

# **University of Pune**

## **Two Year M.Sc. Degree Course in Computer Science**

**M.Sc. Computer Science**

**(Credit and Semester based Syllabus for affiliated colleges to be implemented  
from Academic Year 2013-14)**

### **1) Title of the Course:**

M.Sc. (Computer Science)

### **2) Preamble of the Syllabus:**

This syllabus is the extension of the existing syllabus which is currently being taught to M.Sc. (Computer Science) of University of Pune for the last few years, but modified to be placed within the credit based system to be implemented from the academic year 2013-2014. However, there are few changes incorporated in the existing syllabus.

It is believed that the proposed changes as part of the credit based system will bring a qualitative change in the way M.Sc. (Computer Science) is taught, which will offer a more enriched learning experience. It aims to provide technology-oriented students with the knowledge and ability to develop creative solutions, and better understand the effects of future developments of computer systems and technology on people and society.

The syllabus is about developing skills to learn new technology, grasping the concepts and issues behind its use and the use of computers.

### **3) Introduction:**

#### **Salient Features of the Credit System:**

1. Master's degree in Computer Science would be of 100 credits, where one credit course of theory will be of one clock hour per week running for 15 weeks and one credit for project course will consist of 15 of laboratory hours. Thus, each credit will be equivalent to 15 hours.
2. Student will have to take admission and complete at least 75 credits incorporated in the syllabus structure of Computer Science. The remaining 25 credits can be chosen from courses offered by the other Departments subjects (other than Computer Science courses) of the College with credit system structure.
3. Every student shall complete 100 credits in a minimum of four semesters. All Semesters will have 25 credits each.
4. The student will be declared as failed if s/he does not pass in all credits within a total period of four years. After that such students will have to seek fresh admission as per admission rules prevailing at that time.

5. Academic calendar showing dates of commencement and end of teaching, internal assessment tests and term end examination will be prepared and duly notified before commencement of each semester every year.
6. Project course should not be greater than 10% of the total credits of the degree course. Project course is equivalent to 10 credits.

### **Instructions for the Students**

The students seeking admission to M.Sc. Computer Science course is hereby informed that they are supposed to adhere to the following rules:

1. A minimum of 75 % attendance for lectures / practical is the pre-requisite for grant of term.
2. There shall be tutorial / practical / surprise test / home assignment / referencing of research papers / seminar / industrial visits as a part of internal assessment in each semester. The students are supposed to attend all the tests. The students should note that re-test will not be given to the student absent for the test/s.
3. The students opting for dissertation course shall follow the rules framed for the same.

### **4) Eligibility:**

The candidate should have a B.Sc. degree with Computer Science as principal subject.

**Admission** : Admissions will be given as per the selection procedure / policies adopted by the respective college, in accordance with conditions laid down by the University of Pune. Reservation and relaxation will be as per the government rules.

### **5) Examination**

#### **[A] Pattern of Examination**

Evaluation of Students:

- 1) The In-semester and End-Semester examinations will be of 50 marks each.
- 2) Student has to obtain 40% marks in the combined examination of In-Semester and End-Semester assessment with minimum passing of 30% passing in both assessments separately.
- 3) A student cannot register for third semester if s/he fails to complete the 50% credits of the total expected within two semesters.
- 4) Internal marks will not change. Student cannot repeat internal assessment. If student misses internal assessment examination, s/he will have second chance with the permission of the concerned teacher. But it will not be right of the student. It will be

the discretion of the concerned teacher and internal departmental assessment committee.

- 5) There shall be revaluation of answer script of end semester examination, but not of internal assessment papers.
- 6) Internal assessment answer scripts may be shown to the concerned student but not end semester answer script.

**i. Continuous Assessment:** Internal assessment for each course would be continuous and dates for each tutorials/practical tests will be pre-notified in the time table for teaching or placed separately as a part of time table. Department / College Internal Assessment Committee will coordinate this activity

**a) Theory Courses:** Conducting written tests should not be encouraged. More focus should be on non-written tests. Students should be encouraged to conduct various academic activities. A teacher must select a variety of the procedures for internal assessment suggested as follows.

- a) Mid-term test
- b) On-line test
- c) Open book test (concerned teacher will decide the allowed books)
- d) Tutorial
- e) Surprise test
- f) Oral
- g) Theory Assignments
- h) Review of Research paper
- i) Seminar presentation
- j) Journal/Lecture/Library notes
- k) Group Discussion
- l) Programming Assignments

Student has to preserve the documentation of the internal assessment except midterm test answer script. It is the responsibility of the student to preserve the documents.

**Project Courses :** 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.

**ii. University Examination :** End-Semester examination for 50 marks per course would be held as per the scheduled given by University of Pune..

### **[B] Standard of Passing**

Student has to obtain 40% marks in the combined examination of In-Semester and End-Semester assessment with minimum passing of 30% passing in both assessments separately.

### **[C] ATKT Rules**

A student cannot register for third semester if s/he fails to complete the 50% credits of the total credits expected to be ordinarily completed within two semesters.

### **[D] Award of Class**

Grades will be awarded from grade point average (GPA) of the credits.

#### **GPA Rules:**

1. The formula for GPA will be based on Weighted Average. The final GPA will not be printed unless a student passes courses equivalent to minimum 100 credit hours (Science). Total credits hours means the sum of credit hours of the courses which a student has passed.
2. A seven point grade system [guided by the Government of Maharashtra Resolution No. NGO – 1298 / [4619] / UNI 4 dt. December 11, 1999 and University regulations] will be followed. The corresponding grade table is attached herewith.
3. If the GPA is higher than the indicated upper limit in the third decimal digit then the student be awarded higher final grade (e.g. a student getting GPA of 4.492 may be awarded 'A')
4. For Semester I, II, III examinations, only the grade points will be awarded for each subject. Final GPA along with final grade will be awarded only at the end of IV semester. There is also a provision for verification and revaluation. In case of verification, the existing rules will be applicable. The revaluation result will be adopted if there is a change of at least 10% marks and in the grade of the course.
5. After the declaration of result, for the improvement of Grade, the student can reappear for the examination of 30 credits worth theory courses.
6. Grade improvement programme will be implemented at the end of the academic year. A student can opt for grade improvement programme only after the declaration of final semester examination i.e. at the end of next academic year after passing M.Sc. (Computer Science) examination and within two years of completion of M.Sc. (Computer Science). A student can appear for grade improvement programme only once.

