**Anekant Education Society's**

# Tuljaram Chaturchand College, Baramati

***(Autonomous)***

**Two Year Post Graduate Degree Program in Computer Science**

**(Faculty of Science & Technology)**

**CBCS Syllabus**

**M.Sc. (Computer Science) Part-I Semester -I**

**For Department of Computer Science**

**Tuljaram Chaturchand College, Baramati**

**Choice Based Credit System Syllabus (2023 Pattern)**

**(As Per NEP 2020)**

**To be implemented from Academic Year 2023-2024**

**(Eligibility: B.Sc. Computer Science)**

# Title of the Programme: M.Sc. (Computer Science)

## Preamble

AES's Tuljaram Chaturchand College has made the decision to change the syllabus of across various faculties from June, 2023 by incorporating the guidelines and provisions outlined in the National Education Policy (NEP), 2020. The NEP envisions making education more holistic and effective and to lay emphasis on the integration of general (academic) education, vocational education and experiential learning. The NEP introduces holistic and multidisciplinary education that would help to develop intellectual, scientific, social, physical, emotional, ethical and moral capacities of the students. The NEP 2020 envisages flexible curricular structures and learning based outcome approach for the development of the students. By establishing a nationally accepted and internationally comparable credit structure and courses framework, the NEP 2020 aims to promote educational excellence, facilitate seamless academic mobility, and enhance the global competitiveness of Indian students. It fosters a system where educational achievements can be recognized and valued not only within the country but also in the international arena, expanding opportunities and opening doors for students to pursue their aspirations on a global scale.

In response to the rapid advancements in science and technology and the evolving approaches in various domains of Computer Science and related subjects, the Board of Studies in Computer Science at Tuljaram Chaturchand College, Baramati - Pune, has developed the curriculum for the first semester of M.Sc.(CS) Part-I Computer Science, which goes beyond traditional academic boundaries. The syllabus is aligned with the NEP 2020 guidelines to ensure that students receive an education that prepares them for the challenges and opportunities of the 21st century. This syllabus has been designed under the framework of the Choice Based Credit System (CBCS), taking into consideration the guidelines set forth by the National Education Policy (NEP) 2020, LOCF (UGC), NCrF, NHEQF, Prof. R.D. Kulkarni's Report, Government of Maharashtra's General Resolution dated 20th April and 16th May 2023, and the Circular issued by SPPU, Pune on 31st May 2023.

A degree in Computer Science subject equips students with the knowledge and skills necessary for a diverse range of fulfilling career paths.Career in Computer Science is considered one of the most high-paying jobs and is full of opportunities; particularly when India's prowess in information technology industry is recognized across the globe. The pool of talented computer professionals working in IT companies of the USA, Canada and other countries shows that IT can take a person to higher levels. Numerous IT companies from India employ huge number of computer professionals in their Indian and overseas offices. Students who are interested in programming**,** software development, and have good analytical and reasoning skills may pursue this course. Job opportunities are available for Graduates and Post Graduates in Government as well as Private sector. Graduates may take up the following job posts- Software Engineer, Software Tester, Data Analyst, Project Manager, Network Administrator, database administrator and Application Developer.

Overall, revising the Computer Science syllabus in accordance with the NEP 2020 ensures that students receive an education that is relevant, comprehensive, and prepares them to navigate the dynamic and interconnected world of today. It equips them with the knowledge, skills, and competencies needed to contribute meaningfully to society and pursue their academic and professional goals in a rapidly changing global landscape.

# Programme Specific Outcomes (PSOs)
# for
# M.Sc. (Computer Science)

After completing M.Sc. Computer Science Program students will be able to:

**PSO1:** 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. Choose to study any one subject among recent trends in IT provided in the optional subjects.

**PSO2:** Students understand all dimensions of the concepts of software application and projects.

**PSO3:** Students understand the computer subjects with demonstration of all programming and theoretical concepts with the use of ICT.

**PSO4:** Developed in-house applications in terms of projects.

**PSO5:** Interact with IT experts & knowledge by IT visits.

**PS06:** Get industrial exposure through the 6 months Industrial Internship in IT industry.

**PS07:** To make them employable according to current demand of IT Industry and responsible citizen.

### Anekant Education Society's

# TuljaramChaturchand College, Baramati

*(Autonomous)*

## Board of Studies (BOS) in Computer Science

**From 2022-23 to 2024-25**

| Sr.No. | Name | Designation |
|---|---|---|
| 1. | Mr. Upendra Choudhari | Chairman |
| 2. | Dr. Vilas Kardile | Member |
| 3. | Mr. Abhijeet Mankar | Member |
| 4. | Mr. Vishal Shaha | Member |
| 5. | Mrs. Prajakta Kulkarni | Member |
| 6. | Mrs. Asmita Bhagat | Member |
| 7. | Mr. Rahul Shah | Member |
| 8. | Mr. Shashikant Nakate | Member |
| 9. | Mr. Purushottam Dixit | Member |
| 10. | Mr. Swapnil Chemte | Member |
| 11. | Mrs. Kalyani Londhe | Member |
| 12. | Mrs. Poornima Gavimath | Member |
| 13 | Dr. Kavita A. Khobragade | Vice-Chancellor Nominee |
| 14 | Dr. Sudhakar Bhoite | Expert from other University |
| 15 | Dr. Suhas S. Satonkar | Expert from other University |
| 16 | Mr. Rohit Shah | Industry Expert |
| 17 | Mr. Yogesh More | Meritorious Alumni |
| 18 | Mr. Abhijeet Chopade | Student Representative |
| 19 | Miss. Rutuja Harihar | Student Representative |
| 20 | Mr. Akshada Kulkarni | Student Representative |
| 21 | Mr. Prajwal Nimbalkar | Student Representative |

### Anekant Education Society's

## TuljaramChaturchand College of Arts, Science and Commerce, Baramati

## (Autonomous)

## Credit Distribution Structure for (M.Sc. (Computer Science)) Part-I (2023 Pattern)

| Year | Level | Sem. | Major | | Research Methodology (RM) | OJT/ FP | RP | Cum. Cr. |
|---|---|---|---|---|---|---|---|---|
| | | | Mandatory | Electives | | | | |
| I | 6.0 | Sem-I | COS-501-MJM: Principles of Programming Language (**Credit 04**)<br>COS-502-MJM: Cryptography and Cyber Forensics (**Credit 04**)<br>COS-503-MJM: Database Technologies (PR) (**Credit 02**)<br>COS-504-MJM: DotNet (Basic) (PR) (**Credit 02**) | COS-511-MJE(A): Design and Analysis of Algorithms (**Credit04**) | COS-521-RM: Research Methodology in Computer Science (**Credit 04**) | -- | -- | **20** |
| | | | | | | | | |
| | | Sem- II | COS-551-MJM: Digital Image Processing (**Credit 04**)<br>COS-552-MJM: Data Mining and Data Warehousing (**Credit 04**)<br>COS-553-MJM: Python Programming-I (Basic) (PR) (**Credit 02**)<br>COS-554-MJM: Dot Net (Advanced) (PR) (**Credit 02**) | COS-561-MJE (A): Artificial Intelligence (**Credit04**)<br>*** 1 Credit = 15 Hr.** | -- | COS-581-OJT/FP **Credit 04** | -- | **20** |
| **Cum. Cr.** | | | **24** | **8** | **4** | **4** | **--** | **40** |

**Anekant Education Society's**

## TuljaramChaturchand College of Arts, Science and Commerce, Baramati (Autonomous)

## Course Structure for (M.Sc. (Computer Science) Part-I (2023 Pattern)

| Sem | Course Type | Course Code | Title of Course | Theory / Pract. | No. of Credits |
|---|---|---|---|---|---|
| I | Major (Mandatory) | COS-501-MJM | Principles of Programming Language | Theory | 4 |
|  |  | COS-502-MJM | Cryptography and Cyber Forensics | Theory | 4 |
|  |  | COS-503-MJM | Database Technologies | Practical | 2 |
|  |  | COS-504-MJM | Dot Net (Basic) | Practical | 2 |
|  | Major (Elective) | COS-511-MJE(A) | Design and Analysis of Algorithms | Theory | 4 |
|  | RM | COS-521-RM | Research Methodology in Computer Science | Theory | 4 |
|  |  |  | **Total Credits:** |  | **20** |
| II | Major (Mandatory) | COS-551-MJM | Digital Image Processing | Theory | 4 |
|  |  | COS-552-MJM | Data Mining and Data Warehousing | Theory | 4 |
|  |  | COS-553-MJM | Python Programming–I (Basic) | Practical | 2 |
|  |  | COS-554-MJM | Dot Net(Advanced) | Practical | 2 |
|  | Major (Elective) | COS-561-MJE (A) | Artificial Intelligence | Theory | 4 |
|  | OJT/FP | COS-581-OJT/FP | On Job Training / Field Projects | ------ | 4 |
|  |  |  | **Total Credits:** |  | **20** |
|  | **Cumulative Credits of Semester – I and II** |  |  |  | **40** |

## SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)

## (w. e. from June, 2023)

| | |
|---|---|
| **Name of the Programme** | : M.Sc. Computer Science |
| **Program Code** | : PSCOS |
| **Class** | : M.Sc. (Computer Science) |
| **Semester** | : I |
| **Course Type** | : Major |
| **Course Name** | : Principle of Programming Language |
| **Course Code** | : COS-501-MJM |
| **No. of Lectures** | : 60 |
| **No. of Credits** | : 04 |

### Course Objectives:

1. To introduce the various programming paradigms.
2. To understand the evolution of programming languages.
3. To understand the concepts of OOPs languages, functional languages, logical and scripting languages.
4. To introduce the principles and techniques involved in design and implementation of modern programming languages.
5. To introduce the notations to describe the syntax and semantics of programming languages.
6. To introduce the concepts of concurrency control and exception handling.
7. To introduce the concepts of ADT and OOP for software development
8. Knowledge of, and ability to use, language features used in current programming languages.
9. An understanding of the key concepts in the implementation of common features of programming languages.
10. Increase the ability to learn new programming languages
11. Increase the capacity to express programming concepts and choose among alternative ways to express things

