Anekant Education Society's

TuljaramChaturchand College, Baramati

(Autonomous)

Two Year M.Sc. Degree Program in Computer Science

(Faculty of Science & Technology)

CBCS Syllabus

M.Sc.(Computer Science) Sem- I

For Department of Computer Science

Tuljaram Chaturchand College of Arts, Science & Commerce, Baramati

Anekant Education Society's

Tuljaram Chaturchand College of Arts, Science and Commerce, Baramati, (Autonomous)

M.Sc.(Computer Science)Academic Year 2019-2020

M.Sc. (Computer Science) I - Credit Structure

Subject	Semester I	Semester II	Total
Paper – I	4	4	8
Paper – II	4	4	8
Paper – III	4	4	8
Paper - IV	4	4	8
Paper – V	4	4	8
Practical	4	4	8
Practical (Project)	---	4	4
Intro. to Cyber Security – I& II	2	2	4
Human Rights	2	--	2
Certificate Course- I	--	**2**	**2**
Total ====	**28**	**32**	**60**

M.Sc. (Computer Science) II - Credit Structure

Subject	Semester III	Semester IV	Total
Paper – I	4	**Industrial Training Project / Internship (IT) 16**	4
Paper – II	4		4
Paper – III	4		4
Paper - IV	4		8
Paper – V	4		8
Practical / Paper VI (Sem IV)	4		8
Practical (Project)	4		8
Certificate Course- II	2	-------	2
Skill Development I & II	2	2	4
			2
Total ====	**32**	**18**	**50**

Extra Credits:

1	Human Rights	2 Credits
2	Cyber Security Module I & II	4 Credits
3	Certificate Courses I & II	4 Credits
4	Skill Development I & II	4 Credits
Total Extra Credits =		14 Credits

Total Credits: Academic Credits (24+28+28+16 = 96) + Extra Credits (14) =110

Paper wise Course Structure For M.Sc. (Computer Science) (2019 Pattern)

No	Class	Sem	Code	Paper	Paper Title	Credit	Exam	Marks
1	M.Sc. - I	I	COMP4101	Theory	Principles of Programming Languages (C)	4	I / E	60 + 40
2			COMP4102	Theory	Cryptography and Network Security (C)	4	I / E	60 + 40
3			COMP4103	Theory	Database Technologies (C)	4	I / E	60 + 40
4			COMP4104	Theory	Design and Analysis of Algorithms (C)	4	I / E	60 + 40
5			COMP4105	Theory	Programming with DOT NET (C)	4	I / E	60 + 40
6			COMP4106	Pract.	Lab Course on DOT NET, PPL & Database Technologies (C)	4	I / E	60 + 40
7			HR-101	----	Human Rights – I	2	----	----
8			CYS-101	----	Introduction to Cyber Security – I	2	----	----
Note: Credit: 24. Core subjects is compulsory and Extra credits (2+2=4) is also compulsory.								
9	M.Sc. - I	II	COMP4201	Theory	Digital Image Processing (C)	4	I / E	60 + 40
10			COMP4202	Theory	Data Mining and Data Warehousing (C)	4	I / E	60 + 40
11			COMP4203	Theory	Python Programming (C)	4	I / E	60 + 40
12			COMP4204	Theory	**Advanced Operating System (EI)**	4	I / E	60 + 40
13			COMP4205	Pract.	Lab Course on Python Programming and Advance Operating System (C)	4	I / E	60 + 40
14			COMP4206	Pract.	**Project (EII)**	4	I / E	60 + 40
15			COMP4207	Theory	**Artificial Intelligence (EIII)**	4	I / E	60 + 40
16			CC-12	----	Certificate Course – I	2	----	---
17			CYS-102	----	Introduction to Cyber Security – II	2	----	----
Note: : Credit: 28. Core subjects is compulsory and Extra credits (4) is also compulsory.								
18	M.Sc. - II	III	COMP5301	Theory	Mobile Technologies (C)	4	I / E	60 + 40
19			COMP5302	Theory	Soft Computing (C)	4	I / E	60 + 40
20			COMP5303	Theory	Web Services (C)	4	I / E	60 + 40
21			COMP5304	Theory	**Software Architecture& Design Pattern (EI)**	4	I / E	60 + 40
22			COMP5305	Pract.	Lab Course-on Mobile Technologies and Web Services (C)	4	I / E	60 + 40
23			COMP5306	Pract.	**Project (EII)**	4	I / E	60 + 40
24			COMP5307	Theory	**Recent Trends in IT (Internet of Things) (EIII)**	4	I / E	60 + 40
25			CC-23	----	Certificate Course – II	2	----	----

26			SD-23	----	Skill Development – I	2	----	----
Note: Credit: 28. Core subjects is compulsory and Extra credits (2+2) is also compulsory.								
27	M.Sc. - II	IV	COMP5401	Project	Industrial Training/ Institutional Project **(IT) (Core)**	16	I / E	60 + 40
28			SD-23	----	Skill Development – II	2	----	----
Note: Credit:16. Core subject is compulsory, **Total Credits:Academic Credits(24+28+28+16 = 96) + Extra Credits (14) = 110**								

M.Sc. (Computer Science)- I

Semester-I

Syllabus

A.Y. 2019-20 (2019 Pattern)

SYLLABUS (CBCS) FOR M.Sc. (Computer Science)-I Sem- I
(w.e.f from june 2019) Academic Year 2019-2020

Class: M.Sc. (Computer Science) (Semester-I) Paper Code: **COMP4101**
Title of paper: Principles of Programming Language Paper: **I**
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.