Grade and Grade Point Average		
Marks	Obtained Grade	Grade Points
100 – 75	'O' Outstanding	06
74 – 65	'A' Very Good	05
64 – 55	'B' Good	04
54 – 50	'C' Average	03
49 – 45	'D' Satisfactory	02
44 – 40	'E' Pass	01
39 and less	'F' Fail	00

Final Grade Points	
Grade Points	Final Grade
5.00 – 6.00	O
4.50 – 4.99	A
3.50 – 4.49	B
2.50 – 3.49	C
1.50 – 2.49	D
0.50 – 1.49	E
0.00 – 0.49	F

Common Formula for Grade Point Average (GPA):

$$\text{GPA} = \frac{\text{Total of Grade Points earned} \times \text{Credit hours for each course}}{\text{Total Credit hours}}$$

B Grade is equivalent to at least 55% of the marks

**[E] External Students:** There shall be no external students.

**[F] Setting of Question Paper / Pattern of Question Paper**

For core (compulsory) theory courses end semester question papers set by the University of Pune and centralized assessment for theory papers done as per the University guidelines.

**[G] Verification / Revaluation**

There is also a provision for verification and revaluation. In case of verification, the existing rules will be applicable. There shall be revaluation of end semester examination, but not of internal assessment.

## 6) Structure of Course

- Duration : The entire Programme is a Two year and four semester full time Programme.
- No of Courses : For first three semesters there will be Five courses. The fourth semester will be Industrial Training/Institutional Project and two theory courses.

Year/ Semester	Subject	Paper	Title of Paper	Credit	% of Assessment		
					IA	UE	Total
I Year Sem-I	Core	CS-101	Principles of Programming Languages	5	50	50	100
	Core	CS-102	Advanced Networking	5	50	50	100
	Core	CS-103	Distributed Database Concepts	5	50	50	100
	Core	CS-104	Design and Analysis of Algorithms	5	50	50	100
	Core	CS-105	Network Programming	5	50	50	100

Minimum Credit : 25 Maximum Credit : 25. Core Subject is compulsory . IA :- Internal Assessment, UE :- University Examination

Year/ Semester	Subject	Paper	Title of Paper	Hours/ Weak	Credit	% of Assessment		
						IA	UE	Total
I Year Sem-II	Core	CS-201	Digital Image Processing	4	5	50	50	100
	Core	CS-202	Advanced Operating Systems	4	5	50	50	100
	Core	CS-203	Data Mining and Data Warehousing	4	5	50	50	100
	Core	CS-204	Project	4	5	50	50	100
	Elective	CS-205	Programming With DOT NET	4	5	50	50	100
	Elective	CS-206	Artificial Intelligence	4	5	50	50	100
	Elective	CS-207	Advance Design and Analysis of Algorithms	4	5	50	50	100

Minimum Credit : 25 Maximum Credit : 30. Core Subject is compulsory. From elective courses student can select one course for minimum credit and two for maximum credit. IA :- Internal Assessment, UE :- University Examination

Year/ Semester	Subject	Paper	Title of Paper	Credit	% of Assessment		
					IA	UE	Total
II Year Sem-III	Core	CS-301	Software Metrics & Project Management	5	50	50	100
	Core	CS-302	Mobile Computing	5	50	50	100
	Core	CS-303	Soft Computing	5	50	50	100
	Elective	CS-304	Project	5	50	50	100
	Elective	CS-305	Web Services	5	50	50	100
	Elective	CS-306	Database and System Administrator	5	50	50	100
	Elective	CS-307	Functional Programming	5	50	50	100
	Elective	CS-308	Business Intelligence	5	50	50	100

Minimum Credit : 25    Maximum Credit : 35, Core Subject is compulsory, From elective courses student can select two course for minimum credit and four for maximum credit. IA :- Internal Assessment, UE :- University Examination

Year/ Semester	Subject	Paper	Title of Paper	Credit	% of Assessment		
					IA	UE	Total
II Year Sem-IV	Core	CS-401	Industrial Training /Institutional project	15	50	50	100
	Elective	CS-402	Parallel Computing	5	50	50	100
	Elective	CS-403	Embedded System	5	50	50	100
	Elective	CS-404	Software Quality Assurance	5	50	50	100
	Elective	CS-405	Modeling and Simulation	5	50	50	100

Core Subject is compulsory. If student had completed 85 credit within three semesters then no need to select any elective course otherwise student should select appropriate number of elective courses to minimum complete 100 credits.

IA :- Internal Assessment, UE :- University Examination

### 7) Equivalence of Previous Syllabus:

Not Applicable

### 8) University Terms:

### 9) Qualification of Teacher:

### 10) Detail Syllabus with Recommended Books



**M.Sc.**  
**(Computer Science)**

**First Year Semester 1**

## CS-101(New): Principles of Programming Languages

[Total Lectures: 48 Hours]

### Course 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.

### Why to study this course?

- To allow Informed Design Decisions
- Gives insight when debugging
- Permits effective use of compilers/linkers interpreters and language oriented tools.
- Helps to understand how language features work.
- Learn features, emulate missing features.
- Develop a greater understanding of the issues involved in programming language design and implementation
- Develop an in-depth understanding of functional, logic, and object-oriented programming paradigms
- Implement several programs in languages other than the one emphasized in the core curriculum (Java/C++)
- Understand design/implementation issues involved with variable allocation and binding, control flow, types, subroutines, parameter passing
- Develop thorough understanding of the compilation process
- To introduce several different paradigms of programming
- To gain experience with these paradigms by using example programming languages
- To understand concepts of syntax, translation, abstraction, and implementation

### Course 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

### Unit 1. Introduction [ T1 chap. 1] [2]

- The Art of Language Design [ T1 1.1]
- The Programming Language Spectrum [ T1 1.2]
- Why Study Programming Languages? [ T1 1.3]
- Compilation and Interpretation [ T1 1.4]
- Programming Environments [ T1 1.5]

### Unit 2. Non-Imperative Programming Models: Functional, Logic Languages

[ Text books 3, 4]

[10]

### 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

[T 4]

### Turbo Prolog

Introduction, facts, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates

[ T3 chapter 1 through 9 except chapter 2 ]

### Unit 3. Names, Scopes, and Bindings [ T1 chap.3]

[5]

The Notion of Binding Time [ T1 chap.3.1]

Object Lifetime and Storage Management : [ T1 chap. 3.2]

Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection

Scope Rules T1 chap. 3.3]

Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping

The meaning of Names in a Scope [ T1 chap. 3.5]

Aliases, Overloading, Polymorphism and Related Concepts

The Binding of Referencing Environments [ T1 chap. 3.6]

Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures

Macro Expansion [ T1 chap. 3.7]

### Unit 4. Control Flow [ T1 chap.6] [5]

Expression Evaluation [ T1 6.1]

Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation

Structured and Unstructured Flow [ T1 6.2]

Structured Alternatives to goto

Sequencing [ T1 6.3]

Selection [ T1 6.4]

Short-Circuited Conditions, Case/Switch Statements

Iteration [ T1 6.5]

Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops

Recursion [ T1 6.6]

Iteration and Recursion, Applicative- and Normal-Order Evaluation

### Unit 5. Data Types [ T2 chap.6]

[8]

Introduction T2 6.1]

Primitive Data Types [T2 6.2]

Numeric Types [T2 6.2.1]

Integer [T2 6.2.1.1]

Floating point [T2 6.2.1.2]

