Department of Computer Science

University of Delhi

Curriculum

Bachelor in Computer Science (Honors)

under UGCF 2022

Table of Contents

Preamble

The new curriculum of the four year undergraduate program under NEP, called as UGCF, for computer science aims to develop the core competence in computing and problem solving amongst its graduates. Informally, “Learning to learn” has been the motto of the department since its inception. The curriculum thus focuses on building theoretical foundations in computer science to enable its pupils to think critically when challenged with totally different and new problems. It imbibes the following **Student-Centric** features of NEP 2020:

Flexibility to Exit: In order to support early exits, the curriculum aims to develop employability skills early. This has been done so that the outcomes of the 4 yr degree is not compromised as we believe that all but a few students will go for the full 4 year degree. As programming is at the heart of computing it is proposed to have two programming courses early so that the students can develop good programming skills in the first year. At the same time students are familiarized with the hardware of computers early on.

Employability: Industry demand in the IT sector has changed considerably in the past few years. With the humongous amount of data coming from all the domains like medical data, social networking data, astronomical data, education, etc., automating information extraction and analysis of data is the only way forward to leverage the available data for the future. The curriculum aims to equip the students with tools and techniques of Artificial Intelligence, Machine Learning and a pathway on Data Science if the student so desires.

Having said this, there is no replacement for the foundational courses like programming, data structures and algorithms. With two courses on programming and three courses on data structures and algorithms together, a strong foundation will be laid down for problem solving.

Flexibility to Choose: The curriculum provides multiple pathways (discipline specific electives) for the students to choose their micro-specialization, if any. This includes tracks on Data-Science, Machine Learning and its application, Security, Networks, Theory and Internet Technologies. Each pathway provides learning progression in the chosen area of specialization. The project based pathway on “problem solving with computers” will inculcate the high order thinking skills in the students. Students will be expected to model real life problems mathematically and use their programming skills to solve them.

Multidisciplinarity: The curriculum provides two pathways (general elective), one of computer science (CS) and the other of information technology (IT), to the students from other disciplines. Those who want to earn a minor in CS will be required to choose the first pathway whereas those who simply want to apply IT in the domain of their interest can choose the second pathway.

Research: With the option to obtain specialization in an area of their choice, the curriculum prepares the students to take up research projects in their final year. Strong mathematical background built up with the help of three papers on “Mathematics for Computing” prepares the students to choose the path of higher studies.

Advance Courses: Students with high order thinking skills can do advanced courses early on if they fulfill the requirement of the pre-requisites. For example, students can do DCS01, DSC02, DSC03, DSC07 in Semester I, DCS04, DSC05, DSC06, DSC10 in Semester II, DCS08, DSC09, DSC13, DSC14 in Semester III, DCS11, DSC12, DSC16, DSC17 in Semester IV, DCS15, DSC19 and two electives (of Semester VII) in Semester V, DCS18, DSC20 and two electives (of Semester VIII) in Semester V. This will enable students to focus more on their internships/research projects in the fourth and the last year of the program.

Discipline Specific Graduate Attributes (DSGA)

1. Proficiency in writing readable, correct, efficient, and secure programs of modest complexity.
2. Ability to design efficient algorithms using appropriate data structures for new problems.
3. Understanding of computer architecture, operating systems, computer networks and database management systems and their role in the performance of software applications.
4. Understanding of theoretical foundations and limits of computing.
5. Ability to develop good quality software by following the processes of software development life cycle.
6. Ability to extract information and analyze large volumes of data employing a range of techniques for artificial intelligence and learning.
7. Ability to design parallel algorithms to exploit the strength of multiple computing units in a computer.
8. Ability to develop an end to end compiler using compiler designing tools and techniques.
9. Ability to protect the data and software from various types of cyber attacks.