### Course Outcomes:

After completing this course, the student must demonstrate the knowledge and able to:

**CO1:** Evaluate to enhance and express the syntax and semantics of programming language.

**CO2:** Ability to express syntax and semantics in formal notation.

**CO3:** Ability to apply suitable programming paradigm for the application

**CO4:** Ability to compare the features of various programming languages

**CO5:** Able to understand the programming paradigms of modern programming languages.

**CO6**: Able to understand the concepts of ADT and OOP.

**CO7:** Ability to program in different language paradigms and evaluate their relative benefits.

| Unit | Title and Contents | No. of lectures |
|---|---|---|
| **Unit-I** | **Programming Domains**<br>✓The Art of Language Design - The Programming Language Spectrum, Why Study Programming Languages?<br>✓Types of Programming Language Domains<br># Scientific Applications – Large Number of Floating Point Computations – FORTRAN<br># Business Applications – Produce Reports, Use decimal numbers and characters – COBOL<br># Artificial Intelligence – Symbols rather than numbers manipulated – LISP<br># Systems Programming – Need Efficiency because of continuous use – C<br># Web Software – Eclectic Collection of Languages: Markup (e.g., XHTML), Scripting (e.g., PHP), General-Purpose (e.g., Java)<br># Data Analytics Applications – R Programming, Python Programming | **9** |
| **Unit-II** | **Names, Scopes and Bindings**<br>✓Meaning of Names in Scope-Aliases, Object Lifetime and Storage Management: Static Allocation, Stack-based Allocation, Heap-Based Allocation, Garbage Collection<br>✓The Binding of Referencing Environments - Subroutine Closures, Object Closures, Nested Subroutines, Declaration Order<br>✓Scope Rules, Static Scoping, Dynamic Scoping<br>✓Overloading, Polymorphism and related concepts, Macro Expansion | **8** |
| **Unit-III** | **Data Types**<br>✓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, Implementation of Character String Types.<br>✓User defined Ordinal types - Enumeration types, Designs, Evaluation, Subrange types, Evaluation, Implementation of User defined ordinal types<br>✓Array types - Array initialization, Array operations, Rectangular and Jagged arrays, Slices, Evaluation, Implementation of Array Types<br>✓Associative Arrays – Structure and operations, Implementing Associative arrays<br>✓Record Type – Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types<br>✓Union Type – Design issues, Discriminated versus Free unions, | 9 |

| | | |
|---|---|---|
| | Evaluation, Implementation of Union types<br>✓Pointer and Reference Types - Design issues, Pointer operations, Pointer problems – Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation, Implementation of pointer and reference types, Representation of pointers and references, Solution to dangling pointer problem, Heap management | |
| **Unit-IV** | **Control Flow**<br>✓Expression Evaluation-Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation<br>✓Structured and Unstructured Flow – Structured Alternatives to GOTO Sequencing<br>✓Selection – Short Circuited Conditions, Case/Switch Statements<br>✓Iteration – Enumeration Controlled Loops, Combination, Loops, Iterators, Logically Controlled Loops<br>✓Recursion-Iteration and Recursion, Applicative and Normal Order Evaluation | 9 |
| **Unit-V** | **Subroutines and Control Abstraction**<br>✓Subprograms – Fundamentals of Subprograms, Design Issues for subprograms, Overloaded Subprograms, Nested Subprograms<br>✓Generic Subroutines – Generic Functions in C++, Generic Methods in Java<br>✓Design Issues for Functions, User Defined Overloaded Operators Coroutines<br>✓Parameter Passing Methods, Local Referencing Environments, The General Semantics of Calls and Returns | **9** |
| **Unit-VI** | **Data Abstraction and Object Orientation**<br>✓Encapsulation and Inheritance - Modules, Classes, Nesting, Type, Extensions, Extending without Inheritance<br>✓InitializationandFinalization-ChoosingaConstructor,Referencesand Values, Execution Order<br>✓Dynamic Method Binding-Virtual and Non-Virtual Methods, Abstract Classes and Interfaces, Member Lookup, Polymorphism, Object Closures<br>✓Multiple Inheritance-Semantic Ambiguities, Shared Inheritance, Replicated Inheritance, Mix-In Inheritance | **9** |

**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

## Mapping of this course with Programme Outcomes

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

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

### Mapping of PO1 with All CO'S With Justification:

**CO1:** PSO1 explicitly mentions enriching knowledge in the "Paradigm of Programming Language" as well as other core computing subjects. Syntax and semantics are key to understanding and working within programming paradigms.

**CO2:** PSO1's inclusion of the "Paradigm of Programming Language" and core computing subjects supports understanding formal notation used in programming languages. Formal notations are often taught in conjunction with programming paradigms and related topics like algorithms.

**CO3:** PSO1 the "Paradigm of Programming Language" mentioned in PSO1 directly ties to the ability to choose and apply suitable programming paradigms. It is essential in multiple core computing areas, such as "Software Project Management" and "Advanced Operating Systems."

**CO4:**PSO1 Knowledge of the "Paradigm of Programming Language" allows for direct comparison of features between various languages, which is essential for this outcome. PSO1 helps enrich students' ability to understand these comparisons.

**CO5:** PSO1's emphasis on programming paradigms and recent trends in IT (which include modern programming languages) is closely related to CO5. Understanding modern paradigms is critical in fields like Artificial Intelligence and Cloud Computing.

**CO6:** PSO1 touches on core computing subjects and paradigms of programming languages, which include understanding ADTs and OOP. While it may not focus specifically on these areas, the broader context provides moderate relevance.

**CO7:** PSO1's focus on programming paradigms and the ability to choose among modern programming languages directly supports this outcome. Understanding various paradigms is essential for evaluating their relative benefits.

### Mapping of PO2 with All CO'S With Justification:

**CO1:** PSO2 Understanding software applications and projects involves knowing the programming languages used to build them. Syntax and semantics are foundational to building functional software, making this moderately related.

**CO2:** PSO2 Formal notation is crucial when developing robust software applications. However, it is more specific to language design rather than a broad understanding of software applications, making this partially related.

**CO3:**PSO2 Choosing the right programming paradigm is essential in software development, impacting project success. Thus, this outcome is strongly related to understanding software applications and projects.

**CO4:**PSO2 Comparing programming languages is important for selecting the most suitable one for a software project. This knowledge helps students make informed decisions when developing software, making it moderately related

**CO5:**PSO2 Understanding modern programming paradigms is crucial in developing contemporary software applications, especially when designing large or complex projects. This is moderately related to understanding software concepts.

**CO6:**PSO2 ADT and OOP are fundamental concepts in software development. Software applications often rely on these principles, especially in object-oriented design. This is strongly related to understanding software projects.

**CO7:**PSO2 Being able to program in various paradigms and evaluate their benefits is directly useful in software development and project implementation. Different paradigms offer different strengths depending on the type of software being developed.

**Mapping of PO3 with All CO'S With Justification:**

**CO1:**POS3 Understanding syntax and semantics can be demonstrated using Information and Communication Technology (ICT) tools such as code editors, debuggers, and compilers. PSO3 supports demonstrating programming concepts effectively, making this moderately related.

**CO2:** POS3 Expressing syntax and semantics in formal notation can be demonstrated using ICT tools, like software that formalizes programming languages (e.g., mathematical and theoretical computing platforms).

**CO3:** POS3 Demonstrating different programming paradigms using ICT tools is crucial to helping students understand which paradigm is most suitable for a particular application.

**CO4:** POS3 ICT tools such as IDEs and programming environments can be used to effectively compare programming languages, particularly by demonstrating differences in syntax, semantics, and paradigms.

**CO5:** POS3 ICT tools can be employed to demonstrate the use of modern programming paradigms in practical settings. This supports students in learning about contemporary paradigms in a more interactive way, making this strongly related.

**CO6:** POS3 Concepts like ADT and OOP can be effectively taught and demonstrated using ICT tools (e.g., simulators, IDEs). These demonstrations can enhance understanding of both theoretical and practical aspects of these concepts. This is strongly related.

**CO7:** POS3 Programming in different paradigms and evaluating their benefits can be shown through ICT tools like development environments or simulation platforms. Demonstrating various paradigms using technology helps students compare and evaluate them effectively.

**Mapping of PO4 with All CO'S With Justification:**

**CO1:**POS4 Developing in-house applications requires a deep understanding of syntax and semantics to ensure code correctness and functionality. However, the emphasis on project development may not always directly focus on language evaluation itself.

**CO2:**POS4 While formal notation is useful in some aspects of software development, most in-house applications focus on practical implementation rather than formal language notation. This makes it partially related.

**CO3:**POS4 Choosing and applying the appropriate programming paradigm is key to developing functional and efficient in-house applications. This is strongly related to the success of project-based learning and implementation.

**CO4:**POS4 When developing in-house applications, comparing features of different programming languages is helpful in selecting the most suitable language for a specific project. This is moderately related, as it supports language choice during development.

**CO5:**POS4 Understanding modern programming paradigms is crucial in building robust and efficient in-house applications, especially when using newer or specialized technologies.

**CO6:**POS4 ADT and OOP are foundational for many in-house applications, particularly in object-oriented projects. Developing these types of applications often involves understanding and applying these concepts.

**CO7:**POS4 Programming in different paradigms and evaluating their benefits is directly related to in-house project development. The ability to evaluate which paradigm suits a specific project can improve the effectiveness of the application.

**Mapping of PO5 with All CO'S With Justification:**

**CO1:**POS5 Interacting with IT experts during visits can provide insights into how professionals approach language design and implementation, including the evaluation of syntax and semantics. However, the interaction is indirect and more focused on applied learning.

**CO2:**POS5 While IT visits may offer exposure to industry practices, the ability to express syntax and semantics in formal notation is generally more academic and less directly linked to these interactions. Therefore, it is only partially related.