- Complex [T2 6.2.1.3]
- Decimal [T2 6.2.1.4]
- Boolean Types [T2 6.2.2]
- Character Types [T2 6.2.3]
- Character String Types [T2 6.3]
  - Design Issues [T2 6.3.1]
  - Strings and Their Operations [T2 6.3.2]
  - String Length Operations [T2 6.3.3]
  - Evaluation [T2 6.3.4]
  - Implementation of Character String Types [T2 6.3.5]
- User defined Ordinal types [T2 6.4]
  - Enumeration types [T2 6.4.1]
    - Designs
    - Evaluation
  - Subrange types [T2 6.4.2]
    - Ada's design
    - Evaluation
  - Implementation fo used defined ordinal types [T2 6.4.3]
- Array types [T2 6.5]
  - Design issues [T2 6.5.1]
  - Arrays and indices [T2 6.5.2]
  - Subscript bindings and array categories [T2 6.5.3]
  - Heterogeneous arrays [T2 6.5.4]
  - Array initialization [T2 6.5.5]
  - Array operations [T2 6.5.6]
  - Rectangular and Jagged arrays [T2 6.5.7]
  - Slices [T2 6.5.8]
  - Evaluation [T2 6.5.9]
  - Implementation of Array Types [T2 6.5.10]
- Associative Arrays [T2 6.6]
  - Structure and operations [T2 6.6.1]
  - Implementing associative arrays [T2 6.6.2]
- Record types [T2 6.7]
  - Definitions of records [T2 6.7.1]
  - References to record fields [T2 6.7.2]
  - Operations on records [T2 6.7.3]
  - Evaluation [T2 6.7.4]
  - Implementation of Record types [T2 6.7.5]
- Union Types [T2 6.8]
  - Design issues [T2 6.8.1]
  - Discriminated versus Free unions [T2 6.8.2]
  - Evaluation [T2 6.8.4]
  - Implementation of Union types [T2 6.8.5]
- Pointer and Reference Types [T2 6.9]
  - Design issues [T2 6.9.1]
  - Pointer operations [T2 6.9.2]
  - Pointer problems [T2 6.9.3]

- Dangling pointers
- Lost heap dynamic variables
- Pointers in C and C++ [T2 6.9.5]
- Reference types [T2 6.9.6]
- Evaluation [T2 6.9.7]
- Implementation of pointer and reference types [T2 6.9.8]
  - Representation of pointers and references
  - Solution to dangling pointer problem
- Heap management

**Unit 6. Subroutines and Control Abstraction [ T2 chap.9,10] [5]**

- Fundamentals of Subprograms [ T2 9.2 (excluding 9.2.4)]
- 1Design Issues for subprograms [ T2 9.3]
- Local Referencing Environments [ T2 9.4]
- Parameter-Passing Methods [ T2 9.5]
- Parameters That are Subprograms [ T2 9.6]
- Overloaded Subprograms [ T2 9.7]
- Generic Subroutines [ T2 9.8]
  - Generic Functions in C++ [ T2 9.8.2]
  - Generic Methods in Java [ T2 9.8.3]
- Design Issues for Functions [ T2 9.9]
- User-Defined Overloaded Operators [ T2 9.10]
- Coroutines [ T2 9.10]
- The General Semantics of Calls and Returns [ T2 10.1]
- Implementing “Simple” Subprograms [ T2 10.2]
- Implementing Subprograms with Stack-Dynamic Local Variables [ T2 10.3]
- Nested Subprograms [ T2 10.4]
- Blocks [ T2 10.5]
- Implementing Dynamic Scoping [ T2 10.6]

**Unit 7. Data Abstraction and Object Orientation [ T1 chap.9] [8]**

- Object-Oriented Programming [ T1 9.1]
- Encapsulation and Inheritance [ T1 9.2]
  - Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance
- Initialization and Finalization [ T1 9.3]
  - Choosing a Constructor, References and Values, Execution Order, Garbage Collection
- Dynamic Method Binding [ T1 9.4]
  - Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures
- Multiple Inheritance [ T1 9.5]
  - Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance

**Unit 8. Concurrency [T2 chap. 13] [5]**

- Introduction
  - Multiprocessor Architecture [T2 13.1.1]
  - Categories of concurrency [T2 13.1.2]
  - Motivations for studying concurrency [T2 13.1.3]

- Introduction to Subprogram-level concurrency
  - Fundamental concepts [T2 13.2.1]
  - Language Design for concurrency. [T2 13.2.2]
  - Design Issues [T2 13.2.3]
- Semaphores
  - Introduction [T2 13.3.1]
  - Cooperation synchronization [T2 13.3.2]
  - Competition Synchronization [T2 13.3.3]
  - Evaluation [T2 13.3.4]
- Monitors
  - Introduction [T2 13.4.1]
  - Cooperation synchronization [T2 13.4.2]
  - Competition Synchronization [T2 13.4.3]
  - Evaluation [T2 13.4.4]
- Message Passing
  - Introduction [T2 13.5.1]
  - The concept of Synchronous Message Passing [T2 13.5.2]
- Java Threads
  - The **Thread** class [T2 13.7.1]
  - Priorities [T2 13.7.2]
  - Competition Synchronization [T2 13.7.3]
  - Cooperation Synchronization [T2 13.7.4]
  - Evaluation [T2 13.4.5]

**Text Books:**

T1. Scott Programming Language Pragmatics, 3e(With CD) ISBN 9788131222560  
Kaufmann Publishers, An Imprint of Elsevier, USA

T2. Concepts of Programming Languages, Eighth Edition by Robert W. Sebesta,  
Pearson Education.

T3. Introduction to Turbo Prolog by Carl Townsend

T4. LISP 3rd edition by Patrick Henry Winston & Berthold Klaus Paul Horn (BPB)

**Additional Reading:**

Programming Languages: Principles and Paradigms, M. Gabbrielli, S. Martini, Springer,  
ISBN: 9781848829138

