I	Course Code	CS 242002
II	Course Title	Foundation of Data Science
III	Credit Structure	L: 3, T: 1, P: 0, C: 4
IV	Prerequisites:	Math - III Exposure required
V	Learning Outcomes:	This course provides a mathematical foundation for data analytics and machine learning. After completing this course, the students will be able to • understand the mathematical concepts used in the domain of data science. • apply statistical methods to carry out data analysis. • understand the role of linear algebra and optimization techniques in the design of machine learning algorithms
VI	Course Content	Sources and Types of Data, Descriptive and Inferential statistics, Data Modelling and Visualization using Python/R. Measures of central tendency, Quartiles and percentiles. Measures of dispersion. Review of probability Theory and Random Variables, Expectation and Variance. Sampling and Estimation, confidence Intervals. Applied Linear Algebra: Vector Spaces, Eigen Value and Singular Value Decommissions, Dimensionality Reduction. Applied Optimization: Convexity and Gradient Descent towards minimum. Typology of data science problems and solution frameworks. Learning from Data towards construction of neural networks.
VII	Text / Reference books:	1. Applied statistics and probability for engineers – by Douglas Montgomery 2. Linear Algebra and Learning from Data (2019)-by Gilbert Strang 3. Mastering python for data science- by Samir Madhavan **4.** https://onlinecourses.nptel.ac.in/noc21_cs69/preview **5.** https://study.iitm.ac.in/ds/course_pages/BSSE2002.html

<table>
<tr><td>I</td><td>Course Code</td><td>CS 213001</td></tr>
<tr><td>II</td><td>Course Title</td><td>Design and Analysis of Algorithms</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>1</td><td>0</td><td>4</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Data Structure, Basic Programming</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, Students shall learn the algorithm analysis techniques and different algorithm design techniques</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Introduction, Asymptotic Notations, Analysis of Insertion sort and Merge Sort, Sorting and search- insertion sort, selection sort, merge sort, quicksort, binary search, Design techniques: divide and conquer, greedy, dynamic programming, Data structures: heaps, union of disjoint sets, search trees, Algorithms on graphs: exploration, connectivity, shortest paths, directed acyclic graphs, spanning trees, Intractability: NP completeness, Reductions, Examples; Misc topics.</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>Textbook:
T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd Edition, PHI, 2012.
References book:
E. Horowitz, S. Sahni, S. Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, 2007, Universities Press.
J.Kleinberg, E. Tardos, Algorithm Design, Pearson, 2013.</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 203101</td></tr>
<tr><td>II</td><td>Course Title</td><td>Theory of Computation</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>1</td><td>0</td><td>4</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Data Structure</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>This is a basic course on the theory of computation which will introduce the concept of formal languages, automata and the fundamental concepts of Turing machines.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>o Grammars, Production systems, Chomsky Hierarchy, Right linear grammar and Finite state automata, Context free grammars, Normal forms, Subfamilies of CFL, Derivation trees and ambiguity.
o Finite state Automata, Non deterministic and deterministic FSA, NFSA with ε- moves, Regular Expressions, Equivalence of regular expression and FSA.
o Pumping lemma, closure properties and decidability, Myhill - Nerode theorem and minimization Finite automata.
o Pushdown automata, Acceptance by empty store and final state, Equivalence between pushdown automata and context-free grammars, Closure properties of CFL, Deterministic pushdown automata.
o Turing Machines, Techniques for Turing machine construction, Generalized and restricted versions equivalent to the basic model, Godel numbering, Universal Turing Machine, Recursively enumerable sets and recursive sets, Computable functions, time space complexity measures, context sensitive languages and linear bound automata.
o Decidability, Post's correspondence problem, Rice's theorem, decidability of membership, emptiness and equivalence problems of languages.
o Time and tape complexity measures of Turing machines, Random access machines, the classes P and NP, NP-Completeness, satisfiability and Cook's theorem, Polynomial reduction and some NP-complete problems.</td></tr>
<tr><td>VII</td><td>Text / Reference books:</td><td>Textbook:
1. Peter Linz; An Introduction to Formal Languages and Automata, Sixth Edition (2017) .
2. M.Sipser; Introduction to the Theory of Computation; Wadsworth Publishing Co Inc; 3rd edition 2012.