**CO3:**POS5 During IT visits, students may gain valuable insights into how experts choose and apply programming paradigms for specific applications, giving this outcome practical relevance.

**CO4:**POS5 IT visits can expose students to how experts in the field evaluate and compare different programming languages for their projects, helping them understand the practical reasons behind choosing one language over another.

**CO5:**POS5 Interaction with IT experts during visits can provide first-hand knowledge of the programming paradigms commonly used in the industry today, including modern languages and frameworks. This is strongly related.

**CO6:**POS5 Industry visits can help students see how abstract data types and object-oriented programming are applied in real-world scenarios. Although not the primary focus of most IT visits, they are part of many modern applications.

**CO7:**POS5 IT visits often expose students to diverse programming paradigms used in the field and the rationale behind their use in specific projects.

**Mapping of PO6 with All CO'S With Justification:**

**CO1:**POS6 During a 6-month industrial internship, students would get hands-on experience with programming languages, learning how syntax and semantics are applied in real-world projects. This would give them exposure but may not focus entirely on syntax/semantics evaluation.

**CO2:**POS6 Internships typically emphasize practical applications rather than formal notation. While formal notation may come into play, it is usually not the central focus in industrial settings, making this partially related.

**CO3:**POS6 Internships in the IT industry will provide students with opportunities to apply appropriate programming paradigms to real-world problems. This is a critical skill in project-based work environments.

**CO4:**POS6 Industrial exposure allows students to see how professionals compare programming languages to choose the best one for a specific project, which can enhance their comparative analysis skills.

**CO5:**POS6 Internships provide exposure to modern languages and paradigms used in current IT projects, allowing students to deepen their understanding of modern programming practices.

**CO6:**POS6 Concepts like ADT and OOP are heavily used in industry, especially in object-oriented design and data structures. Industrial internships will help students apply these concepts in real-world

scenarios.
**CO7:**POS6 Industrial exposure allows students to work with different programming paradigms and evaluate their advantages in actual projects, which is a core skill for any IT professional.

**Mapping of PO7 with All CO'S With Justification:**

**CO1:**POS7 Being able to evaluate and express syntax and semantics can improve employability, especially in roles that require a deep understanding of programming languages (e.g., software development). However, it is not the most direct employability skill in a broad sense.
**CO2:**POS7 Formal notation, while important in some specific IT roles (e.g., language design, formal verification), is not a core skill that is widely in demand across all IT industry jobs. This makes it only partially related to employability.
**CO3:**POS7 The ability to apply the correct programming paradigm is essential for employability in IT. Whether developing software, managing projects, or solving technical problems, the ability to choose and apply the right paradigm is highly valued by employers.
**CO4:**POS7 Comparing programming languages is an important skill in many IT jobs, especially when deciding the best technology stack for a project. This enhances employability, particularly for roles in software engineering and architecture.
**CO5:**POS7 Understanding modern programming paradigms is crucial for employability, as these paradigms are the foundation of current IT projects. Knowing these paradigms ensures that students are well-prepared for modern IT roles.
**CO6:**POS7 ADT and OOP are foundational to many programming tasks in the IT industry, especially in software development. Mastery of these concepts is essential for employability in many technical roles, making this strongly related.
**CO7:**POS7 The ability to program in different paradigms and evaluate their benefits is highly valued in the IT industry, especially for roles that require versatility in choosing the right tool or approach for a given problem. This significantly improves employability.

## SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)

## (w. e. from June, 2023)

| | |
|---|---|
| **Name of the Programme** | : M.Sc. Computer Science |
| **Program Code** | : PSCOS |
| **Class** | : M.Sc. (Computer Science) |
| **Semester** | : I |
| **Course Type** | :Major |
| **Course Name** | : Cryptography and Cyber Forensics |
| **Course Code** | : COS-502-MJM |
| **No. of Lectures** | :60 |
| **No. of Credits** | : 4 |

### Course Objectives:

1. To enable students to get sound understanding of Info-Sys-Security, Network Security, Cryptography and cyber forensics.
2. To equip with knowledge and skills necessary to support for their career in Network Security.
3. To encourage them to do further academic studies / research in this area.
4. To develop IT professionals skilled in information/network security and forensic analysis of compromised systems and who are efficient in documentation pertaining to cyber forensic analysis to be provided to the courts of law.
5. Understand principles of web security and to guarantee a secure network by monitoring and analysing the nature of attacks through cyber/computer forensics software/tools.
6. Understand key terms and concepts in Cryptography, Governance and Compliance.
7. To make the student learn different encryption techniques along with hash functions, MAC, digital signatures and their use in various protocols for network security and system security.

### Course Outcomes:

**CO1:** Learn the security concepts and techniques.

**CO2:** In future these experts will be an asset to this country for serving in the fields of information security and digital forensics.

**CO3:** Understand and analyse data encryption standard.

**CO4:** Analyse and evaluate the cyber security needs of an organization.

**CO5:** Determine and analyse software vulnerabilities and security solutions to reduce the risk of exploitation.

**CO6:** Implement cyber security solutions and use of cyber security, information assurance, and cyber/computer forensics software/tools.

**CO7:** Implement various networking protocols.

**TOPICS / CONTENTS:**

| Units | Title & Contents | No.of Lectures |
|---|---|---|
| **Unit–I** | **Introduction to Security, Cryptography and techniques:**<br>The Need for Security, Security Approaches, Principles of Security, Types of Attacks.<br>Introduction to Cryptography, Plain Text and Cipher Text, Substitution Techniques, Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric key cryptography, Steganography. | **09** |
| **Unit–II** | **Symmetric Key Algorithms and AES:**<br>Algorithm Types and Modes, Overview of Symmetric Key Cryptography, DES, IDEA, Blowfish | **09** |
| **Unit–III** | **Asymmetric Key Algorithms, Digital Signature and RSA:** Brief History of Asymmetric Key Cryptography, overview, RSA Algorithm, Comparison between Symmetric & Asymmetric Key Algorithms, Digital Signature | **06** |
| **Unit–IV** | **Digital Certificates and Public Key Infrastructure (PKI):**<br>Introduction, Digital Certificates, private key management. | **04** |
| **Unit–V** | **Introduction to Cyber forensics:** Information Security Investigations, Corporate Cyber Forensics, Scientific method in forensic analysis, investigating large scale Data breach cases. Analyzing malicious software. Types of Computer Forensics Technology, Types of Military Computer Forensic Technology, Types of Law Enforcement: Computer Forensic Technology, Types of Business Computer Forensic Technology, Specialized Forensics Techniques, Hidden Data and How to Find It, Spyware and Adware, Encryption Methods and Vulnerabilities, Protecting Data from Being Compromised Internet Tracing Methods, Security and Wireless Technologies, Avoiding Pitfalls with Firewalls Biometric Security Systems. | **12** |

| | | |
|---|---|---|
| **Unit–VI** | **Types of Computer Forensics Systems:** Internet Security Systems, Intrusion Detection Systems, Storage Area Network Security Systems, Network Disaster Recovery Systems, Satellite Encryption Security Systems, Instant Messaging (IM) Security Systems, Net Privacy Systems, Identity Management Security Systems, Identity Theft, Router Forensics. Cyber forensic stools and case studies. Ethical Hacking: Essential Terminology, Windows Hacking, Malware, Scanning, Cracking. | **10** |

## References:

1. Atul Kahate,“CryptographyandNetworkSecurity”,Second/Thirtd/ForthEdition,McGraw Hill Publication.
2. JohnR. Vacca, ”Computer Forensics: Computer CrimeSceneInvestigation”,2ndEdition,Charles RiverMedia,2005
3. Ravi Kumar & B Jain, “Cyber Forensics - Concepts and Approaches”, icfai universitypress,2006
4. ChristofPaar, JanPelzl, Understanding Cryptography: A Text book forStudentsandPractitioners”,SecondEdition,Springer’s,2010
5. “LiveHacking: The Ultimate Guideto Hacking Techniques & Countermeasures for Ethical Hackers & IT Security Experts”, Ali Jahangiri, Firstedition,2009
6. Kizza, Springer, “Computer Network Security” Harrington, Elsevier, “Network Security

**NOTE: 50 LECTURES FOR CURRICULUM (TEACHING) &10LECTURES FORLEARNING**

### Mapping of this course with Programme Outcomes

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

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

**Mapping of PO1 with All CO’S With Justification:**

**CO1:**POS1 Security concepts and techniques are critical to many advanced areas like cloud computing and web services, and they contribute to core computing knowledge.

**CO2:** POS1 Information security and digital forensics are specialized fields, strongly related to enhancing knowledge in cyber security, a growing trend in IT.

**CO3:** POS1 Data Encryption Standard (DES) is directly tied to understanding encryption algorithms,

which are important in securing data in AI, cloud computing, and databases.
**CO4:** POS1 Evaluating cyber security needs aligns with IT trends like cloud computing, mobile technologies, and database security.
**CO5:** POS1 Software vulnerabilities and security solutions are directly relevant to core computing subjects, particularly in areas like advanced operating systems and web services.
**CO6:** POS1 Cyber security solutions and tools are vital across several emerging IT trends, including cloud computing and AI.
**CO7:** POS1 Networking protocols are fundamental to many core computing subjects, including web services, cloud computing, and mobile technologies.

**Mapping of PO2 with All CO'S With Justification:**

**CO1:** POS2 Security concepts are crucial for building secure software applications, ensuring that projects meet confidentiality, integrity, and availability requirements.
**CO2:** POS2 Knowledge in information security and digital forensics enhances the ability to manage security-focused projects and applications.
**CO3:** POS2 Understanding and analyzing DES is essential for implementing secure encryption in software applications, contributing to the security aspect of software projects.
**CO4:** POS2 Evaluating an organization's cyber security needs is critical in designing and managing secure software applications and project planning.
**CO5:** POS2 Identifying and addressing software vulnerabilities is directly related to developing robust and secure software applications and projects.
**CO6:** POS2 Implementing cyber security solutions is a key aspect of designing secure software applications and managing secure software projects.
**CO7:** POS2 Networking protocols are integral to building software applications that rely on communication, especially in projects that involve distributed systems.