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

Course Outcome

CO1. Knowledge of, and ability to use, language features used in current programming languages
CO2. An ability to program in different language paradigms and evaluate their relative benefits.
CO3. An understanding of the key concepts in the implementation of common features of programming languages.
CO4. To apply suitable programming paradigm for the application.CO5. To program in different language paradigms and evaluate their relative benefits
CO6. The programming paradigms of modern programming languages.
CO7. Knowledge to compare the features of various programming languages

Unit	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,ASSC) ✓ Procedure definition and binding, DEFUN, LET ✓ Predicates and Conditional, EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM,NUMBERP,	**10**

	SYMBOLP, NIL, NULL, IF, WHEN,UNLESS,COND, CASE ✓ Procedure Abstraction and RecursionTurbo Prolog ✓ Introduction, facts, Objects and Predicates, Variables, ✓ Using Rules, Controlling execution fail and cut predicates.	
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	**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-Enumeration types, Designs, Evaluation, Subrange types, Ada's design, Evaluation, Implementation of user defined ordinal types ✓ Array types-Array initialization, Array operations, Rectangular and Jagged arrays, Slices, Evaluation, Implementation of Array Types ✓ Associative Arrays-Structure and operations, Implementing associative arrays ✓ Record type-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	**8**
Unit - V	**Control Flow** ✓ Expression Evaluation-Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation ✓ Structured and Unstructured Flow-Structured Alternatives to goto	**5**

	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	
Unit -VI	**Subroutines and Control Abstraction** ✓ Fundamentals of Subprograms ✓ Design Issues for subprograms ✓ 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	**5**
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-Introduction, Cooperation synchronization, Competition Synchronization, Evaluation ✓ Message Passing-Introduction, The concept of Synchronous Message Passing ✓ Java Threads-The Thread class, Priorities, Competition	**5**

	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, SpringerISBN: 9781848829138

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	2	3	3	1	2	3
CO2	3	2	3	3	1	2	3
CO3	3	3	3	3	1	3	3
CO4	3	2	3	3	1	2	3
CO5	3	2	3	3	1	2	3
CO6	3	3	3	3	1	3	3
CO7	3	2	3	3	1	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification

PO1 with All CO's : CO1, CO2, CO3, CO4, CO5, CO6, and CO7 are strongly related to PO1. They align directly with the objective of enriching knowledge in various IT areas, including programming languages, paradigms, syntax, semantics, and their practical applications.PO2 with All CO's: CO1, CO2, CO4, and CO5 are moderately related to PO2. While they contribute to the understanding of programming languages and their features, they may not directly cover all dimensions of software applications and projects.CO3 and CO7 are strongly related to PO2. These outcomes directly address the ability to apply programming paradigms for applications and evaluate their benefits in different language paradigms, which align with understanding the concepts of software applications and projects.CO6 is moderately related to PO2. Understanding the concepts of ADT and OOP contributes to the broader understanding of software concepts, but it may not cover all dimensions on its own.PO3 with All CO's All COs (CO1 through CO7) are strongly related to PO3. Each of these course outcomes involves understanding, applying, and demonstrating various programming and theoretical concepts, which align with the broader objective of understanding computer subjects with the use of ICT.PO4 with All CO's All COs (CO1 through CO7) are strongly related to PO4. Each of these course outcomes contributes to the development of in-house applications in terms of projects. The ability to evaluate, express, apply

programming paradigms, and understand various programming concepts are essential skills for developing applications.PO5 with All CO’s All COs (CO1 through CO7) are partially related to PO5. While the course outcomes focus on enhancing programming language skills and understanding various programming.PO6 with All CO’s: CO3, CO6, and CO7 are strongly related to PO6. These outcomes directly contribute to the ability to apply programming paradigms, understand concepts of ADT and OOP.CO1, CO2, CO4, and CO5 are moderately related to PO6. While these outcomes involve evaluating and expressing syntax, semantics, and comparing features of programming languages, they may not be as directly tied to the practical experience gained during an industrial internship.

PO7 with All CO’s All COs (CO1 through CO7) are strongly related to PO7. The course outcomes, which involve evaluating and enhancing programming language skills, understanding programming paradigms, and applying concepts like ADT and OOP, contribute directly to making students employable in the current IT industry demands.

Class: M.Sc. (Computer Science) (Semester-I) **Paper Code: COMP4102**
Title of paper: Cryptography and Network Security **Paper: II**
Credit -4 **No. of Lectures: 50**

Learning Objectives:

- To enable students to get sound understanding of Info-Sys-Security, Network Security, Cryptography.
- To equip with knowledge and skills necessary to support for their career in Network 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 PG Course.

Course Outcomes:

1. **Understanding of Cryptographic Fundamentals:** Define and explain basic cryptographic terms and concepts and understand the principles of encryption and decryption.
2. **Knowledge of Cryptographic Algorithms:** Identify and compare various cryptographic algorithms, including symmetric and asymmetric algorithms, also analysis of the strengths and weaknesses of different encryption techniques.
3. **Network Security Protocols:** Describe common network security protocols (e.g., SSL/TLS, IPsec) and understand the role of protocols in securing communication.
4. **Firewall and Intrusion Detection Systems:** Understand the role of firewalls and intrusion detection systems in network security and configure and manage firewall rules.
5. **Incident Response and Forensics:** Develop skills in detecting and responding to security Incidents and understand the basics of digital forensics.
6. **Network Security Protocols, Policies and Compliance:** Understand the role of protocols in securing communication and understand and adhere to legal and regulatory compliance requirements.
7. **Cryptographic Applications:** Apply cryptographic techniques in real-world scenarios and analyze the security of existing cryptographic implementations.

Units	**Title & Contents**	**No. of Lectures**
Unit – I	**Introduction to Security:** The Need for Security, Security Approaches, Principles of Security, Types of Attacks	02
Unit – II	**Cryptography and techniques:** Introduction, Plain Text and Cipher Text, Substitution Techniques, Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric key cryptography, steganography.	06
Unit – III	**Symmetric Key Algorithms and AES:** Algorithm Types and Modes, Overview of Symmetric Key Cryptography, DES, IDEA, Blowfish	06
Unit – IV	**Asymmetric Key Algorithms, Digital Signature and RSA:** Brief	06

	History of Asymmetric Key Cryptography, overview, RSA Algorithm, Comparison between Symmetric & Asymmetric Key Algorithms, Digital Signature	
Unit – V	**Digital Certificates and Public Key Infrastructure (PKI):**Introduction, Digital Certificates, private key management,	03
Unit – VI	**Internet Security Protocols:** Basic Concepts, SSL, TLS, SHTTP, TSP, SET, SSL v/s SET, 3-D Secure Protocol, Electronic Money, Email Security, WAP Security, Users Authentication and Kerberos: Authentication Basics, Password, Authentication Tokens, Certificate based Authentication, Biometric Authentication, Kerberos, Security Handshake pitfalls, SSO Approaches.	14
Unit – VII	**Network Security**, Firewalls and Virtual Private Network (VPN): Brief introduction to TCP/IP, Firewalls, IP Security, VPN, Intrusion	05
Unit – VIII	**Case studies on Cryptography and Security:** Introduction, Cryptographic solutions, Secure inter branch payment transaction, Denial of services(DOS) attacks, IP Spooling Attack, Cross site scripting vulnerability(CSSV), Contract Signing, Secret Splitting, Virtual Electronics, Cookies and Privacy.	08
Reference: ➢ Cryptography and Network Security : Atul Kahate ➢ Computer Network Security : Kizza, Springer ➢ Network Security , Harrington, Elsevier		

8. Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)						
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**
CO1	3	2	3	3	2	2	3
CO2	3	2	3	3	2	2	3
CO3	2	2	3	3	2	2	3
CO4	2	2	3	3	2	2	3
CO5	2	2	3	3	2	2	3
CO6	3	2	3	3	2	2	3
CO7	3	2	3	3	2	2	3

9. Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

10. Justification of above mapping:

PO1 – CO1: Cryptographic fundamentals are foundational to the program's knowledge, and a strong understanding is crucial.
PO1 – CO2: Knowledge of cryptographic algorithms is a key aspect of program knowledge, contributing significantly.
PO1 – CO3: Understanding network security protocols is essential in demonstrating broad knowledge of the program.
PO1 – CO4: Knowledge of firewalls and intrusion detection systems is vital for a comprehensive understanding of the program.

PO1 – CO5: Incident response and forensics knowledge enhances the practical understanding of the program's knowledge.
PO1 – CO6: Understanding protocols, policies, and compliance contributes to the holistic knowledge of network security.
PO1 – CO7: Applying cryptographic techniques in real-world scenarios demonstrates a deep understanding of program knowledge.
PO2 – CO1, CO2, CO3, CO4, CO5, CO6, CO7 (communication aspects): Communication and collaboration skills are essential for social competence, making these outcomes significant.
PO3 – CO1, CO2, CO3, CO4, CO5, CO6, CO7(analytical skills) : Critical thinking is exercised in problem-solving, making these course outcomes crucial for PO3.
PO4 - CO1, CO2, CO3, CO4, CO5, CO6, CO7 (individual and team work): Personal and professional competence is directly linked to individual and collaborative performance, emphasizing these outcomes.
PO5 - CO1, CO2, CO3, CO4, CO5, CO6, CO7 (research and ethics): Emphasizing academics, research ethics, and awareness of intellectual property rights align with scientific temper.
PO6 - CO1, CO2, CO3, CO4, CO5, CO6, CO7 (technology-oriented): Continuous learning in the context of technological changes is crucial, making these outcomes directly align with PO6.
PO7 - CO1, CO2, CO3, CO4, CO5, CO6, CO7 (social concern, ethics): Demonstrating empathetic social concern, fair national development, and ethical commitment align with PO7.

Class: M.Sc. (Computer Science) (Semester-I) Paper Code: COMP4103

Title of paper: Database Technologies **Paper: III**

Credit -4 **No. of Lectures 52**

Prerequisites: Knowledge of RDBMS

Course objectives:

1. To study the usage and applications of Parallel and Distributed databases, Object relational

database and emerging systems.

2. To acquire knowledge on NoSQL databases.

Course outcomes:

CO1. Compare and contrast NoSQL databases with RDBMS.

CO2.Comparedifferentdatabasetechnologies.

CO3.CompareandcontrastNoSQLdatabaseswithRDBMS

CO4. Identify the data models for relevant problems.

CO5.Demonstrate the basic elements of a relational database management system

CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.

CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach

Units	Title & Contents	No. of Lectures
Unit – I	**Parallel and Distributed Databases** 1.1 Database System Architectures: Centralized and Client-Server Architectures, Server System Architectures, Parallel Systems, Distributed Systems 1.2 Parallel Databases : I/O Parallelism, Inter and Intra Query Parallelism, Inter and Intra operation Parallelism 1.3 Distributed Database Concepts: Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control, Distributed Query Processing, Three Tier Client Server Architecture, Case Studies	10
Unit – II	**Object and Object Relational Databases** 2.1 Concepts for Object Databases: Object Identity, Object Structure, Type Constructors, Encapsulation of Operators, Methods, Persistence, Type and Class Hierarchies, Inheritance, Complex Objects, Object Database Standards 2.2 Languages and Design : ODMG Model, ODL, OQL 2.3 Object Relational and Extended Relational Systems: Object Relational features in SQL/Oracle	10

	2.4 Case Studies	
Unit – III	**XML Databases** 3.1 XML Data Model 3.2 DTD 3.3 XML Schema 3.4 XML Querying 3.5 Web Databases 3.6 Information Retrieval 3.7 Data Warehousing 3.8 Data Mining	06
Unit – IV	**Mobile Databases** 4.1 Location and Handoff Management 4.2 Effect on Mobility on Data Management 4.3 Location Dependent Data Distribution 4.4 Mobile Transaction Models 4.5 Concurrency Control 4.6 Transaction Commit Protocols 4.7 Mobile Database Recovery Schemes 4.8 Examples: Oracle Database Lite, Microsoft SQL Server Compact	10
Unit – V	**Introduction to NoSQL** 5.1 Concepts and and evolution 5.2 History of NoSQL 5.3 Different NoSQL products : MongoDB, Couch DB, Cassandra 5.4 Exploring MongoDB 5.5 Advantages of MongoDB over RDBMS 5.6 Interfacing and Interacting with NoSQL 5.7 Sharding 5.8 Replication	08
Unit – VI	**Working with NoSQL** 6.1 NoSQL Storage Architecture 6.2 CRUD operations with MongoDB 6.3 Querying, Modifying and Managing NoSQL data stores 6.4 Indexing and ordering datasets 6.5 Surveying database internals 6.6 Migrating from RDBMS to NoSQL 6.7 Implementing NoSQL with PHP	08

References:

1. Henry Korth, Abraham Silberschatz and S. Sudarshan, “Database System Concepts”, Sixth Edition ,McGraw Hill, 2011.
2. M. Tamer Ozsu and Patrick Valduriez, “Principles of Distributed Database

Systems", Third Edition, Springer, 2011.
3. Thomas Connolly and Carolyn Begg, "Database Systems, A Practical Approach to Design, Implementation and Management", Fourth Edition, Pearson Education, 2008, Fifth Edition, Pearson Education, 2010, Sixth Edition, Pearson Education, 2015.
4. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education, 2017.
5. C.J.Date, A. Kannan, S. Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.
6. Dan Sullivan, "NoSQL for Mere Mortals", First Edition, Pearson Education, 2015.
7. Kristina Chodorow, "MongoDB-The Definitive Guide", Second Edition, O'Reilly, 2013.