<b>CS102 (New) - Advanced Networking</b>	
<b>Unit 1. Review of Basic Concepts</b>	<b>[3]</b>
TCP/IP Protocol Suite [T1 2.3]	
Underlying Technologies : LAN (802.3) T 1 3.1	
Wireless Lans (802.11) T 1 3.2	
Point-to-point WANS T 1 3.3	
Switched WANS T 1 3.4	
<b>Unit 2. The Internet Layer Protocols</b>	<b>[4]</b>
Review of IPv4 Protocol T 1 7.1,7.2,7.3,7.4,7.5	
IPv6 T 1 27.1,27.2	
Transition from IPv4 to IPv6 T 1 27.3	
ICMPv4 T 1 9.1,9.2,9.3,9.4	
ICMPv6 T 1 28.1,28.2,28.3,28.4	
<b>Unit 3. Routing Protocols</b>	<b>[6]</b>
Forwarding T 1 6.2	
Structure of a Router T 1 6.3	
Routing Tables T 1 11.1	
Intra – And Inter-Domain Routing T 1 11.2	
Distance Vector Routing T 1 11.3	
RIP T 1 11.4	
OSPF T 1 11.6	
BGP T 1 11.8	
Multicast Routing T 1 .4	
<b>Unit 4. The Transport Layer</b>	<b>[6]</b>
The Transport Service T 2 6.1	
Elements of Transport Protocols T 2 6.2	
UDP T 2 6.4.1	
TCP T 2 6.5.1 to 6.5.9	
<b>Unit 5. Multimedia</b>	<b>[3]</b>
Digitizing Audio and Video T 1 25.2	
Streaming stored Audio / Video T 1 25.4	
Streaming Live Audio / Video T 1 25.5	
Real-Time Interactive Audio / Video T 1 25.6	
RTP T 1 25.7	
RTCP T 1 25.8	
Voice Over IP T 1 25.9	
<b>Unit 6. Introduction To Security</b>	<b>[2]</b>
The need for Security T 3 1.2	
Security Approaches T 3 1.3	
Principles of Security T 3 1.4	
Types of Attacks T 3 1.5	
<b>Unit 7. Cryptography: Concepts and Techniques</b>	<b>[3]</b>
Introduction T 3 2.1	
Plain Text and Cipher Text T 3 2.2	
Substitution Techniques T 3 2.3.1,2.3.2,2.3.3,2.3.7	
Transposition Techniques T 3 2.4.1,2.4.2,2.4.3	
Symmetric and Asymmetric key cryptography	T 3 2.6.1,2.6.2

<b>Unit 8. Symmetric Key Algorithms</b>	<b>[3]</b>
Algorithms types and modes T 3 3.2.1,3.2.2	
DES T 3 3.4	
<b>Unit 9. Asymmetric key Algorithms</b>	<b>[2]</b>
RSA T 3 4.4	
Symmetric and Asymmetric key Cryptography T 3 4.5	
Digital Signatures T 3 4.6.1,4.6.2	
<b>Unit 10. Digital Certificates</b>	<b>[2]</b>
Introduction T 3 5.1	
Digital Certificates T 3 5.2	
<b>Unit 11. Internet Security Protocols</b>	<b>[10]</b>
Secure Socket Layer T 3 6.3	
TLS T 3 6.4	
SHTTP T 3 6.5	
TSP T 3 6.6	
SET T 3 6.7	
SSL Verses SET T 3 6.8	
3-D Secure Protocol T 3 6.9	
Electronic Money T 3 6.10	
Email Security T 3 6.11	
Firewalls T 3 9.3	
IP Security T 3 9.4	
VPN T 3 9.5	
<b>Unit 12. User Authentication</b>	<b>[4]</b>
Passwords T 3 7.3	
Certificate-based Authentication T 3 7.5	
Kerberos T 3 7.7	
Security Handshake Pitfalls T 3 7.9	

**Text Books:**

**T1 : TCP / IP Protocol Suite Fourth Edition – Behrouz A. Forouzan**

**T2 : Computer Networks Fourth Edition – Andrew Tanenbaum**

**T3 : Cryptography and Network Security Second Edition – Atul Kahate**

**Supplementary but very useful references/texts:** (Few of the references below contain latest research and trends related to Networks and Security and are useful for seminar/ presentations by the students.)

1. Computer Network Security, Kizza, Springer, 9780387204734
2. Guide to Computer Network Security, Kizza, Springer, 978-1-84800-916-5
3. Network Security, Harrington, Elsevier, ISBN 9788131202166
4. Douglas E. Comer, Internetworking with TCP/IP, Vol. 1, Principles, Protocols and Architecture Fifth Edition, Prentice Hall, 2000, ISBN 0-13-018380-6.
5. William Stallings, Data and Computer Communications , Seventh Edition, Pearson Education
6. Douglas E. Comer, Internetworking with TCP/IP, Vol. 2, Design, Implementation and Internals, Prentice Hall Publisher.



7. Internetworking with TCP/IP, Vol. 3, Client-server Programming and Applications by Douglas E. Comer, Prentice Hall Publisher. (Excellent reference for distributed programming over TCP/IP networks)
8. Richard Stevens, TCP/IP Illustrated, Vol. 1, by, Addison Wesley (A very practical book with lots of useful network diagnostic tools and programs.)
9. Craig Hunt, TCP/IP Network Administration O'Reilly & Associates, Inc. (A must for network and system administrators dealing with internetworking.)
10. L. Peterson and B. Davie. Morgan , Computer Networks: A Systems Approach by Kaufmann Publishers Inc., ISBN 9788131210451
11. J. Kurose, K. Ross ``Computer Networking: A Top-Down Approach Featuring the Internet" Addison-Wesley, '00
12. William Stallings," Cryptography And Network Security" Prentice Hall /Pearson Education

**Guidelines to paper setters:**

Frame formats of protocols are not expected

Problems should be asked on Routing Protocols , TCP, Cryptography, RSA

## CS-103(New): Distributed Database Concepts

**Pre-requisites:** 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.

- Unit 1. Distributed databases: An overview** [2]
  - 1.1 Features of distributed Vs centralized databases Chapter 1 from Book 2
  - 1.2 Why DDB? DDBMS
  - 1.3 Promises / problem areas in implementing a DDB Section 1.3,1.5 from Book 1
- Unit 2. DDBMS Architecture** [4]
  - 2.1 DBMS Standardization Chapter 4 from Book 1
  - 2.2 Architectural models for DDBMS
  - 2.3 DDBMS architecture
  - 2.4 Distributed catalog management Section 21.8 from Book 3
- Unit 3. Distributed database design** [10]
  - 3.1 Alternative design strategies Chapter 5 from book 1
  - 3.2 Distributed design issues
  - 3.3 Concepts of join graphs Section 4.2.1.2 from book 2
  - 3.4 Fragmentation and allocation Chapter 5 from Book1
- Unit 4. Overview of Query processing** [4]
  - 4.1 Query processing problems
  - 4.2 Objectives of query processing Chapter 7 from book 1
  - 4.3 Complexity of relational algebra operators
  - 4.4 Characterization of query processors
  - 4.5 Layers of query processing
- Unit 5. Query decomposition & data localization** [2]
  - 5.1 Query decomposition
  - Chapter 5.2 Localization of distributed data 8 from book 1
- Unit 6. Optimization of distributed queries** [10]
  - 6.1 Query optimization
    - Centralized query optimization Join ordering in Chapter 9 from book1
    - fragment queries. Distributed query optimization algorithms
  - 6.2 Centralized query optimization
  - 6.3 Join ordering in fragment queries
  - 6.4 Distributed query optimization algorithms
- Unit 7. Management of distributed transactions** [2]
  - 7.1 Framework for transaction management Chapter 7 from book 2
  - 7.2 Supporting atomicity of distributed transactions
  - 7.3 Concurrency control of distributed transactions
  - 7.4 Architectural aspects of distributed transactions
- Unit 8. Concurrency control** [6]
  - 8.1 Foundations of distributed concurrency control Chapter 8 from book 2
  - 8.2 Distributed deadlocks