**Mapping of PO3 with All CO'S With Justification:**

**CO1:** POS3 Security concepts and techniques are critical theoretical foundations in computer science and are often demonstrated using ICT tools in teaching environments.
**CO2:** POS3 ICT tools are frequently used for digital forensics and information security training, demonstrating practical skills in a programming and theoretical context.
**CO3:** POS3 Learning and analyzing DES involves both programming exercises (to implement or simulate DES) and theoretical understanding, often demonstrated using ICT tools.
**CO4:** POS3 Analyzing cyber security needs involves theoretical and practical approaches that can be effectively demonstrated through ICT tools, showing real-time evaluations.
**CO5:** POS3 Identifying and mitigating vulnerabilities requires programming knowledge and is frequently demonstrated using ICT platforms for testing and simulation.
**CO6:** POS3 Implementing cyber security solutions and using forensic tools directly aligns with practical demonstrations using ICT in both programming and theory-based environments.
**CO7:** POS3 Networking protocols require both theoretical understanding and practical implementation, often demonstrated using ICT labs and network simulation tools.

**Mapping of PO4 with All CO'S With Justification:**

**CO1:** POS4 Understanding security concepts is fundamental for developing secure in-house applications, ensuring that projects meet security standards.
**CO2:** POS4 Skills developed in information security and digital forensics can be applied to in-house projects, particularly in developing security-focused applications.
**CO3:** POS4 Encryption techniques like DES are often implemented in in-house applications to secure

data, making it relevant for project development.
**CO4:** POS4 Evaluating cyber security needs is essential when developing in-house applications, especially those that need to meet organizational security standards.
**CO5:** POS4 Identifying and mitigating software vulnerabilities is critical in the development of secure in-house applications, making it a strong fit for project development.
**CO6:** POS4 Implementing cyber security solutions is key when building secure in-house projects, as these solutions are integrated into the applications developed.
**CO7:** POS4 Networking protocols are integral to many in-house projects, especially those involving communication between systems or distributed applications.

**Mapping of PO5 with All CO'S With Justification:**
**CO1:** POS5 Interacting with IT experts can enhance understanding of security concepts and techniques,as real-world practices and expert insights are often shared during visits.
**CO2:** POS5 IT visits and interactions with professionals in the field of information security and digital forensics provide valuable insights into the industry, making it strongly related**.**
**CO3:** POS5 Discussing encryption standards like DES with IT experts during visits can give students practical perspectives and industry best practices, enhancing their understanding.
**CO4:** POS5 IT visits offer opportunities to observe and discuss real-world cyber security challenges faced by organizations, making it highly relevant.
**CO5:** POS5 Expert interactions and visits can provide real-life examples of vulnerabilities and security measures, offering hands-on knowledge to students.
**CO6:** POS5 Observing cyber security implementations during IT visits and discussing tools with experts can help students understand practical aspects of security solution deployment.
**CO7:** POS5 IT visits can expose students to networking protocols in action, especially in large-scale networks, providing real-world understanding of their implementation.

**Mapping of PO6 with All CO'S With Justification:**
**CO1:** POS6 Security concepts and techniques are directly applied in the IT industry, making them essential for practical tasks during the internship.
**CO2:** POS6 The internship provides valuable skills for future roles in national security, but its immediate focus is on foundational learning rather than directly contributing to national service.
**CO3:** POS6 The use of encryption, particularly standards like DES, is vital for data protection in real- world IT environments, strongly tying it to industrial tasks.
**CO4:** POS6 Identifying and assessing organizational cyber security needs is a central part of industry internships, where interns often work on security audits or threat assessments.
**CO5:** POS6 A key aspect of cyber security internships involves identifying vulnerabilities and applying solutions, directly aligning with this outcome.
**CO6:** POS6 Interns typically get hands-on experience with cyber security tools, making this outcome closely tied to real-world applications.
**CO7:** POS6 Networking protocols are relevant but may not be the primary focus in all cyber security internships, which could emphasize other security aspects more.

**Mapping of PO7 with All CO'S With Justification:**
**CO1:** POS7 A solid understanding of security concepts and techniques is essential for employability In the IT industry, particularly in cyber security roles, and fosters responsible practices in protecting data.
**CO2:** POS7 While this outcome focuses on future contributions to national security, the skills learned will also make students highly employable in the IT industry, particularly in cyber security.

**CO3:** POS7 Knowledge of encryption standards like DES is vital for IT industry roles involving data protection, making this outcome closely related to employability.
**CO4:** POS7 Being able to assess and address cyber security needs is a critical skill in the IT industry, directly enhancing employability and contributing to the responsible management of organizational security.
**CO5:** POS7 The ability to identify vulnerabilities and implement security solutions is highly sought after in the IT industry, directly improving employability and supporting ethical, responsible cyber security practices.
**CO6:** POS7 Practical skills in cyber security tools are highly valued in the IT industry, making this outcome strongly related to employability and ensuring that students can act responsibly in protecting digital assets.
**CO7:** POS7 While knowledge of networking protocols is important for certain IT roles, it may not be As critical for all cyber security-related jobs, making it moderately related to employability in the border IT industry.

**SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)**

**(w. e. from June, 2023)**

| | |
|---|---|
| **Name of the Programme** | : M.Sc. Computer Science |
| **Program Code** | : PSCOS |
| **Class** | : M.Sc. (Computer Science) |
| **Semester** | :I |
| **Course Type** | :Major |
| **Course Name** | : Database Technologies |
| **Course Code** | : COS-503-MJM |
| **No. of Lectures** | : 60 Hours (15 Practical) |
| **No. of Credits** | : 02 Credits |

## Course Objectives:

1. Students will gain knowledge about unstructured database and its importance.
2. Students will understand the structure of MongoDB and various operations of it.
3. Students will study and analyze the difference between structured and unstructured database.
4. Students will study aggregation operations in MongoDB.
5. Students will understand the front-end connectivity with MongoDB.
6. Students will gain knowledge about index and its importance in queries.

**Course Outcomes:** After completing this course, students will be able to

**CO1:** Solve the problem query by using appropriate command in MongoDB.

**CO2:** Categorize the different operations into appropriate groups.

**CO3:** Distinguish between writing query in RDBMS and in MongoDB.

**CO4:** Choose the proper aggregation operations in query.

**CO5:** Create the proper index for solving queries in MongoDB.

**CO6:** Demonstrate how to connect MongoDB with front end.

**CO7:** Design database for an application with MongoDB as backend.

**TOPICS/CONTENTS:**

| | |
|---|---|
| **UNIT 1:** MongoDB Installation Basics | (04 hours) |
| **UNIT 2:** Creating database, collections in MongoDB | (04 hours) |
| **UNIT 3:** Basic commands in MongoDB | (08 hours) |
| **UNIT 4:** CRUD Operations (find, insert) | (12 hours) |
| **UNIT 5:** CRUD Operations (update, delete) | (12 hours) |
| **UNIT 6:** Indexing | (04 hours) |
| **UNIT 7:** Aggregation | (08 hours) |
| **UNIT 8:** Connecting with Front-End | (04 hours) |
| **UNIT 9:** MongoDB Administration | (04 hours) |

## References:

1. Bradshaw, Brazil, Chodorow(2020).MongoDB: The Definitive Guide. Shroff/O’Reilly. Third Edition.
2. Banker, Bakkum, Verch, Garrett, Hawkins(2016). MongoDB in Action. Dreamtech Press. Second Edition.
3. MongoDB Manual: https://docs.mongodb.com/manual/

### Mapping of this course with Programme Outcomes

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

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

**Mapping of PO1 with All CO'S With Justification:**

**CO1**:PO1Solving queries in MongoDB enhances knowledge in database technologies.

**CO2:**PO1Categorizing operations aids in understanding various database functionalities relevant to IT trends.

**CO3:**PO1 Distinguishing between RDBMS and MongoDB promotes a deeper understanding of database paradigms, integral to PSO1.

**CO4:**PO1 Choosing proper aggregation operations demonstrates advanced database management skills aligned with PSO1.

**CO5:**PO1 Creating indexes optimizes query performance, reflecting knowledge in advanced database concepts.

**CO6:**PO1 Demonstrating connectivity between MongoDB and the front end showcases integration skills crucial in modern IT.

**CO7:**PO1 Designing a database with MongoDB as backend encompasses comprehensive understanding of database technologies, strongly related to PO1.

**Mapping of PO2 with All CO'S With Justification:**

**CO1:**PO2 Solving queries in MongoDB contributes to understanding software applications through practical problem-solving.

**CO2:**PO2 Categorizing operations enhances comprehension of software processes and their application in projects.

**CO3:**PO2 Distinguishing between query languages aids in grasping software development intricacies relevant to project work.

**CO4:**PO2 Choosing aggregation operations is essential for effective data handling in applications, linking to project requirements.

**CO5:**PO2 Creating indexes is a vital aspect of optimizing software applications, aligning with best practices in project management.

**CO6:**PO2 Demonstrating front-end connectivity illustrates the integration of backend databases into applications, a core aspect of project development.

**CO7:**PO2 Designing a database for an application directly relates to understanding software project requirements and architecture.

**Mapping of PO3 with All CO'S With Justification:**

**CO1:**PO3 Solving queries in Mongo DB involves practical application of programming concepts, demonstrating theoretical knowledge in action.

**CO2:**PO3 Categorizing operations reflects understanding of programming logic, enhancing theoretical foundations.

**CO3:**PO3 Distinguishing between query languages showcases a deep understanding of theoretical concepts in database management.

**CO4 :**PO3 Choosing aggregation operations requires application of programming principles, linking theory to practice.

**CO5:**PO3 Creating indexes involves both theoretical knowledge and practical skills, demonstrating effective database management techniques.

