CURRICULUM AND SYLLABI

for

Minor Programme

(Applicable to 2022 admission onwards)

http://www.nitgoa.ac.in

राष्ट्रीय प्रौधोगिकी संस्थान गोवा

NATIONAL INSTITUTE OF TECHNOLOGY GOA

कुंकोलिम, जिला दक्षिण गोवा, गोवा, पिन – ४०३७०३, इंडिया

Cuncolim, South Goa District, Goa, Pin – 403703, India

Minor Specialization

in

Computer Science and Engineering

Offered by the

Department of Computer Science and Engineering

Semester Offered	Course Code	Course Name	Type	L-T-P	Credits
IV	CS250M	Fundamentals of Data Structures	MR	3-0-2	4
V	CS300M	Design and Analysis of Algorithms	MR	3-0-0	3
VI	CS350M	Database Management System	MR	3-0-2	4
VII	CS400M	Data Science Fundamentals with Python	MR	2-0-2	3
VIII	CS450M	Artificial Intelligence and Machine Learning	MR	3-1-0	4
Total Credits					**18**

Detailed Syllabi of courses

Course Code	Course Name	L	T	P	Credits
CS250M	Fundamentals of Data Structures	3	0	2	4

Course Objective

The objective of the course is to introduce the basic concepts of data structures and to develop skills to apply appropriate data structures for designing algorithms to solve problems. The objective of the course is also to illustrate the implementation of basic data structures and to develop programming skills to apply appropriate data structures for problem solving

Course Outcomes

At the completion of this course, the student shall acquire knowledge and ability

CO1. Select an appropriate data structure for a particular problem.

CO2. Implement linear and non-linear data structures.

CO3. Implement sorting and searching algorithms using relevant data structures.

CO4. Write programs that makes suitable use of queues, stacks, linked lists, trees, and graphs.

Relationship of Course Outcomes to Program Outcomes

H = High correlation; M = Medium correlation; L = Low correlation

POs → / COs ↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	H	H	H	H	M							
CO2	H	H	H	H	H							
CO3	H	H	H	H	H							
CO4	H	H	H	H	H							

Syllabus

Module 1: Introduction to data structures, Arrays: one dimensional, multi-dimensional, Structures, Union, Recursion, Searching and Sorting Algorithms.

Module 2: Queues: Simple Queue, Circular Queue, Elementary Operations, Applications of Queue. Stacks: Elementary operations, Applications such as infix to postfix expression conversion, postfix expression evaluation, parenthesis matching.

Module 3: Linked lists: Linear, circular and doubly linked lists, Implementation of stack and queue using linked list.

Module 4: Trees: Basic terminologies, Binary tree, Binary search tree, Balanced trees. Graphs: Basic terminologies, Representation of graphs, Search Algorithms, Shortest path algorithms, Minimum spanning tree.

List of Experiments

1. Implementation of array operations, Structures & Unions
2. Stacks, Queues, Circular Queues, Priority Queues, Multiple stacks, and queues
3. Infix to postfix expression using stack.
4. Implementation of linked lists: stacks, queues
5. Implementation of doubly linked lists
6. Tree traversals
7. Searching and sorting

Reference Books/Material

1. Alfred V Aho, John E Hopcroft, Jeffrey D. Ullman, “Data structures & algorithms”, Pearson, 2013.
2. Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein, “Data Structures using C”, Third Edition, Pearson, 2009.

3. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Second Edition, Pearson, 2006.

Course Code	Course Name	L	T	P	Credits
CS300M	Design and Analysis of Algorithms	3	0	0	3

Course Objective

The aim of this course is to provide engineers and scientists with a strong foundation in Design and Analysis of Algorithms. The first part of this course is intended to make students familiar with Asymptotic Notations to analyze algorithms and Divide and Conquer. The second part of this course will provide a detailed introduction to Dynamic Programming. The third part of this course is looking at designing algorithms using Greedy Method. The last part of this course will deal with computationally hard problems and tackling them using approximation algorithms.

Course Outcomes

At the completion of this course, the student shall acquire knowledge and ability to

CO1. Analyze time and space complexities of algorithms

CO2. Identify algorithm design methodology to solve problems.