Table of Core Courses

Semester	DSC - No.	Title	L	T*	P*	Total credits	Pre-requisites	DSGA Contributed to
I	DSC 01	Programming using Python	3	0	1	4	Nil	DSGA1
I	DSC 02	Computer System Architecture	3	0	1	4	Nil	DSGA3
I	DSC 03	Mathematics for computing	3	0	1	4	Nil	DSGA4
II	DSC 04	Object Oriented Programming with C++	3	0	1	4	Nil	DSGA1
II	DSC 05	Discrete Mathematical Structures	3	0	1	4	Nil	DSGA4
II	DSC 06	Probability for Computing	3	0	1	4	Nil	DSGA4
III	DSC 07	Data Structures	3	0	1	4	DSC 04/ a course in C/C++	DSGA1, DSGA2

							at plus 2 level/**	
III	**DSC 08**	**Operating Systems**	**3**	**0**	**1**	**4**	DSC 04/a course in C/C++ at plus 2 level/**	DSGA3
III	**DSC 09**	**Numerical Optimization**	**3**	**0**	**1**	**4**	Nil	DSGA4
IV	**DSC10**	**Design and Analysis of Algorithms**	**3**	**0**	**1**	**4**	DSC07	DSGA1, DSGA2
IV	**DSC 11**	**Database Management System**	**3**	**0**	**1**	**4**	DSC01/ a course in Python at plus 2 level/** , DSC08	DSGA3
IV	**DSC 12**	**Computer Networks**	**3**	**0**	**1**	**4**	DSC04/a course in C/C++ at plus 2 level/** , DSC 07, DSC 08	DSGA3
V	**DSC 13**	**Algorithms and Advanced Data Structures**	**3**	**0**	**1**	**4**	DSC07, DSC10	DSGA1, DSGA2.
V	**DSC 14**	**Theory of**	**3**	**0**	**1**	**4**	DSC04/a	DSGA4

		Computation					course in C/C++ at plus 2 level/** DSC05	
V	**DSC 15**	**Software Engineering**	**3**	**0**	**1**	**4**	DSC01,/ DSC04,/ a course in C/C++/Python at plus 2 level/**	DSGA5
VI	**DSC 16**	**Artificial Intelligence**	**3**	**0**	**1**	**4**	DSC 01 DSC03 DSC 06 DSC 07 DSC 09 DSC10 DSC14	DSGA6
VI	**DSC 17**	**Machine Learning**	**3**	**0**	**1**	**4**	DSC 01 DSC03 DSC 06 DSC 07 DSC 09 DSC10	DSGA6
VI	**DSC 18**	**Introduction to Parallel Programming**	**3**	**0**	**1**	**4**	DCS 02, DSC04, DSC 07, DSC 08	DSGA7
VII	**DSC 19**	**Compiler Design**	**3**	**0**	**1**	**4**	DSC-14:	DSGA8
VIII	**DSC 20**	**Information Security**	**3**	**0**	**1**	**4**	DSC 01 DSC04, DSC 07 DSC 08 DSC 11	DSGA1, DSGA9

							DSC 12	

Note

1. **Batch size for Practicals will be (8-10) and Tutorials will be (12-15).**
2. **Wherever DCS04 is a prerequisite, a course in C/C++ at plus 2 level will be acceptable.**
3. **Wherever DCS01 is a prerequisite, a course in Python at plus 2 level will be acceptable.**

Syllabi of Core Courses

This section gives the detailed syllabus of the core courses. Each course describes the course objective, learning outcomes, the units and the reading material. The reading material has 2 -3 components: main resource(/s), additional text material, and online resources. Main resources are kept to a minimum possible and no more than 3. Additional resources and the online material may be used to enhance the knowledge of the subject.

DSC 01: Programming using Python

Tentative weekly teaching plan is as follows:

Week 1-2	**Unit 1 Introduction to Programming:** Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.
Week 3-6	**Unit 2 Creating Python Programs:** Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.
Week 7-11	**Unit 3 Built-in data structures:** Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations;

	Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.
Week 12-13	**Unit 4 Object Oriented Programming:** Introduction to classes, objects and methods; Standard libraries.
Week 14-15	**Unit 5 File and exception handling:** File handling through libraries; Errors and exception handling.