**CO6 :**PO3 Demonstrating front-end connectivity integrates programming concepts with practical application, aligning closely with ICT usage.

**CO7:**PO3 Designing a database for an application shows mastery of theoretical concepts and their practical implementation in software projects.

**Mapping of PO4 with All CO'S With Justification:**

**CO1:**PO4 Solving queries in MongoDB is crucial for project development, enabling effective data

management within applications.
**CO2:**PO4 Categorizing operations is important for structuring project functionalities, enhancing application organization.
**CO3:**PO4 Understanding differences between RDBMS and MongoDB is vital for making informed design choices in application projects.
**CO4:**PO4 Choosing appropriate aggregation operations is necessary for efficient data handling in project applications.
**CO5:**PO4 Creating proper indexes directly impacts application performance, which is essential for developing effective in-house solutions.
**CO6:**PO4 Demonstrating front-end connectivity is key for full-stack application development, ensuring smooth user interaction with back-end data.
**CO7:**PO4 Designing a database for an application is fundamental to project development, directly linking theoretical concepts to practical implementation.

**Mapping of PO5 with All CO'S With Justification:**
**CO1:**PO5 Solving MongoDB queries can be a topic of discussion during IT visits, facilitating interaction with experts on practical database issues.
**CO2:**PO5 Categorizing operations provides foundational knowledge that may be reinforced during discussions with IT professionals.
**CO3:**PO5 Understanding the differences between RDBMS and MongoDB can lead to insightful conversations with experts about best practices.
**CO4:**PO5 Choosing proper aggregation operations may be explored in IT visits, enhancing practical knowledge through expert insights.
**CO5:**PO5 Discussing indexing strategies during IT visits can help students understand real-world applications of performance optimization
**CO6:**PO5 Demonstrating front-end connectivity with MongoDB directly relates to industry practices, making it a relevant topic for expert interactions.
**CO7:**PO5 Designing a database for an application aligns closely with industry expectations, fostering rich discussions with IT professionals.

**Mapping of PO6 with All CO'S With Justification:**
**CO1:**PO6 Solving MongoDB queries is directly applicable in real-world projects, providing valuable hands-on experience during the internship.
**CO2:**PO6 Categorizing operations is a useful skill for understanding project requirements in an industrial setting.
**CO3:**PO6 Distinguishing between RDBMS and MongoDB queries is essential for adapting to various database systems encountered in the industry.
**CO4:**PO6 Choosing proper aggregation operations is relevant for data analysis tasks that interns may face in real projects.
**CO5:**PO6 Creating proper indexes is crucial for performance optimization in production databases, a key skill for interns
**CO6:**PO6 Demonstrating how to connect MongoDB with front end is a vital competency for full-stack development roles in internships.
**CO7:**PO6 Designing a database for an application is a core task interns will likely engage in, providing practical exposure to software development.

**Mapping of PO7 with All CO'S With Justification:**
**CO1:**PO7 Solving MongoDB queries equips students with in-demand skills, enhancing their employability in data-driven roles.

**CO2:**PO7 Categorizing operations demonstrates analytical thinking, which is valued by employers in the IT industry.

**CO3:**PO7 Understanding differences between RDBMS and MongoDB is essential for versatility in various jo roles in database management.

**CO4:**PO7 Choosing appropriate aggregation operations prepares students for tasks involving data manipulatio and reporting, increasing employability.

**CO5:**PO7 Creating proper indexes is a critical skill for optimizing database performance, making students mo attractive to employers.

**CO6 :**PO7 Demonstrating front-end connectivity with Mongo DB is crucial for full-stack development roles, enhancing job readiness.

**CO7:**PO7 Designing a database for applications directly aligns with job requirements in software developmen ensuring employability.

## SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)

## (w. e. from June, 2023)

| | |
|---|---|
| **Name of the Programme** | : M.Sc. Computer Science |
| **Program Code** | : PSCOS |
| **Class** | : M.Sc. (Computer Science) |
| **Semester** | : I |
| **Course Type** | : Major |
| **Course Name** | : Dot Net (Basic) |
| **Course Code** | : COS-504-MJM |
| **No. of Lectures** | : 60 Hours (15 Practical) |
| **No. of Credits** | : 02 Credits |

### Course Objectives:

1. Able to understand the DOTNET framework
2. C# language features and Windows application development using C#.Net
3. C# is used to understand, diagram, and implement programming concepts.
4. C# decision structures use iteration, class methods, fields, and properties to find logistical alternatives.
5. Creating Desktop Applications using .Net Controls
6. Able to understand the Entity framework
7. Use of Entity Framework in the programming environment

### Course Outcomes:

**CO1:** Understand the Microsoft .NET Framework and C#.NET structure

**CO2:** Design application with variety of controls

**CO3:** Access the data using inbuilt data access tools.

**CO4:** Use Microsoft ADO.NET to access data in Application

**CO5:** Configure and deploy C# Application

**CO6:** Develop secured C# application

**CO7:** Identify and resolve problems (debug /trouble shoot) in C#.NET window-based application

*TOPICS/CONTENTS:*

| | |
|---|---|
| **UNIT1:** Parameter Modifiers (ref, out, params) | **(04 hours)** |
| **UNIT2:** Delegate and Events | **(04 hours)** |
| **UNIT3:** Inheritance and Interface | **(04 hours)** |
| **UNIT4:** Polymorphism (Method Overloading , Operator Overloading and Method Overriding | **(04 hours)** |
| **UNIT5:** Exception Handling | **(04 hours)** |
| **UNIT6:** Collections | **(04 hours)** |
| **UNIT7:** Generics | **(04 hours)** |
| **UNIT8:** Use of Basics Form Controls | **(04 hours)** |
| **UNIT9:** Use of Dialogue Boxes | **(04 hours)** |
| **UNIT10:** Simple Database Operations | **(04 hours)** |
| **UNIT11:** Advanced Database Operations | **(04 hours)** |
| **UNIT12:** Simple Crystal Report | **(04 hours)** |
| **UNIT13:** Advanced Crystal Report | **(04 hours)** |
| **UNIT14:** Event Handling (Calculator) | **(04 hours)** |
| **UNIT15:** Entity Framework | **(04 hours)** |

## References:

1. Beginning Visual C#, Wrox Publication
2. Professional Visual C#, Wrox Publication
3. Database Programming with C#, By Carsten Thomsen, Apress
4. Beginning C# Object-Oriented Programming By Dan Clark , Apress
5. Beginning C# Object-Oriented Programming By Dan Clark , Apress

## Mapping of this course 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 | 2 | 3 | 2 | 3 | 3 |
| **CO4** | 3 | 3 | 2 | 3 | 2 | 3 | 3 |
| **CO5** | 3 | 3 | 2 | 3 | 3 | 3 | 3 |
| **CO6** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **CO7** | 2 | 3 | 3 | 3 | 3 | 3 | 3 |

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

**Mapping of PO2 with All CO’S With Justification:**

**CO1:**POS2 A deep understanding of the .NET Framework and C# structure is fundamental to comprehending the architecture and functionality of software applications, strongly relating to PSO2’s goal of mastering all aspects of software projects.

**CO2:**POS2 Designing applications with various controls is a key component of software application development, and it contributes significantly to understanding user interface design and functionality, a crucial aspect of software projects.

**CO3:**POS2 Data access is a core element of most software applications, and understanding how to efficiently use built-in tools is essential to building effective, data-driven applications, aligning with PSO2’s focus on full project understanding.

**CO4:**POS2 ADO.NET is a critical tool for data access and management in software projects, enabling students to develop applications with robust data-handling capabilities, which is integral to understanding the full lifecycle of software applications.

**CO5:**POS2 The ability to configure and deploy applications is crucial for understanding the end-to-end process of software development and delivery, directly contributing to the PSO2 objective of mastering all dimensions of a software project.

**CO6:**POS2 Developing secure applications ensures students grasp the importance of security in software projects, a critical dimension of modern software development, making it strongly related to understanding full project requirements.

**CO7:**POS2 Debugging and troubleshooting are essential for ensuring the quality and functionality of software, representing a key part of the software development lifecycle and strongly related to understanding all dimensions of software projects.

**Mapping of PO3 with All CO’S With Justification:**

**CO1:**POS3 Understanding the .NET framework and C# structure is foundational to grasping programming concepts, and learning this with ICT tools (like IDEs, tutorials, and simulators) helps in demonstrating both theoretical and practical aspects of the subject.

**CO2:**POS3 Designing applications with controls integrates programming logic with visual demonstration tools, making it easier for students to conceptualize and demonstrate programming concepts using ICT.

**CO3:**POS3 Accessing data with built-in tools demonstrates practical aspects of database management, which can be effectively taught and understood using ICT. This helps students understand data-driven programming, aligning with PSO3’s focus on practical concepts.

**CO4:**POS3 ADO.NET provides the practical means to handle database operations, a concept that can be demonstrated and explored through ICT tools (like Visual Studio). However, it is more focused on data handling than on broader computer subject understanding.
**CO5:**POS3 Configuring and deploying applications involves both theoretical understanding (deployment strategies) and practical skills (using ICT tools for deployment). Demonstration of these activities directly involves ICT.
**CO6:**POS3 Secure application development covers both theory (security principles) and practice (implementing security in C# apps). Demonstrating security-related concepts can benefit greatly from ICT tools for simulating attacks or testing security protocols.
**CO7:**POS3 Debugging and troubleshooting are key skills in understanding programming, and they are inherently tied to the use of ICT tools such as debuggers, IDEs, and log analyzers, which provide a hands-on approach to resolving issues.

