State management Recommendations	**3**
Unit –VII	**Web Services** a. HTTP, XML & Web services b. SOAP c. Building ASP.NET web service d. Consuming a web service	**2**

References:

1. Beginning Visual C#, Wrox Publication
2. Professional Visual C#, Wrox Publication
3. Tom Archer Inside C#, ISBN: 0735612889 Microsoft Press © 2001
4. Beginning ASP.NET 3.5, Wrox Publication
5. Jesse Liberty, Dan Maharry, Dan Hurwitz Programming ASP.NET 3.5, O’Reilly
6. Illustrated C# 2008, Solis, Publication APRESS, ISBN 978-81-8128-958-2
7. Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner, Professional C# 4.0 and .NET 4 WROX
8. Dan Clark, Beginning C# Object-Oriented Programming, Apress
9. Peter D. Blackburn, ADO.NET Examples and Best Practices for C# Programmers, Apress
10. Carsten Thomsen, Database Programming with C#, Apress

PAPER CODE: CSC4204
PAPER–IV: Information System and Security
[Credit -4: No. of Lectures 48]

Prerequisites:

- Student should have knowledge of basic networking concepts.
- Student should have some mathematical background to understand the various cryptography algorithms.

Objectives:

- To enable students to get sound understanding of Info-Sys-Security, Net-Security, Cryptography.
- To equip with knowledge and skills necessary to support for their career in Information Security.
- To develop attitude and interest along with necessary knowledge and skills among the students to encourage them to do further academic studies / research in this area, after the completion of their M.Sc. Course.

	Title and Contents	**No. of Lectures**
Unit -I	**Introduction To Security** • The need for Security • Security Approaches • Principles of Security • Types of Attacks	**2**
Unit -II	**Cryptography: Concepts and Techniques** • Introduction • Plain Text and Cipher Text • Substitution Techniques • Transposition Techniques • Symmetric and Asymmetric key cryptography	**4**
Unit –III	**Symmetric Key Algorithms** • Algorithms types and modes • Overview of symmetric key cryptography • DES (Data Encryption Standard) • Double DES, Triple DES • IDEA (International Data Encryption Algorithm) • RC4 & RC5 • Blowfish • AES (Advanced Encryption Standard)	**8**
Unit –IV	**Asymmetric key / Public Key Encryption** • History & overview of asymmetric key cryptography • Deffie-Hellman key exchange	**4**

	• RSA algorithm • Symmetric and Asymmetric key cryptography • Digital Signature	
Unit –V	**Message Integrity techniques** • Message Digest • MD5 • SHA • Message Authentication Code (MAC) & HMAC • Digital Signature techniques - Digital Signatures using DSA (Digital Signature Algorithm) - DSS (Digital Signature Standard) and RSA	**6**
Unit –VI	**Digital Certificates and PKI (Public Key Infrastructure)** • Digital Certificates • Private key management • PKIX Model • Public key cryptography standards (PKCS)	**4**
Unit –VII	**Internet Security Protocols** • Secure Socket Layer • TLS • SHTTP • TSP • SET • SSL Verses SET • 3-D Secure Protocol • Electronic Money • Email Security	**10**
Unit –VIII	**User Authentication and Kerberos** • Passwords • Certificate-based Authentication • Kerberos • Security Handshake Pitfalls	**4**
Unit –IX	**Server Security & Firewalls** • Firewall • DMZ networks • IP security • VPN • Intrusion Detection, IDS, Intrusion Prevention Systems (IPS)	**6**
References: 1. Andrew Tanenbaum, Computer Networks Fourth Edition 2. Atul Kahate, Cryptography and Network Security Second Edition		

PAPER CODE: CSC4205
PAPER–V: Elective – I Artificial Intelligence
[Credit -4: No. of Lectures 48]

Prerequisites:

- Concepts of Data structures and Design and Analysis of algorithms

Objectives:

- To understand and gain the knowledge of the subject

	Title and Contents	No. of Lectures
Unit -I	**Introduction to Artificial Intelligence** • What is AI? • Early work in AI • AI and related fields • AI problems and Techniques	**2**
Unit -II	**Problems, Problem Spaces and Search** • Defining AI problems as a State Space Search: example • Production Systems • Search and Control Strategies • Problem Characteristics • Issues in Design of Search Programs • Additional Problems	**6**
Unit –III	**Heuristic Search Techniques** • Generate-and-test • Hill Climbing • Best First Search • Problem Reduction • Constraint Satisfaction • Mean-Ends Analysis	**12**
Unit –IV	**Knowledge Representation** • Representations and Mappings • Approaches to Knowledge Representation • Knowledge representation method • Propositional Logic • Predicate logic • Representing Simple facts in Logic • Representing Instances and Isa relationships • Computable Functions and Predicates • Resolution • Forward and backward chaining	**12**

Unit –V	**Slot – and – Filler Structures** • Weak Structures - Semantic Networks - Frames • Strong Structures - Conceptual Dependencies - Scripts	**8**
Unit –VI	**Game Playing** • Minimax Search Procedures • Adding alpha-beta cutoffs • Baysian Classification, Certainty Factor Theory, Dempster Shafar Theory.	**5**
Unit –VII	**Learning** • What is learning? • Rote Learning • Learning by taking advice • Learning in problem solving • Learning from examples • Explanation based learning	**3**

References:

1. Elaine Rich and Kevin Knight , Artificial Intelligence, Tata McGraw Hill, 2nd Edition
2. Eberhart, Computational Intelligence, Elsevier, ISBN 9788131217832
3. Nilsson, Artificial Intelligence: A New Synthesis, Elsevier, ISBN 9788181471901
4. Dan Patterson, Introduction to Artificial Intelligence and Expert System, Prentice Hall of India Pvt. Ltd., New Delhi, 1997, 2nd Printing.
5. Carl Townsend , Introduction to Turbo Prolog

PAPER CODE: CSC4206
PAPER–V: Elective- II Web Services
[Credit -4: No. of Lectures 48]

Prerequisites:

- Strong knowledge about Java programming.
- Good Understanding of Object Oriented Programming concepts.
- Must be familiar with XML

Objectives:

- To Understand Web Services and implementation model for SOA
- To Understand the SOA, its Principles and Benefits
- Understanding cloud computing as a web service
- Discuss the concept of virtualization and data in cloud

	Title and Contents	No. of Lectures
Unit -I	**Web Service and SOA fundamentals** Introduction, Concept of Software as a Service(SaaS), Web services versus Web based applications, Characteristics of Web services, Service interface and implementation, The Service Oriented Architecture(SOA), Quality of service (QoS), Web service interoperability, Web services versus components, RESTful services , Impact and shortcomings of Web services.	**7**
Unit -II	**Web Services Architecture** Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services, developing web services enabled applications.	**7**
Unit –III	**SOAP: Simple Object Access Protocol** Inter-application communication and wire protocols, SOAP as a messaging protocol, Structure of a SOAP message, SOAP communication model, Building SOAP Web Services, developing SOAP Web Services using Java, Error handling in SOAP, Advantages and disadvantages of SOAP	**10**

Unit –IV	**Describing and Discovering Web Services** WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL, Service discovery, role of service discovery in a SOA, service discovery mechanisms, UDDI – UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, support for categorization in UDDI Registries, Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, deleting information in a UDDI Registry, limitations of UDDI	**10**
Unit –V	**Emerging trends: Cloud Computing** What is Cloud Computing?, SOA meets the Cloud, Cloud Service Models, SaaS-Salesforce.com, PaaS-Google App Engine, IaaS-Amazon EC2, Cloud Deployment Models – Public, Community, Private, Hybrid. Virtu alization , Virtual Machine(VM) Technology, Virtual Machine Monitor or Hypervisor - KVM, Xen, VMware hypervisors and their features, Multi-tenancy, Architecture model for Cloud Computing . Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery: next generation Cloud Applications. Case Study: Use Cloud Services – Amazon EC2, Googl e App Engine, Salesforce.com	**10**
Unit –VI	**Cloud Applications** Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages	**4**