Mapping of this course with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	3	3	3	3	3	3
CO2	3	3	3	3	3	3	3
CO3	3	3	3	3	3	3	3
CO4	3	3	3	3	3	3	3
CO5	3	3	3	3	3	3	3
CO6	3	3	3	3	3	3	3
CO7	3	3	3	3	3	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of mapping of PO1 with all COs

PO1: Enrich the knowledge in the areas like Artificial Intelligence, Web Services, Cloud Computing, Paradigm of Programming language, Design and Analysis of Algorithms, Database Technologies Advanced Operating System, Mobile Technologies, Software Project Management, and core computing subjects.
CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Understanding the differences between NoSQL databases and Relational Database Management Systems (RDBMS) is crucial in the broader context of Database Technologies, one of the areas specified in PO1.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: This outcome aligns directly with the goal of enriching knowledge in Database Technologies, a specified area in PO1, by requiring a comparison of various database technologies.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: The detailed comparison between NoSQL databases and RDBMS is a specific aspect within the broader topic of Database Technologies, which falls under the areas specified in PO1.

CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Identifying appropriate data models for specific problems is an integral part of Database Technologies and aligns with the goal of enriching knowledge in this area as mentioned in PO1.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: Understanding and demonstrating the basic elements of a relational database management system are fundamental aspects of Database Technologies, which is a part of the enriched knowledge specified in PO1.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Familiarity with relational database theory and the ability to write relational algebra expressions are key components of Database Technologies, contributing to the enriched knowledge specified in PO1.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Mastering design principles, including the Entity-Relationship (E-R) method and normalization approach, are essential in the context of Database Technologies, which aligns with the knowledge enrichment goal in PO1.

Justification of mapping of PO2 with all COs
PO2: Students understand all dimensions of the concepts of software application and projects.
CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Understanding the differences between NoSQL databases and RDBMS is crucial for students to comprehend various dimensions of software applications, especially when dealing with diverse data storage solutions.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: Comparing different database technologies is directly linked to enhancing students' understanding of software applications by providing insights into the various options available for data storage and management in projects.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: This comparison is integral to developing a comprehensive understanding of database technologies, contributing to the broader perspective required for software application concepts and projects.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Identifying appropriate data models is essential for students to grasp the conceptual dimensions of software applications and projects, as it directly influences how data is structured and managed in the context of specific problems.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: Demonstrating the basic elements of an RDBMS is a foundational aspect that aligns with students' understanding of software application concepts, particularly in terms of data organization and management.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related

Justification: Being familiar with relational database theory and the ability to write relational algebra expressions contribute significantly to students' understanding of database concepts, which is crucial for effective software application development.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Mastering design principles, such as the Entity-Relationship method and normalization, is directly tied to understanding the logical design of databases, which is essential for students comprehending software application concepts and projects.

Justification of mapping of PO3 with all COs
PO3: Students understand the computer subjects with a demonstration of all programming and theoretical concepts with the use of ICT.
CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Understanding and demonstrating the comparison between NoSQL databases and RDBMS aligns with the goal of comprehensively understanding computer subjects through practical application, integrating both programming and theoretical concepts.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: Comparing diverse database technologies with the use of Information and Communication Technology (ICT) supports students in practically demonstrating their understanding of computer subjects, emphasizing the application of theoretical knowledge.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Demonstrating the comparison between NoSQL databases and RDBMS with the use of ICT reinforces the practical application of theoretical concepts, enhancing students' understanding of computer subjects.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Using ICT to identify data models for specific problems provides a practical demonstration of theoretical concepts, contributing to students' understanding of computer subjects through real-world application.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: Demonstrating the basic elements of an RDBMS with the use of ICT directly integrates practical programming concepts with theoretical knowledge, aligning with the goal of understanding computer subjects comprehensively.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Familiarity with relational database theory and the ability to write relational algebra expressions, demonstrated through ICT, emphasizes the practical application of theoretical concepts in computer subjects.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Mastering design principles, including the E-R method and normalization, with the use of ICT, ensure a practical demonstration of theoretical concepts, enhancing students' understanding of computer subjects.

Justification of mapping of PO4 with all COs
PO4: Developed in-house applications in terms of projects.

CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Developing in-house applications often involves database selection, and comparing NoSQL databases with RDBMS is crucial for making informed decisions, directly impacting the success of the projects.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: Developing in-house applications requires an understanding of various database technologies to choose the most suitable one for a given project, aligning with the need to compare and select technologies outlined in CO2.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: The comparison between NoSQL databases and RDBMS is directly relevant to the decision-making process in developing in-house applications, ensuring that the chosen database technology aligns with project requirements.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Identifying appropriate data models is crucial in the development of in-house applications, as it directly influences how data is structured and managed to meet the specific problems or requirements of the projects.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: Demonstrating the basic elements of an RDBMS is essential for developers working on in-house applications, as relational databases are commonly used, and understanding their elements is fundamental to effective application development.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Familiarity with relational database theory and the ability to write relational algebra expressions is directly applicable to in-house application development, aiding in the design and implementation of efficient database queries.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Mastering design principles, including the E-R method and normalization, are crucial for the logical design of databases in in-house application development, ensuring data integrity and optimal performance.

Justification of mapping of PO5 with all COs

PO5: Interact with IT experts & knowledge by IT visits.
CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Interacting with IT experts during IT visits provides opportunities to discuss real-world implementations, including the choice between NoSQL databases and RDBMS, contributing to a deeper understanding of database technologies.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: IT visits offer direct exposure to various database technologies in practical scenarios, aligning with the goal of comparing and contrasting different technologies outlined in CO2.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related

Justification: Engaging with IT experts during visits provides a platform to discuss and understand the practical implications of choosing between NoSQL databases and RDBMS, reinforcing the comparison aspect emphasized in CO3.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Interacting with IT experts allows students to gain insights into real-world problems and the corresponding data models used to address them, aligning with the identification of data models outlined in CO4.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: IT visits provide an opportunity for students to witness and demonstrate the basic elements of relational database management systems in practical settings, reinforcing the theoretical knowledge outlined in CO5.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Interacting with IT experts during visits facilitates discussions on relational database theory and practical demonstrations of writing relational algebra expressions, directly supporting the goals outlined in CO6.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Engaging with IT experts during visits allows students to explore and discuss sound design principles, including the E-R method and normalization, in the context of real-world database design, reinforcing the objectives outlined in CO7.