## Mapping of PO4 with All CO'S With Justification:

**CO1:**POS4 A thorough understanding of the .NET framework and C# is essential to begin any project using this platform. Without this foundational knowledge, developing in-house applications would not be feasible.
**CO2:**POS4 Designing applications with various controls is crucial for developing user-friendly and functional in-house applications. This CO directly supports the ability to create projects with intuitive interfaces and necessary functionality.
**CO3:**POS4 In-house applications often need to manage and access data effectively. Learning how to use data access tools is key to building functional software, especially those that interact with databases or manage large sets of data.
**CO4:**POS4 ADO.NET is a fundamental tool for handling data in C# applications, which makes it highly relevant for in-house projects that require robust data access and manipulation.
**CO5:**POS4 Configuring and deploying applications is the final step in project development. This outcome ensures that students can not only develop but also deliver fully functional applications as part of their in-house projects.
**CO6:**POS4 Security is a critical aspect of any project, especially in-house applications that handle sensitive data or internal processes. This CO directly supports the development of secure applications, which is vital in any professional environment.
**CO7:**POS4 Debugging and troubleshooting are essential for successful project development. Any in-house application will inevitably face issues during development, and the ability to resolve these problems efficiently is crucial for delivering a functional project.

## Mapping of PO5 with All CO'S With Justification:

**CO1:**POS5 Understanding the structure of .NET and C# can enhance conversations with IT experts regarding industry standards, best practices, and the technical architecture of enterprise solutions. It provides a foundational understanding that enables meaningful discussions.
**CO2:**POS5 Designing applications with various controls allows students to discuss user interface design and experience (UI/UX) with IT experts. This CO can help students ask relevant questions and understand advanced design concepts during IT visits.
**CO3:**POS5 Data access is a key component in many IT systems. Having knowledge of data access tools provides a basis for engaging with professionals about database management, enterprise systems, and data architecture during IT visits.
**CO4:**POS5 ADO.NET is a commonly used tool in the industry. During IT visits, students can interact with professionals about database interaction, performance optimization, and integration practices in enterprise applications. This CO helps facilitate deeper technical discussions.

**CO5:**POS5 Deploying applications is a vital aspect of software development. Engaging with IT experts on the deployment practices, cloud deployment strategies, and DevOps practices would be highly relevant. This CO gives students the knowledge to ask insightful questions on these topics.
**CO6:**POS5 Security is a crucial concern in IT, and professionals often focus on it. Students can interact with IT experts about security standards, best practices, and challenges in securing applications. Knowledge from this CO enables students to understand and discuss real-world security issues.
**CO7:**POS5 Debugging and troubleshooting are frequent topics in IT. Discussing problem-solving strategies and best practices for debugging with IT experts is essential. Students can engage in conversations about tools, techniques, and approaches used in professional environments.

**Mapping of PO6 with All CO'S With Justification:**

**CO1:**POS6 Understanding the structure of the .NET framework and C# is essential for working on projects during the internship, especially if students are placed in roles that involve software development using these technologies. It helps students navigate the technical environment of many IT companies.
**CO2:**POS6 Designing applications with various controls is a practical skill highly useful in internships where students may be required to contribute to front-end development or create user interfaces for applications. This hands-on ability is directly applicable to internship tasks.
**CO3:**POS6 Data access is crucial for many projects in the IT industry, especially in enterprise or web applications. Being proficient in accessing data with inbuilt tools is vital for internships that involve working with databases or data-driven applications.
**CO4:**POS6 ADO.NET is widely used in the industry for handling data operations in applications. During the internship, students will likely work on projects that require them to connect applications to databases, making this CO highly relevant to their industrial exposure.
**CO5:**POS6 Configuring and deploying applications is a common task during internships, especially in environments where software is developed, tested, and delivered to clients. Knowing how to deploy C# applications is directly applicable to working on live projects.
**CO6:**POS6 Security is a key concern in the IT industry, and many internship projects will involve ensuring that applications are secure. This CO prepares students to think about security while developing software, making it a valuable skill in industry settings.
**CO7:**POS6 Debugging and troubleshooting are essential skills in the industry, especially during internships where students may need to fix issues in existing projects or test new developments. This CO ensures students are equipped to resolve technical issues effectively.

**Mapping of PO7 with All CO'S With Justification:**

**CO1:**POS7 The .NET framework and C# are widely used in the IT industry for developing a variety of applications. A strong understanding of these technologies is crucial for employability in many development roles, making this knowledge directly aligned with current industry demands.
**CO2:**POS7 Designing applications with various controls is a skill sought after in the IT industry, especially in UI/UX development. Knowledge of building user-friendly, functional applications is key to becoming employable in roles like front-end or full-stack development.
**CO3:**POS7 Efficient data access and manipulation are essential for developing enterprise applications and working with databases. Many IT jobs, especially in back-end and full-stack development, require this competency, making students highly employable in data-driven roles.
**CO4:**POS7 ADO.NET is a key technology for handling data access in C# applications, which aligns with industry needs for developing scalable and efficient applications. Employers value candidates who are skilled in accessing and managing data, making this a crucial employability skill.
**CO5:**POS7 Application deployment is a critical skill for IT professionals, as it demonstrates the ability to deliver completed software projects. This outcome enhances employability by preparing students for

roles that involve end-to-end software development, including deployment.

**CO6:**POS7 Security is a top concern in today's IT industry. Knowing how to develop secure applications ensures that students are prepared to meet the industry's increasing focus on cyber security. This skill is highly valuable for employability, especially in industries handling sensitive data.

**CO7:**POS7 Debugging and troubleshooting are essential skills in the IT industry. Being able to resolve issues efficiently makes candidates highly employable, as problem-solving is a critical aspect of software development and maintenance.

## SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)

## (w. e. from June, 2023)

| | |
|---|---|
| **Name of Programme** | **:** M.Sc.(Computer science) |
| **ProgramCode** | **:** PSCOS |
| **Class** | **:** M.Sc.(Computer science) |
| **Semester** | **:** I |
| **Course Type** | **:** Elective |
| **Course Name** | **:** Design& Analysis of Algorithm |
| **Course Code** | **:** COS-511-MJE(A) |
| **No. of Lectures** | **:** 60 |
| **No. of Credits** | **:** 4 |

**Course Objectives:** Student successfully completing this course will be able to

1. Understand Basic Algorithm Analysis techniques and the use o-asymptotic notation
2. Understand different design strategies
3. Understand the use of data structure proving algorithm performance
4. Understand classical problem and solutions
5. Learn a variety of useful algorithms
6. Understand classification of problems

**Course Outcomes:** At the end of the course, students should be able to:

**CO1:** Understand Tree Traversal method and Greedy Algorithms

**CO2:** Understand algorithm design techniques

**CO3:** Learn how to analyze algorithm and estimate their worst case and average case behavior.

**CO4:** Identify and understand various Time and Space complexities of various algorithms.

**CO5:** Find optimal solution by applying various methods CO6: Design optimal solution by applying various methods.

**CO7:** learn how to apply their theoretical knowledge in practice

| Units | Title and Contents | No. of Lectures |
|---|---|---|
| **Unit–I** | **Analysis & Design Strategies**<br>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). Divide and Conquer-control abstraction, binary search,merge sort,Quick sort,Strassen's matrix Multiplication | **10** |
| **Unit–II** | **Greedy Method**<br>Knapsack problem,job sequencing with deadlines ,minimum-costs Spanning trees, Kruskal's and Prim's algorithm, optimal storage on tapes, optimal merge patterns, Huffman coding | **10** |
| **Unit–III** | **Dynamic programming**<br>Matrix chain multiplication, single source shortest paths,Dijkstra's algorithm, Bellman-ford algorithm, all pairs shortest path, longest common sub sequence,string editing ,0/1knapsack problem, Traveling salesperson problem. | **10** |
| **Unit–IV** | **Decrease and conquer**<br>DFS and BFS, Topological sorting, strongly connected components | **6** |
| **Unit-V** | **Backtracking & Branch and Bound Technique**<br>General method, 8 queen's problem, sum of subset problem, graph coloring problem, Hamiltonian cycle.FIFO,LIFO,LCBB,TSP problem, 0/1 knapsack Problem. | **8** |
| **Unit–VI** | **Transform and Conquer & Problem Classification**<br>Horner's Rule and Binary Exponentiation– Problem Reduction Non deterministic algorithm, The class of P,NP,NP-hard and NP- Complete problems, significance of Cook's theorem. | **6** |

**NOTE:50 LECTURES FOR CURRICULUM (TEACHING) &10 LECTURES FOR LEARNING**

**References:**

1. EllisHorowitz,SartajSahni&SanguthevarRajasekaran,ComputerAlgorithms,Galgotia.

2. T.Cormen,C.Leiserson,&R.Rivest,Algorithms,MITPress,19901

3. A.Aho,J.Hopcroft,&J.Ullman,TheDesignandAnalysisofComputerAlgorithms, AddisonWesley,1974

4. DonaldKnuth,TheArtofComputerProgramming(3vols.,variouseditions,1973-81),AddisonWesley

5. StevenSkiena,TheAlgorithmManual,SpringerISBN:9788184898651

6. Jungnickel,Graphs,NetworksandAlgorithms,Springer,ISBN:3540219056

### Mapping of this course with Programme Outcomes

| Course Outcomes | Programme Outcomes (POs) | | | | | | |
|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 |
| CO1 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| 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**

**Mapping of PO1 with All CO'S With Justification:**

**CO1:**POS1 Tree traversal and greedy algorithms are essential components of algorithm design, which are foundational to several areas mentioned in PSO1, including Artificial Intelligence, Web Services, and core computing subjects. This CO directly contributes to the deep understanding of algorithmic approaches, particularly in the design and analysis of algorithms.

**CO2:**POS1 Algorithm design techniques are at the core of developing efficient solutions in computing areas such as AI, cloud computing, and mobile technologies. A deep understanding of algorithm design techniques is crucial for advanced computing subjects and aligns well with PSO1.

**CO3:**POS1 Analyzing algorithm efficiency is vital in the areas like design and analysis of algorithms, database technologies, and advanced operating systems. This CO enables students to critically evaluate algorithm performance, directly enhancing knowledge in core computing subjects.