CO3. Distinguish between P and NP classes of problems.

CO4. Design and analyze approximation algorithms for NP-hard problems

Relationship of Course Outcomes to Program Outcomes

H = High correlation; M = Medium correlation; L = Low correlation

POs → COs ↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	H	H	H	M	H	M		L	L	L	L	H
CO2	H	H	H	M	H	M		L	L	L	L	H
CO3	H	H	H	M	H	M		L	L	L	L	H

CO4	H	H	H	H	H	M		L	L	L	M	H

Syllabus

Module 1: Introduction to Algorithm Analysis, Asymptotic Notations, Divide and Conquer – Master Theorem, Maximum Element in an Unimodal Array, Maximum Subarray Sum Problem, Expected Running Time of Randomized Quick Sort, Strassen's Matrix Multiplication Algorithm, Karatsuba's Large Integer Multiplication, Selection in Worst Case Linear Time

Module 2: Dynamic Programming - Matrix Chain Multiplication Problem, Optimal Binary Search Tree, Rod-Cutting Problem, 0-1 Knapsack Problem, Travelling Salesman Problem, All-Pairs Shortest Paths Problem, Optimal Vertex Cover of a Tree.

Module 3: Greedy Method - Activity Selection Problem, Fractional Knapsack Problem, Correctness and Running Time Analysis of Prim's and Kruskal's Algorithms for Finding Minimum Spanning Tree, Backtracing, Branch and Bound.

Module 4: Complexity Classes - P, NP, NP-hard, NP-complete, Example NP-complete Problems – Clique, Independent Set, Vertex Cover, Approximation Algorithms - Vertex Cover Problem

Reference Books/Material

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI, 2009.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.
3. Michael R. Garey and David S. Johnson, "Computers and Intractability: A Guide the theory of NP-Incompleteness", W.H. Freeman & Co., 1979.
4. Herbert S. Wilf, "Algorithms and Complexity", AK Peters Ltd., 2003.
5. https://www.algorist.com/

Course Code	Course Name	L	T	P	Credits
CS350M	Database Management Systems	3	0	2	4

Course Objective

This course covers the relational database systems RDBS - the predominant system for business, scientific and engineering applications at present. The course includes relational model, entity-relation model, relational algebra, normalization and data access queries as well as an introduction to SQL.

Course Outcomes

At the completion of this course, the student shall acquire knowledge and ability to

CO1. Model Entity-Relationship diagrams for enterprise level databases

CO2. Formulate Queries using SQL and Relational Formal Query Languages

CO3. Apply different normal forms to design the Database and formal design techniques to produce a database schema

CO4. Summarize concurrency control protocols and recovery algorithms

CO5. Hands on with SQL and PL/SQL

Relationship of Course Outcomes to Program Outcomes

H = High correlation; M = Medium correlation; L = Low correlation

POs → / COs ↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	H	H	H	H	M	M	L	L	H	M	M	M
CO2	H	H	H	H	H	M	L	L	H	M	H	M
CO3	H	H	H	M	M	M	L	L	M	M	M	M
CO4	H	H	H	M	M	M	L	L	M	M	M	M
CO5	H	H	M	M	H	M	L	L	M	L	M	M

Syllabus

Module 1 : Introduction: An overview of database management system, database system vs file system, database system concept and architecture, data model schema and instances, data independence and database language and interfaces, (DDL,DML,DCL), overall database structure, database users. Data modelling using the Entity Relationship model: ER model concepts, notation for ER diagram, mapping constraints, keys, specialization, generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationship of higher degree.

Module 2 : Relational data Model and Language: Relational data model concepts, integrity constraints, entity integrity, referential integrity, key constraints, domain constraints, relational algebra, Introduction on SQL: Characteristics of SQL, SQL data type and literals, types of SQL commands, SQL operators and their procedure, tables, views, queries and sub queries, aggregate functions, insert, update and delete operations, joins, unions, intersection, minus, cursors, triggers, procedures in SQL/PL SQL.