DSC 02: Computer System Architecture

Tentative weekly teaching plan is as follows:

Week 1-2	**Unit 1 Digital Logic Circuits:** Logic Gates, Truth Tables, Boolean Algebra, Digital Circuits, Combinational Circuits, Introduction to Sequential Circuits, Circuit Simplification using Karnaugh Map, Don't Care Conditions, Flip-Flops, Characteristic Tables, Excitation Table.
Week 3-5	**Unit 2 Digital Components (Fundamental building blocks):** Designing of combinational circuits- Half Adder, Full Adder, Decoders, Encoders, Multiplexers, Registers and Memory (RAM , ROM and their types) , Arithmetic Microoperations, Binary Adder, Binary Adder-Subtractor.
Week 6-7	**Unit 3 Data Representation and Basic Computer Arithmetic**: Number System, r and (r-1)'s Complements, data representation and arithmetic operations.
Week 8-10	**Unit 4 Basic Computer Organization and Design**: Bus organization, Microprogrammed vs Hardwired Control , Instruction Codes, Instruction Format, Instruction Cycle, Instruction pipelining, Memory Reference, Register Reference and Input Output Instructions, Program Interrupt and Interrupt Cycle.
Week 11-12	**Unit 5 Processors**: General register organization, Stack Organization, Addressing Modes, Overview of Reduced Instruction Set Computer (RISC) , Complex Instruction Set Computer (CISC), Multicore processor and Graphics Processing Unit (GPU)
Week 13-15	**Unit 6 Memory and Input-Output Organization**: Memory hierarchy (main, cache and auxiliary memory), Input-Output Interface, Modes of Transfer: Programmed I/O, Interrupt initiated I/O, Direct memory access.

S.No.	Unit Name	Chapter Number and Name	Section Numbers	Reference	No. of Lectures*	Week No.	Chapter wise Weightage* (Marks)
1	**Unit 1:** Digital Logic Circuits	Ch 1: Digital Logic Circuits	1.1, 1.2, 1.3, 1.4, 1.5, 1.6 1.7 (up to pg. 28)	[1]	6	1-2	14
2	**Unit 2:** Digital Components (Fundamental building blocks):	Ch 2: Digital Components	2.2, 2.3, 2.7		5	3-5	6
		Ch 4: Register Transfer and Micro-operations	4.4 (up to fig. 4.7)		4		5
3	**Unit 3:** Data Representation and Basic Computer Arithmetic	Ch 3: Data Representation	3.1, 3.2, 3.3		6	6-7	8
4	**Unit 4:** Basic Computer	Ch 5: Basic Computer Organizatio	5.1, 5.2, 5.3,		10		15

	Organization and Design	n and Design	5.4 (up to pg. 137), 5.5, 5.6, 5.7			8-11	
		Ch 9: Pipeline and Vector Processing	9.2		2		4
5	**Unit 5:** Processors:	Ch 8: Central Processing Unit	8.1, 8.2, 8.3 (up to pg. 247), 8.5, 8.8 (only characteristics, i.e., pg. 282 – 284)		4	12-13	10
		Ch-7: Multicores, Multiprocessors, and Clusters	7.1 (page 632- Introduction of Multicore Processor) 7.7 (page 654-656) Characteristics of GPU Vs. CPU	[2]	1		3
6	**Unit 6:** Memory and Input-Output Organization	Ch 11: Input Output Organization	11.2 (up to pg-388), 11.4, 11.6 (up to pg-416)	[1]	6	14-15	8
		Ch 12: Memory	12.1 (up to pg-446)		1		2

		Organizatio n						

*The number of lectures and chapter wise weightage may be treated as indicative only.

References:

[1] Computer System Architecture: Morris M. Mano (Pearson Education, 3rd Edition)

[2] Patterson and John L. Hennessy. "Computer Organization and Design: The Hardware/Software interface", 5th edition, Elsevier, 2012.

DSC 03: Mathematics for computing

Course Objective

This course introduces the students to the fundamental concepts and topics of linear algebra and vector calculus, whose knowledge is important in other computer science courses. The course aims to build the foundation for some of the core courses in later semesters.

Course Learning Outcomes

After successful completion of this course, the student will be able to:

1. Perform operations on matrices and sparse matrices
2. Compute the determinant, rank and eigenvalues of a matrix
3. Perform diagonalization
4. Perform operations on vectors, the dot product and cross product
5. Represent vectors geometrically and calculate the gradient, divergence, curl
6. Apply linear algebra and vector calculus to solve problems in sub-disciplines of computer science.