**CO4:**POS1 Time and space complexity are essential metrics in evaluating algorithm performance, relevant to all the computing fields mentioned in PSO1, such as AI, cloud computing, and mobile technologies. Understanding complexity is crucial for developing optimal solutions.

**CO5:**POS1 The ability to find optimal solutions is key in areas like AI, database technologies, and software project management, where efficiency and optimal resource use are critical. This CO ties directly into applying algorithmic knowledge to solve real-world computing problems.

**CO6:**POS1 Designing optimal solutions is central to several topics under PSO1, particularly in software development, cloud computing, and AI, where efficiency and optimization are vital. This CO ensures students can apply theoretical knowledge to practical challenges in advanced computing fields.

**CO7:**POS1 Applying theoretical knowledge to practice is crucial for all areas listed under PSO1, including software project management, AI, and advanced operating systems. This CO ensures that students are prepared to bridge the gap between theoretical understanding and practical implementation.

**Mapping of PO2 with All CO'S With Justification:**

**CO1:**POS2 Tree traversal and greedy algorithms are foundational in many software applications, such as in search engines, database indexing, and optimization problems. Understanding these concepts helps

students grasp the underlying structures and techniques used in real-world software projects.
**CO2:**POS2 Algorithm design techniques are critical for building efficient software applications. Knowledge of different design approaches, such as divide and conquer, dynamic programming, and greedy algorithms, enables students to address various challenges in software development.
**CO3:**POS2 Understanding the performance of algorithms in terms of worst-case and average-case behavior is crucial in developing software applications that are both robust and efficient. This ability is essential for software project development and maintenance.
**CO4:**POS2 Time and space complexity analysis is vital for designing scalable and optimized software applications. It allows students to understand the resource implications of their algorithm choices, which is a key consideration in software project development.
**CO5:**POS2 Finding optimal solutions is directly related to creating effective software applications. This skill ensures that students can solve complex problems in software projects by choosing the most efficient approach.
**CO6:**POS2 Designing optimal solutions is critical in the development of high-performance software. This CO is strongly related to understanding and implementing all dimensions of software applications, as it involves critical thinking and the application of theoretical knowledge to practical software challenges.
**CO7:**POS2 Applying theoretical knowledge in practical settings is key to successfully working on software projects. This CO bridges the gap between understanding theory and using it to solve real-world software development challenges, making it essential for comprehensive project development.

**Mapping of PO3 with All CO'S With Justification:**
**CO1:**POS3 Tree traversal methods and greedy algorithms are fundamental algorithmic techniques that can be effectively demonstrated using ICT tools such as coding platforms and visualization software. These tools help students better understand the concepts through interactive learning.
**CO2:**POS3 Algorithm design techniques are crucial for both programming and theoretical understanding. ICT tools can be used to demonstrate the step-by-step process of algorithm design, making this CO highly relevant for students to understand and visualize these techniques.
**CO3:**POS3 Analyzing algorithms, including worst-case and average-case scenarios, can be enhanced using ICT-based simulations and tools that provide students with immediate feedback and visualization of algorithm performance. This greatly aids in understanding theoretical concepts.
**CO4:**POS3 Time and space complexity are theoretical concepts that can be difficult to grasp without practical demonstrations. Using ICT tools to simulate algorithm performance and measure complexity helps students better understand these critical concepts.
**CO5:**POS3 ICT tools can be used to implement and test different optimization methods, allowing students to apply their theoretical knowledge in a practical setting. This CO is important for helping students understand how different methods lead to optimal solutions.
**CO6:**POS3 ICT-based platforms allow for the design, testing, and improvement of solutions, enabling students to apply theoretical knowledge and design optimal algorithms in a practical, interactive environment. This is a key skill for understanding and demonstrating programming concepts.
**CO7:**POS3 ICT tools play a crucial role in bridging the gap between theory and practice. Demonstrating theoretical concepts through programming exercises, visual tools, and simulations makes this CO strongly related to PSO3, as it allows students to apply theoretical knowledge in practical scenarios.

**Mapping of PO4with All CO'S With Justification:**
**CO1:**POS4 Tree traversal and greedy algorithms are commonly used in real-world applications, such as in project development involving optimization problems, data structures, and search algorithms. Understanding these methods helps in the design of efficient in-house applications.
**CO2:**POS4 Algorithm design is crucial when developing in-house applications, as it equips students with

the knowledge needed to structure efficient and scalable solutions. This CO directly supports the project development process by ensuring a strong foundation in designing software algorithms.
**CO3:**POS4 In project development, understanding the performance of algorithms in different scenarios is essential for creating robust applications. This CO ensures that students can evaluate the efficiency of their solutions, which is key when building reliable in-house applications.
**CO4:**POS4 Knowledge of time and space complexities allows students to optimize their projects in terms of resource usage. This is highly important for in-house application development, where efficiency and scalability are critical factors in project success.
**CO5:**POS4 Finding optimal solutions is essential in project development to ensure that applications are efficient, fast, and resource-conscious. This CO enables students to explore and implement different methods to solve real-world problems during their in-house projects.
**CO6:**POS4 Designing optimal solutions is a key aspect of developing successful in-house applications. This CO helps students apply algorithmic techniques to design software solutions that are not only functional but also optimized for performance, a critical requirement in project development.
**CO7:**POS4 Applying theoretical knowledge to practical projects is the core of PSO4. This CO strongly aligns with PSO4 as it directly involves students using their algorithmic and theoretical understanding to build functional in-house applications, turning abstract concepts into tangible results.

**Mapping of PO5 with All CO'S With Justification:**

**CO1:**POS5 Tree traversal methods and greedy algorithms are fundamental concepts in computer science, and discussing them with IT experts can provide practical insights into their applications in real-world projects. However, this CO may not be directly emphasized during IT visits.
**CO2:**POS5 Algorithm design techniques are crucial in the IT industry, and interactions with experts can deepen students' understanding of these techniques. Experts can share their experiences and applications of various design techniques, making this CO relevant during IT visits.
**CO3:**POS5 Understanding algorithm analysis is essential in software development, and IT professionals can provide valuable insights into real-world applications and how they handle algorithm efficiency. This interaction enhances students' theoretical knowledge with practical examples.
**CO4:**POS5 Time and space complexity is a critical aspect of software engineering. Engaging with IT experts can help students appreciate the importance of optimizing algorithms for performance and resource usage, making this CO highly relevant during industry visits.
**CO5:**POS5 IT experts often discuss practical approaches to solving problems, including finding optimal solutions. Learning from their experiences can help students understand the application of different methods in real-world scenarios, making this CO significant.
**CO6:**POS5 Designing optimal solutions is crucial in the IT industry, and interactions with professionals can expose students to industry-standard practices and methodologies. This CO is highly relevant as students can learn from the experiences and strategies of experts.
**CO7:**POS5 Engaging with IT experts provides students with opportunities to see how theoretical concepts are applied in the industry. This CO is strongly related to PSO5, as practical applications reinforce theoretical learning and prepare students for real-world challenges.

**Mapping of PO6 with All CO'S With Justification:**

**CO1:**POS6 Tree traversal and greedy algorithms are fundamental concepts often applied in various real-world software applications. During an internship, students may encounter projects that utilize these algorithms, helping them understand their practical applications. However, this understanding might not be the main focus of an internship.
**CO2:**POS6 Algorithm design techniques are essential in the IT industry, and internships provide students with hands-on experience in designing and implementing algorithms for real projects. This CO is highly

relevant to students' experiences in an internship setting.
**CO3:**POS6 Analyzing algorithms is a critical skill in the industry. During an internship, students will likely work on optimizing existing algorithms or analyzing new ones, directly applying this knowledge. This CO is very relevant to their practical experience.
**CO4:**POS6 Understanding time and space complexity is crucial when developing efficient software. Internships provide students with opportunities to work on projects where these concepts are applied, helping them understand their significance in a practical context.
**CO5:**POS6 Finding optimal solutions is a common task in internships, where students must evaluate multiple approaches to problem-solving. This CO is directly related to the practical experience gained during an internship.
**CO6:**POS6 Designing optimal solutions is a key part of software development. Internships often require students to apply theoretical knowledge to create effective solutions, making this CO very relevant to their practical experience.
**CO7:**POS6 This CO is highly relevant as internships are designed to bridge the gap between theory and practice. Students get the opportunity to apply what they have learned in a real-world environment, making this the most critical aspect of their internship experience.

**Mapping of PO7 with All CO'S With Justification:**

**CO1:**POS7 Knowledge of tree traversal and greedy algorithms is essential for many IT roles, especially in software development and data analysis. Understanding these concepts can make students more attractive candidates in technical interviews, making this CO relevant to employability.
**CO2:**POS7 Algorithm design techniques are fundamental skills sought after by employers in the IT industry. Mastery of these techniques not only enhances problem-solving abilities but also improves employability by aligning students' skills with industry requirements.
**CO3:**POS7 The ability to analyze algorithms is a critical skill in software development, as employers often seek candidates who can assess the efficiency and performance of their code. This CO directly supports students' employability.
**CO4:**POS7 Understanding time and space complexity is crucial for developing efficient software, a key competency in the IT industry. This knowledge enhances students' problem-solving skills and makes them more competitive in the job market.
**CO5:**POS7 Employers value candidates who can identify and implement optimal solutions to problems. This CO directly aligns with industry demands and enhances employability by equipping students with essential problem-solving skills.
**CO6:**POS7 Designing optimal solutions is a vital skill in software development, contributing to students' ability to create effective and efficient applications. This CO is highly relevant to making students employable in the IT sector.
**CO7:**POS7 The ability to apply theoretical knowledge in practical settings is essential for employability. Employers look for candidates who can translate their understanding into real-world applications, making this CO critically import.