Justification of mapping of PO6 with all Cos

PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.

CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: The industrial internship provides an opportunity for practical exposure, allowing students to observe and participate in real-world scenarios where the choice between NoSQL databases and RDBMS is a crucial decision.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: During the industrial internship, students are likely to encounter various database technologies in use, providing a practical context for comparing and contrasting different technologies, aligning with the goals outlined in CO2.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Industrial exposure allows students to witness and participate in projects where the comparison between NoSQL databases and RDBMS is essential, contributing directly to the goals specified in CO3.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: Working on real-world projects during the industrial internship requires students to identify and apply appropriate data models for specific problems, aligning closely with the skills outlined in CO4.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related

Justification: The industrial internship offers a practical environment for students to actively demonstrate their understanding of the basic elements of relational database management systems, reinforcing the theoretical knowledge outlined in CO5.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Exposure to real-world scenarios during the industrial internship allows students to become familiar with relational database theory and practice writing relational algebra expressions for queries, directly supporting the objectives in CO6.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Engaging in industrial projects provides a practical platform for students to master sound design principles, including the E-R method and normalization, contributing directly to the goals specified in CO7.

Justification of mapping of PO7 with all COs

PO7: To make them employable according to the current demand of the IT Industry and responsible citizen.
CO1. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: Being employable in the IT industry requires a comprehensive understanding of database technologies, including the ability to compare and contrast NoSQL databases with RDBMS, aligning with the objectives in CO1.
CO2. Compare different database technologies.
Mapping: 3 - Strongly related
Justification: The demand in the IT industry often involves knowledge of various database technologies, making the ability to compare and contrast them crucial for employability, which directly aligns with the goals in CO2.
CO3. Compare and contrast NoSQL databases with RDBMS.
Mapping: 3 - Strongly related
Justification: The skill of comparing and contrasting database systems, specifically NoSQL and RDBMS, is directly relevant to meeting the current demand of the IT industry, contributing to employability as specified in CO3.
CO4. Identify the data models for relevant problems.
Mapping: 3 - Strongly related
Justification: The ability to identify appropriate data models for specific problems is a fundamental skill for employability in the IT industry, aligning closely with the goals in CO4.
CO5. Demonstrate the basic elements of a relational database management system.
Mapping: 3 - Strongly related
Justification: Demonstrating proficiency in the basic elements of a relational database management system is crucial for employability in IT, meeting the demands of industry practices as outlined in CO5.
CO6. Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.
Mapping: 3 - Strongly related
Justification: Industry demand for individuals familiar with relational database theory and the ability to write relational algebra expressions makes this directly relevant to employability, aligning with the goals specified in CO6.
CO7. Master sound design principles for logical design of databases, including the E-R method and normalization approach.
Mapping: 3 - Strongly related
Justification: Mastering design principles, including the E-R method and normalization, is essential for meeting industry standards and demands, directly contributing to employability as specified in CO7.

Class: M.sc.(Computer science) (Semester-I) Paper Code: COMP4104
Title of Paper: Design & Analysis of Algorithm Paper: IV
Credit:4 No. of lectures:52

Learning Objectives: Student successfully completing this course will be able to

- Understand Basic Algorithm Analysis techniques and 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

Learning Outcome

CO1: Understand the Basic algorithms and data structure concepts.
CO2: Understand Basic programming concepts.

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.

CO6: To be able to do performance analysis of simple approximation algorithms.

CO7: Able to analyze String matching algorithms.

Units	Title and Contents	No. of Lectures
Unit –I	**Design strategies** Algorithm definition, space complexity, time complexity, worst case –best case –average casecomplexity, asymptotic notation, sorting algorithms (insertion sort, heap sort) sorting in linear time, searching algorithms, recursive algorithms (Tower of Hanoi , Permutations). **Divide and conquer** control abstraction, binary search, merge sort, Quick sort, Strassen's matrix Multiplication	**12**
Unit –II	**Advanced Design and Analysis Techniques** **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 **-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.	**16**
Unit –	**Decrease and conquer**	**6**

III	DFS and BFS, Topological sorting, connected components	
Unit – IV	**Backtracking** General method, 8 Queen's problem, Sum of subsets problem, graph coloring problem, Hamiltonian cycle	**4**
Unit – V	**Branch and Bound Technique** FIFO, LIFO, LCBB, TSP problem, 0/1 knapsackProblem	**4**
Unit – VI	**Transform and conquer** Horner's Rule and Binary Exponentiation – Problem Reduction	**4**
Unit – VII	**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&SanguthevarRajasekaran, 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

Mapping of this course with Program Outcomes

Course Outcomes	**Program Outcomes (POs)**						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	2	3	2	1	3	3
CO2	3	2	3	2	1	3	3
CO3	3	2	3	2	1	3	3
CO4	3	2	3	3	1	3	3
CO5	3	2	3	3	1	3	3
CO6	3	2	3	2	1	3	3
CO7	3	2	3	2	1	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

1. **Justification of PO1 to ALL CO's:**

CO1: Understand the Basic algorithms and data structure concepts.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.

Mapping: 3 (Strongly related)

CO2: Understand Basic programming concepts.
PO1: Enrich the knowledge in the areas like Paradigm of Programming language.
Mapping: 3 (Strongly related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.
Mapping: 3 (Strongly related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.

Mapping: 3 (Strongly related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.
Mapping: 3 (Strongly related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.
Mapping: 3 (Strongly related)

CO7: Able to analyze String matching algorithms.
PO1: Enrich the knowledge in the areas like Design and Analysis of Algorithms.
Mapping: 3 (Strongly related)

Overall, all the Course Outcomes (CO) is strongly related to the Program Outcome (PO) - "Enrich the knowledge in the areas like Design and Analysis of Algorithms."