- 8.3 Concurrency control based on timestamps
- 8.4 Optimistic methods for distributed concurrency control

**Unit 9. Distributed DBMS reliability**

**[8]**

- 9.1 Reliability concepts & measures
- 9.2 Failures & fault tolerance in distributed systems from book 1
- 9.3 Failures in DDBMS
- 9.4 Local reliability protocols
- 9.5 Distributed reliability protocols
- 9.6 Dealing with site failures
- 9.7 Network partitioning

**Reference Books:**

1. Principles of Distributed Database Systems; 2nd Edition By M. Tamer Ozsu and Patrick Valduriez Publishers: Pearson Education Asia ISBN: 81-7808-375-2
2. Distributed Database; Principles & Systems By Stefano Ceri and Giuseppe Pelagatti Publications: McGraw-Hill International Editions ISBN: 0-07-010829-3
3. Database systems (2nd edition) By Raghuramakrishnan and Johannes

## CS-104(New): Design and Analysis of Algorithms

### 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 of 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

### Unit 1. 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).

[T1 1.1 , 1.2, 1.3 ] [6]

### Unit 2. Design strategies

**Divide and conquer**-control abstraction, binary search, merge sort, Quick sort, Strassen's matrix multiplication [ T1 3.1, 3.2, 3.4,3.5,3.7] [6]

**Unit 3. 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 [ T1 4.1, 4.2, 4.4, 4.5, 4.6,4.7, 4.8] [8]

**Unit 4. 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, Traveling salesperson problem.

[T1 5.1, 5.3, 5.6, 5.7, 5.9] [8]

**Unit 5. Decrease and conquer**: - DFS and BFS, Topological sorting, connected components

[T6.1, 6.2, 6.3, 6.4] [6]

**Unit 6. Backtracking**: General method, 8 Queen's problem, Sum of subsets problem, graph coloring problem, Hamiltonian cycle

[T1 7.1 , 7.2, 7.3, 7.4, 7.5] [4]

**Unit 7. Branch and Bound Technique** : FIFO, LIFO, LCBB, TSP problem, 0/1 knapsack problem

[T1 8.1.1, 8.2, 8.3 ] [4]

**Unit 8. Transform and conquer**:- Horner's Rule and Binary Exponentiation – Problem Reduction –

[T1 9.1, 9.2 ,9.3] [4]

### Unit 9. Problem classification

Nondeterministic algorithm, The class of P, NP, NP-hard and NP- Complete problems, significance of Cook's theorem

[T1 11.1] [2]

### Text Books

**T1.** Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, Computer Algorithms, Galgotia.

**T2** T. Cormen, C. Leiserson, & R. Rivest, Algorithms, MIT Press, 1990

### References Texts

1) A. Aho, J. Hopcroft, & J. Ullman, The Design and Analysis of Computer Algorithms,

M.Sc.(CS) syllabus for affiliated colleges

- Addison Wesley, 1974
- 2) Donald Knuth, The Art of Computer Programming (3 vols., various editions, 1973-81),  
Addison Wesley
  - 3) The Algorithm Manual, Steven Skiena, Springer ISBN:9788184898651
  - 4) Graphs, Networks and Algorithms, Jungnickel, Springer, ISBN: 3540219056

## CS-105 (New) : Network Programming

### Prerequisites:

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

### Syllabus:

[Total Lectures: 48]

#### UNIT 1: Introduction

[2]

- A Simple Daytime Client, Protocol Independence, Error Handling: Wrapper Functions, A Simple Daytime Server [Book-1]

#### UNIT 2: Sockets Introduction

[6]

- Socket Address Structures, 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 [Book-1]
- What is a Socket?, Using Sockets [Book-2]

#### UNIT 3: Elementary TCP Sockets

[4]

- socket Function, connect Function, bind Function, listen Function, accept Function, fork and exec Functions, Concurrent Servers, close Function, getsockname and getpeername Functions [Book-1]

#### UNIT 4: TCP Client-Server Example

[6]

- 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 [Book-1]

#### UNIT 5: I/O Multiplexing: The select and poll Functions

[6]

- I/O Models, select Function, str\_cli Function (Revisited), Batch Input, shutdown Function, str\_cli Function (Revisited Again), TCP Echo Server (Revisited), pselect Function, poll Function, TCP Echo Server (Revisited Again) [Book-1]

#### UNIT 6: Socket Options

[4]

- 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, IPv6 Socket Options, TCP Socket Options [Book-1]

**UNIT 7: Elementary UDP Sockets****[8]**

- 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 [Book-1]
- User Datagram Protocol, File Transfer, Error Handling [Book-2]

**UNIT 8: Protocols, Sessions, State, and Implementing Custom Protocols****[4]**

- State vs. Stateless, Methods for Maintaining State, What Is a Protocol?, Designing a Custom Protocol, Our Chat Protocol, Protocol Registration [Book-2]

**UNIT 9: Elementary Name, Address Conversions and design decisions****[8]**

- 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 [Book-1]
- TCP vs. UDP, Application Protocol Choices, Client-Server Architecture, Client-Side Considerations, Server-Side Considerations [Book-2]

**References:**

**T1: Unix Network Programming, Volume 1: The Sockets Networking API, 3/E by W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, PHI**

**T2: The Definitive Guide to Linux Network Programming by KEIR DAVIS, JOHN W. TURNER, AND NATHAN YOCOM, Apress.**

**M.Sc.**  
**(Computer Science)**

**First Year Semester 2**



## CS-201: Digital Image Processing

### Syllabus:

[Total Lectures: 48]

#### UNIT 1. Introduction

[3]

- What is Digital Image Processing?
- The origins of Digital Image Processing
- Examples of Fields that use Digital Image Processing
  - Gamma-Ray Imaging
  - X-Ray Imaging
  - Imaging in the Ultraviolet Band
  - Imaging in the Visible and Infrared Bands
  - Imaging in the Microwave Band
  - Imaging in the Radio Band
- Fundamental steps in Digital Image Processing
- Components of an Image Processing System

#### UNIT 2. Digital Image Fundamentals

[6]

- Elements of Visual Perception
- Light and the Electromagnetic Spectrum
- Image sensing and Acquisition
- Image Sampling and Quantization
- Some Basic Relationships between Pixels
- An Introduction to the Mathematical Tools Used in Digital Image Processing
  - Array versus Matrix Operations
  - Linear versus Nonlinear Operations
  - Arithmetic Operations
  - Set and Logical Operations

#### UNIT 3. Intensity Transformation and Spatial Filtering

[7]

- Background
- Some Basic Intensity Transformation Functions
- Histogram Processing
  - Histogram Equalization
  - Histogram Matching (Specification)
  - Local Histogram Processing
- Fundamentals of Spatial Filtering
- Smoothing Spatial Filters
- Sharpening Spatial Filters
- Combining Spatial Enhancement Methods