Module 3 : Data Base Design & Normalization: Functional dependencies, primary key, foreign key, candidate key, super key, normal forms, first, second, third normal forms, BCNF, 4th Normal form,5th normal form, loss less join decompositions, canonical cover, redundant cover, MVD, and JDs, inclusion dependence, transaction processing concept, transaction system, testing of serializability, serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, deadlock handling.

Module 4 : Concurrency Control Techniques: Concurrency control, locking techniques for concurrency control, 2PL, time stamping protocols for concurrency control, validation based protocol, multiple granularity, multi version schemes and recovery with concurrent transaction. Storage: Introduction, secondary storage devices, tertiary storage, buffering of blocks, structure of files, file organization, indexing and hashing, types of single level ordered indexes, multilevel indexes, dynamics multilevel indexes using B-trees and B+-Trees, database security.

List of Experiments:

1. Defining schemas for applications.
2. Creating tables, Renaming tables, Data constraints (Primary key, Foreign key, Not Null), Data insertion into a table.
3. Grouping data, aggregate functions, Oracle functions (mathematical, character functions).
4. Sub-queries, Set operations, Joins.
5. Creation of databases, writing SQL and PL/SQL queries to retrieve information from the databases.
6. Procedures, Functions, Cursors, Triggers, Packages, Views and Assertions.

Reference Books/Material

1. Korth, Silberschatz, "Database System Concepts", 4th ed., TMH, 2003.
2. Elmsari and Navathe, "Fundamentals of Database Systems", 4th ed., A. Wesley, 2004
3. Raghu Ramakrishnan , Johannes Gehrke, " Database Management Systems", 3rd Edition, McGraw- Hill, 2003.
4. J D Ullman, "Principles of database systems", Computer Science Press, 2001.

Course Code	Course Name	L	T	P	Credits
CS400M	Data Science Fundamentals with Python	2	0	2	3

Course Objective

Introduce the fundamental concepts of data science. The course will introduce mathematical theory of probability to handle uncertainty that is inherent in the data that needs to be handled today. The probability theory part of the course provides the mathematical basis and, in the statistics part, one learns how to use probability to make sense of data, which is important in a world which is largely data driven currently. The important aspect of the course is the illustration of theoretical concepts using Python.

Course Outcomes

At the completion of this course, the student shall acquire knowledge and ability to

CO1. Understand the intricacies of Python programming to write basic programs
CO2. Understand how uncertainty in data is handled using probability theory
CO3. Using statistical principles to understand the data
CO4. Apply the learnt concepts to any real-world problems from engineering or science domain

Relationship of Course Outcomes to Program Outcomes
H = High correlation; M = Medium correlation; L = Low correlation

POs → / COs ↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L	M	M	L	H							
CO2	H	H	M	M	M							
CO3	H	H	M	M	H							
CO4	H	H	H	H	H							

Syllabus

Module 1:

Introduction to Python programming: Data types, scalar, composite types, arrays/lists, expressions, functions, decisions and iterations, lists, dictionaries, NumPy arrays, strings, File I/O, Data visualization: Plotting functions, making use of color.

Probability: Why probability and what is it? (give real life situations which demands use of probability). Counting, combinations, permutations, binomial and multinomial coefficients, Stirling's formula. Discrete probability spaces (with examples). Axiomatic definition of probability, inclusion-exclusion formula, independence, condition probability, Bayes' rule. Discussion of these concepts in the context of data science

Illustration of counting, basic probability -simulation of simple experiments and conditional probability through Python programming

Module 2:
Random variables: definition, distribution function and its properties, probability mass function (binomial, Bernoulli, Poisson, geometric), probability density function (uniform, exponential, Gaussian). Joint distributions, independence and conditioning of random variables. Function of random variables, change of variable formula. Discussion of these concepts in the context of data science

Illustration of Generating random variables following a given pdf/pmf and engineering application of functions of random variables through Python programming.

Module 3:
Measures of central tendency, dispersion and association – expectation, median, variance, standard deviation, mean absolute deviation, covariance, correlation and entropy (definition and guidelines on how to choose a particular measure). Discussion of these concepts in the context of data science

Illustration of Scatter plot (for independent, correlated, uncorrelated random variables), through Python programming.