References:

1. Michael P. Papazoglou, Web Services & SOA Principles and Technology, Second Edition,.
2. R. Nagappan, R. Skoczylas, R.P. Sriganesh, Developing Java Web Services, Wiley India.
3. S. Chatterjee, J. Webber, Developing Enterprise Web Services, Pearson Education.
4. Gautam Shroff, “Enterprise Cloud Computing” Cambri dge.
5. S. Graham and others, Building Web Services with Java, 2nd Edition, , Pearson Edn., 2008.
6. D.A. Chappell & T. Jewell, Java Web Services, O’Reilly,SPD.

7. Richard Monson-Haefel, J2EE Web Services, Pearson Education.
8. R.Mogha, V.V.Preetham,Java Web Services Programming, Wiley India Pvt.Ltd.
9. Ronald Krutz and Russell Dean Vines, “Cloud Secur ity”, Wiley-India
10. F.P.Coyle XML, Web Services, and the Data Revolution, , Pearson Education.
11. Dr. Kumar Saurabh, ”Cloud Computing”, Wiley Publication
12. Borko Furht, “Handbook of Cloud Computing”, Springer
13. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”,Cambridge University Press; 1 edition,[ISBN: 978-0521137355], 2010

PAPER CODE: CSC4207

PAPER–V: Elective- III Advanced Design and Analysis of Algorithm

[Credit -4: No. of Lectures 48]

Prerequisites:

- Strong knowledge about DAA.

Objectives:

- To Understand Advanced Algorithms
- To Understand and use Advanced data structures

	Title and Contents	No. of Lectures
Unit -I	**Advanced data structures** Advanced data structures (Fibonacci heaps, splay trees, dynamic trees, B-Trees) in-memory representations and persistence of DS, Revision of Graph algorithms: Network flows (max flow and min-cost flow/circulation)	**10**
Unit -II	**String algorithms** String searching - (Knuth–Morris–Pratt algorithm, Rabin–Karp string search algorithm) Suffix trees - mathematical properties of suffix trees Applications of Suffix trees: Regular expression searches using suffix trees; Finding all maximal pairs and maximal repeats, Patricia trees	**8**
Unit –III	**Polynomials and FFT** Representation of Polynomials, DFT and FFT, Efficient FFT implementations	**5**
Unit –IV	**Intractable problems: approximation algorithms** Steiner tree and TSP, Steiner forest, Group Steiner trees , Set cover via primal-dual k-median on a cycle	**12**
Unit –V	**Integer programming and optimization algorithms** Formulations, complexity and relaxations, discrete optimization, cutting plane methods, enumerative and heuristic methods, Convex programming algorithms: ellipsoid method, interior-point methods, proximal point methods.	**13**

References:

1. Cormen, T.H., C.E. Leiserson, R.L. Rivest, and C. Stein Introduction to Algorithms: by; MIT Press; ISBN: 9780262032933

2. Ste ven Skiena, The Algorithm Manual, Springer ISBN:9788184898651

3. Theory of Linear and Integer Programming:. ISBN: 9780471982326

4. Schrijver; John Wiley & Sons Convex Optimization: by Boyd and Vandenberghe; Cambridge University Press; ISBN: 9780521833783

5. Vazirani; Approximation Algorithms: Springer-Verlag: ISBN: 9783540653677

6. Ding-Zhu Du (Editor), J.M.Smith (Editor), J. Hyam Rubinstein (Editor); Springer; Advances in Steiner Trees (Combinatorial OptimizationISBN: 978-0792361107

7. D. Gusfield, Algorithms On Strings, Trees, And Sequences; Cambridge University Press,(ISBN 052158519)

8. Bach and Shallit, Algorithmic Number Theory: MIT Press; ISBN: 9780262024051

CSC4208: Lab Course –I (Based on DOT NET)

Practical Assignment based on DOT NET

Continuous Internal Assessment

1	Journal	10Marks
2	Viva	20Marks
3	Internal test – for skill assessment	10 Marks
4	Attendance + Active participation	10 Marks
	Total	**50 Marks**