Syllabus

Unit 1 Introduction to Matrix Algebra: Echelon form of a Matrix, Rank of a Matrix, Determinant and Inverse of a matrix, Solution of System of Homogeneous & Non-Homogeneous Equations: Gauss elimination and Solution of System of Homogeneous Equations: Gauss Jordan Method.

Unit 2 Vector Space and Linear Transformation: Vector Space, Sub-spaces, Linear Combinations, Linear Span, Convex Sets, Linear Independence/Dependence, Basis & Dimension, Linear transformation on finite dimensional vector spaces, Inner Product Space, Schwarz Inequality, Orthonormal Basis, Gram-Schmidt Orthogonalization Process.

Unit 3 EigenValue and EigenVector: Characteristic Polynomial, Cayley Hamilton Theorem, Eigen Value and Eigen Vector of a matrix, Eigenspaces, Diagonalization, Positive Definite Matrices, Applications to Markov Matrices

Unit 4 Vector Calculus: Vector Algebra,Laws of Vector Algebra, Dot Product, Cross Product,Vector and Scalar Fields, Ordinary Derivative of Vectors, Space Curves, Partial Derivatives, Del Operator, Gradient of a Scalar Field, Directional Derivative, Gradient of Matrices, Divergence of a Vector Field, Laplacian Operator, Curl of a Vector Field.

References

1. Gilbert Strang, *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2021.
2. Erwin Kreyszig, *Advanced Engineering Mathematics*, 10th Edition, Wiley, 2015.
3. Gilbert Strang, *Linear Algebra and Learning from Data*, 1st Edition, Wellesley-Cambridge Press, 2019.
4. R. K. Jain, S. R. K. Iyengar, *Advanced Engineering Mathematics*, 5th Edition, Narosa, 2016.

Additional References

(i) Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong, *Mathematics for Machine Learning*, 1st Edition, Cambridge University Press, 2020.

(ii) Seymour Lipschutz and Marc Lipson, *Schaum's Outline of Linear Algebra*, 6th Edition, McGraw Hill, 2017.

Weekly Plan

Week	Topic
Week 1 – 2	Introduction to Matrix Algebra: Echelon form of a Matrix, Rank of a Matrix, Determinant and Inverse of a matrix, Solution of System of Homogeneous & Non-Homogeneous Equations: Gauss elimination and Gauss Jordan Method.
Week 3 – 5	Vector Space and Linear Transformation: Vector Space, Sub-spaces, Linear Combinations, Linear Span, Convex Sets, Linear Independence/ Dependence, Basis & Dimension.
Week 6 – 8	Linear transformation on finite dimensional vector spaces, Inner Product Space, Schwarz Inequality, Orthonormal Basis, Gram-Schmidt Orthogonalization Process.
Week 9 – 11	Eigenvalue and Eigenvector: Characteristic Polynomial, Cayley Hamilton Theorem, Eigen Value and Eigen Vector of a matrix, Eigenspaces, Diagonalization, Positive Definite Matrices, Applications to Markov Matrices
Week 12 – 15	Vector Calculus: Vector Algebra, Laws of Vector Algebra, Dot Product, Cross Product, Vector and Scalar Fields, Ordinary Derivative of Vectors, Space Curves, Partial Derivatives, Del Operator, Gradient of a Scalar Field, Directional Derivative, Gradient of Matrices, Divergence of a Vector Field, Laplacian Operator, Curl of a Vector Field.

Suggested Practical List

1. Create and transform vectors and matrices (the transpose vector (matrix) conjugate transpose of

a vector (matrix))

2. Generate the matrix into echelon form and find its rank.

3. Find cofactors, determinant, adjoint and inverse of a matrix.

4. Solve a system of Homogeneous and non-homogeneous equations using Gauss elimination method.

5. Solve a system of Homogeneous equations using the Gauss Jordan method.

6. Generate basis of column space, null space, row space and left null space of a matrix space.

7.Check the linear dependence of vectors. Generate a linear combination of given vectors of Rn/ matrices of the same size and find the transition matrix of given matrix space.

8.Find the orthonormal basis of a given vector space using the Gram-Schmidt orthogonalization process.

9.Check the diagonalizable property of matrices and find the corresponding eigenvalue and

verify the Cayley- Hamilton theorem.