2. Justification of PO2 to ALL CO's:
CO1: Understand the Basic algorithms and data structure concepts.
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)
CO2: Understand Basic programming concepts.
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO2: Students understand all dimensions of the concepts of software application and projects.

Mapping: 2 (Moderately related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)

CO7: Able to analyze String matching algorithms
PO2: Students understand all dimensions of the concepts of software application and projects.
Mapping: 2 (Moderately related)
Overall, all the Course Outcomes (CO) are moderately related to the Program Outcome (PO) - "Students understand all dimensions of the concepts of software application and projects."

3. Justification of PO3 to ALL CO's:
CO1: Understand the Basic algorithms and data structure concepts
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.
Mapping: 3 (Strongly related)

CO2: Understand Basic programming concepts.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT
Mapping: 3 (Strongly related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.
Mapping: 3 (Strongly related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.

Mapping: 3 (Strongly related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.
Mapping: 3 (Strongly related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.
Mapping: 3 (Strongly related)

CO7: Able to analyze String matching algorithms.
PO3: Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT.
Mapping: 3 (Strongly related)

Overall, all the Course Outcomes (CO) are strongly related to the Program Outcome (PO) - "Students understand the computer subjects with the demonstration of all programming and theoretical concepts with the use of ICT."

4.Justification of PO4 to ALL CO's:
CO1: Understand the Basic algorithms and data structure concepts.
PO4: Developed in-house applications in terms of projects.
Mapping: 2 (Moderately related)

CO2: Understand Basic programming concepts.

PO4: Developed in-house applications in terms of projects.
Mapping: 2 (Moderately related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO4: Developed in-house applications in terms of projects.
Mapping: 2 (Moderately related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO4: Developed in-house applications in terms of projects.
Mapping: 3 (Strongly related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO4: Developed in-house applications in terms of projects.

Mapping: 3 (Strongly related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO4: Developed in-house applications in terms of projects.
Mapping: 2 (Moderately related)

CO7: Able to analyze String matching algorithms.
PO4: Developed in-house applications in terms of projects
Mapping: 2 (Moderately related)
Overall, CO4 and CO5 are strongly related to PO4, as they directly involve the design and development of algorithms for computational problems. CO1, CO2, CO3, CO6, and CO7 are moderately related to PO4, as they contribute to the foundational understanding and skills needed for software development projects.

5.Justification of PO5 to ALL CO's:

CO1: Understand the Basic algorithms and data structure concepts.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)
CO2: Understand Basic programming concepts.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)
CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)

CO7: Able to analyze String matching algorithms.
PO5: Interact with IT experts & knowledge by IT visits.
Mapping: 1 (Partially related)

Overall, all the Course Outcomes (CO) are partially related to the Program Outcome (PO) - "Interact with IT experts & knowledge by IT visits." The theoretical and algorithmic understanding gained through the courses may not be directly tied to the experiential learning and interaction with IT experts during visits.

6.Justification of PO6 to ALL CO's:

CO1: Understand the Basic algorithms and data structure concepts.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO2: Understand Basic programming concepts.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO6: To be able to do performance analysis of simple approximation algorithms.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

CO7: Able to analyze String matching algorithms.
PO6: Get industrial exposure through the 6 months Industrial Internship in IT industry.
Mapping: 3 (Strongly related)

Overall, all the Course Outcomes (CO) are strongly related to the Program Outcome (PO) - "Get industrial exposure through the 6 months Industrial Internship in IT industry." The skills and knowledge acquired in the courses directly contribute to the practical experience gained during the industrial internship.

7.Justification of PO7 to ALL CO's:

CO1: Understand the Basic algorithms and data structure concepts.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens. Aware them to publish their work in reputed journals.

Mapping:
CO1 with PO7: 3 (Strongly related)

CO2: Understand Basic programming concepts.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens. Aware them to publish their work in reputed journals.
Mapping:
CO2 with PO7: 3 (Strongly related)

CO3: To develop the ability to analyze the running time and prove the correctness of basic algorithms.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens.Aware them to publish their work in reputed journals.
**Mapping:
CO3 with PO7: 3 (Strongly related)

CO4: To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens. Aware them to publish their work in reputed journals.
**Mapping:
CO4 with PO7: 3 (Strongly related)

CO5: To be able to prove the hardness of NP-Hard problems using simple reductions.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens.Aware them to publish their work in reputed journals.
Mapping:
CO5 with PO7: 3 (Strongly related)
CO6: To be able to do performance analysis of simple approximation algorithms.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens.Aware them to publish their work in reputed journals.
Mapping:
CO6 with PO7: 3 (Strongly related)
CO7: Able to analyze String matching algorithms.
PO7: To make them employable according to the current demand of the IT Industry and responsible citizens.Aware them to publish their work in reputed journals.
**Mapping:
CO7 with PO7: 3 (Strongly related)
Overall, all the Course Outcomes (CO) are strongly related to the Program Outcome (PO7) - "To make them employable according to the current demand of the IT Industry and responsible citizens." However, when considering the awareness to publish work in reputed journals (PS08), the mapping is partially related (weightage 1).

Class: M.sc.(Computer science) (Semester-I) **Paper Code: COMP4105**

Title of Paper: Programming with DOT NET **Paper: V**

Credit:4 **No. of lectures:50**

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

Learning Objectives:

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

Learning Outcome:

CO1.Ability to write the Visualized programming and design different real life problems

CO2.Explain the three pillars of object-oriented programming.

CO3.Develop working knowledge of C# programming constructs and the .NET Framework.

CO4.Write an object-oriented program using custom classes.

CO5.Build and debug well-formed Web Forms with ASP.

CO6.Perform form validation with validation controls.