#### UNIT 4. Filtering in the Frequency Domain

[10]

- Background
- Preliminary Concepts
- Sampling and the Fourier Transform of Sampled Functions
- The Discrete Fourier Transform (DFT) of One variable
- Extension to Functions of Two Variables

- Some Properties of the 2-D Discrete Fourier Transform
- The Basics of Filtering in the Frequency Domain
- Image Smoothing Using Frequency Domain Filters
- Image Sharpening Using Frequency Domain Filters
- Selective Filtering

UNIT 5. Image Restoration and Reconstruction

[6]

- A Model of the Image Degradation / Restoration Process
- Noise Models
- Restoration in the Presence of Noise Only- Spatial Filtering
- Periodic Noise Reduction by Frequency Domain Filtering
  - Bandreject Filters
  - Bandpass Filters
  - Notch Filters
- Estimating the Degradation Function
- Inverse Filtering
- Minimum Mean Square Error(Wiener) Filtering
- Geometric Mean Filter

UNIT 6. Morphological Image Processing

[5]

- Preliminaries
- Erosion and Dilation
- Opening and Closing
- The Hit-or-Miss Transformation
- Some Basic Morphological Algorithms
  - Boundary Extraction
  - Hole Filling
  - Extraction of Connected Components
  - Convex Hull
  - Thinning
  - Thickening
  - Skeletons
  - Pruning
  - Morphological Reconstruction

UNIT 7. Image Segmentation

[7]

- Fundamentals
- Point, Line, and Edge Detection
  - Background
  - Detection of Isolated Points
  - Line Detection
  - Edge Models
  - Basic Edge Detection
  - Edge Linking and Boundary Detection
- Thresholding
  - Foundation
  - Basic Global Thresholding
  - Optimum Global Thresholding Using Otsu's Method

- Using Image Smoothing to Improve Global Thresholding
- Using Edges to Improve Global Thresholding
- Region-Based Segmentation

UNIT 8. Representation and Description

[4]

- Representation
  - Boundary (Border) Following
  - Chain Codes
  - Polygonal Approximations Using Minimum-Perimeter Polygons
  - Other Polygonal Approximation Approaches
  - Signatures
  - Boundary Segments
  - Skeletons
- Boundary Descriptors
  - Some Simple Descriptors
  - Shape Numbers
  - Fourier Descriptors
- Regional Descriptors
  - Some Simple Descriptors
  - Topological Descriptors
  - Texture

**Text Book:**

**1. Gonzalez, R. C. and Woods, R. E. [2002/2008], Digital Image Processing, 3rd ed., Prentice Hall**

Reference Books:

1. Sonka, M., Hlavac, V., Boyle, R. [1999]. Image Processing, Analysis and Machine Vision (2nd edition), PWS Publishing, or (3rd edition) Thompson Engineering, 2007
2. Gonzalez, R. C., Woods, R. E., and Eddins, S. L. [2009]. Digital Image Processing Using MATLAB, 2nd ed., Gatesmark Publishing, Knoxville, TN.
3. Anil K. Jain [2001], Fundamentals of digital image processing (2nd Edition), Prentice-Hall, NJ
4. Willian K. Pratt [2001], Digital Image Processing (3rd Edition), , John Wiley & Sons, NY
5. Burger, Willhelm and Burge, Mark J. [2008]. Digital Image Processing: An Algorithmic Introduction Using Java, Springer
6. Digital Image Analysis (With CD-ROM), Kropatsch, Springer, ISBN 978038795066
7. Digital Image Processing, 6e (With CD), Jähne, Springer, ISBN:978-3-540-24035-8 2

## CS-202(New): Advanced Operating Systems

### Prerequisites:

- Working knowledge of C programming.
- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts.

### Course Objectives:

This course teaches Advanced Operating Systems Concepts using Unix/Linux and Windows as representative examples. This course strikes a delicate balance between theory (covered in TextBook-2, 3) and practical applications (covered in TextBook-1, 4). In fact, most Units start with the theory and then switches focus on how the concepts are implemented in a C program. This course describes the programming interface to the Unix/Linux system - the system call interface. It is intended for anyone writing C programs that run under Unix/Linux. Finally, it concludes with an overview of Windows Threads Management. This course provides an understanding of the functions of Operating Systems. It also provides provide an insight into functional modules of Operating Systems. It discusses the concepts underlying in the design and implementation of Operating Systems.

### Syllabus:

#### Unit 1. Introduction to UNIX/Linux Kernel

[03]

- System Structure, User Perspective, Assumptions about Hardware, Architecture of UNIX Operating System (TextBook-3: Chapter Topics: 1.2, 1.3, 1.5, 2.1)
- Concepts of Linux Programming- Files and the Filesystem, Processes, Users and Groups, Permissions, Signals, Interprocess Communication (TextBook-1: Chapter 1- relevant topics)

#### Unit 2. File and Directory I/O

[13]

- Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, inodes, structure of regular file, open, read, write, lseek, close, pipes, dup (TextBook- 3: Chapter Topics: 3.1-3.4, 4.1, 4.2, 5.1-5.3, 5.5-5.7, 5.12, 5.13)
- open, creat, file sharing, atomic operations, dup2, sync, fsync, and fdatasync, fcntl, /dev/fd, stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod, sticky bit, chown, fchown, and lchown, file size, file truncation, file systems, link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions, file times, utime, mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files (TextBook-4: Chapter Topics: 3.3, 3.4, 3.10-3.14, 3.16, 4.2-4.23)
- Scatter/Gather I/O, Mapping Files into Memory, Advice for Normal File I/O, I/O Schedulers and I/O Performance, Directories, Copying and Moving files, Device Nodes, Out-of-Band Communication (TextBook-1: Chapters: 4 and 7-relevant topics)

Unit 3. Process Environment, Process Control and Process Relationships [14]

- Process states and transitions, layout of system memory, the context of a process, saving the context of a process, sleep, process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, changing the size of the process, The Shell, Process Scheduling (TextBook-3: Chapter Topics: 6.1-6.4, 6.6, 7.1-7.8, 8.1)
- Process termination, environment list, memory layout of a C program, shared libraries, environment variables, setjmp and longjmp, getrlimit and setrlimit, process identifiers, fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4, race conditions, exec, changing user IDs and group IDs, system function, user identification, process times (TextBook-4: Chapter Topics: 7.3, 7.5-7.7, 7.9-7.11, 8.2-8.11, 8.13, 8.15, 8.16)
- The Process ID, Running a New Process, Terminating a Process, Waiting for Terminated Child Processes, Users and Groups, Daemons, Process Scheduling, Yielding the Processor, Process Priorities, Processor Affinity (TextBook-1: Chapter 5 and 6 [Relevant Topics])

Unit 4. Memory Management [06]