References book:
3. K.Krithivasan and R.Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education, 2009.
4. J.E.Hopcroft, R.Motwani and J.D.Ullman , "Introduction to Automata Theory Languages and computation", 3 edition 2008.</td></tr>
</table>

I	Course Code	CS 242003
II	Course Title	Computer Organization and Architecture
III	Credit Structure	L: 3, T: 0, P: 3, C: 4.5
IV	Prerequisites:	Digital Logic Design
V	Learning Outcomes:	After completing this course, the students will be able to • understand various architectural components of a microprocessor. • design and implement assembly language programs • design a basic microprocessor, understand various performance enhancements and issues related to that understand the functioning of Memory and Input/Output subsystem
VI	Course Content	Introduction and history of computers and their architecture; Architecture of 8085/x86/ARM processor. Assembly language programming: Instruction set, instruction formats and addressing modes, data representation; data transfer, arithmetic and logic operations, unconditional and conditional branching, subroutines and interrupts; Hardware support for subroutines, interrupt service, task switching; Design of a basic microprocessor: Design of micro-programmed control, micro-instruction format and micro-programming; Design of central processing unit: Arithmetic and logic unit; register and stack organization; format of instructions for data transfer and manipulation, control transfer; Memory subsystem: memory hierarchy - main memory, cache, translation look aside buffer, and virtual memory; Hardware support for segmentation and paging Performance enhancement through pipeline and vector processing: pipelined arithmetic units, Instruction level parallelism, branch prediction, multi core processing and Cache coherence, multiprocessors IO organization: Interfacing and communication, buses, interrupts; I/O protocols and devices
VII	Text / Reference books:	• Computer Architecture: A Quantitative Approach by John L. Hennessy, David A. Patterson, Morgan Kaufmann publisher • Computer Organization and Design: The Hardware/Software Interface by David A. Patterson and John L. Hennessy, Morgan Kaufmann publisher • Computer Organization and Architecture – designing for performance, William Stallings, Prentice Hall • Computer Architecture and Organization by John P. Hayes, McGraw-Hill • Computer System Architecture by M. Morris Mano, Pearson edu-cation

<table>
<tr><td>I</td><td>Course Code</td><td>CS 242001</td></tr>
<tr><td>II</td><td>Course Title</td><td>Database Management System</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>3</td><td>4.5</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>Basic understanding of design and analysis of data structures and algorithms</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>after completing this course, the students will be able to
• Understand relational data models in terms of data structure, data integrity, and data manipulation.
• Understand and create conceptual database models utilizing entity-relationship.
• Design data structures that will limit redundancy and enforce data integrity while conforming to organizational requirements utilizing normalization methodology.
• Understand the theory behind the relational data model as it applies to interactions with current database management systems.
• Interpret a given data model to query the database and transform the data into information using SQL (Structured Query Language).</td></tr>
<tr><td>VI</td><td>Course Content</td><td>Database system architecture Data Abstraction, Data Independence, Data Definition, and Data Manipulation Languages.

Data models Entity-relationship, network, relational and object-oriented data models, integrity constraints, and data manipulation operations.

Relational query languages Relational algebra, tuple, and domain relational calculus, SQL.

Relational database design Domain and data dependency, Armstrong's axioms, normal forms, dependency preservation, lossless design. Concept of relations, schema-instance distinction, keys, referential integrity, and foreign keys, relational algebra operators: selection, projection, cross product, various types of joins, division, example queries, tuple relation calculus, domain relational calculus, converting the database specification in E/R notation to the relational schema.

Query processing and optimization and SQL Introduction, data definition in SQL, table, key and foreign key definitions, and update behaviors. Querying in SQL - basic select-from-where block and its semantics, nested queries - correlated and uncorrelated, notion of aggregation, aggregation functions group by and having clauses, embedded SQL.

Dependencies and Normal forms Importance of a good schema design, problems encountered with bad schema designs, motivation for normal forms, dependency theory - functional dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers, definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.

Data storage and indexes File organizations, primary, secondary index structures, various index structures - hash-based, dynamic hashing</td></tr>
</table>

		techniques, multi-level indexes, B+ trees. **Transaction processing and Error recovery** Concepts of transaction processing, ACID properties, concurrency control, locking-based protocols for CC, error recovery and logging, undo, redo, undo-redo logging, and recovery methods. Recovery and concurrency control, locking and timestamp-based schedulers, multi version and optimistic Concurrency Control schemes. **Advanced topics** Object-oriented and object-relational databases, logical databases, web databases, distributed databases, data warehousing, and data mining. **Laboratory** Database schema design, database creation, SQL programming, and report generation using a commercial RDBMS like ORACLE/SYBASE/DB2/SQL-Server/INFORMIX. Students are to be exposed to front end development tools, ODBC and CORBA calls from application Programs, internet based access to databases and database administration.
VII	Text / Reference books:	• Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill. • Connolly, T. and C. Begg, "Database Systems: A Practical Approach to Design, Implementation, and Management," 6th edition, Pearson, 2014 • Coronel, C. and S. Morris, "Database Systems: Design, Implementation, & Management," 12th edition, Cengage, 2016 • Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.