10.Application of Linear algebra: Coding and decoding of messages using nonsingular matrices.

eg code "Linear Algebra is fun" and then decode it.

11.Compute Gradient of a scalar field.

12.Compute Divergence of a vector field.

13.Compute Curl of a vector field.

DSC 04: Object Oriented Programming with C++

References

1. Prata, S. *C++ Primer Plus*, 6th edition, Pearson India, 2015
2. Balaguruswamy, E. *Object Oriented Programming with C++*, 8th edition, McGraw-Hill Education, 2020
3. Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*, 6th edition, Cengage Learning, 2013

Additional References

(i) Schildt, H. *C++: The Complete Reference*, 4th edition, McGraw Hill, 2003
(ii) Forouzan, A. B., Gilberg, R. F. *Computer Science: A Structured Approach using C++*, 2nd edition, Cengage Learning, 2010

Weekly Plan

Week	**Topic**
Week 1	**Unit 1 Introduction to C++:** Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++
Week 2 – 5	**Unit 2 Programming Fundamentals:** Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision making constructs, Iteration, Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters
Week 6 – 10	**Unit 3 Object Oriented Programming:** Concepts of Abstraction, Encapsulation. Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Template functions and classes
Week 11 – 13	**Unit 4 Pointers and References:** Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references
Week 14 – 15	**Unit 5 Exception and File Handling:** Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations

DSC 05: Discrete Mathematical Structures

Week 1-2	**Unit 1 Sets, Functions, Sequences and Summations, Relations:** Sets: Set Operations, Computer Representation of Sets, Countable and Uncountable Set, Principle of Inclusion and Exclusion, Multisets; Functions: One-to-one and Onto Functions, Inverse Functions and Compositions of Functions, Graphs of Functions Sequences and Summations: Sequences, Special Integer Sequences, Summations; Relations: Properties of Binary Relations, Equivalence relations and Partitions, Partial Ordering Relations and Lattices.
Week 3-5	**Unit 2 Logic and Proofs:** Propositional Logic, Propositional Equivalences, Use of first-order logic to express natural language predicates, Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategies, Mathematical Induction.
Week 6-8	**Unit 3 Number Theory:** Division and Integers, Primes and Greatest Common Divisors, Representation of Integers, Algorithms for Integer Operations, Modular Exponentiation, Applications of Number Theory.
Week 9-10	**Unit 4 Combinatorics/Counting:** The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations.
Week 11-13	**Unit 5 Graphs and Trees:** Graphs: Basic Terminology, Multigraphs and Weighted Graphs, Paths and Circuits, Eulerian Paths and Circuits, Hamiltonian paths and Circuits, Shortest Paths, Spanning Trees, Graph Isomorphism, Planar Graphs; Trees: Trees, Rooted Trees, Path Lengths in Rooted Trees.
Week 14-15	**Unit 6 Recurrence:** Recurrence Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their solution.

DSC 06: Probability for Computing

Course Objective

This course introduces the students to the fundamental concepts and topics of probability and statistics, whose knowledge is important in other computer science courses. The course aims to build the foundation for some of the core courses in later semesters.

Course Learning Outcomes

After successful completion of this course, the student will be able to:

1. Use probability theory to evaluate the probability of real-world events.
2. Describe discrete and continuous probability distribution functions and generate random numbers from the given distributions.
3. Find the distance between two probability distributions
4. Define and quantify the information contained in the data.
5. Perform data analysis in a probabilistic framework.
6. Visualize and model the given problem using mathematical concepts covered in the course.

Week	Topic
Week 1 – 3	Basic Probability: Introduction to the notion of probability, Random experiment, Sample space and Events, Probability defined on events, Algebra of events. Conditional probabilities, independent events, Bayes' theorem.
Week 4 – 7	Random Variables: Introduction to Random Variables, Probability mass/density functions, Cumulative distribution functions. Discrete Random Variables (Bernoulli, Binomial, Poisson, Multinomial and Geometric). Continuous Random Variables (Uniform, Exponential and Normal). Expectation of a Random Variable, Expectation of Function of a Random Variable and Variance. Markov inequality, Chebyshev's inequality, Central Limit Theorem, Weak and Strong Laws of Large Numbers.
Week 8 – 10	Joint Distributions: Jointly distributed Random Variables, Joint distribution functions, Independent Random Variables, Covariance of Random Variables, Correlation Coefficients, Conditional Expectation.