CO7.Create simple data binding applications.Mapping of this Course Outcomes with Programme Outcomes

Part I : C#		
1.	**Introduction to 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 Collector : Faster Memory allocation, Optimizations **d. Common Language Integration** i. Common type system ii. Reflection API **e. User and Program Interface**	**2**
2.	**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	**8**

	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	
3.	**Window Programming** **a. Window Controls** i. Common Controls ii. Container Controls iii. Menus and Toolbars iv. Printing v. Dialogs vi. **Data tools** **b. Deploying Window Application** i. Deployment Overview ii. Adding setup project iii. Building the project : Installation	**6**
4.	**Data Handling** **a. File System Data** **b. XML Data** **c. Databases and ADO.NET**	**6**
5.	**Reporting Tools** **a. Data Report** **b. Crystal Report**	**4**
6.	**Dot NET Assemblies** a. Components b. .NET Assembly features c. Structure of Assemblies d. Calling assemblies, private and shared assemblies	**3**
	Part II : ASP.NET	
1.	**Introduction to ASP.NET** **a. History of Web Programming** **b. Basic of Web programming**	**1**
2.	**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.** Datatypes : Numeric, text, arrays, datacollections **e.** Overview of Control structures	**4**
3.	**Even Driven Programming and PostBack** **a.** HTML events **b.** ASP.NET page events **c.** ASP.NET Web control events	**3**

	d. Event driven programming and postback	
4.	**Reading from Databases** **a.** Data pages , **b.** ADO.NET	**3**
5.	**ASP.NET Server Controls** **a.** ASP.NET Web Controls **b.** HTML Server Controls **c.** Web Controls	**4**
6.	**DOTNET assemblies and Custom Controls** **a.** Introduction to Cookies, Sessions **b.** Session events **c.** State management Recommendations	**2**
7.	**Web Services** **a.** HTTP, XML & Web services **b.** SOAP **c.** Building ASP.NET web service **d.** Consuming a web service	**2**

Recommended Text and Reference books:

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

Mapping of this Course Outcomes with Programme Outcomes

Course Outcomes	**Programme Outcomes (POs)**						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	3	3	3	2	3	3
CO2	3	3	3	3	3	3	3
CO3	3	3	3	3	2	3	2
CO4	3	3	3	3	2	2	3
CO5	3	2	3	3	2	3	3
CO6	2	2	2	2	3	3	3
CO7	2	1	2	2	2	3	2

Mapping of PO1 With All CO’s with Justification :
CO1: with PO1: as it aligns with the goal of enriching knowledge in areas like Artificial Intelligence, Web Services, Cloud Computing, and other core computing subjects.
CO2: with PO1: as understanding the principles of object-oriented programming is fundamental in areas like software design, which is part of the enriched knowledge in programming languages and design paradigms.
CO3: With PO1: as C# and .NET Framework are key components in the study of programming Languages, web services, and software project management.
CO4: With PO1: as it directly involves applying knowledge of object-oriented programming inCreating software solutions, contributing to the enriched knowledge in programming languages andsoftware design.
CO5: With PO1: as it involves practical skills in web development and aligns with the areas of webservices, cloud computing, and mobile technologies.
CO6: With PO1 : as form validation is part of web development and contributes to knowledge in webservices, but it's not as broad as some other aspects covered in PO1.
CO7: With PO1: as data binding is a specific aspect of web development and contributes to knowledge in web services and database technologies, but it may not cover a broad range of areasmentioned in PO1.
Mapping of PO2 With All CO’s with Justification :
CO1: With PO2 : as the ability to write visualized programs and design real-life problems contributes to a comprehensive understanding of software application concepts and projects.
CO2: With PO2 : as understanding the three pillars of object-oriented programming is fundamental to grasping the conceptual dimensions of software application and projects.
CO3: With PO2: as acquiring a working knowledge of C# and the .NET Framework is essential forunderstanding software application concepts and projects.
CO4: With PO2 : as writing object-oriented programs with custom classes demonstrates practicalapplication, contributing to a comprehensive understanding of software application concepts andprojects.
CO5: With PO2 : as building and debugging web forms with ASP contributes to understandingsoftware applications, but it may not cover all dimensions of software projects.
CO6: With PO2: as form validation is a specific aspect of software applications and contributes tounderstanding certain dimensions of projects, but it's not as comprehensive as other aspects.
CO7: With PO2 : as creating data binding applications contributes to understanding certain aspectsof software applications but may not cover all dimensions of software projects.
Mapping of PO3 With All CO’s with Justification :
CO1: With PO3 : as the ability to write visualized programs and design real-life problems Demonstrates understanding through practical application, facilitated by the use of Information andCommunication Technology (ICT).
CO2: With PO3 : as explaining the three pillars of object-oriented programming involves theoreticalunderstanding and can be demonstrated through ICT tools, enhancing the comprehension ofcomputer subjects.

CO3: With PO3 : as developing working knowledge using ICT tools in C# and .NET contributes to apractical and theoretical understanding of computer subjects.
CO4: With PO3 : as writing object-oriented programs using custom classes, especially with the use of ICT, demonstrates a practical application of theoretical concepts in computer subjects.
CO5: With PO3: as building and debugging web forms using ASP with the aid of ICT tools enhances the practical and theoretical understanding of computer subjects, especially in web development.
CO6: With PO3: as performing form validation involves practical application with ICT tools, Contributing to the understanding of certain dimensions of computer subjects.
CO7: With PO3 : as creating data binding applications with the use of ICT tools contributes to a practical understanding of certain aspects of computer subjects.
Mapping of PO4 With All CO's with Justification :
CO1: With PO4 : as the ability to write visualized programs and design real-life problems is crucial in the development of in-house applications as it involves translating conceptual ideas into practical solutions.
CO2: With PO4 : as understanding the three pillars of object-oriented programming is fundamental in the development of in-house applications, guiding the design and structure of the software.
CO3: With PO4: as a working knowledge of C# and the .NET Framework is essential for effectively developing in-house applications using these technologies.
CO4: With PO4: as writing object-oriented programs with custom classes is a practical skill necessary for the development of in-house applications, ensuring the creation of well-structured and maintainable code.
CO5: With PO4 : as building and debugging web forms with ASP is directly applicable to the development of web-based in-house applications.
CO6: With PO4 : as form validation is a specific aspect of developing user-friendly in-house applications, contributing to the overall quality of the software.
CO7:With PO4: as creating data binding applications is relevant to the development of in-house applications, especially when dealing with data integration and user interface design.
Mapping of PO5 With All CO's with Justification :
CO1: With PO5 : as the ability to write visualized programs and design real-life problems contributes to skills useful during interactions with IT experts but may not be the sole focus of such visits.CO2: With PO5: as understanding the three pillars of object-oriented programming provides a foundational knowledge useful for interacting with IT experts but may not be the primary focus of such visits.
CO3: With PO5 : as developing a working knowledge of C# and the .NET Framework is beneficial for discussions with IT experts but might not be the sole purpose of IT visits.CO4: With PO5 : as writing object-oriented programs with custom classes contributes to the practical skills that can be discussed during interactions with IT experts but may not be the central focus ofsuch visits.CO5: With PO5 : as building and debugging web forms with ASP is a practical skill that can be relevant during interactions with IT experts but may not be the primary objective of IT visits.CO6: With PO5 : as performing form validation is a specific skill that can be discussed during interactions with IT experts but may not be the primary purpose of such visits.CO7: With PO5 : as creating data binding applications can be a topic of discussion during interactionswith IT experts but may not be the central theme of IT visits.