- The Process Address Space, Allocating Dynamic Memory, Managing Data Segment, Anonymous Memory Mappings, Advanced Memory Allocation, Debugging Memory Allocations, Stack-Based Allocations, Choosing a Memory Allocation Mechanism, Manipulating Memory, Locking Memory, Opportunistic Allocation (TextBook-1: Chapter 8)
- Swapping, Demand Paging (TextBook-3: Chapter Topics: 9.1, 9.2)

Unit 5. Signal Handling [06]

- Signal concepts, signal function, unreliable signals, interrupted system calls, reentrant functions, SIGCLD semantics, reliable-signal technology, kill and raise, alarm and pause, signal sets, sigprocmask, sigpending, sigsetjmp and siglongjmp, sigsuspend, abort, system function revisited, sleep (TextBook-4: Topics: 10.2-10.13, 10.15-10.19)
- Signal Concepts, Basic Signal Management, Sending a Signal, Reentrancy, Signal Sets, Blocking Signals, Advanced Signal Management, Sending a Signal with a Payload (TextBook-1: Chapter 9)

Unit 6. Windows Thread Management (TextBook-2: Chapter 5 [relevant topics]) [06]

- Thread Internals
  - Data Structures, Kernel Variables, Performance Counters, Relevant Functions, Birth of a Thread Examining Thread Activity : Limitations on Protected Process Threads, Worker Factories (Thread Pools)
- Thread Scheduling
  - Overview of Windows Scheduling, Priority Levels, Windows Scheduling APIs, Relevant Tools, Real-Time Priorities, Thread States, Dispatcher Database, Quantum, Scheduling Scenarios, Context Switching, Idle Thread, Priority Boosts

**Recommended Text:**

1. Linux System Programming, O'Reilly, by Robert Love.
2. Windows Internals, Microsoft Press, by Mark E. Russinovich and David A. Solomon.
3. The Design of the UNIX Operating System, PHI, by Maurice J. Bach.
4. Advanced Programming in the UNIX Environment, Addison-Wesley, by Richard Stevens.

## **CS-203(New): Data Mining and Data Warehousing**

<b>Unit 1. Introduction to Data Mining</b>	<b>[4]</b>
<ul style="list-style-type: none"><li>• Basic Data Mining Tasks</li><li>• DM versus Knowledge Discovery in Databases</li><li>• Data Mining Issues</li><li>• Data Mining Metrics</li><li>• Social Implications of Data Mining</li><li>• Overview of Applications of Data Mining</li></ul>	
<b>Unit 2. Introduction to Data Warehousing</b>	<b>[4]</b>
<ul style="list-style-type: none"><li>• Architecture of DW</li><li>• OLAP and Data Cubes</li><li>• Dimensional Data Modeling-star, snowflake schemas</li><li>• Data Preprocessing – Need, Data Cleaning, Data Integration &amp; Transformation, Data Reduction</li><li>• Machine Learning</li><li>• Pattern Matching</li></ul>	
<b>Unit 3. Data Mining Techniques</b>	<b>[4]</b>
<ul style="list-style-type: none"><li>• Frequent item-sets and Association rule mining: Apriori algorithm, Use of sampling for frequent item-set, FP tree algorithm</li><li>• Graph Mining: Frequent sub-graph mining, Tree mining, Sequence Mining</li></ul>	
<b>Unit 4. Classification &amp; Prediction</b>	<b>[16]</b>
<ul style="list-style-type: none"><li>• Decision tree learning: [3 hrs] Construction, performance, attribute selection Issues: Over-fitting, tree pruning methods, missing values, continuous classes Classification and Regression Trees (CART)</li><li>• Bayesian Classification: [6 hrs]</li><li>• Bayes Theorem, Naïve Bayes classifier,</li><li>• Bayesian Networks</li><li>• Inference</li><li>• Parameter and structure learning</li><li>• Linear classifiers [4 hrs]</li><li>• Least squares, logistic, perceptron and SVM classifiers</li><li>• Prediction [3 hrs]</li><li>• Linear regression</li><li>• Non-linear regression</li></ul>	
<b>Unit 5 Accuracy Measures</b>	<b>[4]</b>
Precision, recall, F-measure, confusion matrix, cross-validation, bootstrap	
<b>Unit 6. Software for data mining and applications of data mining</b>	<b>[4]</b>
R, Weka, Sample applications of data mining	
<b>Unit 7. Clustering</b>	<b>[4]</b>
<ul style="list-style-type: none"><li>• k-means</li></ul>	

- Expectation Maximization (EM) algorithm
- Hierarchical clustering, Correlation clustering

Unit 8. **Brief overview of advanced techniques**

[4]

- Active learning
- Reinforcement learning
- Text mining
- Graphical models
- Web Mining

**Reference Books:**

1. Data Mining: Concepts and Techniques, Han, Elsevier ISBN:9789380931913/9788131205358
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), ISBN:9789380501864
8. [Research-Papers]: Some of the relevant research papers that contain recent results and developments in data mining field



### **CS-204 Project**

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,  
Deptt. of Computer Science

- You should prepare design document using SE/UML techniques depends on your project
- **Project Report Content should as follow :**
  1. College certificate
  2. Acknowledgement
  3. Problem Definition
  4. Existing System and need for the new system
  5. Scope of the work
  6. Feasibility study (Including H/W & S/W setup requirements)
  7. Requirement Analysis (including fact finding methods used)
  8. E-R diagrams
  9. Decision trees/Decision tables
  10. Normalized Database Design & Data Dictionary.
  11. Data flow Diagrams (if applicable)
  12. Use-case Diagrams
  13. Class Diagrams
  14. Object Diagrams
  15. Sequence Diagrams
  16. Collaboration Diagram
  17. Activity Diagram
  18. State Chart (if applicable)
  19. Component Diagram
  20. Deployment Diagram (if applicable)
  21. Use interface design
    - Menus
    - Input Screens using sample data
    - Reports, Graphs using sample data
  22. Testing & Implementation plan (Should contain testing strategies, techniques used & implementation approach used.)
  23. User manual

24. Drawbacks, Limitations & Proposed enhancement
25. Abbreviations used (if any)
26. Bibliography/Reference (Including book titles, authors name, editions, publications, etc)

**About project Report: -**

The report should be typed on A4 size, executive bond paper for the final submission. The report should be in the good quality Rexene bound. We suggest, using one-and-half spaced printing, Times New Roman 12 font sizes for the normal text, 14-16 font sizes for headings & page titles.

**Number of copies:**

For one project you should prepare 2 copies of the project report. One for yourself, one for college.