## SYLLABUS (CBCS as per NEP 2020) FOR M. Sc. (Computer Science)

## (w. e. from June, 2023)

| | |
|---|---|
| **Name of the Programme** | **:** M.Sc. Computer Science |
| **Program Code** | **:** PSCOS |
| **Class** | **:** M.Sc. (Computer Science) |
| **Semester** | **:** I |
| **Course Type** | **:** RM |
| **Course Name** | **:** Research Methodology(Theory) |
| **Course Code** | **:** COS-521-RM |
| **No. of Lectures** | **:**60 |
| **No. of Credits** | **:**4 |

### Course Objectives:

1. Identify and discuss the role and importance of research in the social sciences.
2. Identify and discuss the issues and concepts salient to the research process.
3. Identify and discuss the complex issues inherent in selecting a research problem, selecting an appropriate research design, and implementing a research project.
4. Identify and discuss the concepts and procedures of sampling, data collection, analysis and reporting.
5. Students should be able to distinguish between the writing structure used for a quantitative study and one used for a qualitative study
6. Develop skills in qualitative and quantitative data analysis and presentation
7. Develop advanced critical thinking skills

### Course Outcomes:

**CO1:** Equip themselves with ethical issues related to Research and Publication.

**CO2:** Build a strong foundation for future research work in a systematic manner by applying notions of Research Methodology.

**CO3:** Gain ability to apply knowledge of Computer Science to research in real-world issues.

**CO4:** Get familiar with current research trends in various core areas of Computer Science.

**CO5:** Know the knowledge, general competence, and analytical skills in Research Methodology and Research & Publication Ethics.

**CO6:** Build their foundation for research in Computer Science.

**CO7:** Provide hands-on experience to carry out research work in Computer Science as well as interdisciplinary areas.

**TOPICS/CONTENTS:**

| Units | Title & Contents | No.of Lectures |
|---|---|---|
| **Unit–I** | **Foundations of Research**<br>Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process | 06 |
| **Unit–II** | **Problem Identification & Formulation**<br>Research Question – Investigation Question – Measurement Issues – Hypothesis – Qualities of a good Hypothesis –Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance | 06 |
| **Unit–III** | **Research Design: Concept and Importance in Research**<br>Features of a good research design – Exploratory Research Design – concept, types and uses, Descriptive Research Designs – concept, types and uses. Experimental Design: Concept of Independent & Dependent variables. | 06 |
| **Unit–IV** | **Qualitative and Quantitative Research**<br>Qualitative research – Quantitative research – Concept of measurement, causality, generalization, replication. Merging the two approaches. | 04 |
| **Unit -V** | **Measurement**<br>Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval,Ratio | 04 |

| | | |
|---|---|---|
| Unit VI | **Sampling**<br>Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non-Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample – Practical considerations in sampling and sample size. | 07 |
| Unit VII | **Data Analysis**<br>Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association. | 05 |
| Unit VIII | **Interpretation of Data and Paper Writing**<br>Layout of a Research Paper, Journals in Computer Science, Impact factor of Journals, When and where to publish? Ethical issues related to publishing, Plagiarism and Self-Plagiarism. | 06 |
| Unit IX | **Use of Encyclopedias, Research Guides, Handbook etc., Academic Databases for Computer Science Discipline.** | 04 |
| Unit X | **Use of tools / techniques for Research: methods to search required information effectively, Reference Management Software like Zotero/Mendeley, Software for paper formatting like LaTeX/MS Office, Software for detection of Plagiarism** | 06 |

**Book References:**

1. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches by John W. Creswell
2. Research in Education by Best and Kahn
3. Research and methodology byC.R.Kothar
4. Understanding the research problem by Paul Oliver
5. Research Methods by Rashmi Agrawal
6. An Introduction to Qualitative Research by Uwe Flick

## Mapping of this course with Programme Outcomes

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

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

## Mapping of PO1 with All CO’S With Justification:

**CO1:** PO1 Understanding ethical issues in research is moderately related to enriching knowledge in IT areas, emphasizing responsible conduct in computing.
**CO2:** PO1 Building a foundation for future research through systematic research methodology directly supports advanced understanding in computing disciplines
**CO3:** PO1 Applying Computer Science knowledge to real-world issues aligns strongly with the practical application of theoretical concepts in IT.
**CO4:** PO1 Familiarity with current research trends in core areas of Computer Science enhances students' awareness of developments in the field.
**CO5:** PO1 Knowledge and skills in Research Methodology and Ethics are crucial for a comprehensive understanding of responsible research practices in computing.
**CO6:** PO1 Building a research foundation in Computer Science directly enriches knowledge in various core computing subjects.
**CO7:** PO1 Providing hands-on experience in research supports practical applications of knowledge, essential for exploring new IT trends and methodologies.

## Mapping of PO2 with All CO’S With Justification:

**CO1:**PO2 Understanding ethical issues in research is moderately related to software applications, emphasizing responsible practices in project development.
**CO2:** PO2 Building a systematic foundation for research directly supports understanding software application development processes and methodologies.
**CO3:** PO2 Applying Computer Science knowledge to real-world issues is crucial for developing practical software solutions in projects.
**CO4:** PO2 Familiarity with current research trends helps students understand evolving software application concepts and technologies.
**CO5:** PO2 Knowledge and analytical skills in Research Methodology are vital for effectively managing software projects and ensuring quality outcomes.
**CO6:** PO2 Establishing a solid research foundation in Computer Science directly enriches the understanding of software application concepts and project management.
**CO7:** PO2 Hands-on experience in research enables students to apply theoretical knowledge in software development projects, bridging the gap between theory and practice.

## Mapping of PO3 with All CO’S With Justification:

**CO1:**PO3 Understanding ethical issues in research is moderately related to demonstrating responsible practices in programming and theoretical concepts.
**CO2:** PO3 Building a systematic foundation for research enhances the application of theoretical and programming concepts in Computer Science.
**CO3:** PO3 Computer Science knowledge to real-world issues strongly aligns with the practical application of ICT in addressing problems.
**CO4:** PO3 Familiarity with current research trends in core areas of Computer Science helps students stay updated with programming and theoretical advancements.
**CO5:** PO3Knowledge and analytical skills in Research Methodology are crucial for effective application of programming concepts in research.
**CO6:** PO3Building a research foundation in Computer Science supports understanding and demonstrating programming and theoretical concepts effectively.
**CO7:** PO3Providing hands-on experience in research allows students to apply ICT and programming skills in real-world and interdisciplinary contexts.

## Mapping of PO4 with All CO'S With Justification:

**CO1:**PO4 Understanding ethical issues in research is moderately related to developing in-house applications, as ethical considerations are important in software development.

**CO2:**PO4 Building a systematic foundation for research supports the structured development of in-house applications, aligning closely with project methodologies.

**CO3:**PO4 The ability to apply Computer Science knowledge to real-world issues is essential for creating effective in-house applications that solve practical problems.

**CO4:**PO4 Familiarity with current research trends is moderately related to developing applications, as it can inform the features and technologies used in projects.

**CO5:**PO4 Knowledge and skills in Research Methodology and Ethics are important for ensuring quality and responsibility in developing in-house applications.

**CO6:**PO4 Building a foundation for research in Computer Science directly supports the innovation and improvement of in-house application development.

**CO7:**PO4 Providing hands-on experience in research enables practical application of theoretical knowledge, which is vital for successfully developing in-house projects.

## Mapping of PO5 with All CO'S With Justification:

**CO1:**PO5 Understanding ethical issues is moderately related to interactions with IT experts, as ethical discussions are common in professional settings.

**CO2:** PO5 Building a strong foundation in research methodology is moderately relevant, as it informs discussions about best practices during IT visits.

**CO3:** PO5 The ability to apply Computer Science knowledge to real-world issues strongly supports meaningful interactions with IT professionals about practical applications.

**CO4:** PO5 Familiarity with current research trends is crucial for engaging effectively with IT experts and understanding industry advancements..

**CO5:** PO5 Knowledge of research methodology and ethics is moderately relevant for discussions on responsible practices in the IT industry.

**CO6:** PO5 Building a research foundation is moderately related as it prepares students to engage with IT experts about ongoing and future research endeavors.

**CO7:** PO5 Hands-on experience in research allows students to discuss practical applications and interdisciplinary approaches, enhancing their interactions with IT professionals.

## Mapping of PO6 with All CO'S With Justification:

**CO1:**PO6 Understanding ethical issues is moderately related to industrial exposure, as ethical considerations often arise in real-world IT practices.

**CO2:**PO6 Building a systematic foundation for research is crucial for applying structured approaches during the internship in project work.

**CO3:**PO6 The ability to apply Computer Science knowledge to real-world issues is vital during the internship, enhancing practical skills and problem-solving abilities.

**CO4:**PO6 Familiarity with current research trends is moderately relevant, as it can inform project decisions and technologies encountered during the internship.

**CO5:**PO6 Knowledge and analytical skills in Research Methodology are moderately related to ensuring quality and rigor in internship projects.

**CO6:**PO6 Building a foundation for research in Computer Science strongly supports innovative thinking and practical application during the internship.

**CO7:**PO6 Providing hands-on experience in research is directly applicable to the internship, allowing students to implement theoretical knowledge in practical settings.

# Examination Pattern / Evaluation Pattern

### Teaching and Evaluation (for Major, Minor, AEC, VEC, IKS courses)

| Course Credits | No. of Hours per Semester<br>Theory/Practical | No. of Hours per Week<br>Theory/Practical | Maximum Marks | CE<br>40 % | ESE<br>60% |
|---|---|---|---|---|---|
| 1 | 15 / 30 | 1 / 2 | 25 | 10 | 15 |
| 2 | 30 / 60 | 2 / 4 | 50 | 20 | 30 |
| 3 | 45 / 90 | 4 / 6 | 75 | 30 | 45 |
| 4 | 60 / 120 | 4 / 8 | 100 | 40 | 60 |

### Teaching and Evaluation (for VSC, SEC & CC courses)

- Evaluation to be done by Internal & External Experts.
- No descriptive end semester written examination.
- Evaluation to be done at Department level preferably prior to commencement of Theory /Practical Examinations.
- Evaluation to be done on the Skills gained by student.