Module 4:
Statistics: Using probability to understand data (give real life examples). Frequentist approach - point and range estimates, confidence intervals, hypothesis testing p-values, significance level, power and t-test. Bayesian inference – maximum likelihood estimation. Discussion of these concepts in the context of data science

Illustration of Parameter estimation, hypothesis testing, regression through Python programming.

Module 5:
Matrices and vectors and operations on them, Norms of matrix and vectors, Eigen analysis and singular value decomposition, liner regression-least squares method. Discussion of these concepts in the context of data science
Illustration of the concepts through Python programming, application of Eigen analysis through principal component analysis.

A case study of a large dataset from any Engineering / Science domain

Reference Books/Material

1. Sheldon Ross, Introduction to Probability and Statistics for Engineers, 5/e (2014), Elsevier
2. Python programming for the Absolute Beginners, Third Edition, Michael Dawson, Cengage Learning, Inc.
3. Morris H. DeGroot and Mark J. Schervish, Probability and Statistics (4/e)(2012), Addison-Wesley.
4. Blitzstein and Hwang, Introduction to Probability (2015), CRC Press.
5. Gilbert Strang, Linear Algebra and Its Applications, Fourth Edition

Course Code	Course Name	L	T	P	Credits
CS450M	Artificial Intelligence and Machine Learning	3	1	0	4

Course Objective
The course introduces the variety of concepts in the field of artificial intelligence and Machine Learning. It discusses the philosophy of AI, and how to model a new problem as an AI problem. Key concepts on Problem-solving by search, Constraint Satisfaction Problems and Logic are discussed in AI, In addition to that algorithms in Machine learning and also concepts like deep

learning are also introduced. By the end of this course, students will develop a deep understanding of AI and ML.

Course Outcomes

At the completion of this course, the student shall acquire knowledge and ability to

CO1. Solve searching problems using A*, Mini-Max algorithms.

CO2. Formulate valid solutions for problems involving uncertain inputs or outcomes by using decision making techniques

CO3. Understand Bayesian Networks to do probabilistic reasoning

CO4. Understand Reinforcement Learning and Deep Learning

Relationship of Course Outcomes to Program Outcomes

H = High correlation; M = Medium correlation; L = Low correlation

POs → / COs ↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	H	H	H	H	M	L		L	L	M	M	H
CO2	H	H	H	H	M	L		L	L	M	M	H
CO3	H	H	H	H	M	L		L	L	M	M	H
CO4	H	H	H	H	M	L		L	L	M	M	H

Syllabus

Module 1: What Is AI? The inception of artificial intelligence. Agents and Environments, The Nature of Environments, The Structure of Agents: Problem-solving by search: Problem-Solving Agents, Search Algorithms: Uninformed Search Strategies : Best-first search, Dijkstra's algorithm or uniform-cost search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Search contours, Satisficing search: Inadmissible heuristics and weighted A*, Bidirectional heuristic search, Adversarial search: Minimax Algorithm for two player games, An Example of Minimax Search, Alpha Beta Pruning and its analysis.

Module 2: Constraint Satisfaction Problems: Representation of the atomic state, Map coloring and other examples of CSP, Backtracking Search, Variable and Value Ordering in Backtracking Search, Inference for detecting failures early, Exploiting problem structure

Module 3: Logic in AI: Different Knowledge Representation systems, syntax, semantics, Forward Chaining, Resolution, Reduction to Satisfiability Problems, SAT Solvers : DPLL Algorithm, WalkSAT Algorithm, Decision Theory: Markov Decision Processes:

Module 4: Learning: Basics of Probability, Conditional Independence & Bayes Rule, Bayesian Networks, Bayesian Learning, Structure Learning and Expectation Maximization, Reinforcement Learning: Model-based Learning for policy evaluation (Passive Learning), TD Learning and Computational Neuroscience, Q Learning, Deep Learning: Perceptrons and Activation function, Backpropagation, Thin Deep Vs Fat Shallow Networks, Convolutional Neural Networks

Reference Books/Material

1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, Third Edition (2009)
2. Ian GoodFellow, Yoshua Bengio & Aaron Courville, Deep Learning, MIT Press (2016).