## **Elective Course [CS-205]: Programming with DOT NET**

Objectives:

- To understand the DOTNET framework, C# language features and Web development using ASP.NET

### **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

### **Topics to be covered:**

#### **Part I : C#**

#### **Unit 1. DOTNET Framework (2)**

- 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 Collector : Faster Memory allocation, Optimizations
- d. Common Language Integration
  - i. Common type system
  - ii. Reflection API
- e. User and Program Interface

#### **Unit 2. Introduction to C# (8)**

- 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, 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 3. Window Programming (6)**

- a. Window Controls
  - i. Common Controls
  - ii. Container Controls
  - iii. Menus and Toolbars
  - iv. Printing
  - v. Dialogs
- b. Deploying Window Application
  - i. Deployment Overview
  - ii. Visual studio setup and Deployment project types
  - iii. Microsoft windows installer architecture
  - iv. Building the project : Installation

#### **Unit 4. Data Access (6)**

- a. File System Data
- b. XML
- c. Databases and ADO.NET
- d. Data Binding

#### **Unit 5. Web Programming (6)**

- a. Basic Web programming
- b. Advanced Web programming
- c. Web Services
- d. Deployment Web applications

#### **Unit 6. .NET Assemblies (3)**

- a. Components
- b. .NET Assembly features
- c. Structure of Assemblies
- d. Calling assemblies, private and shared assemblies

#### **Unit 7. Networking (2)**

- a. Networking overview
- b. Networking programming options
  - i. WebClient
  - ii. WebRequest and WebResponse
  - iii. TcpListener & TcpClient

#### **Unit 8. Introduction to GDI+ (2)**

- a. Overview of Graphical Drawing
- b. Pen Class, Brush Class, Font Class
- c. Using Images
- d. Clipping, Drawing2D, Imaging

### **Part II : ASP.NET**

#### **Unit 1. Introduction to ASP.NET (1)**

#### **Unit 2. Server Controls and Variables, control Structures & Functions (4)**

- a. Forms, webpages, HTML forms, Webforms
- b. Request & Response in Non-ASP.NET pages
- c. Using ASP.NET Server Controls
- d. Datatypes : Numeric, text, arrays, datacollections
- e. Overview of Control structures
- f. Functions : web controls as parameters

#### **Unit 3. Even Driven Programming and PostBack (3)**

- a. HTML events
- b. ASP.NET page events
- c. ASP.NET Web control events
- d. Event driven programming and postback

**Unit 4. Reading from Databases (3)**

- a. Data pages
- b. ADO.NET

**Unit 5. ASP.NET Server Controls (4)**

- a. ASP.NET Web Controls
- b. HTML Server Controls
- c. Web Controls

**Unit 6. DOTNET assemblies and Custom Controls (2)**

- a. Introduction to Cookies, Sessions
- b. Session events
- c. State management Recommendations

**Unit 7. Web Services (2)**

- a. HTTP, XML & Web services
- b. SOAP
- c. Building ASP.NET web service
- d. Consuming a web service

**Recommended Text and Reference books:**

Beginning Visual C#, Wrox Publication  
 Professional Visual C#, Wrox Publication  
 Inside C#, by Tom Archer ISBN: 0735612889 Microsoft Press © 2001, 403 pages  
 Beginning ASP.NET 3.5, Wrox Publication  
 Programming ASP.NET 3.5 by Jesse Liberty, Dan Maharry, Dan Hurwitz, O'Reilly  
 Illustrated C# 2008, Solis, Publication Apress, ISBN 978-81-8128-958-2  
 Professional C# 4.0 and .NET 4 by Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson,  
 Morgan Skinner, WROX  
 Beginning C# Object-Oriented Programming By Dan Clark , Apress  
 ADO.NET Examples and Best Practices for C# Programmers, By Peter D. Blackburn Apress  
 Database Programming with C#, By Carsten Thomsen, Apress

## **Elective Course [CS-206]: Artificial Intelligence**

### **Prerequisites –**

- Concepts of Data structures and Design and Analysis of algorithms

### **Objectives-**

- To understand and gain the knowledge of the subject

### **Course contents –**

#### **Unit 1. Introduction to Artificial Intelligence**

- What is AI?
- Early work in AI
- AI and related fields
- AI problems and Techniques

#### **Unit 2. 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

#### **Unit 3. Heuristic Search Techniques**

- Generate-and-test
- Hill Climbing
- Best First Search
- Problem Reduction
- Constraint Satisfaction
- Mean-Ends Analysis

#### **Unit 4. 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

#### **Unit 5. Slot – and – Filler Structures**

- Weak Structures
- Semantic Networks
- Frames
- Strong Structures
- Conceptual Dependencies
- Scripts

#### **Unit 6. Game Playing**

- Minimax Search Procedures
- Adding alpha-beta cutoffs
- Uncertainty Reasoning: Basic Probability Axioms, Baye's



Rule, Bayesian Classification, Certainty Factor Theory, Dempster Shafar Theory.

Unit 7. Learning

- What is learning?
- Rote Learning
- Learning by taking advice
- Learning in problem solving
- Learning from examples
- Explanation based learning

**Internal evaluation**

- To implement the AI concepts using programming language PROLOG.

**Reference books –**

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

## **Elective Course [CS-207]: Advance Algorithms**

Unit 1 : 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 Hrs)**

Unit 2 . String algorithms: **(10 Hrs)**  
1 String searching - (Knuth–Morris–Pratt algorithm, Boyer–Moore string search algorithm, Rabin–Karp string search algorithm)  
2. Suffix trees - mathematical properties of suffix trees  
3. Applications of Suffix trees:  
    Regular expression searches using suffix trees;  
    Finding all maximal pairs and maximal repeats, Patricia trees

Unit 3 : Intractable problems: approximation algorithms **(14 Hrs)**  
1. Steiner tree and TSP  
2. Steiner forest  
3. Group Steiner trees  
4. Set cover via primal-dual  
5. k-median on a cycle

Unit 4: Integer programming and optimization algorithms **(14 Hrs.)**  
1. Formulations, complexity and relaxations  
2. discrete optimization,  
3. cutting plane methods,  
4. enumerative and heuristic methods  
5. Convex programming algorithms: ellipsoid method, interior-point methods, proximal point methods.

### **Preliminary reading:**

- Introduction to Algorithms: by Cormen, T.H., C.E. Leiserson, R.L. Rivest, and C. Stein; MIT Press; ISBN: 9780262032933
- The Algorithm Manual, Steven Skiena, Springer ISBN:9788184898651

### **Reference Books:**

- Theory of Linear and Integer Programming: by Schrijver; John Wiley & Sons. ISBN: 9780471982326
- Convex Optimization: by Boyd and Vandenberghe; Cambridge University Press; ISBN: 9780521833783
- Approximation Algorithms: by Vazirani; Springer-Verlag: ISBN: 9783540653677
- Advances in Steiner Trees (Combinatorial Optimization) by Ding-Zhu Du (Editor), J.M. Smith (Editor), J. Hyam Rubinstein (Editor); Springer; ISBN: 978-0792361107
- Algorithms On Strings, Trees, And Sequences; by D. Gusfield; Cambridge University Press,(ISBN 052158519)

### **Additional reading:**

- Algorithmic Number Theory: by Bach and Shallit; MIT Press; ISBN: 9780262024051