Mapping of PO6 With All CO's with Justification :CO1: With PO6: as the ability to write visualized programs and design real-life problems is crucial forsuccessfully undertaking and contributing to industrial internship projects in the IT industry.CO2: With PO6: as understanding the three pillars of object-oriented programming is essential for grasping software design principles and applying them in real-world industrial projects during theinternship.CO3: With PO6: as a working knowledge of C# and the .NET Framework is directly applicable to many industrial projects in the IT industry, making this

competency highly relevant during the internship.CO4: With PO6: as writing object-oriented programs with custom classes is a practical skill that canbe directly applied in the development of software solutions during the industrial internship.CO5: With PO6: as building and debugging web forms with ASP is a practical skill that aligns with many web-based projects encountered in the IT industry during the internship.

CO6: With PO6: as performing form validation is a practical aspect that contributes to the quality ofuser interfaces, which is often a requirement in industrial projects undertaken during the internship.
CO7: With PO6: as creating data binding applications is directly applicable to projects involving data integration, a common aspect of industrial IT projects during the internship.
Mapping of PO7 With All CO's with Justification :
CO1: With PO7: as the ability to write visualized programs and design real-life problems is a fundamental skill that directly contributes to employability in the IT industry by demonstrating practical problem-solving capabilities.
CO2: With PO7: as understanding the three pillars of object-oriented programming is a foundational knowledge that is highly relevant to the demands of the IT industry, contributing to the employability of graduates.
CO3: With PO7: as developing a working knowledge of C# and the .NET Framework is directly alignedwith the current demands of the IT industry, enhancing the employability of students.
CO4: With PO7: as the ability to write object-oriented programs with custom classes is a practicalskill that is in high demand in the IT industry, making graduates more employable.
CO5: With PO7: as building and debugging web forms with ASP is a practical skill that is directlyapplicable to web development, a field with significant demand in the IT industry.
CO6: With PO7 : as form validation is a practical aspect that contributes to the development ofrobust and user-friendly applications, enhancing the employability of graduates.
CO7: With PO7 : as creating data binding applications is a practical skill that aligns with the industrydemand for professionals who can work with data integration, making graduates more employable.

Class: M.sc.(Computer science) (Semester-I) Paper Code: COMP4106
Title of Paper: Practical On DOTNET, PPL, Database Technologies
Paper: VI (Lab Course)
Credit:4(3 Hr. Practical /week/batch) No.ofPracticals:12

Course Outcomes:

CO1.Ability to write the Visualized programming and design different real life problems
CO2.Explain the three pillars of object-oriented programming.
CO3.Develop working knowledge of C# programming constructs and the .NET Framework.
CO4.Write an object oriented program using custom classes.
CO5.Build and debug well-formed Web Forms with ASP.
CO6.Perform form validation with validation controls.
CO7.Create simple data binding applications.

Console Application	
Assignment 1	Parameter Modifiers (ref , out , params)
Assignment 2	Delegate and Events
Assignment 3	Properties and Indexers
Assignment 4	Inheritance and Interface
Assignment 5	Polymorphism (Method Overloading , Operator Overloading and Method Overriding
Assignment 6	Exception Handling
Assignment 7	Collections
Assignment 8	Generics
Windows Application	
Assignment 1	Use of Basics Form Controls
Assignment 2	Use of List Box
Assignment 3	Event Handling (Calculator)
Assignment 4	Use of Dialogue Boxes
Assignment 5	Simple Database Operations
Assignment 6	Advanced Database Operations
Assignment 7	Simple Crystal Report
Assignment 8	Advanced Crystal Report
ASP.Net Web Applications	
Assignment 1	Use of Web Controls
Assignment 2	Validation Controls
Assignment 3	Use of CSS

Assignment 4	Database Connectivity
Assignment 5	Database Connectivity (Stored Procedure)
Assignment 6	Use of Master Pages
Assignment 7	Use of Master Pages
Assignment 8	Use of State Management (Cookies, Sessions…)
PPL Assignments	
Assignment 1	LISP
Assignment 2	PROLOG
Database Technologies Assignments	
Assignment 1	Creating database, collections, insert, update & delete documents in NoSQL
Assignment 2	Querying documents in NoSQL

Mapping of this Course Outcomes with Programme Outcomes

Course Outcomes	**Programme Outcomes (POs)**						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	3	3	3	2	3	3
CO2	3	3	3	3	2	3	3
CO3	3	3	3	3	2	3	3
CO4	3	3	3	3	2	3	3
CO5	3	3	3	3	2	3	3
CO6	3	3	3	3	2	3	3
CO7	3	3	3	3	2	3	3

Mapping of PO With All CO's with Justification:
All CO: with PO1: as it aligns with the goal of enriching knowledge in areas like Artificial Intelligence, Web Services, Cloud Computing, and other core computing subjects.
All CO: with PO2: as understanding the principles of object-oriented programming is fundamental in areas like software design, which is part of the enriched knowledge in programming languages and design paradigms.
All CO: With PO3: as C# and .NET Framework are key components in the study of programming Languages, web services, and software project management.
All CO: With PO4: as it directly involves applying knowledge of object-oriented programming increating software solutions, contributing to the enriched knowledge in programming languages andsoftware design.
All CO: With PO5: as it involves practical skills in web development and aligns with the areas of webservices, cloud computing, and mobile technologies.
All CO: With PO6 : as form validation is part of web development and contributes to knowledge in webservices, but it's not as broad as some other aspects covered in PO1.

All CO: With PO7: as data binding is a specific aspect of web development and contributes to knowledge in web services and database technologies, but it may not cover a broad range of areasmentioned in PO1.