Week 11 – 15	Markov Chain and Information Theory: Introduction to Stochastic Processes, Chapman–Kolmogorov equations, Classification of states, Limiting and Stationary Probabilities. Random Number Generation, Pseudo Random Numbers, Inverse Transformation Method, Rejection Method, Uncertainty, Information and Entropy, Mutual Information, KL Divergence.

Syllabus

Unit 1 Basic Probability: Introduction to the notion of probability, Random experiment, Sample space and Events, Probability defined on events, Algebra of events. Conditional probabilities, independent events, Bayes' theorem.

Unit 2 Random Variables: Introduction to Random Variables, Probability mass/density functions, Cumulative distribution functions. Discrete Random Variables (Bernoulli, Binomial, Poisson, Multinomial and Geometric). Continuous Random Variables (Uniform, Exponential and Normal). Expectation of a Random Variable, Expectation of Function of a Random Variable and Variance. Markov inequality, Chebyshev's inequality, Central Limit Theorem, Weak and Strong Laws of Large Numbers.

Unit 3 Joint Distributions: Jointly distributed Random Variables, Joint distribution functions, Independent Random Variables, Covariance of Random Variables, Correlation Coefficients, Conditional Expectation.

Unit 4 Markov Chain and Information Theory: Introduction to Stochastic Processes, Chapman–Kolmogorov equations, Classification of states, Limiting and Stationary Probabilities. Random Number Generation, Pseudo Random Numbers, Inverse Transformation Method, Rejection Method, Uncertainty, Information and Entropy, Mutual Information, KL Divergence.

References

1. **S**heldon Ross, *Introduction to Probability Models*. 12th Edition, Elsevier, 2019.

2. K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. 2nd Edition, Wiley, 2015.
3. Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong, *Mathematics for Machine Learning*. 1st Edition. Cambridge University Press, 2020.
4. Ian F. Blake, *An Introduction to Applied Probability*. John Wiley.

Additional References

(i) James L. Johnson, *Probability and Statistics for Computer Science*, 6th Edition, Wiley, 2004.

(ii) David Forsyth, *Probability and Statistics for Computer Science*, 1st Edition, Springer, 2019.

(iii) Freund J.E., *Mathematical Statistics with Applications*, 8th Edition, Pearson Education, 2013.

(iv) Jay L. Devore, *Probability and Statistics for Engineering and the Sciences*, 9th Edition, Cengage Learning, 2020.

Suggested Practical List

The goal of this lab is to develop data interpretation skills. Following exercises are designed to enable students to understand data characteristics either by visualization or by interpreting computed measures. All the exercises are to be completed using MS Excel functions and graphs. At the end of each exercise, the student should be able to draw a conclusion and state in a concise manner.Teachers are expected to guide students to obtain real data available through the internet for the following exercises.

1. Plotting and fitting of Binomial distribution and graphical representation of probabilities.
2. Plotting and fitting of Multinomial distribution and graphical representation of probabilities.
3. Plotting and fitting of Poisson distribution and graphical representation of probabilities.
4. Plotting and fitting of Geometric distribution and graphical representation of probabilities.
5. Plotting and fitting of Uniform distribution and graphical representation of probabilities.
6. Plotting and fitting of Exponential distribution and graphical representation of probabilities.
7. Plotting and fitting of Normal distribution and graphical representation of probabilities.
8. Calculation of cumulative distribution functions for Exponential and Normal distribution.

9. Given data from two distributions, find the distance between the distributions.
10. Application problems based on the Binomial distribution.
11. Application problems based on the Poisson distribution.
12. Application problems based on the Normal distribution.
13. Presentation of bivariate data through scatter-plot diagrams and calculations of covariance.
14. Calculation of Karl Pearson’s correlation coefficients.
15. To find the correlation coefficient for a bivariate frequency distribution.
16. Generating Random numbers from discrete (Bernoulli, Binomial, Poisson) distributions.
17. Generating Random numbers from continuous (Uniform, Normal) distributions.
18. Find the entropy from the given data set.