End Semester Assessment

1.	DOT NET Programs	**50 Marks**

	PAPER CODE: CSC4208 **PAPER –VI: Lab Course –I (Based on DOT NET)** **[Credit -4: No. of Practicals 12]**
	Title of Experiment/ Practical
1	Write a program to work with StringBuilder • Create a string Assign it with large string value consisting of no of words • Access the string character by character and print • Access the string word by word and print • Find a pattern in the string such as "AB" and replace it with some other string
2	Demonstrate multicast delegate with help of following • Define a class Maths which has following methods like o int Add(int , int) o int Multiply(int , int) o int Subtract(int , int) • Define a delegate MathOp, which accepts 2 integers and returns an integer int MathOp(int , int). • Using this delegate invoke methods Add, Subtract, multiply.
3	Write a program to implement Custom Exception. Create InvalidStudentNameException class in a school application, which does not allow any **special character** or **numeric value** in a name of any of the students. Use **Regex("^[a-zA-Z]+$")** to check Student Name
4	Write a form based program offering binary calculator having following functionality • Add, multiply, subtract, divide • Left shift, right shift
5	Write a program which implements following classes a. Write a class Earth (Producer), which exposes static event EarthQuake b. Implement classes hospital, NGO who respond to EarthQuake event c. Execution: • On Click of a button on Form, EarthQuake event should be triggered. • Message should be shown that NGO and Hospital have responded to it
6	To implement reflection do following a. Implement a class library as follows. • Car class - 2 methods, 2 member variables b. Write an application (Console based) performing following tasks using reflection • Load Class library using reflection • Iterate class – types, display type details.

7	Create base class Customer and subclasses SilverCustomer and GoldCustomer • Define discount() method in Customer class which returns 20% discount Overload discount method in the subclasses and return different discount value Define base class variable as "Customer cust" Assign different objects of Customer, SilverCustomer and GoldCustomer to variable cust one after other and invoke discount method each time. What is the discount % returned each time?
8	Write a console based program to create a linked list of Employee objects using the generic class LinkedList <>. Perform following operations on the list: • Add a new employee • Display the list of employee • Total number of employee in a list Employee class has members EmpNo, EmpName, Salary, BirthDate, Addres
9	Design a form which offers User Registration Form On Click of OK, registered user data should get saved in XML
10	Write a program to create a magic square using Windows Forms? Accept square dimension from user.
11	Implement a Simple Editor which has following features • Menu : File, New, Save, Print Preview, Print • Toolbar: Formatting for Bold, Italic, Underline
12	Design a web site offering Teachers Feedback. • Clicking on save, should save data in database. • Display list of feedback entered using standard data controls e.g. GridView, DataList, ListView

PAPER CODE: CSC4209
PAPER–VI: Lab Course –II (Project)

Objectives:

The objective is to develop industry oriented projects with high end design structure. The application should be well tested with industry guidelines.

The Project can be platform, Language and technology independent. Project will be evaluated by project guide. Assessment will be done weekly in the respective batch. Evaluation will be on the basis of weekly progress of project work, progress report, oral, results and documentation and demonstration.

You should fill your status of the project work on the progress report and get the Signature of project guide regularly. Progress report should sharply focus how much time you have spent on specific task. (The format of progress report is given as follow.) You should keep all signed progress report. Project will not be accepted if progress report is not submitted and all responsibility remains with student.

Project Progress Report

Roll No & Name Of the student	
Title of the Project	
Project guide Name	

SN	From Date	To Date	Details of Project work	Project guide sign (with date)

Head,
Dept. of Computer Science

PAPER CODE: CSC4210 PAPER–VII : Emerging Technology [Credit -2: No. of Lectures 30] Objective: • To make students aware of the new emerging trends and technologies.		
	Title and Contents	**No. of Lectures**
Unit -I	**Introduction to technology**	**5**
Unit -II	**Learning Techniques**	**10**
Unit –III	**Hands On**	**15**
Note: The evaluation will be for 20 Marks.		